Controlled Language Generation for Language Learning Items

Kevin Stowe*, **Debanjan Ghosh***, **Mengxuan Zhao**
Educational Testing Service
{kstowe, dghosh}@ets.org, mzhao@etscanada.ca

Abstract

This work aims to employ natural language generation (NLG) to rapidly generate items for English language learning applications: this requires both language models capable of generating fluent, high-quality English, and to control the output of the generation to match the requirements of the relevant items. We experiment with deep pretrained models for this task, developing novel methods for controlling items for factors relevant in language learning: diverse sentences for different proficiency levels and argument structure to test grammar. Human evaluation demonstrates high grammatically scores for all models (3.4 and above out of 4), and higher length (24%) and complexity (9%) over the baseline for the advanced proficiency model. Our results show that we can achieve strong performance while adding additional control to ensure diverse, tailored content for individual users.[1]

1 Introduction

Recent advancement of transformer (Vaswani et al., 2017) based pre-trained language models (LM) (Lewis et al., 2020; Brown et al., 2020; Raffel et al., 2020) have resulted in unprecedented success in generating large amounts of fluent English text. One possible area where text generation can be applied is item generation for English language learning applications (LLAs). LLAs are popular apps used by millions of people all over the world.[2] These apps often include multiple choice items for vocabulary tests, flashcards, grammar lessons, and more. Typically, such items are created manually (Service, 2010) or curated from crowd-sourced sentence database, e.g., Tatoeba (Settles et al., 2020).[3]

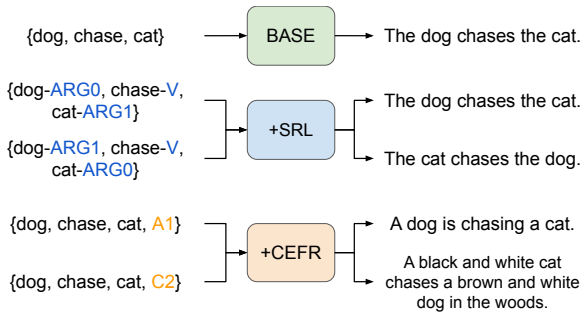

Figure 1: Controlling the concept2seq generation process, using semantic role labels and CEFR levels.

On the contrary, our goal is to make this process scalable by employing LMs, enabling developers of LLAs to be able to implement a much broader array of learning items quickly and efficiently.

This is accomplished by using a *concept2seq* framework: a sequence-to-sequence architecture in which, given a set of relevant *concepts*, we aim to generate sentences that minimally contain those concepts.[4] There is a wide body of work relating to this framework (Lin et al., 2020; Carlsson et al., 2022) to generate sentences: we experiment with a number of adaptations that are applicable to LLAs. First, the importance of providing diverse content based on student/user needs and different skill sets is well established in learning sciences (Morgan, 2014; Clarke and Miles, 2003). We generate sentences according to different skill sets by conditioning on the Common European Framework of Reference for Languages (CEFR) levels in English.[5] We use a document level CEFR predictor (Montgomorie, 2022) to predict the CEFR levels in the training data and in turn use the level in a controlled generation framework based on BART (Lewis et al., 2020) to generate different CEFR level-specific sentence (Section 3.1). Second, to

*Equal Contribution.

[1]Code and datasets made available at https://github.com/EducationalTestingService/concept-control-gen

[2]https://www.businessofapps.com/data/language-learning-app-market/

[3]https://tatoeba.org/en/

[4]Concepts are lemmatized tokens in text extracted by using the ConceptNet knowledge base (Speer et al., 2017).

[5]CEFR is an international standard for measuring user' ability within a language.

test the grammatical proficiency of the users we attach syntactic and semantic information in the form of semantic role labels (SRL) to the concept inputs for controlled generation where we can specify the semantic roles (e.g., ARG0) in the generation (Section 3.2). Such generations can be used in LLAs to ask grammar questions and further expand the diversity of items.

Consider the examples in Figure 1. The BASE model generates a sentence using the concepts "dog", "chase", and "cat". The CEFR model demonstrates two different generations for two skill levels: a simple short sentence (e.g. "a cat is chasing a dog") for the A1 beginner level and a long complex sentence (e.g., "a black and white ... in the woods) for the C2 proficiency level (Section 3.1). Likewise, the SRL model presents examples where we conditioned the generations on specific semantic roles (Section 3.2).

We evaluate our models using automatic metrics relevant to our goals (perplexity, concept coverage in the generated sentences, length, and lexical diversity), as well as utilizing Amazon Mechanical Turk (MTurk) to get human judgments of important factors: grammaticality, complexity, and semantic plausibility. The CEFR model generates less complex, shorter sentences than the baseline when targeting the A1 level (complexity score of 2.45, average length of 11.6) compared to the C2 level (complexity score of 2.73, average length of 15.3 words). The SRL model generates the targeted words in the correct argument slot significantly more than the baseline (improving from 6% to 32% based on the targeted role). All models are within 3% of the baseline in terms of grammaticality and semantic plausibility, indicating that we can effectively generate sentences from concepts while adding additional control.

