Generating Scientific Definitions with Controllable Complexity

Anonymous ACL submission

Abstract

Unfamiliar terminology and complex language can present barriers to understanding science. Natural language processing stands to help address these issues by automatically defining unfamiliar terms. We introduce a new task and dataset for defining scientific terms and *controlling* the complexity of generated definitions as a way of adapting to a specific reader's background knowledge. We test four definition generation methods for this new task, finding that a sequence-to-sequence approach is most successful. We then explore the version of the task in which definitions are generated at a target complexity level. We introduce a novel reranking approach and find in human evaluations that it offers superior fluency while also controlling complexity, compared to several controllable generation baselines.

1 Introduction

Unfamiliar concepts and complex language can make understanding scientific information difficult for readers (Brossard and Shanahan, 2006; Shea, 2015; Martínez and Mammola, 2021), especially because understanding such terms is highly dependent on their domain knowledge. Given the wide variation in such knowledge, providing a one-size-fits-all definition may not be sufficiently understandable for all readers.

We envision a software tool designed to aid readers with varying domain knowledge by automatically defining scientific terms. Such a tool would afford readers control over generated definitions, including their complexity. This hypothetical system motivates research on automated generation of scientific definitions and generation-time control of definition complexity.

Prior work in generating definitions and personalizing generations to a reader falls short of these goals. Most definition generation has focused on common, general-usage words in English (Noraset et al., 2017; Balachandran et al., 2018); however, these approaches and models may not be suitable for generating scientific definitions (Beltagy et al., 2019). Scientific terms rarely reach common usage (Shea, 2015; Britt et al., 2014) and the contexts in which their definitions might appear (e.g., a research paper) are often much more complex than general-purpose resources for definitions (e.g., dictionaries or standard word embeddings). Previous methods focused on reader personalization have aimed at generating based on a reader's prior knowledge and interests (Acharya et al., 2018; Murthy et al., 2021). These approaches work well when models can leverage a reader profile (Murthy et al., 2021) or incorporate reader feedback over time. However, in many cases a model might not have access to this additional information, such as for newcomers in an online forum discussing scientific findings (August et al., 2020a). We are interested instead in providing readers the ability to explicitly set definition complexity suited to their technical comfort (McNamara and Kintsch, 1996; Kintsch, 1994; Kim et al., 2016).

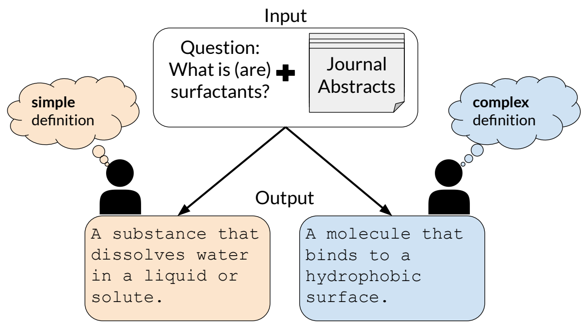


Figure 1: Example of our task. Definitions are generated with a controlled amount of complexity based on the question, "What is (are) X?"

We introduce a new task for generating definitions of scientific and medical terms with varying complexity (§2; Joshi et al., 2017; Fan et al., 2019).

Our dataset (§3) is constructed from consumer medical questions and science glossaries containing words that vary in their complexity and frequency.

We start by evaluating four modeling approaches for generating definitions, finding that, among them, a finetuned BART model is most successful at this new task (§4). As a first step to adjusting definition complexity, we introduce methods to explicitly set definition complexity as either high or low at generation time.

To our knowledge, this is the first paper using decoding-time controllable generation techniques on text complexity. We operationalize complexity based on readability and science communication research (Pitler and Nenkova, 2008; Gardner and Davies, 2013; Leroy et al., 2010) and evaluate several state-of-the-art controllable generation methods on this task (§5). We also develop a new, lightweight method for controlling generation based on discriminator ranking.

Our automatic and human evaluations show that our lightweight method is effective at varying complexity while maintaining high fluency and reducing factual errors. On publication, we will make our dataset, models, and evaluation scripts available to encourage future work on this task.

2 Definition Tasks

Generating definitions has been approached as a word-to-sequence task, where language models used a word's embedding to generate its definition (Noraset et al., 2017). Recent work used a sequence-to-sequence setup for generating definitions instead, where the defined word was a highlighted token in a sequence (Mickus et al., 2019).

This conceptualization of definition modeling is an important starting point for addressing our task. However, new scientific terms are introduced regularly and many never appear in dictionaries or reach common usage (Shea, 2015; Britt et al., 2014), making it difficult to rely on general-purpose dictionaries (Kim et al., 2016). Scientific terms are also notoriously esoteric (e.g., *hidden Markov model*) or else overload definitions of common words (e.g., *transformer* the model architecture versus *transformer* the electrical device), both of which complicate the use of standard word representations from pretrained models (Beltagy et al., 2019).

We address these issues by drawing inspiration from abstractive question answering (QA). Specifically, we frame our task as generating an answer to the question "What is (are) X?" This reframing allows us to leverage scientific definitions from more diverse sources (e.g., QA datasets) and to incorporate domain-specific knowledge into definition generation by including supporting information (§3.2; Chen et al., 2017; Joshi et al., 2017).

3 Dataset Collection

We collect a new dataset of definitions that are answers to the question "What is (are) X?" where X is a scientific term or concept (e.g., *carbon nanotubes*). These questions are roughly equally from an existing QA dataset or templated from scientific glossaries.

3.1 Sources

We draw definitions from two sources.

Medical consumer questions Ben Abacha and Demner-Fushman (2019) collected 47,457 medical questions from 12 National Institutes of Health (NIH) websites and collected them into the MedQuAD dataset. The dataset covers 37 different question types. Three question categories are focused on defining and providing information on medical terms: "Information," "How can I learn more," and "Other information."

Manual inspection of these question categories shows that all questions are of the form "What is (are) X?" or "Do you have more information on X?" Responses to the these questions begin with a brief definition of X. After filtering for this question type and removing questions with no answer due to copyright restrictions, we had 4,525 definitions.

Wikipedia The MedQuAD questions are an excellent source of definitions, but only cover medical terms. Because we are interested in tackling scientific terms more broadly, we augment this set with terms drawn from Wikipedia science glossaries.[1] We extract all science-related terms and their definitions, yielding another 3,738 terms for a total dataset of 8,263 terms.[2]

We split our dataset into training, development, and test sets (8/1/1). Examples of terms in this

[1] https://en.wikipedia.org/wiki/Category:Glossaries_of_science

[2] We explored using other QA datasets that included scientific information to expand our coverage of scientific domains outside of medicine, such as the Explain Like I am Five (Fan et al., 2019) and ARC science exam question datasets (Clark et al., 2018). We found these questions to be less focused on definitions, though future work might find ways to make use of them.

Source	Count	Example Questions	Example Definitions
MedQuAD	4,525	What is (are) complement component 2 deficiency?	Complement component 2 deficiency is a disorder that causes the immune system to malfunction, resulting in a form of immunodeficiency.
Wikipedia	3,738	What is (are) rotation period?	The time that an object takes to complete a single revolution about its own axis of rotation relative to the background stars.
Total	8,263		

Table 1: Dataset statistics and examples.

dataset are in Table 1.

3.2 Support Documents

We next collect scientific abstracts related to each term to allow models to incorporate related scientific knowledge (Fan et al., 2019; Clark et al., 2018). Specifically, given a term question (i.e., "What is (are) X?"), we query S2ORC (Lo et al., 2020), a corpus of over 81 million scientific articles, for the top 10 related abstracts. Query scoring and retrieval is done with Elasticsearch.[3] These abstracts are concatenated together and form the input along with the term question for our models (§4).

We use scientific abstracts, rather than general audience text like Wikipedia or the Common Crawl, for two reasons. First, scientific terms are originally introduced and most commonly used in research papers, making them the most reliable source for these terms. Second, terms can be contextual, having different meanings in common usage. Additional details for collecting the terms and creating the support documents are in Appendices A.1 and A.2.

