# Generative Models can Help Writers without Writing for Them

Kenneth C. Arnold[a], April M. Volzer[a] and Noah G. Madrid[a]

[a]Calvin University, 1740 Knollcrest Circle SE, Grand Rapids, MI, 49546, USA

**Abstract**

Computational models of language have the exciting potential to help writers generate and express their ideas. Current approaches typically provide their outputs to writers in a way that writers can (and often do) appropriate as their own—giving the system more control than necessary over the final outcome of the writing. We present early explorations of two new types of interactions with generative language models; both share the design goal of keeping the writer in ultimate control while providing generative assistance. One interaction enables new kinds of structural manipulation of already-drafted sentences; it keeps the writer in semantic control by conditioning the output to be a paraphrase of human-provided input. The other interaction enables new kinds of idea exploration by offering questions rather than snippets to writers; it keeps the writer in semantic control by providing its ideas in an open-ended form. We present the results of our early experiments on the feasibility and suitability of these types of interactions.

**Keywords**

writing tools, language modeling, interactive paraphrase generation, user interface

## 1. Introduction

Human writers can use computational tools to be more efficient, creative, and effective. As the capability and accuracy of language models has been improving rapidly in recent years, we are excited by the idea that these technologies might help writers in new ways. Recent developments have used language modeling technology to improve text entry (e.g., improving speed and accuracy) and feedback about grammar and style. We envision technology that will go beyond these capabilities to help writers explore ideas and alternatives—to rapidly hone both what to say and how to say it.

A common approach for using language models to help writers is to have the model generate text, often with a seed, constraint, or objective; the writer then uses the resulting text directly or as inspiration (see, for example, a study from last year's HAI-GEN workshop [1]). Although this type of application aligns with the capabilities of the language model (auto-regressive generation of the next token), it does not necessarily support all of the ways that writers may want assistance. Moreover, it feeds words to writers—suggesting that the writer claim the system's words as their own. Several studies have documented conformity effects of predictive text systems on writing content [2, 3]. Thus, such "autocomplete" interactions might not align with the writer's goals or their desire to have an individual voice.

Our design goal is to use the strengths of generative models of language to help writers while keeping them in direct control of the writing. The two approaches we present towards that goal target different stages of writing. To help writers with drafting, most existing approaches generate text; could the system instead generate *questions* that might inspire writers to add details or clarify their arguments? To help writers with editing, most existing approaches try to identify and fix errors; could the system instead enable the writer to edit the structure and content of their documents through direct manipulation of words and phrases?

In this work, we present prototypes and feasibility studies of these two interactions. We first discuss the task of interactive sentence structure manipulation and present two interaction techniques and one NLP approach to power them. We then describe the task of providing open-ended topic ideas to writers and present evidence from an exploratory study that suggests that writers substantially prefer guidance in the form of questions to guidance in the form of examples.

## 2. Interactive Manipulation of Sentence Structure

Our first proposal involves utilizing language modeling technology to enable writers to manipulate individual sentences in their writing to shape their meaning and organization. This manipulation could entail changing both the words and their arrangement within the sentence.

Some current interfaces suggest edits that can be accepted or rejected, typically for grammatical error correction [4] or contextual spelling correction. Other in-

*Joint Proceedings of the ACM IUI 2021 Workshops, April 13-17, 2021, College Station, USA*

ka37@calvin.edu (K. C. Arnold)

0000-0003-3892-9870 (K. C. Arnold)