2 Data

We employ two datasets for concept2seq generation. Each instance in the datasets is a set of concepts paired with a sentence that contains those concepts. First, we use the COMMONGEN data which is based on existing caption corpora (Lin et al., 2020). From this dataset we use 71,408 concept/sentence pairs. Although COMMONGEN is used in related concept2seq generation (Lin et al., 2020), since it is based on image captions, many samples are phrases (not sentences) and less diverse. Thus, we also collect another dataset based on fourteen relevant vocabulary items for language learning that belong to different CEFR levels.[6] Sentences are collected from diverse sources such as the ROCStories (Mostafazadeh et al., 2016), Tatoeba sentence database, and the Google book corpus.[7] We first retrieved sentences containing the vocabulary items from these different sources and then extracted the concepts using the ConceptNet knowledge-base by employing Becker et al. (2021). We extract noun, verbs, and adjective tokens as concepts and keep only those sentences containing 2-5 concepts to keep consistency with COMMONGEN. This dataset is denoted as the VOCABULARY dataset and consists of an additional 218,997 concept/sentence pairs. In total, the COMMONGEN and VOCABULARY gives us a dataset of 290,399 pairs of concepts and sentences.

To evaluate our generation models, we create two test sets to evaluate two different scenarios. Our goal is to evaluate concept sets that occur frequently in the training data, as well those that occur rarely. In order to build these two types of test sets, we first generate the frequency counts of each concept over the entire dataset. We then calculate the frequency of a given concept set as the sum of the frequency counts of each concept within that set. We then sample 500 instances from both the COMMONGEN and VOCABULARY datasets from the top 10% highest frequency concept sets and 500 each from the bottom 10% frequency sets. We split the test data this way to evaluate two likely use cases. The lowest 10% aligns with the case where we have unseen concepts and want to generate something novel; the most frequent 10% matches the everyday use case where we generate from things we've seen before. This gives us a test set of 2000 total sample, half from COMMONGEN and half from VOCABULARY, additionally split into high frequency and low frequency concepts. From the remaining dataset we randomly use 90% as training and 10% as validation.

3 Methods

Here, we present our computational approaches for sentence generation using the concept2seq framework. At its base form, our task is to generate a sentence s that consists of sequence of tokens, $s = \{s_1, ..., s_m\}$ using a list of concepts

[6] This vocabulary list and their word derivatives was recommended by the learning scientists of the LLA.

[7] https://www.english-corpora.org/googlebooks/

$c = \{c_1, ..., c_n\}$. Using the standard autoregressive sequence-to-sequence architecture (Sutskever et al., 2014) we model $P_\theta(s \mid c)$ as follows:

$$P_\theta(s \mid c) = \prod_i P_\theta(s_i \mid s_1, \ldots, s_{i-1}, c) \quad (1)$$

Note, the resulting sentence s should contain relevant vocabulary from the concept set c, but is otherwise unconstrained. We use a pretrained BART model (Lewis et al., 2020) composed of a bidirectional encoder and an autoregressive decoder. In our simplest setup (called BASE), the input for BART is a set of concepts c and the output text s is a sentence that contains those concepts. For the BASE model, we fine-tune the `bart-base` model on our training data described above, and then generate sentences based on the test concepts. Note, this is equivalent to the BART based experiment reported in Lin et al. (2020).[8]

3.1 CEFR-controlled generation

Research in language learning has shown that students' retention of words and texts increases when they encounter increasing diversity in content (Adelman et al., 2006; Frances et al., 2020). Since language learning is affected by many variables (Oxford and Nyikos, 1989), we focus on a specific variable - the CEFR levels - which are an international standard that measures text complexity and has strong correlations with learners' skill sets and language learning ability (Papageorgiou et al., 2015). To this end, we generate sentences guided towards different skill sets of users by conditioning on the CEFR levels. These labels are, in increasing order of proficiency: A1, A2, B1, B2, C1, and C2, where A1 denotes a beginner level and C2 denotes high proficiency. Our goal is to be able to start with a list of concepts and generate a sentence at the appropriate proficiency level. We first use a document level CEFR predictor (Montgomorie, 2022) to predict the CEFR levels for each sentence in the training data. This tagger, which functions by combining lexical, syntactic, and other attested proficiency features, provides a tag from A1 to C2 for each sentence in the training dataset. In turn, we use this predicted CEFR level as control codes to guide sentence generation.

[8] Although Lin et al. (2020) also reported experimental results using other transformer-based LMs such as GPT-2 (Radford et al., 2019) and T5 (Raffel et al., 2020), we notice BART performs better on several metrics, so we continue to use BART.

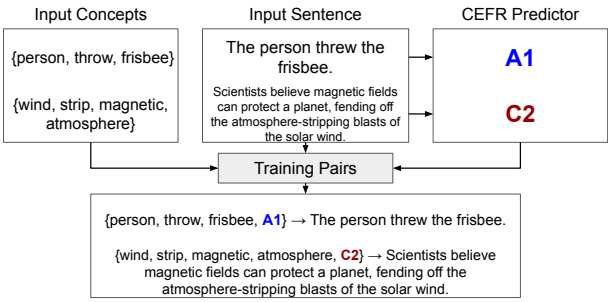

Figure 2: Method for applying CEFR labels to input concepts using the CEFR scorer.

Controlled generation models (Kikuchi et al., 2016; Hu et al., 2017; Ficler and Goldberg, 2017; Tsai et al., 2021) condition on a control code f in addition to the input c to model the distribution of $P_\theta(s \mid c, f)$. Similar to Eq. (1), we can write,

$$P_\theta(s \mid c, f) = \prod_i P_\theta(s_i \mid s_1, \ldots, s_{i-1}, c, f) \quad (2)$$

Text generation conditioned on such control codes, such as sentiment control of movie reviews, style for chatbots, diverse story continuations, question generation etc., have been used effectively in recent research (Tu et al., 2019; Krause et al., 2021; Roller et al., 2021; Gao et al., 2022). We use the same idea for sentence generation by conditioning on the CEFR levels. Figure 2 shows the overview of the process.