3.3 Why Not Standard Dictionaries?

Our goal is to create a definition dataset with (i) coverage of scientific and medical terminology and (ii) diverse levels of complexity, to support the application envisioned in §1. We conjecture that general-purpose dictionaries will lack coverage of such terms and tend to have complex definitions for those terms that they do include. Indeed, we found that less than 20% of the terms (191 out of 1,000) in the medical consumer portion of our dataset have entries in the Merriam Webster Dictionary (MW).[4] The dictionary definitions also use substantially more academic vocabulary: an average of 39% (s.d. 12%) of words in those dictionary definitions were in the Academic Vocabulary List (Gardner and Davies, 2013)—a list of words that occur more frequently in academic writing than common usage—compared to 29% (s.d. 12%) in our definitions. Examples of definitions from our dataset and from MW are in Table 7 in the Appendix.

While complex definitions are not necessarily bad, we want diverse complexity levels in our input. While medical consumer questions tend to use fewer specialized terms than a dictionary, we also find that a random sample of 1,000 Wikipedia terms in our dataset use close to as much specialized terminology as a dictionary (37%, s.d. 12%). This provides us with a wider range of complexity levels than were we to use a single source of scientific definitions. We later explore how this exposure to different complexity levels in the input make it possible to control the complexity of generated definitions (§5.2).

4 Definition Generation: Basic Models

Our first goal is to generate fluent definitions that include relevant and accurate information about the term being defined. Because this is a new task and there are multiple reasonable approaches to generating fluent text (Prabhumoye et al., 2020), we experiment with four methods that have performed strongly in question answering and general-purpose definition generation and evaluate their effectiveness in this novel domain. For additional details on the training setups and hyperparameter tuning for the models described below, see Appendix A.3.

4.1 Methods

Sequence-to-Sequence: Finetuning BART (FT BART) BART (Lewis et al., 2020) has been used to define general English terms in context (Bevilacqua et al., 2020) and reached state-of-the-art results on the Explain Like I am Five (ELI5; Lewis et al., 2020) QA dataset, which includes some questions requiring scientific knowledge (e.g., "What is a Turing Machine and why is it so important?").

[3] https://www.elastic.co/

[4] For this analysis, we exclude the Wikipedia science glossary terms since Wikipedia is also often used as a general-purpose resource of definitions, and the Merriam Webster API restricts us to 1,000 queries.

We experiment with finetuning the BART pretrained model on our task and dataset (referred to as FT BART). During training and generation we concatenate the term question with the supporting document. We use BART-large as our base model.[5]

Out-of-the-Box Causal Language Modeling (OOTB GPT-2 and OOTB GPT-3) Recent work has also shown that large pretrained causal language models, such as GPT-2 and GPT-3, can generate fluent answers to factual questions without finetuning (Brown et al., 2020).

We experiment with using both GPT-2 and GPT-3 out-of-the-box (OOTB GPT-2 and OOTB GPT-3). We use GPT-2 medium[6] and GPT-3 davinci[7] for these experiments. For OOTB GPT-3, we evaluate with 100 terms due to OpenAI API limits. For generation, we follow the few-shot setting proposed in Brown et al. (2020) and prepend two held-out question term and definition pairs before each generation.

We do not include the supporting documents in this few shot setting since doing so extends beyond GPT-2's context window of 1024 tokens and preliminary results showed that the additional text led to fewer definitions and more repetition from the abstracts.

Finetuning GPT-2 (FT GPT-2): Because OOTB GPT-2 and OOTB GPT-3 involve no finetuning or use of the support documents, we suspect that they will underperform FT BART. We experiment with finetuning the GPT-2 medium model (FT GPT-2) with the question and support document, separated by new special tokens.

Information Retrieval (OOTB BIDAF): Information retrieval (IR) methods are an important part of many open-domain QA systems and have presented a strong baseline in scientific question answering (Clark et al., 2018). We experiment using a pretrained BiDAF model (Seo et al., 2017) to extract the highest scoring span in the support document based on the term question (OOTB BIDAF). We use AllenNLP's implementation of BiDAF trained on SQuAD.[8]

[5] https://huggingface.co/facebook/bart-large

[6] https://huggingface.co/gpt2-medium. We obtain similar results when using GPT2-large.

[7] https://beta.openai.com/

[8] https://docs.allennlp.org/models/main/models/rc/predictors/bidaf/

4.2 Results

Table 3 shows the ROUGE scores and BERTscore for each modeling method on the development set of our dataset.[9] FT BART outperforms all other models. OOTB GPT-3 performs surprisingly well, outperforming even FT GPT-2. OOTB BIDAF extracts spans that don't answer the question.

Table 2 provides examples of the generated definitions for each modeling approach. FT BART provides the most concise answer while also remaining informative, compared to FT GPT-2's definition, which is circular (e.g., "Acanthoma (cancer) is a type of cancer"). While most models show impressive background knowledge, there is evidence of incorrect or hallucinated information, such as Acanthoma being a type of skin cancer (OOTB GPT-2), these hallucinations are marked in Table 2. We explore the amount of hallucinated information further in §7.2. For the rest of the paper we use the FT BART model since it outperforms other methods.

5 Controlling Definition Complexity

Automatically generating definitions is an important first step in supporting readers who come across unfamiliar scientific terms, but individuals can have different tolerances for the complexity of a definition depending on their domain knowledge (Britt et al., 2014). The models we tested in Section 4 were not trained to vary the complexity of definitions; they do not adapt definitions to different readers. Here we explore how to control the complexity of generated definitions.

Controlling or guiding text generation is an active research area with important applications like toxicity control (Gehman et al., 2020) and language debiasing (Ma et al., 2020). For a review, see Prabhumoye et al., 2020. To the best of our knowledge, ours is the first work to evaluate decoding-time controllable generation methods for text complexity.

One task that has considered changing text complexity is text simplification. Work on text simplification has mostly used a machine translation setup based on parallel corpora (Zhu et al., 2010; Cao et al., 2020) to translate complex sentences into simple ones. These parallel corpora are rare and often expensive to create (Xu et al., 2015). This setup also assumes an input text to be simplified

[9] We reserve our test for the experiments on complexity control to avoid selecting models based on a test set that they are later evaluated on. Hyperparameter tuning and finetuning were done on split subsets of the training data.

Model	Response
FT BART	Acanthoma is a skin lesion that develops from cells in the skin.
FT GPT-2	Acanthoma **(cancer) is a type of cancer**.
OOTB GPT-2	Acanthoma is a **type of skin cancer** that is caused by **the fungus Acanthamoeba histolytica**. **It is a common skin cancer in the United States, and it is also found in other parts of the world, such as the United Kingdom, Australia, and New Zealand**.
OOTB GPT-3	An Acanthoma is a **form of skin cancer** which can also be termed as a skin tumor that arises from the cells of the epidermis, is usually pinkish in color and may or may not be itchy. Acanthomas are classified in various ways based on their histological appearance, such as:
OOTB BIDAF	**Broad Line Region**

Table 2: Generated definitions from each modeling approach for the question: "What is (are) Acanthoma?" Factually incorrect information is labelled in **bold red**.

Model	ROUGE (↑)			BERT (↑)
	1	2	L	
FT BART	**0.33**	**0.16**	**0.30**	**0.89**
FT GPT-2	0.27	0.08	0.24	0.87
OOTB GPT-2	0.20	0.05	0.16	0.85
OOTB GPT-3	0.30	0.14	0.27	0.87
OOTB BIDAF	0.03	0.00	0.03	0.80

Table 3: ROUGE and BERT scores for basic definition generation methods..

(Surya et al., 2019), whereas our task expects that the text will be generated with varying complexity.

5.1 Baseline Generation Control Methods

Below we describe prior methods, used as baseline generation control methods. In each case, we focus on a binary distinction between "low complexity" and "high complexity" definitions, leaving more fine-grained distinctions to future work. We also introduce a novel lightweight approach based on reranking candidate generations in §5.2. Additional details for training are in Appendix A.4.

Plug-and-play language models PPLM (Dathathri et al., 2020) is a technique to guide generation using the gradients of a classifier for a particular desired text attribute. At each generation step, the classifier's gradients are used to update the language model's hidden representations. Due to the computational expense of PPLM, we evaluate with 100 randomly sampled test set terms.

