Soft Lexical Control of Language Models to Generate Personalized Level-Appropriate Vocabulary Examples for Language Learners

William Walker
walk0950@umn.edu

Philip Nguyen
nguy4652@umn.edu

Emilie Bourget
bourg103@umn.edu

Abstract

Large language models hold potential as conversational language-learning assistants for language learners. However, a common issue is that LLM assistants can produce output using vocabulary or grammar that is outside of the learner's abilities to understand. In this report we demonstrate a novel decoding-time technique, "lexical logit boost", that encourages an LLM to use words from an arbitrary "vocabulary set" provided by the user at inference time. We specifically focus on the task of generating user-appropriate vocabulary example sentences, and we provide simple evaluations for this task. Tuning was required of our "boost" hyperparameter to balance the LLM's adherence to the vocabulary set and the generation quality. In the end, using lexical logit boost with Llama 3.1 8B, we are able to outperform prompting techniques used on the larger model Gemini 1.5 Flash.

1 Introduction

1.1 Motivation

In language learning, an common and effective way to acquire fluency is through conversational practice. However, not all language learners have access to conversational partners who One of the problems in language learning programs based around a fixed curriculum is that predefined "levels" of proficiency of a language assume that language learning follows a sequential, linear progression. Oftentimes the learning path of a language learner outside the classroom does not fit this predefined progression. Depending on the language learner's individual interests and motivations for learning a language such as travel, casual conversation, or mastery, different vocabulary may be more relevant at different times.

When the vocabulary in a conversation is too advanced, language learners are forced to rely on external aids such as dictionaries and translation tools. This process disrupts the flow of practice and may introduce even more advanced words that can lead the language learner to an overwhelming exposure of unfamiliar words. This is especially problematic for language learners who are just starting or have a limited vocabulary base. This disruption from the flow of practice can lead to discouragement and disengagement from the language learning process. The goal of this project is to address the difference in an individual's vocabulary to make the flow of practice more natural with less disruptions.

LLMs, with their ability to generate diverse and context rich responses, hold promise to become great assets for language learners. They are able to simulate a conversational partner, and through this project's exploration of various sentence generation techniques.

For the scope of this class project, we have chosen to work with the English language since all team members are proficient in English, which would better allow us to prototype and perform initial human evaluations.

In this report, we present the results of our exploration of various methods for sentence generation that may have application in the domain of language learning. We intend for this project to be the first step towards towards eventually building an LLM conversation partner that actively monitors the user's language proficiency and vocabulary and tailors its own language and behavior to challenge the language learner in a level-appropriate manner.

1.2 Formal problem description

Given a set of *vocab words V* that the language learner already knows and a single *target word*, generate a sentence using the target word. This sentence should use the target word in an illustrative manner, such that a language learner unfamiliar with the word could learn the meaning of the word from context. Furthermore, there is a lexical requirement: apart from the target word, the majority of words in the sentence should be *vocab words*. Non-vocab words are allowed, but they should compose only a small fraction of the words in the sentence to avoid overwhelming the language learner.

2 Related work

Jinran et al. (2023) approached the problem of generating example sentences for language learners at appropriate lexical complexity levels. However, their lexical complexity levels were pre-fixed and determined by CEFR scores (e.g. A1, A2, $\cdots$, C2). They accomplished this by adding one "complexity embedding" vector parameter for each complexity level. Next, each token was assigned a complexity level. During tokenization, complexity embeddings are added onto tokens in the same way that positional embeddings are in the typical transformer architecture (Vaswani et al., 2023). The model is fine-tuned on the task with all parameters frozen except for the complexity embeddings. Our work differs from this one because it allows for the set of "complex words" (those outside the vocabulary) to be determined arbitrarily at inference-time without any training or fine-tuning.

Liang et al. (2024) provide a very comprehensive survey of existing techniques for controllable text generation. Notably, they discuss several approaches to decoding-time interventions, including Plug-and-Play Language Models (Dathathri et al., 2020), and FUDGE (**?**). Both of these techniques are classifier-based logit-manipulation techniques, and they focus on the use of a trained classifier to guide generation at decoding time. Our work differs from these techniques in that no neural-network-based classifier is used to manipulate the logits; rather, an inexpensive and flexible rules-based approach are used to decide logit manipulations.

3 Techniques

Our investigations occurred in two phases. The first phase (Section 3.1) was a preliminary investigation, were we simply experimented with generating sentences (without lexical control). In the second stage (Section 3.2), we devised a lexical control system and tested it on the full task.

3.1 Example generation without lexical control, via prompting

To begin addressing the challenge of generating sentences to help language learners learn their desired vocab we began with an initial exploration using purely prompting techniques with Meta's Llama 3.1 8B Instruct model (Dubey et al., 2024), run via Hugging Face's Transformers library (Wolf et al., 2020). The dataset that we used (reference huggingface dataset) consisted of vocabulary words, their definitions, and example sentences that generally contained low contextual information. Using this we explored prompts that would produce high-context sentences that would assist the language learner in figuring out word meanings.

We began by implementing three prompting strategies to test the Llama model with. In all of the cases we instructed the model specifically to "generate a sentence, using the word directly, with enough context clues for someone to understand the meaning of the word without directly using the definition. Respond with only the sentence, nothing else, no explanations."

How these three strategies differed was in the amount and type of reference information from the dataset they were given. The first method was target-word-only, in this strategy the model was provided with only the vocabulary word and was asked to generate a sentence containing the word with enough context clues to convey the meaning of the word to the user.

The second method was target-word-and-definition, in this strategy the model was provided with both the target word as well as the definition. By including the definition in the prompt we hoped to assist the model in generating a sentence that more aligned with the intended definition of the word.

The third method was masked-word-with-definition, in this strategy the model was given the definition of the target word and was asked to use a special token, <vocab>, in place of a word with that definition. After the sentence was generated, the <vocab> token was then replaced with the original target word. The hope for this method was to encourage the model to focus on generating a context rich sentence that was independent of the target word.

By comparing the performance of the model given these three prompts, we attempted to measure the relative importance of the word itself and the word's definition in creating high-quality answers.

By providing only the target word, the language model knows exactly what word it must include. However, this word may have several different senses in which it is used (polysemy), and the language model has no way of disambiguating between these different senses.

By providing only the target word's definition (as in the masked-vocab prompt), the issue of polysemy is reduced (since the exact meaning of the word is specified), however the language model has no way of disambiguating between synonyms.

By evaluating the performance of the model on these three prompts, we hoped to determine what information was most important to supply into the prompt for the language example generation task.

Additionally, these initial investigations provided us with a foundation that could later be incorporated into the more complex modified decoder technique.

3.2 Inference-time lexical control via logit manipulation

Our goal is to "softly" lexically control the language model to preferentially use words from a privileged set, called the "vocab set".

A requirement is that this vocab set must be cheaply modifiable at inference time. For instance, *fine-tuning* the language model to use this vocab set would be infeasible, since the model would need to be repeatedly fine-tuned over time as the language learner's vocabulary expands and drifts.

An additional requirement is that this vocab set can potentially be quite large, consisting of thousands of words as the language learner's vocabulary grows. This rules out straightforward techniques where the full vocabulary set is passed into the LLM via its prompt. If such a purely prompt-based technique were used, prompt sizes would balloon and lead to very high compute costs, and potentially also negative impacts of filling such a large piece of the context with semantically unrelated/meaningless words.

Instead, we propose a third method, which we call "lexical logit boosting" (LLB), wherein we directly modify the logits of tokens in the next-token classification head at the very final stage of the language model's decoder. This method is compute-economical and is *completely prompt-agnostic*, meaning that this method can be applied to a variety of text generation tasks, notably including use in interactive chat assistants.

After some consideration, we have settled on the following implementation of lexical logit boosting. While simple, the method possesses useful mathematical properties that led us to choose it over a more complicated method.

As an introductory simplification, assume that every word in the language model's vocabulary is exactly one token. Then the language learner's familiar vocabulary set may be imagined as a set of tokens $V = \{v_1, v_2, \cdots, v_n\}$ that is a subset of $T = \{1, 2, \cdots, |T|\}$, the set of all of the language model's tokens. In this simple case, our implementation of lexical logit boost can be seen in Algorithm 1. Given a vocab set V and a "boost size" hyperparameter b, the logits corresponding to in-vocabulary words are simply increased by b.