3.2 Argument Structure-controlled generation

As the second task, we use the argument structure of the generated sentences as the control code. We determine for any given concept in the input what semantic role that concept should play in the output text. This gives two key advantages: we can ensure the semantic viability of generations in which the concepts make more sense in particular roles, and we can extend the variety of sentences generated by varying the semantic roles of the arguments. Consider the following generation:

$$\{\text{dog, chase, cat}\} \rightarrow \begin{matrix}\text{(a) the dog chased the cat} \\ \text{(b) the cat chased the dog}\end{matrix}$$

As the concepts are unordered, the model can generate both sentences where (a) the dog is chasing the cat and where (b) the cat is chasing the dog. Stereotypically, we would expect (a), but (b) is a viable reading. By enforcing the semantic roles of the concepts, either with the dog or the cat as the agent, our aim is to be able to more concretely

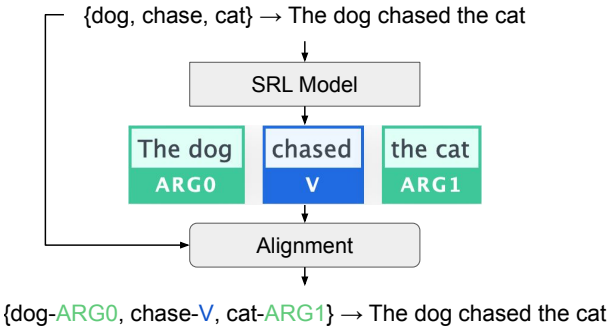

Figure 3: Method for applying SRL parse to the original concepts during training.

choose which output we'd like to see. This in turn can be utilized to check grammar skills of the users by follow-up questions in an LLA (e.g., which is the agent in the sentence?).

To identify the semantic roles, we tag the training dataset with an automatic semantic role labeling system (Stanovsky et al., 2018).[9] For each verb in the input, the system tags each word in the sentence with the argument it takes in that verb's scope. In order to convert these tags to control codes, we first extract each word in the sentence that matches a lemmatized version of the one of the input concepts. For each of these words, we identify all of the possible roles it can play in a sentence (note that words can take multiple roles, when they are arguments of separate verbs). We then iterate through these options, aligning the possible semantic role labels that word can take to the concepts in the input. This yields a new batch of concepts labeled with semantic role information that serve as the inputs for the given sentence. An overview of this process is shown in Figure 3.

Inference For CEFR-controlled generation, we have trained a single model on the full scope of CEFR tagged data: in order to generate sentences at a particularly level, we need to provide that level to the model at inference time. For this, we experiment with the simplest level (A1) and the most advanced level (C2). We add these labels to the concept inputs to generate sentences that should match those levels: these setups are dubbed CEFR-A1 and CEFR-C2. provided different CEFR levels to generate different sentences. For argument structure-controlled generation, we run our SRL tagger on the test data and then apply these to the input sources to generate a SRL-controlled output, as was done in training (Figure 3). We generate using top-K sampling ($k = 50$) with a maximum length of 64 and a length penalty of 1.0.

[9] https://github.com/allenai/allennlp.

3.3 Additional pretraining

The above BART model is originally trained with text as both the input and output. Our task is somewhat different, as the input consists of concepts. While these concepts are superficially a set of keywords, this still differs from what the original BART encoders have seen during training. In order to encourage the model to better handle this concept2seq data formulation, we leverage the power of additional pretraining, which has been shown to further improve model performance on new tasks (Gururangan et al., 2020).

We perform additional pretraining using wikipedia. Starting with a dump of wikipedia data,[10] we first extract sentences which are then run through our concept extraction pipeline (Section 2). We filter down to 10M random concept-sentence pairs with 2-5 concepts/sentence. These 10M pairs are then used to continue training on top of the pretrained BASE model (the WIKI model).[11]

4 Results

Evaluating natural language generation tasks can be difficult, and some automatic metrics can be problematic (Reiter, 2018). To overcome these difficulties, we use metrics specifically tailored to our task, as well as performing manual evaluation to get a concrete understanding of model performance.

4.1 Automatic Evaluation

Standard metrics, e.g., BLEU (Papineni et al., 2002) or ROUGE-L (Lin, 2004) that are often used to evaluate NLG outputs require true reference sentences for evaluation purpose. These methods are insufficient for our approach – our goal here is to generate sentences containing particular concepts conditioned on specific controls (e.g. CEFR) – and the resulting outputs do not need to match any particular gold standard. For that reason, we employ the following reference-free metrics for evaluation.

First, **perplexity** under a language model can indicate the fluency of the text. We report average perplexity per word using the GPT-2-base LM (Radford et al., 2019) in the generated sentences. **Coverage** indicates whether the generated

[10] https://dumps.wikimedia.org/enwiki/latest/

[11] Full details of the training procedures are in Appendix A.3.

Model	Perplexity	Coverage (All)	Coverage (Any)	Length	Diversity
BASE	4.51	50.55	93.34	12.13	43.34
SRL	4.53	51.40	94.10	11.98	43.51
CEFR (A1)	4.52	49.16	92.70	11.58	43.95
(C2)	4.41	39.70	74.61	15.26	37.31
WIKI	4.58	51.51	94.10	12.28	42.88

Table 1: Automatic evaluation of generation models. Lower scores indicate better **Perplexity** and **Diversity**.