We train our attribute classifier on sentences from scientific journal abstracts and scientific news articles. Journal abstracts are sampled from the ArXiv dataset (Clement et al., 2019) and used to guide to more complex language. Scientific news articles are sampled from a corpus of science news articles (August et al., 2020b) and used to guide towards less complex language.

Generative discriminators The GeDi method (Krause et al., 2021) uses a class-conditioned language model trained on text with a certain desired (or undesired) feature (e.g., toxicity) to guide generation. At each generation step, the model provides next token probabilities to the generator via Bayes' rule. We train a new GeDi on the same dataset of science news and journal articles as for PPLM.

Ensemble of language models DExperts (Liu et al., 2021) combines multiple pretrained language models in an ensemble of "experts" and "anti-experts." Specifically, a base language model is combined with a language model trained on text with desirable attributes (expert) and text with undesirable attributes (anti-expert). At generation time, the base model's logits are combined with the difference of the expert's and anti-expert's logits.

Our expert and anti-expert are pretrained BART-large models that we continue to pretrain on the data used to train the PPLM discriminator. One model is pretrained on the journal abstracts and one on the science news articles. To generate more complex definitions, the expert is the model trained on journal abstracts while the anti-expert is the model trained on science news. To generate less complex definitions, the roles are reversed.

5.2 Novel Approach: Reranking

We introduce a new, lightweight method to generate definitions with different complexity via reranking. Past work has explored selecting candidate generations based discriminator scores to control for specific topics or discourse structure but found that it did not provide strong control (Dathathri et al., 2020; Gabriel et al., 2021). Because our generation task does not require topic shifts and our input has naturally varying complexity (§3.3), we adapt this method by scoring and selecting candidates based on complexity discriminators.

Model	AVL ↑	TE ↑	Function Words ↓	GPT ppl. ↑	# Words ↓	Flesch-Kincaid ↑
Rerank-SVM	**0.10**	**0.12**	**–0.04**	**128.71**	**–0.53**	1.60
Rerank-BERT	**0.01**	**0.04**	**–0.01**	–4.36	0.20	0.68
DExpert	–0.06	**0.05**	0.01	**1130.29**	**–3.23**	–4.01
GeDi	–0.01	**0.01**	–0.01	–40.45	**–1.14**	–0.48
PPLM (100)	–0.02	**0.03**	**–0.01**	123.16	–0.67	–0.04

Table 4: Differences between high and low complexity generations. **Bolded** values are statistically significant in the correct direction using independent samples t-test corrected with the Bonferroni-holm correction for multiple hypothesis testing ($p < 0.002$; Weisstein, 2004). Flesch-Kincaid is a single score and so not tested for significance.

Specifically, at test time we use our BART model (FT BART) to generate 100 candidate definitions for each definition. We then rerank these candidate generations based on logits from a discriminator trained to distinguish scientific journal text from science news text. We consider two discriminators. Both are trained on the the same dataset of science news and journal articles as PPLM.

BERT We use the SciBERT uncased pretrained model (Beltagy et al., 2019). For more complex definitions we select definitions with high predicted probability for journal text, and for less complex definitions we select definitions with high prediction probability for science news text.

Linear We also experiment with using a linear SVM classifier. The SVM's features are complexity measures drawn from science communication and readability literature, discussed in §5.3.

5.3 Complexity Measures

The complexity of scientific writing is affected by many factors and it is difficult to operationalize it into a single dimension. We use multiple measures of scientific writing complexity based on prior work in science communication and readability. These measures are not meant to be an exhaustive list (for a review, see Pitler and Nenkova, 2008), but a selection of measures that capture different elements of complexity important to definitions.[10]

We use most of these measures in two different ways. Five of them are the features in our linear SVM reranker. We also use them as a preliminary automatic evaluation of the various controllable generation approaches in §5.1 and §5.2. Obviously, we expect the linear SVM reranker to outperform the other approaches in this automatic evaluation since it was trained with these complexity features; it should be considered something like an upper bound for these complexity measures. Our human evaluations (§6.2 and §7) provide a more complete picture of the systems' performance.

Academic Vocabulary List (AVL) occurrences The AVL is a list of academic vocabulary drawn from corpora spanning many scientific disciplines (Gardner and Davies, 2013). We measure the fraction of AVL words in a generated definition.

***Thing Explainer* out-of-vocabulary** The popular book *Thing Explainer* explains scientific concepts using only the 1,000 most frequent words in English (measured by Wiktionary's contemporary fiction frequency list) (Munroe, 2017).[11] We measure the fraction of words in the definition outside of the top 1,000 used in *Thing Explainer*.

Function words In health communication, function words (e.g., prepositions, auxiliary verbs, or question words) positively correlate with perceived and actual readability (Leroy et al., 2008, 2010).

Sentence length Sentence length is a commonly used metric for document level complexity and is part of many classic readability measures (Pitler and Nenkova, 2008; Feng et al., 2010). While we set a maximum generation length for our definitions (64 tokens), we enable early stopping. While longer sentences are often considered more complex, we hypothesize that in our dataset longer definitions will be associated with less complex language due to elaborative simplification, where complex terms are explained as a way of simplifying them (Srikanth and Li, 2020).

Language model perplexity Language model perplexity has been found to correlate with perceived and actual reading difficulty (Pitler and Nenkova, 2008; Collins-Thompson, 2014). We use the GPT model to measure language model perplexity, as it was trained on common English (as opposed to scientific text).

[10] Table 17 in the Appendix has examples of model outputs that scored either very high or very low for each measure.

[11] https://en.wiktionary.org/wiki/Wiktionary:Frequency_lists/Contemporary_fiction

Control Method	Direction	
	Low (News)	High (Journal)
SVM-Rerank	A type of computing in which there are many computers running at the same time in different parts of the world.	In computer science, distributed computing is the process of computing on a large scale **without a single centralized data center**.
BERT-Rerank	A type of computer system in which there are more than a few computers working together.	In computer science, distributed computing is the process of computing on a large scale **without a single centralized data center**.
GeDi	Is the implementation of computer programs across multiple computers on similar hardware and/or software resources.	In computer science, **a concept that states that data must be shared across computing resources**.
DExpert	An **Internet-driven by-computing** that portion of different computers from start to finish.	In computer science and communication between-Consequently-integrates.
PPLM	**Easeless, self-organized, and often self-organizing** networked computer systems intended for the purposes of optimization.	Multi-purpose, distributed **system software** with or without a single datum storage system.

Table 5: Generated definitions from each complexity control method for the question: **What is (are) distributed computing?** Factually incorrect information is labelled in **bolded red**.

Flesch-Kincaid grade level This score (FK) uses simple calculations based on sentence length, word length, and syllable counts (Kincaid et al., 1975). Although findings are mixed on how well the FK predicts readability in science or medical documents (Leroy et al., 2008), it is a standard, widely used measure of text complexity (Redmiles et al., 2019). The FK expects a document with multiple sentences, but our definitions are a single sentence. To address this, we calculate the FK score based on the concatenation of all definitions generated by a particular method. For the same reason, we do not include the FK score as a feature in our SVM reranker (§5.1).

6 Evaluating Complexity

Here we evaluate how well our baseline and novel generation control methods can vary the complexity of definitions. For each generation method, we generate and evaluate 10 definitions for each term.

6.1 Automatic Evaluation

We automatically evaluate each control method by calculating the difference in each complexity measure (§5.3) for the high and low complexity generations. Table 4 details these differences. While each measure captures a different element of complexity, counting the number of words outside of the top 1,000 most common English words (TE) seems to be one of the most consistent measures, with all higher complexity generations having differences in the expected direction. DExperts and the BERT reranker have the largest differences, with 5% and 4% more words per sentence. Higher complexity generations also have higher GPT perplexity, with DExperts having the largest difference.

The two rerankers (BERT and SVM) perform better than other models on most measures. This is unsurprising for the SVM since it was trained with these complexity features, but it is interesting that reranking with the BERT classifier also provides effective control over complexity. Table 5 provides example generations based on each approach.