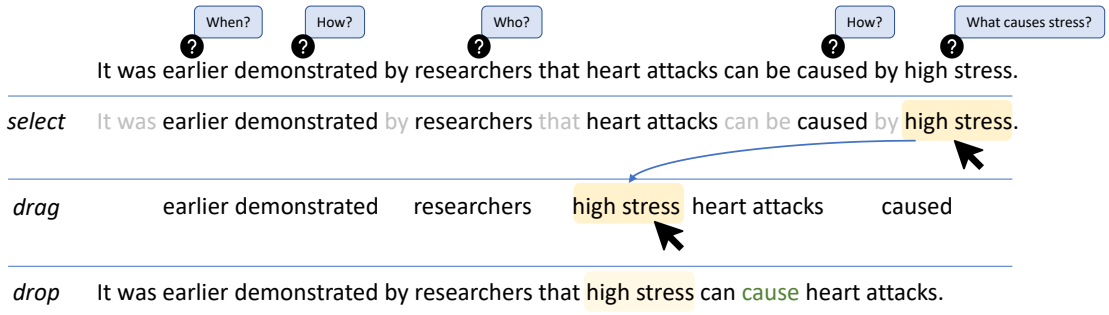


**Figure 1:** Mock-up of two of the interactions proposed in this paper. At top, example of questions generated to encourage the writer to elaborate and clarify. Below, an example direct manipulation interaction for rearranging a sentence by dragging and dropping a selected phrase. During dragging, non-content words are dimmed. Dropping the phrase constrains its relative position; green highlights indicate other words that the language model needed to modify or move.

terfaces allow exploration of alternatives to single words using a contextual thesaurus [5]. Still others provide indirect controls over the the system's edits [6, 7, 8].

Instead, we draw inspiration from systems like GAN-Paint [9] and collaborative summarization [10] that allow direct manipulation of creative outputs using high-level tools. This kind of interaction can be called Collaborative Semantic Inference (CSI) [10]: the interaction provides human-understandable hooks into the model's inference process, enabling the human and the generative model to collaborate in the process of editing an image or writing a summary of an article. Our approach conceptually extends the collaborative summarization system described in the CSI paper to manipulation types that are helpful in different kinds of writing tasks. For example, their work described summarization tasks, in which the selection and paragraph-level organization of information is important; we focus instead on helping the writer craft the organization and expression of information within a given sentence. Moreover, in a summarization task much of the content is determined by the source documents, but we focus on cases where the writer wants to be the ultimate author of most of the content.

## 2.1. Goal: Phrase Reordering

We focus our work here on helping writers vary sentence structure in informative writing tasks, specifically on *reordering phrases in a sentence while preserving the overall meaning*. The Oxford Essential Guide to Writing discusses the importance of both recurrence and variety in sentences. Recurrence, which is the repetition of the same sentence structure, can be used to highlight parallel ideas. However, the overuse of recurrence leads to monotony and a lack of focus. One method of adding variety to a piece of writing is to vary sentence openings. Instead of always starting a sentence with the subject, it could begin with a different construction such as prepositional phrase or subordinate clause [11]. In many sentences, such constructions could be reordered without affecting the general meaning of the sentence. So our design goal is to enable writers to easily rearrange phrases in sentences they have written.

We have explored several potential interaction techniques for manipulating phrase order in sentences. Figure 1 shows a direct manipulation interaction: the writer drags a selected phrase to a new location and the system performs the necessary edits on the rest of the sentence (highlighted in green in the figure). However, in this interaction it is not obvious to the writer which manipulations are likely to be successful, so we also explored interactions in which the system presents several plausible reorderings that the writer can choose among and then refine using further selection operations (Figure 2).

## 2.2. Approach

Round-trip machine translation via a pivot language is a common and effective approach for paraphrase generation (sometimes called sentence rewriting) [12, 13]. The pivot sentence captures much of the original meaning of the source-language sentence without constraining the ordering or word choice. Thus, the model can retain the original meaning as much as possible even when the words it can generate are manipulated and restricted in certain ways. We implemented an English-to-English paraphrase generation model using Spanish as a pivot language (because it was understood by members of our team and it is related to English) and pretrained translation models.[1] Since we can use these models directly

[1]Models were downloaded from https://huggingface.co/Helsinki-NLP/opus-mt-en-ROMANCE and https://huggingface.co/Helsinki-NLP/opus-mt-ROMANCE-en. They are seq2seq models with a BART-like architecture, pre-trained using Marian