Model	V	ARG0	ARG1	ARGM
BASE	88.18	70.48	73.34	58.67
SRL	94.06	88.93	88.01	77.60

Table 2: Automatic evaluation of SRL coverage. Scores refer to percentage of times the label occurred with a given concept in the output over the input.

sentences contains the input concepts. We evaluate the percentage of generations that contain lemmas matching the input concepts in two ways: first, the percentage of outputs containing **any** lemmas matching the input as well as the percentage of those where **all** of the concepts are found in the output. We also measure average **length** of the generated sentences (in number of words). Finally, we measure lexical **diversity**: for this, we use the average tf-idf score of all non-stopwords in the sentence (learned from a recent Wikipedia dump): higher scores indicate more common words, while lower scores indicate more lexical diversity. Relevant results are shown in Table 1.

We note a number of observations from automatic metrics: perplexity remains relatively stable across models, indicating they all can produce fluent sentences. CEFR C2 has the lowest perplexity, indicating that BART can produce complex but still fluent sentences. Coverage, length, and diversity remain relatively stable across models as well. One exception is the CEFR C2 model, which has lower coverage (39.70) and higher length (15.26 words per sequence). Since C2 sentences in the training dataset are longer (averaging 25.1 words per sentence, compared to the overall average of 17.0), it is expected that CEFR C2 model produce longer sentences when higher proficiency is required. Likewise, the CEFR A1 model tends towards shorter sentences (11.58 words per sequence). Finally, the low diversity score of the CEFR C2 model indicate the complex sentences generated by the model have higher lexical diversity.

SRL Overlap Evaluation: Note that for the SRL model, we evaluate the test data with a single set of SRL labels generated from the original SRL model. There are many ways to apply SRL labels to a given set of concepts, and we only evaluate against a single reference.

We use an automatic parser to capture whether the SRL-based inputs are accurately represented in the outputs. For the four most frequent argument types (ARG0, ARG1, ARGM, and V), we evaluate accuracy by comparing its presence in the generated output to its presence in the control codes. We measure the percentage of times the argument type is correctly represented by a concept in the generated sentence over the number of times the instructions indicate it should be. We compare the SRL to the BASE model in Table 2.

We might expect a large, pretrained model like BART to automatically generate the concepts into their expected roles, but we can see that the BASE model actually fluctuates greatly: for non-verbal arguments, it generates them in the semantic role of the reference sentence less than 75% of the time. This isn't necessarily a problem, as the system still is generating sentences with the appropriate concepts, but it highlights the usefulness of argument control: using the SRL model, we can generate concepts into specific semantic roles much more consistently, with scores ranging from 77% to 94%, thus, improving by a large margin over the BASE model.

4.2 Human Evaluation

For human evaluation, we aim to capture three essential criteria that are important for test item generation. These are:

1. **Grammaticality**: The generated sentences/phrases should be grammatical, and should follow normal English syntax.
2. **Complexity**: The complexity of a given sentence as it relates to end users.
3. **Plausibility**: The generated items should describe semantically plausible scenarios, or they

Model	Grammatical		Complex		Plausible	
	Common	Rare	Common	Rare	Common	Rare
BASE	3.50	3.40	2.41	2.58	3.60	3.61
SRL	3.48	3.46	*2.25*	*2.45*	3.58	3.61
CEFR (A1)	3.50	3.44	2.40	2.50	3.63	3.61
(C2)	*3.31*	3.41	**2.67**	**2.78**	3.57	3.61
WIKI	3.44	3.42	2.46	2.60	3.56	3.59

Table 3: MTurk evaluation for each generation model, evaluated for grammaticality, complexity, and plausibility. *Italic* scores are significantly below the baseline; **bold** are significantly above (two-tailed t-test, $p < .01$).

Input Concepts	Model	Sentence	Gram	Comp	Plaus
{competition, agree, strange, participate}	BASE	I agree with you that it's strange to participate in a competition with your dog.	4	3	4
	SRL	They agreed to participate in the strange competition.	4	2	4
	CEFR (A1)	I agreed to participate in the strange competition.	4	2.67	3.67
	CEFR (C2)	In some cases, it might be safer to think of different words to use if you don't want to participate in a strange competition.	4	3	3.67
	WIKI	In a strange twist of luck, i'd like to participate in a competition. I don't agree with you.	2	3.33	2.33
{girl, clothes}	BASE	A girl in black clothes is standing on a ladder and drawing on a wall.	4	2.33	4
	SRL	A girl in black clothes is playing soccer.	4	2.33	4
	CEFR (A1)	A girl in black clothes is playing with a toy lawn mower.	4	1.67	4
	CEFR (C2)	Young people sit in plastic chairs arranged around a set of stairs in a covered concrete area, wearing swim clothes, resting, and waiting.	4	3	4
	WIKI	A girl in black clothes playing soccer.	3.33	2	4

Table 4: Examples of generated sentences via different models with their annotated scores.

risk confusing or even misinforming the user.

We evaluated the generated outputs across these three criteria using Amazon Mechanical Turk (MTurk). We evaluated 800 sentences generated from each model. Three crowd-annotators were employed for each task and were asked to evaluate each sentence on a four-point scale for each criteria. We included many examples in the instructions. Each Human Intelligence Task (HIT) contained ten sentences to judge and we paid $2 per HIT. We obtain three scores for each of the above criteria for each generated sentence, and take their mean as the final score (Table 3).