6.2 Human Evaluation

Automatic classification of text complexity is difficult and domain-specific (Collins-Thompson, 2014; Redmiles et al., 2019); even in combination, we believe the measures in §5.3 are insufficient for a full evaluation of our approaches. We therefore carry out a human evaluation to assess how each method influences perceived definition complexity.

We select the models that performed best overall in our automatic evaluation: DExperts, GeDi, and the SVM reranker.[12] We randomly sample 50 terms from our test split to evaluate. We use a high and low complexity generation from each model, leaving us with $50 \times 2 \times 3 = 300$ definitions.

We broke down complexity into two ratings: how complicated a definition was and how difficult to understand the definition was. For each, participants rated definitions on a 1–4 Likert scale. We recruited participants on Amazon Mechanical Turk. Each participant was payed US$0.50 cents based on US$10 dollars/hour. This study was approved our our institution's internal review board.

Participants 233 participants took part in our evaluation (mean age 35 years, s.d. 11). Table 18 in the Appendix provides more details on their demographics. We removed 4 participants due to

[12]We do not include PPLM in this analysis due to its computational cost and similar performance to GeDi.

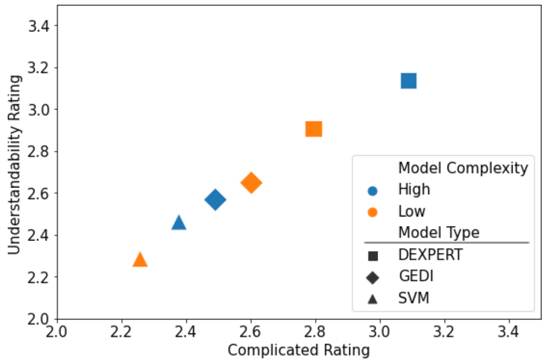


Figure 2: Average ratings for how complicated ("How complicated is the definition's text?") and difficult to understand ("Imagine you are looking up this term, how hard is it for you to understand this definition?") definitions are for each model on each complexity level. Range is from 1 = "Not at all" to 4 = "Very". No differences are statistically significant.

Model	Fluency (s.d.)↑	Relevence (s.d.)↑	Factuality (s.d.)↓
SVM	3.71 (0.59)	3.51 (0.78)	1.81 (0.81)
GeDi	3.20 (1.06)*	2.86 (1.22)*	2.38 (1.12)*
DExpert	2.33 (0.85)*	2.80 (0.91)*	2.59 (0.97)*

Table 6: Fluency, relevance, and factuality ratings from our human evaluation. More details are in Appendices A.7.2 and A.7.3. * =Significant compared to SVM ratings using independent t-tests corrected for multiple hypothesis testing using the Bonferroni-Holm correction.

low effort responses (i.e., responding to all prompts with the same rating within 15 seconds).

Results Figure 2 shows the average ratings for each model type. DExperts generations differentiate most between high and low complexity. GeDi definitions behave in a way that is the opposite of what we expected, with the low complexity generations rated as more complicated and difficult to understand than the high complexity generations. The SVM-reranked definitions perform in the expected direction, with high complexity generations being rated as more complicated and difficult to understand. Examples of ratings and raw counts are in Table 19 and Figure 4 in the Appendix.

7 Evaluating Fluency, Relevance, and Factuality

Our results suggest that our reranking method is a simple intervention that can control complexity with similar performance as other state-of-the-art methods. However, definitions of scientific terms also must be fluent, relevant, and factual. Factuality can be especially difficult to achieve in generations (Maynez et al., 2020). In science communication such failures could spread misinformation with fluent but incorrect definitions (Britt et al., 2019).

We do two additional human evaluations for fluency and relevance (§7.1), and factuality (§7.2). We used two trained annotators, one of them an author, to rate the same 300 definitions used in the complexity evaluation (§6.2). Neither annotator saw the model generations before evaluation or know which method had generated each definition.

7.1 Fluency & Relevance

Annotators rated definitions for fluency and relevance using 1-4 Likert scales (1 = "Not at all" to 4 = "Very"). Table 6 shows the average fluency and relevance ratings. The SVM-reranked definitions were rated close to "Very" fluent and relevant (both above 3.5 on a 4 point scale), and significantly more fluent compared to GeDi ($t_{198} = 5.99$ $p < 0.001$, Cohen's $d = 0.60$) and DExperts ($t_{198} = 18.85$ $p < 0.001$, $d = 1.88$).

7.2 Factuality

For each definition, annotators identified if there was any factually incorrect information in the definition (a binary label) and if so, rated how extensive these errors were on the same 1–4 scale. Table 6 reports on the average rating for how extensive these errors were. Below we report on the binary label.

Overall 60% of our generations were labeled as factually incorrect by at least one annotator (40% by both). The SVM had significantly fewer factual errors (38% by one annotator, 16% by both), compared to GeDi (52% and 33%, $t_{198} = 4.71$ $p < 0.001$, Cohen's $d = 0.47$) and DExperts (86% and 67%, $t_{198} = 12.29$ $p < 0.001$, $d = 1.24$).

8 Conclusion

We introduce a new task and dataset for generating definitions of scientific terms with controllable complexity as a way of adapting to different reader's scientific background. We evaluate conventional generation methods and introduce a lightweight approach of reranking candidate generations based on a discriminator to control complexity. We find that this reranking is effective at controlling text complexity while also maintaining fluency and factuality. We will release our dataset and code on publication to encourage more work on making scientific terms more accessible to readers of diverse background knowledge.

9 Ethical Considerations

The goal of this paper is to enable a wider audience of readers to understand and engage with scientific writing. A risk, though, is that such attempts might instead widen the gap to accessing scientific information. The texts in the datasets we train our models on are in General or Academic American English. Many people, especially those who have been historically underrepresented in STEM disciplines and medicine, may not be comfortable with this dialect of English. This risks further alienating the readers we hope to serve. This is a common issue in NLP systems (Sap et al., 2019), since the majority of datasets are in General American English. An important and exciting direction in NLP is making models more flexible to dialects and low-resource languages (e.g., the ACL 2022 theme being "Language Diversity").

While our results suggest that the lighter control of reranking generations leads to less hallucinated information, strong supervision of definition factuality is important for any future deployment of such a system. While hallucinated information can be damaging in any generation context, incorrect scientific definitions could mislead readers and potentially contribute to broader scientific misinformation. Furthermore, a bad actor could use these models to generate fluent but incorrect definitions at scale, potentially contributing to misinformation campaigns with a veneer of scientific language (Britt et al., 2019). We trained our models on data we believe is trustworthy (e.g., questions and answers from NIH websites); and we release our training data and models to allow for further work on encouraging factuality in these model generations.

References

Sabita Acharya, Barbara Di Eugenio, Andrew Boyd, Richard Cameron, Karen Dunn Lopez, Pamela Martyn-Nemeth, Carolyn Dickens, and Amer Ardati. 2018. Towards generating personalized hospitalization summaries. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Student Research Workshop*, pages 74–82, New Orleans, Louisiana, USA. Association for Computational Linguistics.

Tal August, Dallas Card, Gary Hsieh, Noah A Smith, and Katharina Reinecke. 2020a. Explain like i am a scientist: The linguistic barriers of entry to r/science. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*, pages 1–12.

Tal August, Lauren Kim, Katharina Reinecke, and Noah A. Smith. 2020b. Writing strategies for science communication: Data and computational analysis. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 5327–5344, Online. Association for Computational Linguistics.

Vidhisha Balachandran, Dheeraj Rajagopal, Rose Catherine Kanjirathinkal, and William Cohen. 2018. Learning to define terms in the software domain. In *Proceedings of the 2018 EMNLP Workshop W-NUT: The 4th Workshop on Noisy User-generated Text*, pages 164–172.

Iz Beltagy, Kyle Lo, and Arman Cohan. 2019. SciBERT: A pretrained language model for scientific text. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3615–3620, Hong Kong, China. Association for Computational Linguistics.

Asma Ben Abacha and Dina Demner-Fushman. 2019. A question-entailment approach to question answering. *BMC Bioinform.*, 20(1):511:1–511:23.

Michele Bevilacqua, Marco Maru, and Roberto Navigli. 2020. Generationary or "how we went beyond word sense inventories and learned to gloss". In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 7207–7221, Online. Association for Computational Linguistics.