Algorithm 1 Single-token implementation of lexical logit boost

Input: original logits $\vec{l} \in \mathbb{R}^{|T|}$,
vocab set $V \subseteq T$,
boost size $b \in \mathbb{R}$

```
1: for i = 1, ..., |T| do
2:     if i ∈ V then
3:         l'_i ← l_i + b
4:     else
5:         l'_i ← l_i
6:     end if
7: end for
```

Output: modified logits $\vec{l'}$

This tactic of incrementing logits by a constant results in some nice mathematical properties. Let $P(y)$ refer to the probability of the next token being y (conditioned on some context) without using lexical logit boost, and let $P'(y)$ be the corresponding probability while using lexical logit boost. We can show that LLB preserves properties of the original next-token distribution l, namely the relative abundances of in-vocabulary tokens:

$$\frac{P'(v_i)}{P'(v_j)} = \frac{P(v_i)}{P(v_j)} \, \forall v_i, v_j \in V. \tag{1}$$

This follows because

$$\begin{aligned}\frac{P'(v_i)}{P'(v_j)} &= \frac{\text{softmax}(\vec{l'})[v_i]}{\text{softmax}(\vec{l'})[v_j]} \\ &= \frac{e^{\vec{l'}[v_i]}/\sum_{k=1}^{|T|} e^{\vec{l'}[k]}}{e^{\vec{l'}[v_j]}/\sum_{k=1}^{|T|} e^{\vec{l'}[k]}} \\ &= \frac{e^{\vec{l}[v_i]+b}}{e^{\vec{l}[v_j]+b}} \\ &= \frac{e^{\vec{l}[v_i]}/\sum_{k=1}^{|T|} e^{\vec{l}[k]}}{e^{\vec{l}[v_j]}/\sum_{k=1}^{|T|} e^{\vec{l}[k]}} \\ &= \frac{\text{softmax}(\vec{l})[v_i]}{\text{softmax}(\vec{l})[v_j]} \\ &= \frac{P(v_i)}{P(v_j)}.\end{aligned}$$

Similarly, if two tokens are not in the vocabulary, their relative abundance is also preserved, i.e.

$$\frac{P'(y_i)}{P'(y_j)} = \frac{P(y_i)}{P(y_j)} \; \forall y_i, y_j \notin V. \tag{2}$$

This property is appealing, because it means that while lexical logit boosting influences the probability that the next token generated is within the vocabulary, it does not perturb the conditional probabilities

$$P'(y \mid y \in V) = P(y \mid y \in V) \text{ and} \tag{3}$$
$$P'(y \mid y \notin V) = P(y \mid y \notin V). \tag{4}$$

We will call this property (i.e. the satisfaction of Equations 3 and 4) "transparency". This means that while the LLB decoder increases the probability that the next token will come from the vocab set, when making the decision of exactly *which* token it will be within (or without) the vocab set, it defers completely to the underlying language model. It is for this reason that we call the property "transparency"; the decoder respects the fine-grained next-token preferences of the signal outputted by the underlying model.

Now that we have demonstrated a simple version of lexical logit boost, we present Algorithm 2, which extends it to handle multi-token vocabulary words. That is, instead of V containing individual tokens, it should contain sequences of tokens $V = \{(v_i^1, \cdots v_i^{k_i})\}_{i \in \{1, \cdots n\}} \subseteq T^*$ (where * indicates the Kleene star operation).

In this algorithm, a token y is only given a boost if, when combined onto some suffix of the preceding context, the result $x_{t-m}x_{t-m+1} \cdots x_{t-1}\, y$ is a prefix of a word w in the vocab set.

Algorithm 2 Multi-token implementation of lexical logit boost (note: one-indexed)

Input:
original logits $\vec{l} \in \mathbb{R}^{|T|}$,
vocab set $V \subseteq T^*$,
boost size $b \in \mathbb{R}$,
context $x = x_1 \cdots x_{t-1} \in T^{t-1}$
1: $\vec{\beta} \leftarrow \texttt{zeros}(|T|)$
2: **for** $w = (v^1 \cdots v^{k_i}) \in V$ **do**
3: **for** $m = 0, \cdots, k_i - 1$ **do**
4: **if** $w[1 : m+1] = x[t-m : t]$ **then**
5: $\beta[w[k_i - m]] \leftarrow b$
6: **end if**
7: **end for**
8: **end for**
9: $\vec{l'} \leftarrow \vec{l} + \vec{\beta}$
Output: modified logits $\vec{l'}$

During prototyping, we considered the possibility of the boost added to the token varying depending on the length m of the matching token sequence, however we decided against this because 1. it tended to bias the output towards short words, and 2. the resulting decoder no longer had the transparency property.

An alternate implementation note is that lines 2-10 of Algorithm 2 can equivalently be expressed as a language membership problem decidable by a definite finite automaton. If we let S be the set of reversed suffixes of words $w \in V$, the language being matched is $S \circ T^*$, and the string being checked for membership is $y\; x_{t-1}x_{t-2} \cdots x_1$. This must be checked for every $y \in T$. Under certain circumstances (very large $|V|$, small $|T|$), this implementation may be more efficient than the implementation notated in Algorithm 2.

There also may be yet better formulations/implementations of this algorithm, but this is the extent of what we have found so far. In practice, Algorithm 2 runs very quickly if the lengths of $w \in V$ are reasonably small (which is generally expected to be the case, since most words are generally composed of no more than 5 tokens).

4 Evaluations

Similarly to how Section 3 is broken into two subsections for the two phases of our investigation, this section is also divided into two subsections holding the corresponding evaluations used.

4.1 Evaluations for prompting-based investigation

The evaluations detailed in this subsection correspond to the techniques described in Subsection 3.1.

For evaluation of the initial three strategies, we used DiscoScore (Zhao et al., 2023), a BERT based evaluation framework that provides scores that measures sentences's focus and coherence. The two scores that DiscoScore provides are DS_FOCUS_NN and DS_SENT_NN.

DS_FOCUS_NN is a metric that evaluates the focus drift of a sentence. For our use case, this score was used to determine if the sentence remained centered around the target vocabulary word.

DS_SENT_NN is a metric that evaluates the overall coherence of the generated sentence. As a preliminary step, we used DiscoScore to help measure and ensure that the sentences remain on topic as well as coherent overall before transitioning to manual human evaluations.

This initial step was done to scope out how the model would react to generating these kinds of sentences. Because we are altering the vocabulary output of the generated sentences, it is important to ensure "weights" of certain words are not pushed and favored out of proportion which would then lead to an incoherent sentence.

4.2 Evaluations for lexically controlled investigation

This experiment corresponds to the techniques described in Section 3.2. It tests the performance of Llama 3.1 8B with the modified LLB decoder on the sentence generation task.

The key evaluation objectives were:

1. Control: how well does the generation stay within the inputted vocab set?
2. Quality: is the sample sentence high-quality?
    (a) Does the sentence contain the target word?
    (b) "Mechanics": is the sentence syntactically/grammatically correct?
    (c) "Semantics": Is the semantic meaning of the sentence realistic/plausible?
    (d) "Context": Does the sentence illustrate the meaning of the target word? Can the meaning of the word be inferred from the surrounding context?

Evaluations for objectives 1 and 2(a) were implemented programmatically.

The control score is by word counts. To score control for a generation, the generation is cleaned and split into words. The words are counted, and the percentage of words outside the vocab set constitutes the evaluation result. For the purposes of this evaluation, the target word is also temporarily considered to part of the vocabulary, so that the model is not penalized for using the target word. Generally, a lower score is preferred so as not to overwhelm the language learner. However, ideal generation does not necessarily minimize this score. Instead, the score need only be below a learner-specific threshold of what they find comfortable or otherwise prefer.

To evaluate objective 2(a), the generation was simply checked for the target word as an exact substring match.

For both of these programmatic evaluations, lemmatization was not applied to the words, meaning that that these evaluations did not consider alternate conjugations, declensions, or other inflections of words. This is left as a potential future enhancement.

For the remaining, more subjective evaluations, we collected human evaluations. We elected to do this rather than use automated evaluation metrics for a number of reasons. First, the final goal of the project is human-centric, and directly measuring humans would be simpler than introducing an intermediate metric which would constitute an additional point of failure or uncertainty. Second, we did not find easily available automatic evaluations for objectives 2(b), 2(c), and 2(d). LLM-as-a-judge was an option, but again we would then have the issue of determining whether or not those judge scores were closely related to ground truth human evaluation. In the end, then, we elected to do human evaluation.

The three qualities (mechanics, semantics, and context) were all evaluated on a 0-5 Likert scale, where higher numbers indicated a more desirable score. For the full evaluation rubric, see Appendix B.

4.3 Experimental design

This section describes the design of the experiment testing the lexical logit boosting prototype. It corresponds to Sections 3.2 and 4.2.