We observe a number of key take-aways from the human evaluations. First, the rare concept sets are more likely to yield more complex generations, but otherwise they are fairly similar to the common sets: they exhibit similar grammaticality and plausibility scores. All models score strongly for grammaticality: the CEFR C2 model is lowest, as it is attempting to generate more complex sentences and likely to make more mistakes, but all models average about 3.4. Second, with regard to complexity, the CEFR A1 model scores lower than the BASE while the CEFR C2 model scores higher: this is our expected result, as the lower A1 level instruction yields simpler sentences, while the higher level C2 yields more complex sentences. Third, all models perform similarly with regard to plausibility, with every model being within .04 of the baseline. Finally, we see that additional pretraining doesn't improve performance significantly over the baseline: the BART-base model seems perfectly capable of adapting to concept2seq instructions without additional pretraining.

Table 4 presents two examples from our models along with average human ratings for all three aspects. In general, CEFR C2 has produced long sentences with high complexity for all examples. Likewise, grammaticality and plausibility scores are almost perfect except one example from WIKI.

In general, the methods we implemented to allow for additional control (CEFR and SRL) function as expected: we can manipulate the proficiency and argument structure of the generated sentences to a significant degree, allowing us to develop diverse content for users at different levels for LLAs.

4.3 Performance Time

The model was trained on a single nVidia K80 GPU for approximately 158 minutes, at approximately 81 training samples processed per second. We are then able to generate approximately 54 sentences per second at inference time. While this makes the system capable in some regards of generating live learning items, this is not desirable nor is it our use case. There are substantial risks involved in generating items live and presenting them to users, including possible grammatical and semantic disfluencies, unsuitable content, and biases inherent in generation from language models (Sheng et al., 2021). Rather, this system is designed to be run offline, generating a batch of possible learning items that can then be curated by experts.

5 Related Work

The concept2seq generation problem has been investigated in several recent studies. Lin et al. (2020) released the COMMONGEN dataset and generated sentences using various transformer models, (Carlsson et al., 2022) have proposed prompting for generation, and (Zhou et al., 2020) have conducted instruction tuning for generation using concepts. Our work is related to the above and our novelty is that we utilize this framework to generate LLA items. Although we did not experiment with the ordering of concepts similar to (Zhao et al., 2022), our SRL based generation in fact implicitly control the order of the concepts by offering specific grammar roles.

In prior work on controllable generation, embedding vectors of the control variables were fed into the model to control the output (Kikuchi et al., 2016; Fan et al., 2018), whereas our approach resembles recent efforts where the control variable is concatenated to the main input (Keskar et al., 2019) to control particular style, such as sentiment, style for chatbots, diverse story continuations and argument generation (Tu et al., 2019; Schiller et al., 2021; Krause et al., 2021; Roller et al., 2021).

6 Conclusion and Future Work

We proposed a type-controlled sentence generation framework for LLAs. We generate sentences (a) conditioned on the CEFR levels to provide content for users/students who belong to different skill sets (e.g., beginner or proficient in English), and (b) conditioned with specific argument structures for grammar. In automatic evaluation, the SRL model shows better coverage of input concepts than BASE, whereas human evaluation demonstrates high grammatically scores (3.4 and above) for all the models as well as high complexity for the CEFR C2 model that was designed to generate complex sentences for proficient users. In future, we want to continue a couple of error analyses on the input as well as on the generated sentences. Having taken into account that input data is pre-processed in several ways (e.g., concept extraction (Becker et al., 2021) and analysis of semantic roles (Stanovsky et al., 2018)), we want to select a small subset of data to determine whether such extraction has any error. Likewise, we also want to employ expert content developers to analyze the results of the CEFR predictor. Finally, we plan to employ additional controls such as word senses to guide context specific generations.

Acknowledgments

Thanks to Casey Medlock Paul and Kristen Herrick for suggesting reference materials on learning science, as well as Swapna Somasundaran for helpful comments.

7 Ethical Considerations

We leverage the freely available COMMONGEN dataset for model training. Though we have not exhaustively checked the dataset, given COMMONGEN is based on a variety of caption datasets, we consider them relatively safe and do not find any objectionable content. Likewise, we create another dataset, VOCABULARY, which is based on standard narratives and sentence databases that are used in many recent work. Training is done using large pretrained models that have been shown to have bias. Although the generated content do not appear biased, they may hallucinate content, which is a common problem for neural generation models. In future work, we plan to analyze and identify hallucinations from the generations, and assess possible bias issues within these generations.

Finally, we obtained institutional review board permission to conduct MTurk based evaluations to collect judgments from crowd workers regarding the quality of the sentences.

References

James S Adelman, Gordon DA Brown, and José F Quesada. 2006. Contextual diversity, not word frequency,

determines word-naming and lexical decision times. *Psychological science*, 17(9):814–823.

Maria Becker, Katharina Korfhage, and Anette Frank. 2021. COCO-EX: A tool for linking concepts from texts to ConceptNet. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: System Demonstrations*, pages 119–126, Online. Association for Computational Linguistics.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901.

Fredrik Carlsson, Joey Öhman, Fangyu Liu, Severine Verlinden, Joakim Nivre, and Magnus Sahlgren. 2022. Fine-grained controllable text generation using non-residual prompting. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 6837–6857, Dublin, Ireland. Association for Computational Linguistics.

John Clarke and Sherri Miles. 2003. Changing systems to personalize learning: Introduction to the personalization workshops. Technical report, The Education Alliance at Brown University.

Angela Fan, David Grangier, and Michael Auli. 2018. Controllable abstractive summarization. In *Proceedings of the 2nd Workshop on Neural Machine Translation and Generation*, pages 45–54, Melbourne, Australia. Association for Computational Linguistics.