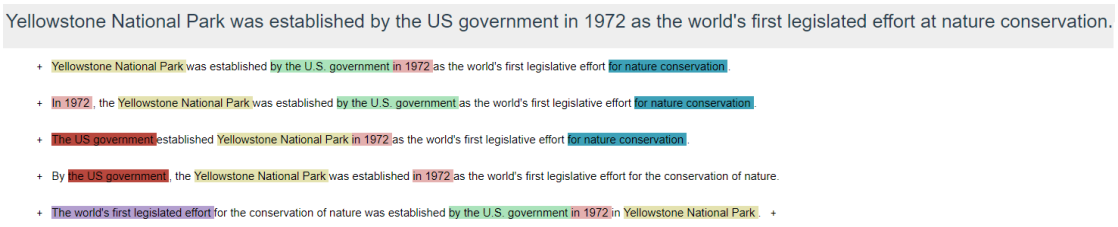


**Figure 2:** A selection interaction for clause reordering within a given sentence. The example (unedited output of our prototype system) shows five different possibilities for which phrase opens the sentence. Phrases are given colored backgrounds to visualize the relationships between the alternatives. Each alternative can be expanded to show alternative completions of that phrase.

without fine-tuning, our approach is "plug-and-play."

All of the interactions we explored require obtaining a set of high-quality paraphrases that are diverse in clause ordering. While in theory these different orderings would all have reasonable probabilities under the conditional generative model, unmodified beam search tends to find a set of very similar outputs, with only minor variations in words towards the end of the sequence. Generating alternatives that meaningfully differed in clause order would have required unreasonably large beam sizes. Although methods such as Diverse Beam Search [16] have been developed to address this problem in general, we focused our implementation effort on approaches that could be directly controlled in interpretable ways.

We started with a simplified reordering task: varying the opening clause of a sentence. Varying openings can express many meaningful variations of a sentence, such as choosing between active and passive voice. We solved this problem using a three-step approach. First, the system identifies clauses that could be moved to be openers, specifically noun phrases, prepositional phrases, and adverbial clauses, using spaCy [17]. Then it applies language-specific heuristics to edit these phrases to be appropriate as sentence prefixes. For example, object pronouns like "me" are replaced with the corresponding subject pronouns ("I"), and the first word is capitalized. Finally, the system uses beam search to generate the most likely completions of each selected starting phrase.

A problem we faced was that some sentence completions would duplicate or skip information from the source sentence. For example, sometimes the phrase selected as a prefix would recur later in the sentence. To avoid such results, we postprocessed the beam search results to penalize discrepancies between the number of occurrences of each content word in the original sentence and in the generated output.[2]

The interface (Figure 2) displays the most likely completion for each prefix. To help the writer understand the alternatives at a glance, it color-codes each clause; the order of colors serves as a glanceable summary of how each alternative sentence has edited the original.

It shows a variety of types of alternatives. First, for each prefix, it offers alternative full completions. Also, for each generated word, it shows the 10 most likely alternatives offered by the language model (conditioned on left context only, because of the limitation of left-to-right generation). Clicking on any option causes it to be chosen (regenerating the rest of the sentence if necessary) and the resulting sentence can continue to be edited.

## 2.3. Initial Evaluation

Although our system's implementation is not yet sufficiently refined for formal evaluation with users (e.g., its response latency is too high), an informal offline evaluation of its outputs shows promise for its ability to support various semantic editing operations.

To test our approach, we drew from a characterization of paraphrases by Bhagat and Hovy, who classify paraphrases and quasi-paraphrases into 25 categories and provide simple examples of each [18]. We used one sentence from each of the example pairs from that study as input and the other as a target. A test is successful if the target sentence can be displayed within one or two actions by a hypothetical user. In most cases, this means it is one of the 5 alternatives displayed after the user changes a word to one of the other top 10 predictions for that word. Of the 25 examples, there were 16 successes, all but one occurring after just one interaction. The categories for these successes are synonym substitution, change of voice, change of person, pronoun substitution, ellipsis, function word variations, actor/action substitution, verb/"semantic-role noun" substitution (successful in 1 of 2 tests), manipulator/device substitution, general/specific substitution, part/whole substitution, verb/noun conversion, noun/adjective conversion, change of tense, and change of aspect.

A limitation of this evaluation approach is that the

NMT [14] on parallel corpora collected by the OPUS project [15] and graciously shared with the public by the Helsinki NLP group.