For this experiment, the vocabulary set was arbitrarily chosen to be a set of the 500 most common

English words from a list scraped from [a website]. This list was chosen because, unlike other lists we checked, it did not lemmatize words before placing them on the list. This was important because, for instance, the word "is" would not appear on a lemmatized list, as all occurrences of "is" would be counted as "be". It is important to note that the architecture of LLB made no requirements that we choose the top 500 words in English as the vocab set; we simply picked this vocabulary because it would be the most basic list to use for prototyping and proofs of concept (especially considering our desired application). (For instance, we could have picked a vocab set of all words starting with the letter "e", however we decided against this more eccentric vocabulary set given the above reasons.)

Next, 20 target words were randomly selected from the list used in the first described in Section 3.1. Then, various LLM configurations (described further below) were prompted to generate example sentences using the target word. These generations were collected, and the evaluations were run on them to produce the final results.

The models used were Llama 3.1 8B Instruct with various values of the boost hyperparameter ($b = 0, 4, 8, 16$), and stock Gemini 1.5 Flash (Team et al., 2024), accessed via Google's API. The $b = 0$ model served as a control (as when $b = 0$, logits are boosted by 0 when they are in the vocab set), the other values of b served to examine the effects of stronger and stronger boost strengths. The Llama models were prompted with instructions and the target word only; the alternate prompts described in Section 3.1 were not used.

Finally, the Gemini model served as the reference "unmodified state of the art model". Unlike the Llama models, the Gemini model was supplied the vocabulary list directly via its prompt. (See Appendix A for the exact prompts and hyperparameters for both models.) Recall that this technique may work well for small vocabularies, but it is more costly and is anticipated to have other issues, especially as the vocabulary size grows.

5 Results

5.1 Prompting results

The results in this section correspond to the preliminary, investigative work described in Sections 5.1 and 4.1.

Figure 1 plots the FOCUS scores of the

Relevant figures: Figures 1, 2.

Visually, the DiscoScore-FOCUS scores of the word-only prompt and the word-and-definition prompt both appear similar. However, the masked-vocab prompt had a greater number of generations with FOCUS scores of 0, indicating a large number of masked-vocab generations had low semantic similarity to the reference sentences using the same target word.

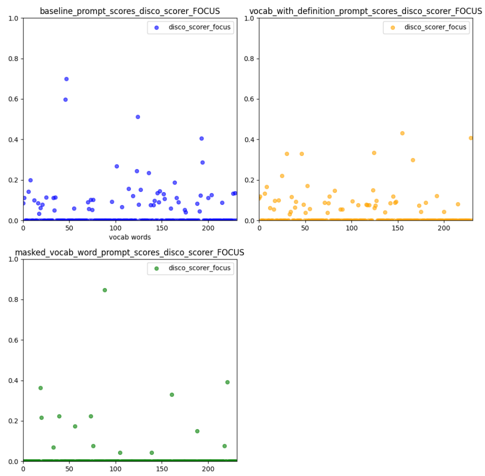

Figure 1: DiscoScore FOCUS scores for the three prompting strategies. Among the subplots, values at the same x position correspond to the same target word.

This result is interesting, as it may suggest that to get a

Polysemy vs synonymy

This suggests that the masked-vocab prompt

Visually, no large differences between the SENT scores was observed (Figure 2).

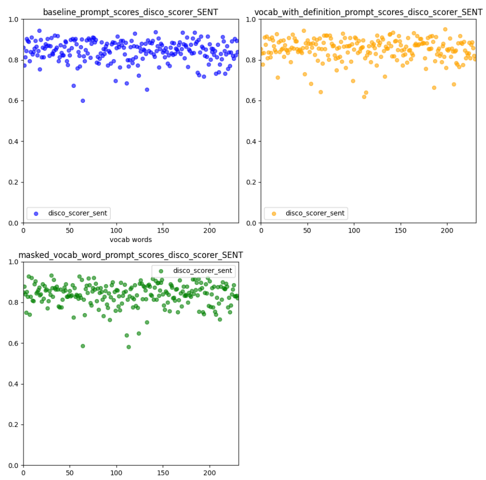

Figure 2: DiscoScore SENT scores for the three prompting strategies

5.2 Lexical logit boost results

These results correspond to the work described in Sections 3.2, 4.2, and 4.3.

The stronger the boost applied, the more strongly the LLM's generation stayed within the vocab set (Figure 3). However, strong boosts also harmed the quality of generation, as seen both in the human evaluations (Figure 5) and in the LLM's ability to follow the prompt's instructions and include the target word (Figure 4). This is a manifestation of the familiar control/quality trade-off discussed in Liang et al. (2024). However, in this case it seemed that the model with a moderate boost value ($b = 4$) presented a satisfactory result. In quality, the $b = 4$ model performed only marginally worse than the control $b = 0$ model and the Gemini model, while its mean non-vocab-set percentage was significantly lower than the Gemini model.

To measure inter-evaluator agreement, Krippendorff α values (Krippendorff, 2011) were calculated for each metric. They were $\alpha = 0.76, 0.66, 0.69$ for the mechanics, semantics, and context human evals, respectively.

Qualitatively, we observed that generations with higher values of b tended to be prone to form long run-on sentences. We hypothesize that the reason for this is that the vocab set did not contain punctuation, so punctuation tokens were not boosted along with the common tokens. Given the transparency property of LLB, punctuation should be generated "as normal" if it is added to the vocab set. This is a topic for future investigation.

Another qualitative observation was that the LLB model appeared to produce lower-quality generations when the target words were more lexically complex. A possible reason for this is that when a target word is lexically complex, it is more difficult to generate a lexically simple example sentence. This hypothesis is also worth future investigation.

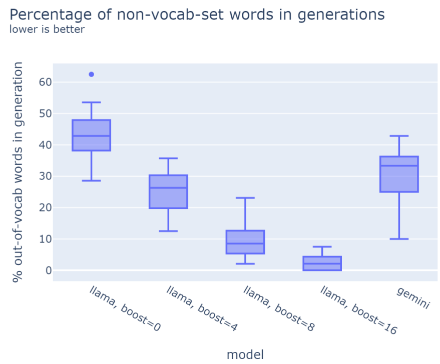

Figure 3: As the boost parameter b was increased, the generation tended more and more strongly towards using words from the vocab set, as intended.

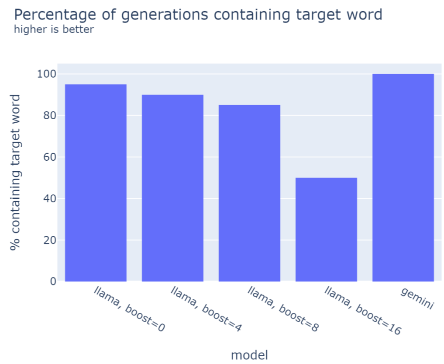

Figure 4: As the boost parameter b was increased, the fraction of generations containing the target word undesirably decreased.

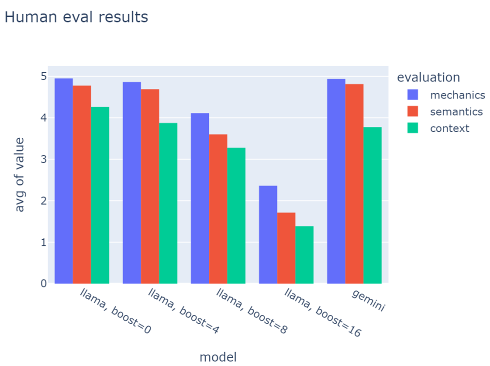

Figure 5: As the boost parameter b was increased, human evaluations of generation quality were initially not heavily affected. However, when b became sufficiently large, evaluation scores dropped heavily.

6 Discussion

6.1 Ethics

Our work aims to support language learners by assisting with their vocabulary learning process through generating sentences. While our goal is educational and is intended to have positive impacts. We have to acknowledge that there are potential risks associated with using large language models. The first is the dependency on already existing large language models. The problem here is that these pretrained models may have inherent biases due to their training data or company values and these biases can be reflected in the output of the model. Such outputs could misinform or reinforce stereotypes when generating sentences for the user. A way to address this problem could be to carefully prompt the model or deploy additional evaluation metrics to ensure the outputs are safe. Another ethical concern relates to user data privacy. Because the ultimate goal for the modified decoder is to learn the user's vocabulary through conversation instead of the user directly providing it themselves, the way this user data is handled can lead to some concerns. There has to be a safeguard set so that users are informed as to how their personal data is tracked, stored, and analyzed even if it is for educational purposes. Through identifying these potential risks, corrective actions can be taken to mitigate the risk and help to ensure a net positive impact on society.