M Anne Britt, Tobias Richter, and Jean-François Rouet. 2014. Scientific literacy: The role of goal-directed reading and evaluation in understanding scientific information. *Educational Psychologist*, 49(2):104–122.

M Anne Britt, Jean-François Rouet, Dylan Blaum, and Keith Millis. 2019. A reasoned approach to dealing with fake news. *Policy Insights from the Behavioral and Brain Sciences*, 6(1):94–101.

Dominique Brossard and James Shanahan. 2006. Do they know what they read? building a scientific literacy measurement instrument based on science media coverage. *Science Communication*, 28(1):47–63.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel Ziegler, Jeffrey Wu, Clemens Winter, Chris Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei.

2020. Language models are few-shot learners. In *Advances in Neural Information Processing Systems*, volume 33, pages 1877–1901. Curran Associates, Inc.

Yixin Cao, Ruihao Shui, Liangming Pan, Min-Yen Kan, Zhiyuan Liu, and Tat-Seng Chua. 2020. Expertise style transfer: A new task towards better communication between experts and laymen. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1061–1071, Online. Association for Computational Linguistics.

Dallas Card, Peter Henderson, Urvashi Khandelwal, Robin Jia, Kyle Mahowald, and Dan Jurafsky. 2020. With little power comes great responsibility. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 9263–9274, Online. Association for Computational Linguistics.

Danqi Chen, Adam Fisch, Jason Weston, and Antoine Bordes. 2017. Reading Wikipedia to answer open-domain questions. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1870–1879, Vancouver, Canada. Association for Computational Linguistics.

Peter Clark, Isaac Cowhey, Oren Etzioni, Tushar Khot, Ashish Sabharwal, Carissa Schoenick, and Oyvind Tafjord. 2018. Think you have solved question answering? try arc, the ai2 reasoning challenge. *ArXiv*, abs/1803.05457.

Colin B. Clement, Matthew Bierbaum, Kevin P. O'Keeffe, and Alexander A. Alemi. 2019. On the use of arxiv as a dataset.

Kevyn Collins-Thompson. 2014. Computational assessment of text readability: A survey of current and future research. *ITL-International Journal of Applied Linguistics*, 165(2):97–135.

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. In *International Conference on Learning Representations*.

Ismail Fahmi and Gosse Bouma. 2006. Learning to identify definitions using syntactic features. In *Proceedings of the Workshop on Learning Structured Information in Natural Language Applications*.

Angela Fan, Yacine Jernite, Ethan Perez, David Grangier, Jason Weston, and Michael Auli. 2019. ELI5: Long form question answering. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3558–3567, Florence, Italy. Association for Computational Linguistics.

Lijun Feng, Martin Jansche, Matt Huenerfauth, and Noémie Elhadad. 2010. A comparison of features for automatic readability assessment. In *Coling 2010: Posters*, pages 276–284, Beijing, China. Coling 2010 Organizing Committee.

Saadia Gabriel, Antoine Bosselut, Jeff Da, Ari Holtzman, Jan Buys, Kyle Lo, Asli Celikyilmaz, and Yejin Choi. 2021. Discourse understanding and factual consistency in abstractive summarization. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 435–447, Online. Association for Computational Linguistics.

Dee Gardner and Mark Davies. 2013. A New Academic Vocabulary List. *Applied Linguistics*, 35(3):305–327. Downloaded from: `https://www.academicvocabulary.info/`.

Samuel Gehman, Suchin Gururangan, Maarten Sap, Yejin Choi, and Noah A. Smith. 2020. RealToxicityPrompts: Evaluating neural toxic degeneration in language models. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 3356–3369, Online. Association for Computational Linguistics.

Mandar Joshi, Eunsol Choi, Daniel Weld, and Luke Zettlemoyer. 2017. TriviaQA: A large scale distantly supervised challenge dataset for reading comprehension. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1601–1611, Vancouver, Canada. Association for Computational Linguistics.

Yea-Seul Kim, Jessica Hullman, Matthew Burgess, and Eytan Adar. 2016. SimpleScience: Lexical simplification of scientific terminology. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1066–1071, Austin, Texas. Association for Computational Linguistics.

J Peter Kincaid, Robert P Fishburne Jr, Richard L Rogers, and Brad S Chissom. 1975. Derivation of new readability formulas (automated readability index, fog count and flesch reading ease formula) for navy enlisted personnel. Technical report, Naval Technical Training Command Millington TN Research Branch.

Walter Kintsch. 1994. Text comprehension, memory, and learning. *American psychologist*, 49(4):294.

Ben Krause, Akhilesh Deepak Gotmare, Bryan McCann, Nitish Shirish Keskar, Shafiq Joty, richard socher, and Nazneen Rajani. 2021. Gedi: Generative discriminator guided sequence generation.

Gondy Leroy, Stephen Helmreich, and James R Cowie. 2010. The influence of text characteristics on perceived and actual difficulty of health information. *International journal of medical informatics*, 79(6):438–449.

Gondy Leroy, Stephen Helmreich, James R Cowie, Trudi Miller, and Wei Zheng. 2008. Evaluating online health information: Beyond readability formulas. In *AMIA Annual Symposium Proceedings*, volume 2008, page 394. American Medical Informatics Association.

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.

Alisa Liu, Maarten Sap, Ximing Lu, Swabha Swayamdipta, Chandra Bhagavatula, Noah A. Smith, and Yejin Choi. 2021. Dexperts: On-the-fly controlled text generation with experts and anti-experts. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics*, Online. Association for Computational Linguistics.

Kyle Lo, Lucy Lu Wang, Mark Neumann, Rodney Kinney, and Dan S. Weld. 2020. S2orc: The semantic scholar open research corpus.

Xinyao Ma, Maarten Sap, Hannah Rashkin, and Yejin Choi. 2020. PowerTransformer: Unsupervised controllable revision for biased language correction. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 7426–7441, Online. Association for Computational Linguistics.

Alejandro Martínez and Stefano Mammola. 2021. Specialized terminology reduces the number of citations of scientific papers. *Proceedings of the Royal Society B*, 288(1948):20202581.

Joshua Maynez, Shashi Narayan, Bernd Bohnet, and Ryan McDonald. 2020. On faithfulness and factuality in abstractive summarization. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1906–1919, Online. Association for Computational Linguistics.

Danielle S. McNamara and Walter Kintsch. 1996. Learning from texts: Effects of prior knowledge and text coherence. *Discourse Processes*, 22(3):247–288.

Timothee Mickus, Denis Paperno, and Matthieu Constant. 2019. Mark my word: A sequence-to-sequence approach to definition modeling. In *Proceedings of the First NLPL Workshop on Deep Learning for Natural Language Processing*, pages 1–11, Turku, Finland. Linköping University Electronic Press.

Randall Munroe. 2017. *Thing explainer complicated stuff in simple words*. John Murray.

Sonia Krishna Murthy, Daniel King, Tom Hope, Daniel Weld, and Doug Downey. 2021. Towards personalized descriptions of scientific concepts. In *Proceedings of The Fifth Widening Natural Language Processing Workshop*, Punta Cana, Dominican Republic. Association for Computational Linguistics.

Thanapon Noraset, Chen Liang, Larry Birnbaum, and Doug Downey. 2017. Definition modeling: Learning to define word embeddings in natural language. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 31.

Emily Pitler and Ani Nenkova. 2008. Revisiting readability: A unified framework for predicting text quality. In *Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing*, pages 186–195, Honolulu, Hawaii. Association for Computational Linguistics.

Shrimai Prabhumoye, Alan W Black, and Ruslan Salakhutdinov. 2020. Exploring controllable text generation techniques. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 1–14, Barcelona, Spain (Online). International Committee on Computational Linguistics.

Elissa Redmiles, Lisa Maszkiewicz, Emily Hwang, Dhruv Kuchhal, Everest Liu, Miraida Morales, Denis Peskov, Sudha Rao, Rock Stevens, Kristina Gligorić, Sean Kross, Michelle Mazurek, and Hal Daumé III. 2019. Comparing and developing tools to measure the readability of domain-specific texts. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 4831–4842, Hong Kong, China. Association for Computational Linguistics.