Jessica Ficler and Yoav Goldberg. 2017. Controlling linguistic style aspects in neural language generation. In *Proceedings of the Workshop on Stylistic Variation*, pages 94–104, Copenhagen, Denmark. Association for Computational Linguistics.

Candice Frances, Clara D Martin, and Jon Andoni Duñabeitia. 2020. The effects of contextual diversity on incidental vocabulary learning in the native and a foreign language. *Scientific reports*, 10(1):1–11.

Lingyu Gao, Debanjan Ghosh, and Kevin Gimpel. 2022. "what makes a question inquisitive?" a study on type-controlled inquisitive question generation. In *Proceedings of the 11th Joint Conference on Lexical and Computational Semantics*, pages 240–257, Seattle, Washington. Association for Computational Linguistics.

Suchin Gururangan, Ana Marasović, Swabha Swayamdipta, Kyle Lo, Iz Beltagy, Doug Downey, and Noah A. Smith. 2020. Don't stop pretraining: Adapt language models to domains and tasks. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8342–8360, Online. Association for Computational Linguistics.

Zhiting Hu, Zichao Yang, Xiaodan Liang, Ruslan Salakhutdinov, and Eric P. Xing. 2017. Toward controlled generation of text. In *Proceedings of the 34th International Conference on Machine Learning, ICML 2017, Sydney, NSW, Australia, 6-11 August 2017*, volume 70 of *Proceedings of Machine Learning Research*, pages 1587–1596. PMLR.

Nitish Shirish Keskar, Bryan McCann, Lav R. Varshney, Caiming Xiong, and Richard Socher. 2019. CTRL: A conditional transformer language model for controllable generation. https://arxiv.org/abs/1909.05858. ArXiv:1909.05858.

Yuta Kikuchi, Graham Neubig, Ryohei Sasano, Hiroya Takamura, and Manabu Okumura. 2016. Controlling output length in neural encoder-decoders. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1328–1338, Austin, Texas. Association for Computational Linguistics.

Ben Krause, Akhilesh Deepak Gotmare, Bryan McCann, Nitish Shirish Keskar, Shafiq Joty, Richard Socher, and Nazneen Fatema Rajani. 2021. GeDi: Generative discriminator guided sequence generation. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 4929–4952, Punta Cana, Dominican Republic. Association for Computational Linguistics.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020. BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7871–7880, Online. Association for Computational Linguistics.

Bill Yuchen Lin, Wangchunshu Zhou, Ming Shen, Pei Zhou, Chandra Bhagavatula, Yejin Choi, and Xiang Ren. 2020. CommonGen: A constrained text generation challenge for generative commonsense reasoning. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 1823–1840, Online. Association for Computational Linguistics.

Chin-Yew Lin. 2004. ROUGE: A package for automatic evaluation of summaries. In *Text Summarization Branches Out*, pages 74–81, Barcelona, Spain. Association for Computational Linguistics.

Adam Montgomorie. 2022. CEFR english level predictor. https://github.com/AMontgomerie/CEFR-English-Level-Predictor.

Hani Morgan. 2014. Maximizing student success with differentiated learning. *The Clearing House: A Journal of Educational Strategies, Issues and Ideas*, 87(1):34–38.

Nasrin Mostafazadeh, Nathanael Chambers, Xiaodong He, Devi Parikh, Dhruv Batra, Lucy Vanderwende,

Pushmeet Kohli, and James Allen. 2016. A corpus and cloze evaluation for deeper understanding of commonsense stories. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 839–849.

Rebecca Oxford and Martha Nyikos. 1989. Variables affecting choice of language learning strategies by university students. *The modern language journal*, 73(3):291–300.

Spiros Papageorgiou, Richard J Tannenbaum, Brent Bridgeman, and Yeonsuk Cho. 2015. The association between toefl ibt® test scores and the common european framework of reference (cefr) levels. *Research Memorandum No. RM-15-06). Princeton, NJ: Educational Testing Service*.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 311–318, Philadelphia, Pennsylvania, USA. Association for Computational Linguistics.

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

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, 21:1–67.

Ehud Reiter. 2018. A structured review of the validity of bleu. *Computational Linguistics*, 44(3):393–401.

Stephen Roller, Emily Dinan, Naman Goyal, Da Ju, Mary Williamson, Yinhan Liu, Jing Xu, Myle Ott, Eric Michael Smith, Y-Lan Boureau, and Jason Weston. 2021. Recipes for building an open-domain chatbot. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume, EACL 2021, Online, April 19 - 23, 2021*, pages 300–325. Association for Computational Linguistics.

Benjamin Schiller, Johannes Daxenberger, and Iryna Gurevych. 2021. Aspect-controlled neural argument generation. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 380–396, Online. Association for Computational Linguistics.

Educational Testing Service. 2010. Toefl ibt® test framework and test development. *TOEFL iBT Research Insight*, 1.

Burr Settles, Geoffrey T LaFlair, and Masato Hagiwara. 2020. Machine learning–driven language assessment. *Transactions of the Association for computational Linguistics*, 8:247–263.

Emily Sheng, Kai-Wei Chang, Prem Natarajan, and Nanyun Peng. 2021. Societal biases in language generation: Progress and challenges. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 4275–4293, Online. Association for Computational Linguistics.

Robyn Speer, Joshua Chin, and Catherine Havasi. 2017. Conceptnet 5.5: An open multilingual graph of general knowledge. In *Thirty-first AAAI conference on artificial intelligence*, pages 4444–4451.