6.2 Limitations

The current limitation of our project is that the stored user vocabulary does not grow and adapt as the user interacts with the model. For the scope of this project all of the evaluations were done using a vocabulary set that contains the 200 most common words, not a set specific to any one person. The ultimate goal of the model is that the model will, on its own, learn the user's vocabulary as they continue to interact with it.

Another limitation of our project is that right now the focus is on generating a single sentence from a single target word. But, to be an effective language learning tool, eventually we want the interaction with the model to be more like a conversation. Meaning instead of providing the vocab one at a time, a list of vocab can be provided beforehand and the model will incorporate them into the conversation as the user interacts with it.

6.3 Future work

- Multiple lexical complexity levels
- Adaptive b
- Investigate run-on sentence problems. Is punctuation the problem?
- For this exact application, constrained decoding may be useful. This way we could force the output to contain the target word.
- More scalable evaluation via perplexity or LLM-as-a-judge.
- Larger human evaluation.
- Create system to monitor user's vocabulary use (the "input half" of the adaptive-vocab chat assistant).

A key future step that we would like to highlight is to create a system that monitors the language learner's vocabulary use during conversations and uses this to build and update the vocabulary set over time. If this tool is developed and combined with lexical logit boost or similar control strategies, the result would be a fully adaptive-vocabulary chat assistant, a boon to language learners.

6.4 Acknowledgements

We extend our thanks to our project mentors Shirley Anugrah Hayati and James Mooney for the guidance they have offered throughout this project. Additionally, we thank Andrew Hale for volunteering to contribute to the human evaluation. Finally we thank our instructors: Dongyeop Kang, Shirley Anugrah Hayati, James Mooney, and Robert Jia for equipping us with the knowledge, both theoretical and practical, that has allowed us to execute this project.

References

Sumanth Dathathri, Andrea Madotto, Janice Lan, Jane Hung, Eric Frank, Piero Molino, Jason Yosinski, and Rosanne Liu. 2020. Plug and play language models: A simple approach to controlled text generation.

Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, Anirudh Goyal, Anthony Hartshorn, Aobo Yang, Archi Mitra, Archie Sravankumar, Artem Korenev, Arthur Hinsvark, Arun Rao, Aston Zhang, Aurelien Rodriguez, Austen Gregerson, Ava Spataru, Baptiste Roziere, Bethany Biron, Binh Tang, Bobbie Chern, Charlotte Caucheteux, Chaya Nayak, Chloe Bi, Chris Marra, Chris McConnell, Christian Keller, Christophe Touret, Chunyang Wu, Corinne Wong, Cristian Canton Ferrer, Cyrus Nikolaidis, Damien Allonsius, Daniel Song, Danielle Pintz, Danny Livshits, David Esiobu, Dhruv Choudhary, Dhruv Mahajan, Diego Garcia-Olano, Diego Perino, Dieuwke Hupkes, Egor Lakomkin, Ehab AlBadawy, Elina Lobanova, Emily Dinan, Eric Michael Smith, Filip Radenovic, Frank Zhang, Gabriel Synnaeve, Gabrielle Lee, Georgia Lewis Anderson, Graeme Nail, Gregoire Mialon, Guan Pang, Guillem Cucurell, Hailey Nguyen, Hannah Korevaar, Hu Xu, Hugo Touvron, Iliyan Zarov, Imanol Arrieta Ibarra, Isabel Kloumann, Ishan Misra, Ivan Evtimov, Jade Copet, Jaewon Lee, Jan Geffert, Jana Vranes, Jason Park, Jay Mahadeokar, Jeet Shah, Jelmer van der Linde, Jennifer Billock, Jenny Hong, Jenya Lee, Jeremy Fu, Jianfeng Chi, Jianyu Huang, Jiawen Liu, Jie Wang, Jiecao Yu, Joanna Bitton, Joe Spisak, Jongsoo Park, Joseph Rocca, Joshua Johnstun, Joshua Saxe, Junteng Jia, Kalyan Vasuden Alwala, Kartikeya Upasani, Kate Plawiak, Ke Li, Kenneth Heafield, Kevin Stone, Khalid El-Arini, Krithika Iyer, Kshitiz Malik, Kuenley Chiu, Kunal Bhalla, Lauren Rantala-Yeary, Laurens van der Maaten, Lawrence Chen, Liang Tan, Liz Jenkins, Louis Martin, Lovish Madaan, Lubo Malo, Lukas Blecher, Lukas Landzaat, Luke de Oliveira, Madeline Muzzi, Mahesh Pasupuleti, Mannat Singh, Manohar Paluri, Marcin Kardas, Mathew Oldham, Mathieu Rita, Maya Pavlova, Melanie Kambadur, Mike Lewis, Min Si, Mitesh Kumar Singh, Mona Hassan, Naman Goyal, Narjes Torabi, Nikolay Bashlykov, Nikolay Bogoychev, Niladri Chatterji, Olivier Duchenne, Onur Çelebi, Patrick Alrassy, Pengchuan Zhang, Pengwei Li, Petar Vasic, Peter Weng, Prajjwal Bhargava, Pratik Dubal, Praveen Krishnan, Punit Singh Koura, Puxin Xu, Qing He, Qingxiao Dong, Ragavan Srinivasan, Raj Ganapathy, Ramon Calderer, Ricardo Silveira Cabral, Robert Stojnic, Roberta Raileanu, Rohit Girdhar, Rohit Patel, Romain Sauvestre, Ronnie Polidoro, Roshan Sumbaly, Ross Taylor, Ruan Silva, Rui Hou, Rui Wang, Saghar Hosseini, Sahana Chennabasappa, Sanjay Singh, Sean Bell, Seohyun Sonia Kim, Sergey Edunov, Shaoliang Nie, Sharan Narang, Sharath Raparthy, Sheng Shen, Shengye Wan, Shruti Bhosale, Shun Zhang, Simon Vandenhende, Soumya Batra, Spencer Whitman, Sten Sootla, Stephane Collot, Suchin Gururangan, Sydney Borodinsky, Tamar