[2]We defined content words as tokens not in spaCy's stopword list and not punctuation.

most of the paraphrases considered by Bhagat and Hovy consist of lexical substitutions rather than phrase reordering. Since our system's automatically generated alternatives were designed to emphasize differences in phrase order, few of the target paraphrases could be made by a single selection. So we also informally tested sentences pulled from other sources. We found that sentences from informative genres (such as Wikipedia articles) often generated acceptable paraphrases, but sentences from narrative genres such as fiction were more likely to suffer from a changed meaning. Nevertheless, results from all sources tended to be grammatical and understandable.

### 2.4. Discussion and Future Work

We believe our work falls in line with many of the design guidelines for co-writing tools suggested by Calderwood et al. at last year's HAI-GEN workshop in their study of novelists employing generative language models within their writing [1]. The writers within this study unanimously drew attention to their co-writing tool's tendency to deviate from their preconceived direction leading the study to synthesize design guidelines which include, in part, providing many suggestions that may be swapped out or replaced frequently, putting these suggestions into categories as writers often already have a certain type of suggestion in mind, and allowing the interface to actively or passively be aware of the type of suggestion being requested. Our proposed system would give the writer a large amount of suggestions while also providing agency in choosing the category of suggestion. The use of drag and drop would also allows writers to actively pursue their desired text suggestions adhering to the guideline of allowing the interface to be passively or actively aware of the type of suggestion being requested.

One direction of improving this work is improving the process of searching for high-quality rewrites given ordering constraints. Techniques such as Diverse Beam Search [16] or the Gumbel Top-k trick [19] could help the search explore a wider range of possibilities that are likely to satisfy the ordering constraints. Improved techniques for tracking sequential constraints would enable more principled ways to control the search process [20, 21].

Improved language modeling could improve both the quality and capability of the outcomes. The use of a single pivot sentence sometimes leads to translationese bleeding into the output; this could be reduced by pivoting through several different target languages as proposed by Mallison [12]. We are also investigating decoder-only architectures such as GPT-3 and the approach of Guo et al. [22], possibly with an approach like prefix-tuning [23] to direct the output sequence. Finally, since the autoregressive setting is limiting for editing operations, we have begun to explore applying language models with flexible ordering such as XLNet.

## 3. Supporting Drafting by Interviewing the Writer

Sentence rearrangement facilitates exploration of how to express an idea already committed to, but by design does not assist the writer in forming ideas in the first place. Generative models have often been used to provide ideas by generating target text [1], but this interaction turns the writer into an editor. To not supplant the human as the primary authors, we propose a different approach, inspired by how humans help each other express their ideas: like a skilled interviewer, the system generates *questions* to encourage the writer to elaborate or clarify their points or to discuss new topics.

Interventions that provide goal-oriented guidance to writers have shown benefits to the quality of the final result. For example, fourth- and sixth-grade students produced more effective essays when provided with a list of subgoals appropriate for argumentative writing, such as "You need to explain why those reasons are good reasons for your opinion" [24]. Structure-based planning, in which writers are given high-level goals to organize their outlining, may improve text quality [25]. IntroAssist [26] uses checklists paired with annotated examples, both generated by experts, to scaffold writers in an uncommon but high-impact writing task.

Existing interventions are either specific to a certain kind of document or provide only shallow support to a range of documents. Language modeling presents an opportunity to scale these kinds of interventions in two ways: (1) to a wider range of document types and (2) to more targeted guidance within those documents.

### 3.1. Design Study

In an exploratory study on encyclopedic writing, we compared the approach of giving guidance in the form of questions ("Questions") with with two alternatives: no guidance, and exemplar sentences ("Snippets") from high-quality related documents. Results of the study, summarized in Figure 3, suggested that while both types of guidance helped productivity, writers found Questions to be more relevant than Snippets and subjectively preferred them.

#### 3.1.1. Task

We designed a scenario for which writers would need to write isolated sentences in encyclopedia-style writing with optional guidance from a "bot." The hypothetical premise was that Wikipedia needs to rewrite many articles because of licensing issues, so they designed two bots (corresponding to "Questions" and "Snippets" but identified to participants by number only) to make suggestions based on existing articles.

| | |
|---|---|
| **Snippet** | "Blade Runner" initially underperformed in North American theaters and polarized critics; some praised its thematic complexity and visuals, while others were displeased with its slow pacing and lack of action. |
| **Questions** | How did it initially perform? How did critics react? What aspects did critics praise? What aspects did critics condemn? |

**Table 1**
An example of the Snippet vs Questions presentations of the same prompt used in our exploratory study. This prompt was from the "film" category, taken in this case from the Blade Runner article.