Gabriel Stanovsky, Julian Michael, Luke Zettlemoyer, and Ido Dagan. 2018. Supervised open information extraction. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 885–895, New Orleans, Louisiana. Association for Computational Linguistics.

Ilya Sutskever, Oriol Vinyals, and Quoc V. Le. 2014. Sequence to sequence learning with neural networks. In *Advances in Neural Information Processing Systems 27: Annual Conference on Neural Information Processing Systems 2014, December 8-13 2014, Montreal, Quebec, Canada*, pages 3104–3112.

Alicia Y. Tsai, Shereen Oraby, Vittorio Perera, Jiun-Yu Kao, Yuheng Du, Anjali Narayan-Chen, Tagyoung Chung, and Dilek Hakkani-Tür. 2021. Style control for schema-guided natural language generation. https://arxiv.org/abs/2109.12211. ArXiv:2109.12211.

Lifu Tu, Xiaoan Ding, Dong Yu, and Kevin Gimpel. 2019. Generating diverse story continuations with controllable semantics. In *Proceedings of the 3rd Workshop on Neural Generation and Translation*, pages 44–58, Hong Kong. Association for Computational Linguistics.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, pages 5998–6008.

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. Transformers: State-of-the-art natural language processing. In

Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations, pages 38–45, Online. Association for Computational Linguistics.

Chao Zhao, Faeze Brahman, Tenghao Huang, and Snigdha Chaturvedi. 2022. Revisiting generative commonsense reasoning: A pre-ordering approach. In *Findings of the Association for Computational Linguistics: NAACL 2022*, pages 1709–1718, Seattle, United States. Association for Computational Linguistics.

Wangchunshu Zhou, Dong-Ho Lee, Ravi Kiran Selvam, Seyeon Lee, Bill Yuchen Lin, and Xiang Ren. 2020. Pre-training text-to-text transformers for concept-centric common sense. https://arxiv.org/abs/2011.07956. ArXiv:2011.07956.

A Appendix

A.1 Vocabulary items

The following fourteen words in Table 5 (see their associated CEFR levels) were suggested by the learning scientists/content developers of the LLA we are involved with.

Word	CEFR
Clothes	A1
Famous	A1
Electric	A2
Return	A2
Lose	B1
Delicious	B1
Entertainment	B1
Literature	B1
Atmosphere	B2
Participate	B2
Awkward	B2
Solar	B2
Devote	B2
Caution	C1

Table 5: Words that are used to generate the VOCABULARY dataset (with their CEFR levels).

A.2 Concept Extraction

We extracted concepts using the concept extractor tool CoCo-Ex (Becker et al., 2021). The tool first parse sentences using standard parsers and then match tokens as found in ConceptNet knowledge base. The resulted concepts are categorized to their parts-of-speech. For our work we use nouns, verbs, and adjective tokens.

A.3 Model Training

We train our models using the HuggingFace platform (Wolf et al., 2020). We use the `bart-base` model as the initial checkpoint from the HuggingFace repository (Wolf et al., 2020). Each model is trained for 5 epochs with a batch size of 32 and a learning rate is 5e-5, as these parameters yielded the best performance on the validation set. For the CEFR generation, the label is added as an additional concept. For the SRL-based generation, the labels are concatenated to individual concepts. For the additional pretraining with the Wikipedia data, we ran pretraining for 3 epochs. All experiments were conducted using NVIDIA K-80 GPUs.

A.4 MTurk Experiments

In order to collect human evaluation for generated sentences, we deployed our data collection pipeline using AWS infrastructures. After reading and confirming the consent page, Turkers are directed to the survey interface where detailed instructions and survey questions are presented (shown in Figure 4). Turkers must complete all questions to be able to submit. We initially began evaluation using 5 samples per HIT, but extended this to 10 as the time necessary for annotators to complete a HIT was extremely short. A "survey code" is returned to the Turker as the submission is successful, and with the code the Turker can submit the HIT and to be qualified for payment. All evaluation results are saved into a DynamoDB table through API Gateway and Lambda.

To ensure the quality of evaluation, we recruited Turkers with more than 10,000 approved HITs and the approval rate or 97% and above. One task is required to be completed within 60 minutes, and most Turkers were able to submit within 10 to 15 minutes.

A.5 Human Evaluation

Table 6 presents more examples of MTurk annotations.