Herman, Tara Fowler, Tarek Sheasha, Thomas Georgiou, Thomas Scialom, Tobias Speckbacher, Todor Mihaylov, Tong Xiao, Ujjwal Karn, Vedanuj Goswami, Vibhor Gupta, Vignesh Ramanathan, Viktor Kerkez, Vincent Gonguet, Virginie Do, Vish Vogeti, Vladan Petrovic, Weiwei Chu, Wenhan Xiong, Wenyin Fu, Whitney Meers, Xavier Martinet, Xiaodong Wang, Xiaoqing Ellen Tan, Xinfeng Xie, Xuchao Jia, Xuewei Wang, Yaelle Goldschlag, Yashesh Gaur, Yasmine Babaei, Yi Wen, Yiwen Song, Yuchen Zhang, Yue Li, Yuning Mao, Zacharie Delpierre Coudert, Zheng Yan, Zhengxing Chen, Zoe Papakipos, Aaditya Singh, Aaron Grattafiori, Abha Jain, Adam Kelsey, Adam Shajnfeld, Adithya Gangidi, Adolfo Victoria, Ahuva Goldstand, Ajay Menon, Ajay Sharma, Alex Boesenberg, Alex Vaughan, Alexei Baevski, Allie Feinstein, Amanda Kallet, Amit Sangani, Anam Yunus, Andrei Lupu, Andres Alvarado, Andrew Caples, Andrew Gu, Andrew Ho, Andrew Poulton, Andrew Ryan, Ankit Ramchandani, Annie Franco, Aparajita Saraf, Arkabandhu Chowdhury, Ashley Gabriel, Ashwin Bharambe, Assaf Eisenman, Azadeh Yazdan, Beau James, Ben Maurer, Benjamin Leonhardi, Bernie Huang, Beth Loyd, Beto De Paola, Bhargavi Paranjape, Bing Liu, Bo Wu, Boyu Ni, Braden Hancock, Bram Wasti, Brandon Spence, Brani Stojkovic, Brian Gamido, Britt Montalvo, Carl Parker, Carly Burton, Catalina Mejia, Changhan Wang, Changkyu Kim, Chao Zhou, Chester Hu, Ching-Hsiang Chu, Chris Cai, Chris Tindal, Christoph Feichtenhofer, Damon Civin, Dana Beaty, Daniel Kreymer, Daniel Li, Danny Wyatt, David Adkins, David Xu, Davide Testuggine, Delia David, Devi Parikh, Diana Liskovich, Didem Foss, Dingkang Wang, Duc Le, Dustin Holland, Edward Dowling, Eissa Jamil, Elaine Montgomery, Eleonora Presani, Emily Hahn, Emily Wood, Erik Brinkman, Esteban Arcaute, Evan Dunbar, Evan Smothers, Fei Sun, Felix Kreuk, Feng Tian, Firat Ozgenel, Francesco Caggioni, Francisco Guzmán, Frank Kanayet, Frank Seide, Gabriela Medina Florez, Gabriella Schwarz, Gada Badeer, Georgia Swee, Gil Halpern, Govind Thattai, Grant Herman, Grigory Sizov, Guangyi, Zhang, Guna Lakshminarayanan, Hamid Shojanazeri, Han Zou, Hannah Wang, Hanwen Zha, Haroun Habeeb, Harrison Rudolph, Helen Suk, Henry Aspegren, Hunter Goldman, Ibrahim Damlaj, Igor Molybog, Igor Tufanov, Irina-Elena Veliche, Itai Gat, Jake Weissman, James Geboski, James Kohli, Japhet Asher, Jean-Baptiste Gaya, Jeff Marcus, Jeff Tang, Jennifer Chan, Jenny Zhen, Jeremy Reizenstein, Jeremy Teboul, Jessica Zhong, Jian Jin, Jingyi Yang, Joe Cummings, Jon Carvill, Jon Shepard, Jonathan McPhie, Jonathan Torres, Josh Ginsburg, Junjie Wang, Kai Wu, Kam Hou U, Karan Saxena, Karthik Prasad, Kartikay Khandelwal, Katayoun Zand, Kathy Matosich, Kaushik Veeraraghavan, Kelly Michelena, Keqian Li, Kun Huang, Kunal Chawla, Kushal Lakhotia, Kyle Huang, Lailin Chen, Lakshya Garg, Lavender A, Leandro Silva, Lee Bell, Lei Zhang, Liangpeng Guo, Licheng Yu, Liron Moshkovich, Luca Wehrstedt, Madian Khabsa, Manav Avalani, Manish Bhatt, Maria Tsimpoukelli, Martynas Mankus, Matan Hasson, Matthew Lennie, Matthias Reso, Maxim Groshev, Maxim Naumov, Maya Lathi, Meghan Keneally, Michael L. Seltzer, Michal Valko, Michelle Restrepo, Mihir Patel, Mik Vyatskov, Mikayel Samvelyan, Mike Clark, Mike Macey, Mike Wang, Miquel Jubert Hermoso, Mo Metanat, Mohammad Rastegari, Munish Bansal, Nandhini Santhanam, Natascha Parks, Natasha White, Navyata Bawa, Nayan Singhal, Nick Egebo, Nicolas Usunier, Nikolay Pavlovich Laptev, Ning Dong, Ning Zhang, Norman Cheng, Oleg Chernoguz, Olivia Hart, Omkar Salpekar, Ozlem Kalinli, Parkin Kent, Parth Parekh, Paul Saab, Pavan Balaji, Pedro Rittner, Philip Bontrager, Pierre Roux, Piotr Dollar, Polina Zvyagina, Prashant Ratanchandani, Pritish Yuvraj, Qian Liang, Rachad Alao, Rachel Rodriguez, Rafi Ayub, Raghotham Murthy, Raghu Nayani, Rahul Mitra, Raymond Li, Rebekkah Hogan, Robin Battey, Rocky Wang, Rohan Maheswari, Russ Howes, Ruty Rinott, Sai Jayesh Bondu, Samyak Datta, Sara Chugh, Sara Hunt, Sargun Dhillon, Sasha Sidorov, Satadru Pan, Saurabh Verma, Seiji Yamamoto, Sharadh Ramaswamy, Shaun Lindsay, Shaun Lindsay, Sheng Feng, Shenghao Lin, Shengxin Cindy Zha, Shiva Shankar, Shuqiang Zhang, Shuqiang Zhang, Sinong Wang, Sneha Agarwal, Soji Sajuyigbe, Soumith Chintala, Stephanie Max, Stephen Chen, Steve Kehoe, Steve Satterfield, Sudarshan Govindaprasad, Sumit Gupta, Sungmin Cho, Sunny Virk, Suraj Subramanian, Sy Choudhury, Sydney Goldman, Tal Remez, Tamar Glaser, Tamara Best, Thilo Kohler, Thomas Robinson, Tianhe Li, Tianjun Zhang, Tim Matthews, Timothy Chou, Tzook Shaked, Varun Vontimitta, Victoria Ajayi, Victoria Montanez, Vijai Mohan, Vinay Satish Kumar, Vishal Mangla, Vítor Albiero, Vlad Ionescu, Vlad Poenaru, Vlad Tiberiu Mihailescu, Vladimir Ivanov, Wei Li, Wenchen Wang, Wenwen Jiang, Wes Bouaziz, Will Constable, Xiaocheng Tang, Xiaofang Wang, Xiaojian Wu, Xiaolan Wang, Xide Xia, Xilun Wu, Xinbo Gao, Yanjun Chen, Ye Hu, Ye Jia, Ye Qi, Yenda Li, Yilin Zhang, Ying Zhang, Yossi Adi, Youngjin Nam, Yu, Wang, Yuchen Hao, Yundi Qian, Yuzi He, Zach Rait, Zachary DeVito, Zef Rosnbrick, Zhaoduo Wen, Zhenyu Yang, and Zhiwei Zhao. 2024. The Llama 3 herd of models.