#### 3.1.2. Participants

We recruited 30 participants from MTurk. Each selected a book, a film, and a travel destination of their choice, then wrote 10 sentences about each. For each sentence, participants were given a fixed set of 10 prompts in a fixed order. Prompt presentation was counterbalanced between the three levels (Questions, Snippets, and a level in which no prompts were given). For each prompt, the writer was first asked whether the prompt gave them an idea about what to write for their article. If they answered Yes, they were then asked to write a sentence. Participants were instructed not to worry about ordering or flow between sentences, and were instructed to invent plausible information if necessary.

#### 3.1.3. Stimuli

For this exploratory study, we chose 30 prompts non-adaptively: For each sentence in a selection of Wikipedia Featured Articles[3] in each of the categories (book, film, or travel destination), one of the authors attempted to identify a single clear question that it answered, which was typically straightforward for these encyclopedic texts. We then picked the 10 sentences for which the identified questions seemed most relevant to similar articles. Table 1 gives an example of a prompt: the Snippets condition presented the original sentences; the Questions condition instead showed the question we wrote based on the snippets. Since the prompts may have been based on a very different genre, the study can measure how *robust* the interaction technique is to relevance failures for a future adaptive prompt generation technique.

#### 3.1.4. Results

We found that prompts shown as Questions gave usable ideas more often than prompts shown as Snippets, and that writers expressed strong preference for Questions over Snippets presentation (Figure 3). Likelihood ratio tests in a binomial mixed model predicting number of prompts marked as "relevant" found a significant effect of Presentation ($\chi^2 = 48.99$, p<.0001) and category relevance ($\chi^2 = 7.35$, p=.007), but no interaction between the two ($\chi^2 = 3.75$, p=.05). In this analysis, both Participant and Task were treated as random effects.

[3]For travel destination, we used used "star city" articles on WikiVoyage. We omitted History sections of Wikivoyage articles and plot summaries in book and film articles since these are much more highly idiosyncratic.

### 3.2. Feasibility of Question Generation

Since participants found prompting questions to be both relevant and useful in drafting tasks, we turn now to investigating approaches to generating questions. The desired system would take a partially written document and generate questions that are topically relevant and not yet answered. These questions could focus on eliciting new types of information (like the prompts of our design study) or on elaboration or clarification of already-written material. The latter task may be well handled by learning to tag phrases with a small set of "wh"-questions. However, the former task requires a more general approach to generating questions. Prior approaches and datasets for generating questions typically focus on specific factual questions, often for reading comprehension assessment (e.g., [27]), which leads to questions that may not generalize to as-yet-unwritten sentences. For example, SQuAD v2 [28] includes questions like "What is produced when the features of passive solar architecture are customized to the environment?" or "Who is Beyoncé's biggest musical influence?", which (presumably by design) have only a small number of possible correct answers—not useful for giving a writer new ideas. Nevertheless, it may be feasible to generalize some of these questions; for example, the second question could be delexicalized to become "Who is the artist's biggest musical influence?".

Since any particular question may be applicable to a wide range of documents, we explored the feasibility of a hybrid (ML + crowdsourcing) algorithm to generate a collection of questions and identify which of those questions are relevant (and not yet answered) in a writing task. We clustered sentences within a collection of documents (e.g., English Wikipedia articles about films), ensuring that each cluster occurred in several different documents. We then picked several sentences close to the cluster centers and had a person (one of the authors) attempt to write a question that many of those sentences would answer; this process successfully produced a canonical question for many clusters. Therefore, if the system could identify