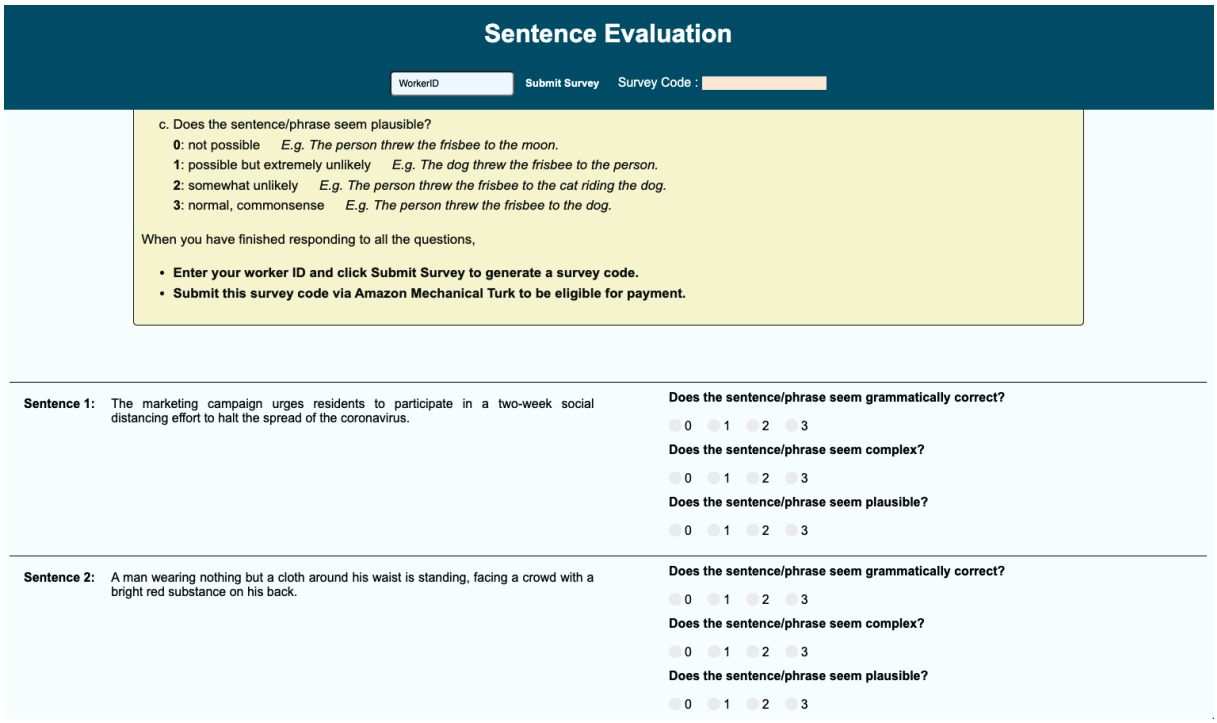

Figure 4: The survey interface where Turkers can read detailed instructions and complete the evaluation of ten sentences.

Input Concepts	Model	Sentence	Gram	Comp	Plaus
{pace, author, contemporary, literature}	BASE	I'm not interested in literature	4	1.33	4
	SRL	I'm not interested in literature.	4	1.33	4
	CEFR (A1)	I'm not very interested in literature.	4	1.67	4
	CEFR (C2)	Although many european researchers have studied ancient persian literature in the nineteenth century, the new world is not paying attention to our contemporary literature.	3	3.33	2.67
	WIKI	He is very interested in literature.	4	2.33	4
{competition, agree, strange, participate}	BASE	I agree with you that it's strange to participate in a competition with your dog.	4	3	4
	SRL	They agreed to participate in the strange competition.	4	2	4
	CEFR (A1)	I agreed to participate in the strange competition.	4	2.67	3.67
	CEFR (C2)	In some cases, it might be safer to think of different words to use if you don't want to participate in a strange competition.	4	3	3.67
	WIKI	In a strange twist of luck, i'd like to participate in a competition. I don't agree with you.	2	3.33	2.33
{girl, clothes}	BASE	A girl in black clothes is standing on a ladder and drawing on a wall.	4	2.33	4
	SRL	A girl in black clothes is playing soccer.	4	2.33	4
	CEFR (A1)	A girl in black clothes is playing with a toy lawn mower.	4	1.67	4
	CEFR (C2)	Young people sit in plastic chairs arranged around a set of stairs in a covered concrete area, wearing swim clothes, resting, and waiting.	4	3	4
	WIKI	A girl in black clothes playing soccer.	3.33	2	4
{intersection, people, clothing}	BASE	A group of people in dark clothing are crossing an intersection in a city.	4	2	4
	SRL	A group of people in dark clothing are corssing an intersection in a city.	4	2	4
	CEFR (A1)	A group of people in heavy clothing congregate near an intersection.	3.667	2	4
	CEFR (C2)	A motorcycle is standing near a decaying building wall, window, and close-line with several clothing items hanging.	3.667	3.333	3.667
	WIKI	A group of people in reflective clothing is crossing an intersection together.	3.667	3	3.667
{stand, rock, water, body}	BASE	The man stands on the rocks near the body of water.	3.333	2.667	4
	SRL	A man stands on a rock near a body of water.	4	2.333	4
	CEFR (A1)	A body of water standing on rocks.	3.333	2	2
	CEFR (C2)	Young man standing on rocks near body of water in the mountains.	3.333	2.667	4
	WIKI	A man stands on a rock near a body of water.	4	2.333	4
{sand, beach, group, sit, large, clothing}	BASE	A man in black clothing and a black hat is walking in front of a yellow wall.	4	2.667	4
	SRL	A man in black clothing walking down a sidewalk.	3	2	4
	CEFR (A1)	A man in very torn, dirty clothing is walking down a city street in front of a well dressed woman.	4	3.333	4
	CEFR (C2)	A young man in colorful clothing and a helmet is walking down the street in front of a well dressed woman and young girl.	4	2.667	4
	WIKI	A woman wearing heavy makeup and a pink jacket is walking past a clothing shop at night.	4	3	4
{kosovo, game, participate}	BASE	In kosovo, they participate in a game of hopscotch.	3	2.333	4
	SRL	Kosovo will participate in the olympic games.	3.667	1.667	3.667
	CEFR (A1)	In 2006, the olympic games were cancelled after kosovo decided to participate in the oclolympic game.	3.667	2.667	1
	CEFR (C2)	The olympic athlete participates in his first match as a professional athlete in the olympics.	3.333	2	4
	WIKI	I will participate in the kosovo olympic games.	3.667	1.667	3.667

Table 6: Examples of generated sentences via different models with their annotated scores.