Nie Jinran, Yang Liner, Chen Yun, Kong Cunliang, Zhu Junhui, and Yang Erhong. 2023. Lexical complexity controlled sentence generation for language learning. In *Proceedings of the 22nd Chinese National Conference on Computational Linguistics*, pages 648–664, Harbin, China. Chinese Information Processing Society of China.

Klaus Krippendorff. 2011. Computing krippendorff's alpha-reliability.

Xun Liang, Hanyu Wang, Yezhaohui Wang, Shichao Song, Jiawei Yang, Simin Niu, Jie Hu, Dan Liu, Shunyu Yao, Feiyu Xiong, and Zhiyu Li. 2024. Controllable text generation for large language models: A survey.

Gemini Team, Petko Georgiev, Ving Ian Lei, Ryan Burnell, Libin Bai, Anmol Gulati, Garrett Tanzer, Damien Vincent, Zhufeng Pan, Shibo Wang, Soroosh Mariooryad, Yifan Ding, Xinyang Geng, Fred Alcober, Roy Frostig, Mark Omernick, Lexi Walker, Cosmin Paduraru, Christina Sorokin, Andrea Tacchetti, Colin Gaffney, Samira Daruki, Olcan Sercinoglu, Zach Gleicher, Juliette Love, Paul Voigtlaender, Rohan Jain, Gabriela Surita, Kareem Mohamed, Rory Blevins, Junwhan Ahn, Tao Zhu, Kornraphop Kawintiranon, Orhan Firat, Yiming Gu, Yujing Zhang, Matthew Rahtz, Manaal Faruqui, Natalie Clay, Justin Gilmer, JD Co-Reyes, Ivo Penchev, Rui Zhu, Nobuyuki Morioka, Kevin Hui, Krishna Haridasan, Victor Campos, Mahdis Mahdieh, Mandy Guo, Samer Hassan, Kevin Kilgour, Arpi Vezer, Heng-Tze Cheng, Raoul de Liedekerke, Siddharth Goyal, Paul Barham, DJ Strouse, Seb Noury, Jonas Adler, Mukund Sundararajan, Sharad Vikram, Dmitry Lepikhin, Michela Paganini, Xavier Garcia, Fan Yang, Dasha Valter, Maja Trebacz, Kiran Vodrahalli, Chulayuth Asawaroengchai, Roman Ring, Norbert Kalb, Livio Baldini Soares, Siddhartha Brahma, David Steiner, Tianhe Yu, Fabian Mentzer, Antoine He, Lucas Gonzalez, Bibo Xu, Raphael Lopez Kaufman, Laurent El Shafey, Junhyuk Oh, Tom Hennigan, George van den Driessche, Seth Odoom, Mario Lucic, Becca Roelofs, Sid Lall, Amit Marathe, Betty Chan, Santiago Ontanon, Luheng He, Denis Teplyashin, Jonathan Lai, Phil Crone, Bogdan Damoc, Lewis Ho, Sebastian Riedel, Karel Lenc, Chih-Kuan Yeh, Aakanksha Chowdhery, Yang Xu, Mehran Kazemi, Ehsan Amid, Anastasia Petrushkina, Kevin Swersky, Ali Khodaei, Gowoon Chen, Chris Larkin, Mario Pinto, Geng Yan, Adria Puigdomenech Badia, Piyush Patil, Steven Hansen, Dave Orr, Sebastien M. R. Arnold, Jordan Grimstad, Andrew Dai, Sholto Douglas, Rishika Sinha, Vikas Yadav, Xi Chen, Elena Gribovskaya, Jacob Austin, Jeffrey Zhao, Kaushal Patel, Paul Komarek, Sophia Austin, Sebastian Borgeaud, Linda Friso, Abhimanyu Goyal, Ben Caine, Kris Cao, Da-Woon Chung, Matthew Lamm, Gabe Barth-Maron, Thais Kagohara, Kate Olszewska, Mia Chen, Kaushik Shivakumar, Rishabh Agarwal, Harshal Godhia, Ravi Rajwar, Javier Snaider, Xerxes Dotiwalla, Yuan Liu, Aditya Barua, Victor Ungureanu, Yuan Zhang, Bat-Orgil Batsaikhan, Mateo Wirth, James Qin, Ivo Danihelka, Tulsee Doshi, Martin Chadwick, Jilin Chen, Sanil Jain, Quoc Le, Arjun Kar, Madhu Gurumurthy, Cheng Li, Ruoxin Sang, Fangyu Liu, Lampros Lamprou, Rich Munoz, Nathan Lintz, Harsh Mehta, Heidi Howard, Malcolm Reynolds, Lora Aroyo, Quan Wang, Lorenzo Blanco, Albin Cassirer, Jordan Griffith, Dipanjan Das, Stephan Lee, Jakub Sygnowski, Zach Fisher, James Besley, Richard Powell, Zafarali Ahmed, Dominik Paulus, David Reitter, Zalan Borsos, Rishabh Joshi, Aedan Pope, Steven Hand, Vittorio Selo, Vihan Jain, Nikhil Sethi, Megha Goel, Takaki Makino, Rhys May, Zhen Yang, Johan Schalkwyk, Christina Butterfield, Anja Hauth, Alex Goldin, Will Hawkins, Evan Senter, Sergey Brin, Oliver Woodman, Marvin Ritter, Eric Noland, Minh Giang, Vijay Bolina, Lisa Lee, Tim Blyth, Ian Mackinnon, Machel Reid, Obaid Sarvana, David Silver, Alexander Chen, Lily Wang, Loren Maggiore, Oscar Chang, Nithya Attaluri, Gregory Thornton, Chung-Cheng Chiu, Oskar Bunyan, Nir Levine, Timothy Chung, Evgenii Eltyshev, Xiance Si, Timothy Lillicrap, Demetra Brady, Vaibhav Aggarwal, Boxi Wu, Yuanzhong Xu, Ross McIlroy, Kartikeya Badola, Paramjit Sandhu, Erica Moreira, Wojciech Stokowiec, Ross Hemsley, Dong Li, Alex Tudor, Pranav Shyam, Elahe Rahimtoroghi, Salem Haykal, Pablo Sprechmann, Xiang Zhou, Diana Mincu, Yujia Li, Ravi Addanki, Kalpesh Krishna, Xiao Wu, Alexandre Frechette, Matan Eyal, Allan Dafoe, Dave Lacey, Jay Whang, Thi Avrahami, Ye Zhang, Emanuel Taropa, Hanzhao Lin, Daniel Toyama, Eliza Rutherford, Motoki Sano, HyunJeong Choe, Alex Tomala, Chalence Safranek-Shrader, Nora Kassner, Mantas Pajarskas, Matt Harvey, Sean Sechrist, Meire Fortunato, Christina Lyu, Gamaleldin Elsayed, Chenkai Kuang, James Lottes, Eric Chu, Chao Jia, Chih-Wei Chen, Peter Humphreys, Kate Baumli, Connie Tao, Rajkumar Samuel, Cicero Nogueira dos Santos, Anders Andreassen, Nemanja Rakićević, Dominik Grewe, Aviral Kumar, Stephanie Winkler, Jonathan Caton, Andrew Brock, Sid Dalmia, Hannah Sheahan, Iain Barr, Yingjie Miao, Paul Natsev, Jacob Devlin, Feryal Behbahani, Flavien Prost, Yanhua Sun, Artiom Myaskovsky, Thanumalayan Sankaranarayana Pillai, Dan Hurt, Angeliki Lazaridou, Xi Xiong, Ce Zheng, Fabio Pardo, Xiaowei Li, Dan Horgan, Joe Stanton, Moran Ambar, Fei Xia, Alejandro Lince, Mingqiu Wang, Basil Mustafa, Albert Webson, Hyo Lee, Rohan Anil, Martin Wicke, Timothy Dozat, Abhishek