Maarten Sap, Dallas Card, Saadia Gabriel, Yejin Choi, and Noah A. Smith. 2019. The risk of racial bias in hate speech detection. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 1668–1678, Florence, Italy. Association for Computational Linguistics.

Minjoon Seo, Aniruddha Kembhavi, Ali Farhadi, and Hannaneh Hajishirzi. 2017. Bidirectional attention flow for machine comprehension. *ArXiv*, abs/1611.01603.

Nicole A Shea. 2015. Examining the nexus of science communication and science education: A content analysis of genetics news articles. *Journal of Research in Science Teaching*, 52(3):397–409.

N. Srikanth and Junyi Jessy Li. 2020. Elaborative simplification: Content addition and explanation generation in text simplification. *ArXiv*, abs/2010.10035.

Sai Surya, Abhijit Mishra, Anirban Laha, Parag Jain, and Karthik Sankaranarayanan. 2019. Unsupervised neural text simplification. In *Proceedings of the*

57th Annual Meeting of the Association for Computational Linguistics, pages 2058–2068, Florence, Italy. Association for Computational Linguistics.

Eric W Weisstein. 2004. Bonferroni correction. *https://mathworld. wolfram. com/*.

Wei Xu, Chris Callison-Burch, and Courtney Napoles. 2015. Problems in current text simplification research: New data can help. *Transactions of the Association for Computational Linguistics*, 3:283–297.

Zhemin Zhu, Delphine Bernhard, and Iryna Gurevych. 2010. A monolingual tree-based translation model for sentence simplification. In *Proceedings of the 23rd International Conference on Computational Linguistics (Coling 2010)*, pages 1353–1361, Beijing, China. Coling 2010 Organizing Committee.

A Appendix

A.1 Data collection

We downloaded all terms from the Wikipedia science glossaries.[13] We included the first definition for each term, and cleaned Wikipedia text of url and image references. Note that since the glossaries provide definitions of all terms on a single page, we did not use the full Wikipedia articles for each term. For each Wikipedia term, X, we format the term as the question "What is (are) X?".

Because our definitions often include additional information beyond a definition (e.g., recommendations for checking if you have the disease being defined), we use the first sentence of each response, which is commonly used in constructing definition datasets (Fahmi and Bouma, 2006).

A.2 Support Documents

Following Fan et al. (2019), we concatenate the abstracts together using a *<P>* token to create a support document for each term question. We filter all retrieved journal abstracts for each question to make sure that none of the same abstracts occur across the train, development, and test splits in our data.

We analyze how often definitions occur in our support documents by searching the documents for the phrase "X is a/an." We find that around 20% of the support documents contain at least one sentence with this phrase. Manual inspection of these sentences revealed that many of them are heavily jargoned, usually containing very few of the same words as our gold definitions. When removing these examples from our test and development set we see no drop in performance. We view these embedded definitions as an additional source of complexity that our models can leverage to vary the generated definitions' complexity.

A.3 Definition generation finetuning

All training and finetuning was done on a NVIDIA Titan X 12GB GPU. We select 1,000 examples from our training dataset and seperate them into a 75/25 split for training and testing each hyperparameter setting. For our model evaluations in §4, we train on a 75/25 split of the full training data and reserve the original development split for testing.

[13] https://en.wikipedia.org/wiki/Category:Glossaries_of_science

Finetuning BART (FT BART) For finetuning the BART model on our dataset, we do a random search for hyperparameter tuning with a subset of our training data. We ran a total of 10 search trials. During training and generation we concatenate the template question with the support document in the format "question: What is (are) X? context: <SUPPORT DOC>".

Table 8 details the final hyperparameters. We use the training code provided by HuggingFace for sequence-to-sequence summarization finetuning.[14]

Out-of-the-Box (OOTB) Language Modeling (OOTB GPT-2 and OOTB GPT-3) For generation, we follow the few-shot setting proposed in Brown et al. (2020). We prepend two held-out question term and definition pairs, shown in Table 9. The two examples are separated by two newlines and a separator token used during generation as the stop symbol (i.e., *###*). At generation time we append the question for the term. Some GPT-3 outputs were empty, which we ignore for evaluation.

Finetuning GPT-2 (FT GPT-2) Each part of the input (supporting document, question, definition) is prepended with a new special symbol (i.e., *<context>*, *<question>*, *<definition>*) and the model is trained in the standard causal language model loss. At generation time, the model is conditioned on the support document, question, and the *<definition>* tag.

We do the same random search for hyperparameter tuning for the GPT-2 model as for BART with the same subset of data. One difference is that we finetune on the standard causal language modeling objective for GPT-2 rather than the sequence-to-sequence summarization task. We use the training code provided by HuggingFace for causal language model training.[15] Table 8 details the final hyperparameters for our GPT-2 model.

A.4 Discriminator training

We filter out all sentences sampled from the journal abstracts and scientific news articles that are less than 5 words, as these sentences are usually bylines or headers, and randomly sample 50k sentences

[14] https://github.com/huggingface/transformers/tree/master/examples/seq2seq

[15] https://github.com/huggingface/transformers/tree/master/examples/language-modeling

Table 7: Example definitions from a general-purpose dictionary (Merriam-Webster) and our dataset.

Term	Dictionary definition	Dataset definition
neuroblastoma	A malignant tumor formed of embryonic ganglion cells	Neuroblastoma is a type of cancer that most often affects children.
cirrhosis	Widespread disruption of normal liver structure by fibrosis and the formation of regenerative nodules that is caused by any of various chronic progressive conditions affecting the liver	Cirrhosis is scarring of the liver.
antibiotics	A substance able to inhibit or kill microorganisms; specifically : an antibacterial substance (such as penicillin, cephalosporin, and ciprofloxacin) that is used to treat or prevent infections by killing or inhibiting the growth of bacteria in or on the body	Summary : Antibiotics are powerful medicines that fight bacterial infections.

Table 8: Final hyperparameters for finetuning the BART and GPT-2 models on definition generation and bounds for hyperparameter tuning random search.

Hyperparameter	BART Assignment	GPT-2 Assignment	Bounds
Number of epochs	3	3	[3, 5]
Effective batch size	8	16	[4, 8, 16]
Learning rate	5e-05	4e-04	[4e-3, 4e-4, 4e-5, 5e-05, 4e-6]
Adam Epsilon	1e-08	1e-07	[1e-7, 1e-8, 1e-9]
Source length/Block size	1024	1024	[1024]
Target length	64	NA	[64]

Table 9: Held out QA pairs for OOTB GPT-2 and OOTB GPT-3.

Question	Answer
What is (are) complement component 2 deficiency?	Complement component 2 deficiency is a disorder that causes the immune system to malfunction, resulting in a form of immunodeficiency.
What is (are) entrepreneurship?	The efforts by a person, known as an ‘entrepreneur,’ in organizing resources for the creation of something new or taking risks to create new innovations and production.

Table 10: Hyperparameters for BART-large PPLM training.

Hyperparameter	Assignment
Batch size	64
Embedding size	1024
Number of steps	10 epochs
Learning rate	1e-4

Table 11: Hyperparameters for BART-large GeDi training.

Hyperparameter	Assignment
Number of epochs	1
Max length	192
Effective batch size	4
Learning rate	2e-5
Lambda	0.80

from each set (100k total) for training, and another 5k each for the development and testing splits.

Even some science news articles require background knowledge not shared among all possible readers (Shea, 2015). We try to address this issue by sampling sentences from science venues that reach a broader audience (e.g., magazines) and have been shown to have lower jargon levels (August et al., 2020b).

PPLM For training the PPLM attribute classifier, we adapt the HuggingFace training code[16] to work with the sequence-to-sequence architecture of BART. Our attribute classifier is trained from the BART-large pretrained model. We use the default training hyperparameters, shown in Table 10.

GeDi For training the GeDi discriminator we adapt the authors original training code[17] to work with the sequence-to-sequence architecture of BART. Our GeDi is trained from the BART-large pretrained model. We use the default training hyperparameters, shown in Table 11.

DExperts For the expert and anti-expert models, we continue to pretrain the BART-large model on

[16] https://github.com/huggingface/transformers/tree/master/examples/research_projects/pplm

[17] https://github.com/salesforce/GeDi/

Table 12: Hyperparameters for additional BART-large pretraining for DExperts.

Hyperparameter	Assignment
Number of epochs	3
Source length	512
Target length	512
Effective batch size	8
Learning rate	5e-05
Learning rate optimizer	Adam
Adam epsilon	1e-08
learning rate scheduler	linear
weight decay	0

science journal text or science news text. Because there is no official script for BART's pretraining, we re-implement the text corruption described in the original paper (Lewis et al., 2020). We specifically create a text-infilling approach, where a number of tokens are masked from each sentence. The number of tokens is drawn from a Poisson distribution ($\lambda = 3$), and they are replaced with a single [MASK] token. We use one mask per sentence in the dataset. We use the default pretraining hyperparameters from HuggingFace's sequence-to-sequence summarization script, detailed in Table 12. We again start from the BART-large pretrained language model.

BERT Reranker We use the SciBERT model (Beltagy et al., 2019) to train our BERT reranker. The training data is identical for training our other discriminators. Table 13 details hyperparameter settings.

SVM Reranker We train our SVM with complexity features from Section 5.3 to classify sentences from academic journal abstracts and science news text using the same dataset for training our discriminators. The SVM reaches 79% accuracy on held out data, showing that these features can be strong differentiators of scientific text.

A.5 Complexity generation hyperparameters

We use the same generation hyperparameters across all models where possible. Shared generation hyperparameters are detailed in Table 14, while those specific to PPLM and GeDi are in Table 15, and

Table 13: Hyperparameters for BERT reranker training.

Hyperparameter	Assignment
Number of epochs	3
Max input length	1024
Effective batch size	16
Learning rate	5e-05
Learning rate optimizer	Adam
Adam epsilon	1e-08
Learning rate scheduler	linear
Weight decay	0.01
Warmup steps	500

Table 14: Hyperparameters shared among all models for generation. For reranking, the top 10 samples are taken out of 100 total returned sequences.

Hyperparameter	Assignment
Number of samples	10
Number of beams	5
Top-p (sampling)	0.9
Top-k	50
Temperature	1
Max length	64
Min length	8

Table 15: Hyperparameters specific to PPLM for generation. Details of each hyperparameters can be found in (Dathathri et al., 2020).

Hyperparameter	Assignment
Number of samples	10
Stepsize	0.06
Gamma	1
GM-scale	0.9
KL-scale	0.01
Repetition penalty	1.0
Grad length	10,000
Horizon length	1
Window length	0

Table 16: Hyperparameters specific to GeDi for generation. Details of each hyperparameters can be found in (Krause et al., 2021).

Hyperparameter	Assignment
Posterior weighting exponent	30
Filter p (1 - p)	0.8
Target p (τ)	0.8
Repetition penalty scale	10
Repetition penalty	1.2

Table 16, respectively. For DExperts, there is one additional hyperparameter, α, which we set to $\alpha = 2.0$ based on the authors original experiments (Liu et al., 2021). For reranking, the top 10 samples are taken out of 100 total returned sequences.

A.6 Complexity Features

To calculate complexity features, we tokenized and lemmatized all generated definitions using Spacy.[18] We lemmatized all words in the AVL and *Thing Explainer* list to search for AVL word occurances and *Thing Explainer* out-of-vocabulary words.

For function words, we used Spacy's POS tags. The following tags we considered function words: ['DET', 'ADP', 'PRON', 'CONJ', 'SCONJ', 'AUX', 'PART', 'INTJ']. For the Flesch-Kincaid grade level, we use the py-readability-metrics package.[19]

Table 17 provides examples of definitions that scored high and low for each complexity feature.

A.7 Human Evaluations

We select our number of samples (50) based on a power analysis with an expected medium effect and power $\beta = 0.8$ (for more information on power and statistical tests in NLP, see Card et al., 2020).

A.7.1 Complexity & Understandability

Participant Demographics The participant demographics for the complexity evaluation (§6.2) are shown in Table 18.

Before beginning, participants filled out a short demographics questionnaire detailing their age,

[18] https://spacy.io/

[19] https://pypi.org/project/py-readability-metrics/

Table 17: Examples of sentences with high or low values of each complexity feature. The Flesch-Kincaid reading level score is not included since it is calculated over all responses for a model.

Feature	High	Low
AVL Occurances	The process by which organic material dissolves in soil.	Your gallbladder is part of your liver.
Thing Explainer OOV	Rock composed mostly yellow tolukalaceous organic material composed mostly marine calcite.	Your brain changes as you age.
Function Words	A place to shelter from the elements of a storm.	See kin genealogy.
LM Perplexity	A metamorphism consisting mainly pyroxenesiloclinic pyroxene.	Your body is made up of many types of muscles.
Word Count	An area of machine-readable digital forerunners or virtual reality-generally enhanced with the goal of gathering, organizing artificial intelligence and guiding artificial neural networks in-depth (machine learning from artificial neural network technology, machine learning and/machine learning and machine learning.	See asteroid impact.

highest degree attained, and STEM (Science, Technology, Engineering, and Math) education. They then reviewed instructions that provided examples of very complex and not at all complex definitions (Figure 5). Each participant rated 3 definitions randomly drawn from different terms. Figure 3 provides an example of the interface for the complexity evaluation. Raw counts of complexity and understandability ratings are provided in Figure 4.

Interrater agreement was relatively low for complexity ($\alpha = 0.14$) and understandability ($\alpha = 0.14$). This is unsurprising given that we used untrained annotators and perceived complexity and understandability are often based on a reader's domain knowledge (Kintsch, 1994).

A.7.2 Fluency & Relevance

Annotators were given examples of very fluent and relevant definitions, and not at all fluent and relevant definitions before starting the task. For fluency, annotators were asked, "How fluent is this definition?" and for relevance, they were asked, "How relevant is this definition for the term?" Interrater agreement was high for both fluency (Krippendorff's $\alpha = 0.63$) and relevance ($\alpha = 0.58$).

A.7.3 Factuality

Annotators were given examples of very extensive factual errors and and not at all extensive factual errors before starting the task. For each definition, annotators checked a box if there was any factually incorrect information in the definition based on the question, "Does this definition contain factually incorrect information?" and if so, rated how extensive these errors were based on the question, "If the definition contains factually incorrect information, how extensive are these errors?" Annotators were encouraged to use the internet if they did not know if a definition was correct.

Interrater agreement was high for both whether a definition contained factually incorrect information (Krippendorff's $\alpha = 0.59$) and how extensive these errors were ($\alpha = 0.55$).

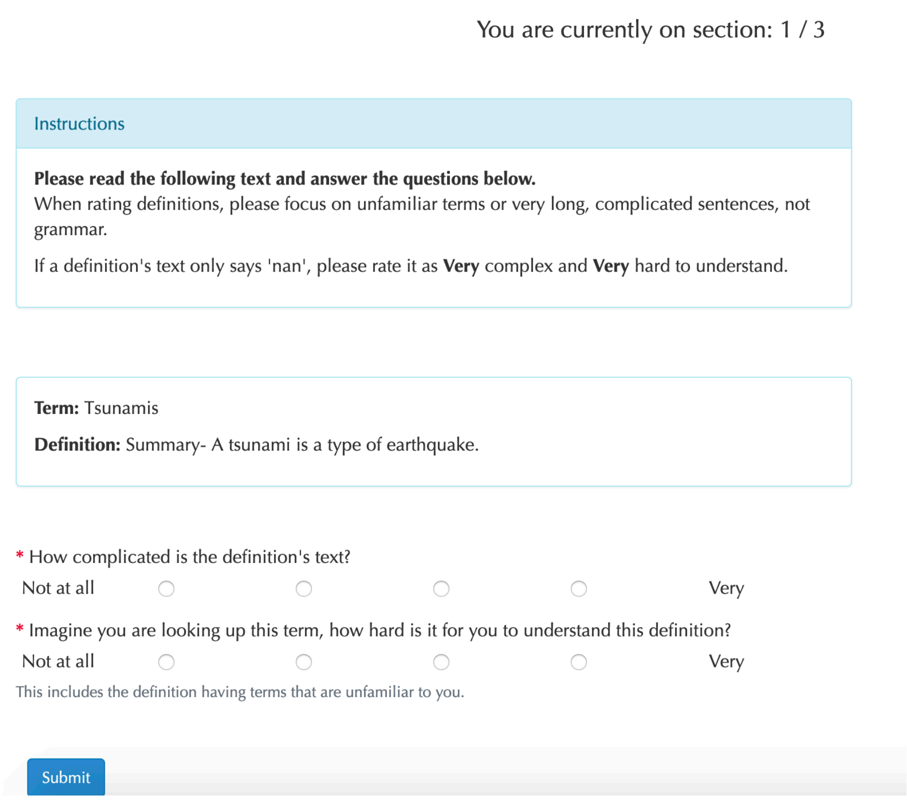


Figure 3: Example of human evaluation interface for definition complexity. The fluency and factuality evaluations had the same interface.