Sinha, Enrique Piqueras, Elahe Dabir, Shyam Upadhyay, Anudhyan Boral, Lisa Anne Hendricks, Corey Fry, Josip Djolonga, Yi Su, Jake Walker, Jane Labanowski, Ronny Huang, Vedant Misra, Jeremy Chen, RJ Skerry-Ryan, Avi Singh, Shruti Rijhwani, Dian Yu, Alex Castro-Ros, Beer Changpinyo, Romina Datta, Sumit Bagri, Arnar Mar Hrafnkelsson, Marcello Maggioni, Daniel Zheng, Yury Sulsky, Shaobo Hou, Tom Le Paine, Antoine Yang, Jason Riesa, Dominika Rogozinska, Dror Marcus, Dalia El Badawy, Qiao Zhang, Luyu Wang, Helen Miller, Jeremy Greer, Lars Lowe Sjos, Azade Nova, Heiga Zen, Rahma Chaabouni, Mihaela Rosca, Jiepu Jiang, Charlie Chen, Ruibo Liu, Tara Sainath, Maxim Krikun, Alex Polozov, Jean-Baptiste Lespiau, Josh Newlan, Zeyncep Cankara, Soo Kwak, Yunhan Xu, Phil Chen, Andy Coenen, Clemens Meyer, Katerina Tsihlas, Ada Ma, Juraj Gottweis, Jinwei Xing, Chenjie Gu, Jin Miao, Christian Frank, Zeynep Cankara, Sanjay Ganapathy, Ishita Dasgupta, Steph Hughes-Fitt, Heng Chen, David Reid, Keran Rong, Hongmin Fan, Joost van Amersfoort, Vincent Zhuang, Aaron Cohen, Shixiang Shane Gu, Anhad Mohananey, Anastasija Ilic, Taylor Tobin, John Wieting, Anna Bortsova, Phoebe Thacker, Emma Wang, Emily Caveness, Justin Chiu, Eren Sezener, Alex Kaskasoli, Steven Baker, Katie Millican, Mohamed Elhawaty, Kostas Aisopos, Carl Lebsack, Nathan Byrd, Hanjun Dai, Wenhao Jia, Matthew Wiethoff, Elnaz Davoodi, Albert Weston, Lakshman Yagati, Arun Ahuja, Isabel Gao, Golan Pundak, Susan Zhang, Michael Azzam, Khe Chai Sim, Sergi Caelles, James Keeling, Abhanshu Sharma, Andy Swing, YaGuang Li, Chenxi Liu, Carrie Grimes Bostock, Yamini Bansal, Zachary Nado, Ankesh Anand, Josh Lipschultz, Abhijit Karmarkar, Lev Proleev, Abe Ittycheriah, Soheil Hassas Yeganeh, George Polovets, Aleksandra Faust, Jiao Sun, Alban Rrustemi, Pen Li, Rakesh Shivanna, Jeremiah Liu, Chris Welty, Federico Lebron, Anirudh Baddepudi, Sebastian Krause, Emilio Parisotto, Radu Soricut, Zheng Xu, Dawn Bloxwich, Melvin Johnson, Behnam Neyshabur, Justin Mao-Jones, Renshen Wang, Vinay Ramasesh, Zaheer Abbas, Arthur Guez, Constant Segal, Duc Dung Nguyen, James Svensson, Le Hou, Sarah York, Kieran Milan, Sophie Bridgers, Wiktor Gworek, Marco Tagliasacchi, James Lee-Thorp, Michael Chang, Alexey Guseynov, Ale Jakse Hartman, Michael Kwong, Ruizhe Zhao, Sheleem Kashem, Elizabeth Cole, Antoine Miech, Richard Tanburn, Mary Phuong, Filip Pavetic, Sebastien Cevey, Ramona Comanescu, Richard Ives, Sherry Yang, Cosmo Du, Bo Li, Zizhao Zhang, Mariko Iinuma, Clara Huiyi Hu, Aurko Roy, Shaan Bijwadia, Zhenkai Zhu, Danilo Martins, Rachel Saputro, Anita Gergely, Steven Zheng, Dawei Jia, Ioannis Antonoglou, Adam Sadovsky, Shane Gu, Yingying Bi, Alek Andreev, Sina Samangooei, Mina Khan, Tomas Kocisky, Angelos Filos, Chintu Kumar, Colton Bishop, Adams Yu, Sarah Hodkinson, Sid Mittal, Premal Shah, Alexandre Moufarek, Yong Cheng, Adam Bloniarz, Jaehoon Lee, Pedram Pejman, Paul Michel, Stephen Spencer, Vladimir Feinberg, Xuehan Xiong, Nikolay Savinov, Charlotte Smith, Siamak Shakeri, Dustin Tran, Mary Chesus, Bernd Bohnet, George Tucker, Tamara von Glehn, Carrie Muir, Yiran Mao, Hideto Kazawa, Ambrose Slone, Kedar Soparkar, Disha Shrivastava, James Cobon-Kerr, Michael Sharman, Jay Pavagadhi, Carlos Araya, Karolis Misiunas, Nimesh Ghelani, Michael Laskin, David Barker, Qiujia Li, Anton Briukhov, Neil Houlsby, Mia Glaese, Balaji Lakshminarayanan, Nathan Schucher, Yunhao Tang, Eli Collins, Hyeontaek Lim, Fangxiaoyu Feng, Adria Recasens, Guangda Lai, Alberto Magni, Nicola De Cao, Aditya Siddhant, Zoe Ashwood, Jordi Orbay, Mostafa Dehghani, Jenny Brennan, Yifan He, Kelvin Xu, Yang Gao, Carl Saroufim, James Molloy, Xinyi Wu, Seb Arnold, Solomon Chang, Julian Schrittwieser, Elena Buchatskaya, Soroush Radpour, Martin Polacek, Skye Giordano, Ankur Bapna, Simon Tokumine, Vincent Hellendoorn, Thibault Sottiaux, Sarah Cogan, Aliaksei Severyn, Mohammad Saleh, Shantanu Thakoor, Laurent Shefey, Siyuan Qiao, Meenu Gaba, Shuo yiin Chang, Craig Swanson, Biao Zhang, Benjamin Lee, Paul Kishan Rubenstein, Gan Song, Tom Kwiatkowski, Anna Koop, Ajay Kannan, David Kao, Parker Schuh, Axel Stjerngren, Golnaz Ghiasi, Gena Gibson, Luke Vilnis, Ye Yuan, Felipe Tiengo Ferreira, Aishwarya Kamath, Ted Klimenko, Ken Franko, Kefan Xiao, Indro Bhattacharya, Miteyan Patel, Rui Wang, Alex Morris, Robin Strudel, Vivek Sharma, Peter Choy, Sayed Hadi Hashemi, Jessica Landon, Mara Finkelstein, Priya Jhakra, Justin Frye, Megan Barnes, Matthew Mauger, Dennis Daun, Khuslen Baatarsukh, Matthew Tung, Wael Farhan, Henryk Michalewski, Fabio Viola, Felix de Chaumont Quitry, Charline Le Lan, Tom Hudson, Qingze Wang, Felix Fischer, Ivy Zheng, Elspeth White, Anca Dragan, Jean baptiste Alayrac, Eric Ni, Alexander Pritzel, Adam Iwanicki, Michael Isard, Anna Bulanova, Lukas Zilka, Ethan Dyer, Devendra Sachan, Srivatsan Srinivasan, Hannah Muckenhirn, Honglong Cai, Amol Mandhane, Mukarram Tariq, Jack W. Rae, Gary Wang, Kareem Ayoub, Nicholas FitzGerald, Yao Zhao, Woohyun Han, Chris Alberti, Dan Garrette, Kashyap Krishnakumar, Mai Gimenez, Anselm Levskaya, Daniel Sohn, Josip Matak, Inaki Iturrate, Michael B. Chang, Jackie Xiang, Yuan Cao, Nishant Ranka, Geoff Brown, Adrian Hutter, Vahab Mirrokni, Nanxin Chen, Kaisheng Yao, Zoltan Egyed, Francois Galilee, Tyler Liechty, Praveen Kallakuri, Evan Palmer, Sanjay Ghemawat, Jasmine Liu, David Tao, Chloe Thornton, Tim Green, Mimi Jasarevic, Sharon Lin, Victor Cotruta, Yi-Xuan Tan, Noah Fiedel, Hongkun Yu, Ed Chi, Alexander Neitz, Jens Heitkaemper, Anu Sinha, Denny Zhou, Yi Sun, Charbel Kaed, Brice Hulse, Swaroop Mishra, Maria Georgaki, Sneha Kudugunta, Clement Farabet, Izhak Shafran, Daniel Vlasic, Anton Tsitsulin, Rajagopal Ananthanarayanan, Alen Carin, Guolong Su, Pei Sun, Shashank V, Gabriel Carvajal, Josef Broder, Iulia Comsa, Alena Repina, William Wong, Warren Weilun Chen, Peter Hawkins, Egor Filonov, Lucia Loher, Christoph Hirnschall, Weiyi Wang, Jingchen Ye, Andrea Burns, Hardie Cate, Diana Gage Wright, Federico Piccinini, Lei Zhang, Chu-Cheng Lin, Ionel Gog, Yana Kulizhskaya, Ashwin Sreevatsa, Shuang Song, Luis C.