Table 18: Participant demographics for the complexity evaluation.

Age	0-19	0
	20-29	74
	30-39	106
	40-49	32
	50-59	10
	60-69	7
	70-79	4
	80+	0
English proficiency	Elementary	6
	Limited working	5
	Professional working	7
	Full professional	25
	Native/bilingual	190
Education	Pre-high school	1
	High School	45
	College	118
	Graduate school	60
	Professional school	9
# STEM courses after high school	0	44
	1-3	84
	4-6	55
	7-9	18
	10+	32

Model	Term	Definition	Complexity	Understandability
DEXPERT High	Bayesian Programming	A formalism for problem-solving in computer programming.	1	4
DEXPERT Low	Zirconium	A rock mineral that crystallises on rock beds or minerals silicate beds.	3	1
GeDi Low	Sexually Transmitted Diseases	There are a number of sexually transmitted diseases.	1	1
GeDi High	Tsunamis	Summary : Tsunamis are oceanic tsunamis.	2	4
SVM Low	Paroxysmal extreme pain disorder	Paroxysmal extreme pain disorder (PEPD) is a rare form of erythromelalgia.	4	2
SVM High	Kelvin–Helmholtz instability	A condition in which the flow of charged particles in a fluid is unstable.	4	4

Table 19: Example generations and their ratings. Examples are selected to show a range of ratings.

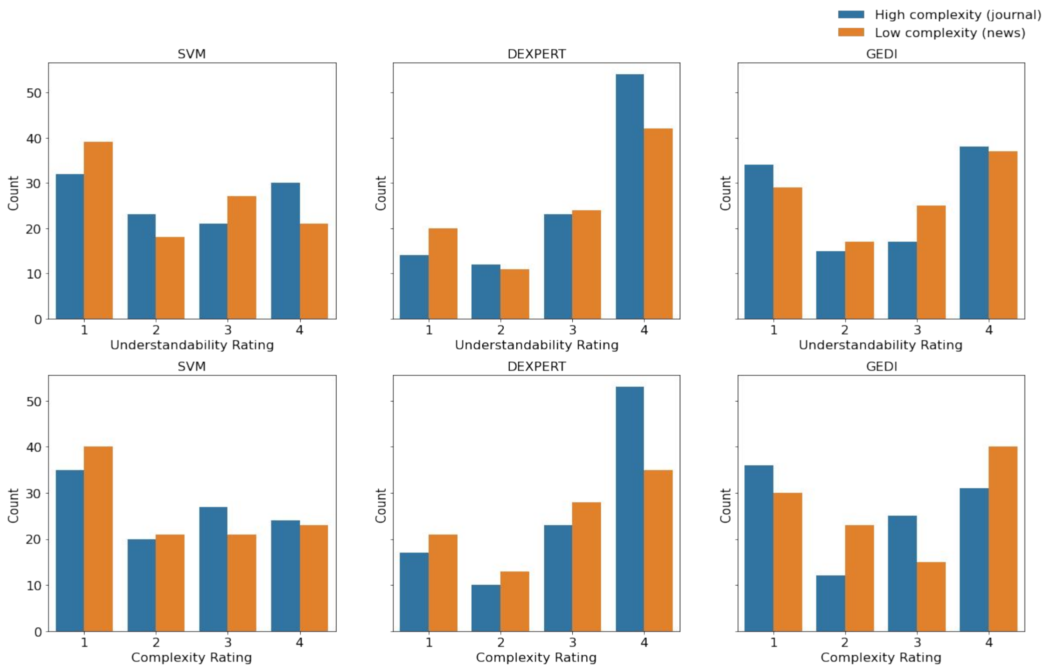
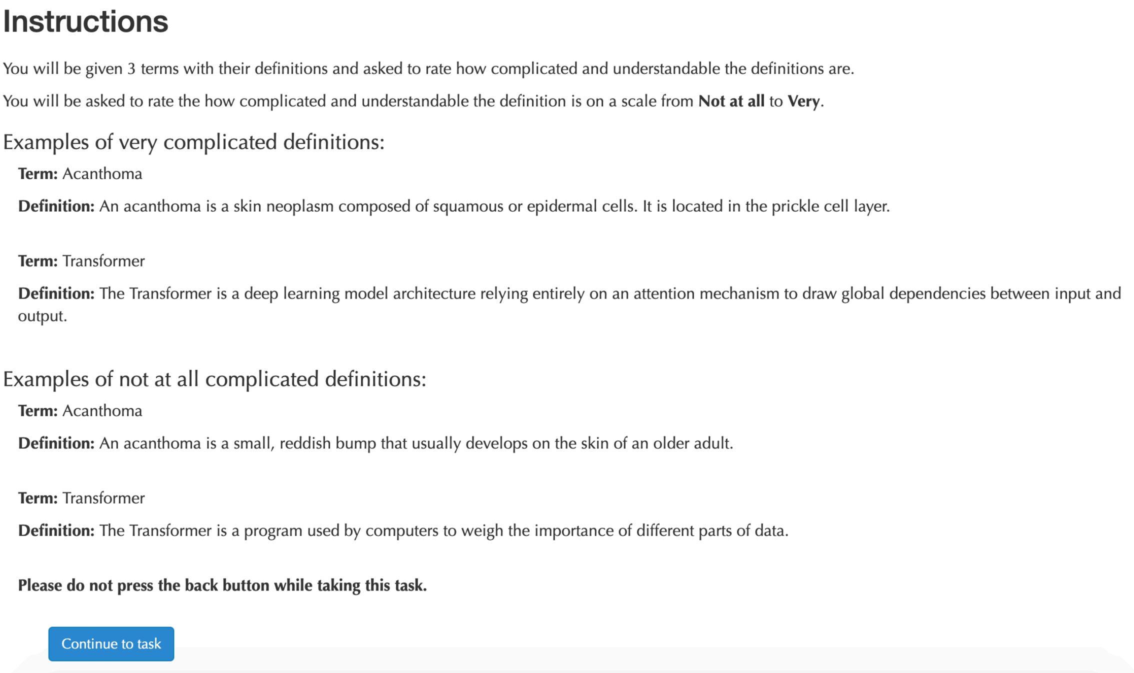


Figure 4: Counts of complexity and understandability ratings for each controllable generation method. 1 = Not at all and 4 = Very



Instructions

You will be given 3 terms with their definitions and asked to rate how complicated and understandable the definitions are.

You will be asked to rate the how complicated and understandable the definition is on a scale from **Not at all** to **Very**.

Examples of very complicated definitions:

Term: Acanthoma

Definition: An acanthoma is a skin neoplasm composed of squamous or epidermal cells. It is located in the prickle cell layer.

Term: Transformer

Definition: The Transformer is a deep learning model architecture relying entirely on an attention mechanism to draw global dependencies between input and output.

Examples of not at all complicated definitions:

Term: Acanthoma

Definition: An acanthoma is a small, reddish bump that usually develops on the skin of an older adult.

Term: Transformer

Definition: The Transformer is a program used by computers to weigh the importance of different parts of data.

Please do not press the back button while taking this task.

Continue to task

Figure 5: Instructions page for the human complexity evaluation.