Cobo, Anand Iyer, Chetan Tekur, Guillermo Garrido, Zhuyun Xiao, Rupert Kemp, Huaixiu Steven Zheng, Hui Li, Ananth Agarwal, Christel Ngani, Kati Goshvadi, Rebeca Santamaria-Fernandez, Wojciech Fica, Xinyun Chen, Chris Gorgolewski, Sean Sun, Roopal Garg, Xinyu Ye, S. M. Ali Eslami, Nan Hua, Jon Simon, Pratik Joshi, Yelin Kim, Ian Tenney, Sahitya Potluri, Lam Nguyen Thiet, Quan Yuan, Florian Luisier, Alexandra Chronopoulou, Salvatore Scellato, Praveen Srinivasan, Minmin Chen, Vinod Koverkathu, Valentin Dalibard, Yaming Xu, Brennan Saeta, Keith Anderson, Thibault Sellam, Nick Fernando, Fantine Huot, Junehyuk Jung, Mani Varadarajan, Michael Quinn, Amit Raul, Maigo Le, Ruslan Habalov, Jon Clark, Komal Jalan, Kalesha Bullard, Achintya Singhal, Thang Luong, Boyu Wang, Sujeevan Rajayogam, Julian Eisenschlos, Johnson Jia, Daniel Finchelstein, Alex Yakubovich, Daniel Balle, Michael Fink, Sameer Agarwal, Jing Li, Dj Dvijotham, Shalini Pal, Kai Kang, Jaclyn Konzelmann, Jennifer Beattie, Olivier Dousse, Diane Wu, Remi Crocker, Chen Elkind, Siddhartha Reddy Jonnalagadda, Jong Lee, Dan Holtmann-Rice, Krystal Kallarackal, Rosanne Liu, Denis Vnukov, Neera Vats, Luca Invernizzi, Mohsen Jafari, Huanjie Zhou, Lilly Taylor, Jennifer Prendki, Marcus Wu, Tom Eccles, Tianqi Liu, Kavya Kopparapu, Francoise Beaufays, Christof Angermueller, Andreea Marzoca, Shourya Sarcar, Hilal Dib, Jeff Stanway, Frank Perbet, Nejc Trdin, Rachel Sterneck, Andrey Khorlin, Dinghua Li, Xihui Wu, Sonam Goenka, David Madras, Sasha Goldshtein, Willi Gierke, Tong Zhou, Yaxin Liu, Yannie Liang, Anais White, Yunjie Li, Shreya Singh, Sanaz Bahargam, Mark Epstein, Sujoy Basu, Li Lao, Adnan Ozturel, Carl Crous, Alex Zhai, Han Lu, Zora Tung, Neeraj Gaur, Alanna Walton, Lucas Dixon, Ming Zhang, Amir Globerson, Grant Uy, Andrew Bolt, Olivia Wiles, Milad Nasr, Ilia Shumailov, Marco Selvi, Francesco Piccinno, Ricardo Aguilar, Sara McCarthy, Misha Khalman, Mrinal Shukla, Vlado Galic, John Carpenter, Kevin Villela, Haibin Zhang, Harry Richardson, James Martens, Matko Bosnjak, Shreyas Rammohan Belle, Jeff Seibert, Mahmoud Alnahlawi, Brian McWilliams, Sankalp Singh, Annie Louis, Wen Ding, Dan Popovici, Lenin Simicich, Laura Knight, Pulkit Mehta, Nishesh Gupta, Chongyang Shi, Saaber Fatehi, Jovana Mitrovic, Alex Grills, Joseph Pagadora, Dessie Petrova, Danielle Eisenbud, Zhishuai Zhang, Damion Yates, Bhavishya Mittal, Nilesh Tripuraneni, Yannis Assael, Thomas Brovelli, Prateek Jain, Mihajlo Velimirovic, Canfer Akbulut, Jiaqi Mu, Wolfgang Macherey, Ravin Kumar, Jun Xu, Haroon Qureshi, Gheorghe Comanici, Jeremy Wiesner, Zhitao Gong, Anton Ruddock, Matthias Bauer, Nick Felt, Anirudh GP, Anurag Arnab, Dustin Zelle, Jonas Rothfuss, Bill Rosgen, Ashish Shenoy, Bryan Seybold, Xinjian Li, Jayaram Mudigonda, Goker Erdogan, Jiawei Xia, Jiri Simsa, Andrea Michi, Yi Yao, Christopher Yew, Steven Kan, Isaac Caswell, Carey Radebaugh, Andre Elisseeff, Pedro Valenzuela, Kay McKinney, Kim Paterson, Albert Cui, Eri Latorre-Chimoto, Solomon Kim, William Zeng, Ken Durden, Priya Ponnapalli, Tiberiu Sosea, Christopher A. Choquette-Choo, James Manyika, Brona Robenek, Harsha Vashisht, Sebastien Pereira, Hoi Lam, Marko Velic, Denese Owusu-Afriyie, Katherine Lee, Tolga Bolukbasi, Alicia Parrish, Shawn Lu, Jane Park, Balaji Venkatraman, Alice Talbert, Lambert Rosique, Yuchung Cheng, Andrei Sozanschi, Adam Paszke, Praveen Kumar, Jessica Austin, Lu Li, Khalid Salama, Wooyeol Kim, Nandita Dukkipati, Anthony Baryshnikov, Christos Kaplanis, Xiang-Hai Sheng, Yuri Chervonyi, Caglar Unlu, Diego de Las Casas, Harry Askham, Kathryn Tunyasuvunakool, Felix Gimeno, Siim Poder, Chester Kwak, Matt Miecnikowski, Vahab Mirrokni, Alek Dimitriev, Aaron Parisi, Dangyi Liu, Tomy Tsai, Toby Shevlane, Christina Kouridi, Drew Garmon, Adrian Goedeckemeyer, Adam R. Brown, Anitha Vijayakumar, Ali Elqursh, Sadegh Jazayeri, Jin Huang, Sara Mc Carthy, Jay Hoover, Lucy Kim, Sandeep Kumar, Wei Chen, Courtney Biles, Garrett Bingham, Evan Rosen, Lisa Wang, Qijun Tan, David Engel, Francesco Pongetti, Dario de Cesare, Dongseong Hwang, Lily Yu, Jennifer Pullman, Srini Narayanan, Kyle Levin, Siddharth Gopal, Megan Li, Asaf Aharoni, Trieu Trinh, Jessica Lo, Norman Casagrande, Roopali Vij, Loic Matthey, Bramandia Ramadhana, Austin Matthews, CJ Carey, Matthew Johnson, Kremena Goranova, Rohin Shah, Shereen Ashraf, Kingshuk Dasgupta, Rasmus Larsen, Yicheng Wang, Manish Reddy Vuyyuru, Chong Jiang, Joana Ijazi, Kazuki Osawa, Celine Smith, Ramya Sree Boppana, Taylan Bilal, Yuma Koizumi, Ying Xu, Yasemin Altun, Nir Shabat, Ben Bariach, Alex Korchemniy, Kiam Choo, Olaf Ronneberger, Chimezie Iwuanyanwu, Shubin Zhao, David Soergel, Cho-Jui Hsieh, Irene Cai, Shariq Iqbal, Martin Sundermeyer, Zhe Chen, Elie Bursztein, Chaitanya Malaviya, Fadi Biadsy, Prakash Shroff, Inderjit Dhillon, Tejasi Latkar, Chris Dyer, Hannah Forbes, Massimo Nicosia, Vitaly Nikolaev, Somer Greene, Marin Georgiev, Pidong Wang, Nina Martin, Hanie Sedghi, John Zhang, Praseem Banzal, Doug Fritz, Vikram Rao, Xuezhi Wang, Jiageng Zhang, Viorica Patraucean, Dayou Du, Igor Mordatch, Ivan Jurin, Lewis Liu, Ayush Dubey, Abhi Mohan, Janek Nowakowski, Vlad-Doru Ion, Nan Wei, Reiko Tojo, Maria Abi Raad, Drew A. Hudson, Vaishakh Keshava, Shubham Agrawal, Kevin Ramirez, Zhichun Wu, Hoang Nguyen, Ji Liu, Madhavi Sewak, Bryce Petrini, DongHyun Choi, Ivan Philips, Ziyue Wang, Ioana Bica, Ankush Garg, Jarek Wilkiewicz, Priyanka Agrawal, Xiaowei Li, Danhao Guo, Emily Xue, Naseer Shaik, Andrew Leach, Sadh MNM Khan, Julia Wiesinger, Sammy Jerome, Abhishek Chakladar, Alek Wenjiao Wang, Tina Ornduff, Folake Abu, Alireza Ghaffarkhah, Marcus Wainwright, Mario Cortes, Frederick Liu, Joshua Maynez, Andreas Terzis, Pouya Samangouei, Riham Mansour, Tomasz Kępa, François-Xavier Aubet, Anton Algymr, Dan Banica, Agoston Weisz, Andras Orban, Alexandre Senges, Ewa Andrejczuk, Mark Geller, Niccolo Dal Santo, Valentin Anklin, Majd Al Merey, Martin Baeuml, Trevor Strohman, Junwen Bai, Slav Petrov, Yonghui Wu, Demis Hassabis, Koray Kavukcuoglu, Jeffrey Dean, and Oriol Vinyals. 2024. Gemini 1.5: Unlocking multimodal understanding across millions of tokens of context.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2023. Attention is all you need.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander M. Rush. 2020. HuggingFace's Transformers: State-of-the-art natural language processing.

Wei Zhao, Michael Strube, and Steffen Eger. 2023. DiscoScore: Evaluating text generation with BERT and discourse coherence. In *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics*, pages 3865–3883, Dubrovnik, Croatia. Association for Computational Linguistics.

A LLM hyperparameters and prompts

A.1 Gemini 1.5 Flash

Prompts:

```
system_prompt = f"You are a helpful language
↪ tutor. Your student is only familiar with the
↪ following words, so please mainly use words
↪ from the following list: {vocab_set |
↪ {target_word}}"
user_prompt = f"Please write a sentence using the
↪ word \"{target_word}\", with enough context
↪ clues that someone can understand the meaning
↪ of the word. However, don't simply define the
↪ word. Respond with only the sentence, nothing
↪ else, no explanations."
```

Initialization hyperparameters:

```
llm = ChatGoogleGenerativeAI(
    model='gemini-1.5-flash',
    temperature=0.5,
    max_tokens=None,
    timeout=None,
    max_retries=2,
    google_api_key=GEMINI_API_KEY
)
```

A.2 Llama 3.1 8B Instruct

Main prompt:

```
user_prompt = f'Please write a sentence using the
↪ word "{target_word}", with enough context
↪ clues that someone can understand the meaning
↪ of the word. However, don\'t simply define
↪ the word. Respond with only the sentence,
↪ nothing else, no explanations.'
```

Alternate prompts (mentioned in Section 3.1):

```
vocab_with_definition_prompt = f"Please write a
↪ sentence using the word \"{target_word}\",
↪ with the definition of \"{definition}\" with
↪ enough context clues that someone can
↪ understand the meaning of the word. However,
↪ don't simply define the word. Respond with
↪ only the sentence, nothing else, no
↪ explanations."
masked_vocab_word_prompt = f"Pretend the word
↪ \"<vocab>\" has the definition
↪ \"{definition}\". Now write a sentence, using
↪ \"<vocab>\", with enough context clues for
↪ someone to understand the meaning of the word
↪ without directly using the definition.
↪ Respond with only the sentence, nothing else,
↪ no explanations."
constrained_beam_search_prompt = "Generate a
↪ sentence containing enough context clues for
↪ someone to understand the meaning of the
↪ sentence entirely. Respond with only the
↪ sentence, nothing else, no explanations."
```

Inference hyperparameters:

```
output = model.generate(
    **inputs,
    max_new_tokens=50,
    no_repeat_ngram_size=3,
    logits_processor=logits_processors,
    pad_token_id=tokenizer.eos_token_id,
)
```

B Human evaluation rubric