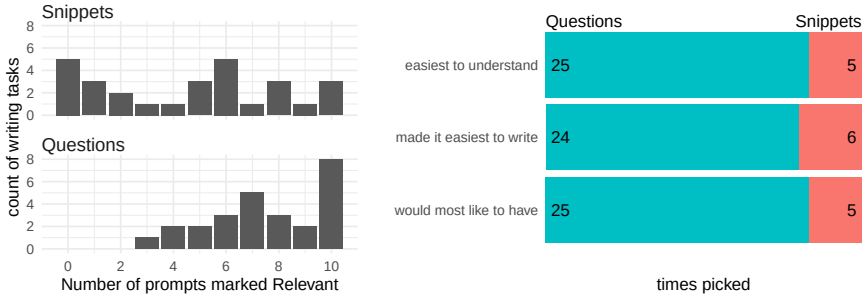


**Figure 3:** Left: Prompts presented as Questions were more often marked as relevant by participants. Right: Participants chose the Questions prompt as most preferable along all three measures asked.

a likely but not-yet-used cluster in a partially written document, it could ask the corresponding question to the writer. We found that a simplistic classification approach (Naive Bayes on cluster unigrams and bigrams) yielded a promising top-1 accuracy of 25% at predicting the cluster of a not-yet-seen sentence. So while much work remains to be able to reliably generate relevant questions, these results encourage us that is is feasible with current technology.

Alternatively, a language model like GPT-3 may have enough examples of interviews in its dataset (e.g., podcast transcriptions) to be able to be primed to generate sensible questions; we have applied for access to GPT-3 to evaluate this potential.

## 4. Discussion and Conclusion

We have presented two design concepts embodying the value that all meaning in the resulting writing should originate with the human author. Our systems eschew generation of novel text that a writer could appropriate directly. Instead, we present novel ideas as questions, not answers—making writing more like a conversation. Refinements are presented as a visual language for manipulating existing text. Generation is constrained to follow the semantics of text that the author provides.

Like many writing assistance technologies, the systems proposed here are dual-use: they can help writers clarify their ideas and make them intelligible to specific audiences, or they can be used to disguise plagiarism or for "article spinning" by content farms. However, existing countermeasures would readily detect paraphrases generated by our systems.

These proposed interactions barely scratch the surface of how high-capacity generative models of language can help writers. For example, our interactions support just a few of the many challenges that writers face when drafting and editing, and none of the challenges that writers face when revising (molding a document to achieve a desired goal) or other tasks such as giving or receiving feedback from others. However, language models can be of great help in these and many more tasks if we continue to think creatively about what we might ask them to generate for us.

## Acknowledgments

Prof. Krzysztof Z. Gajos and members of the Intelligent Interactive Systems group at Harvard provided valuable formative feedback on part of this work and supported the MTurk experiments. This work was funded in part by a Calvin Research Fellowship and a Jansma Family Research Fellowships in the Sciences. We are grateful to the contributors to the Huggingface Transformers project, especially the Helsinki NLP group, for making easy-to-use APIs for pre-trained models.

## References

[1] A. Calderwood, V. Qiu, K. I. Gero, L. B. Chilton, How novelists use generative language models, in: HAI-GEN Workshop at IUI 2020, 2020.

[2] K. C. Arnold, K. Chauncey, K. Z. Gajos, Sentiment Bias in Predictive Text Recommendations Results in Biased Writing, in: Graphics Interface 2018, Toronto, Ontario, Canada, 2018, pp. 8–11. URL: http://graphicsinterface.org/wp-content/uploads/gi2018-7.pdf.

[3] K. C. Arnold, K. Chauncey, K. Z. Gajos, Predictive text encourages predictable writing, in: Proceedings of the 25th International Conference on Intelligent User Interfaces, IUI '20, Association for Computing Machinery, New York, NY, USA, 2020, pp. 128–138. URL: https://doi.org/10.1145/3377325.3377523. doi:10.1145/3377325.3377523.

[4] R. Grundkiewicz, C. Bryant, M. Felice, A crash course in automatic grammatical error correction, in: Proceedings of the 28th International Conference on Computational Linguistics: Tutorial Abstracts, International Committee for Computational Linguistics, Barcelona, Spain (Online), 2020, pp. 33–38. URL: https://www.aclweb.org/anthology/2020.coling-tutorials.6. doi:10.18653/v1/2020.coling-tutorials.6.

[5] K. I. Gero, L. B. Chilton, How a stylistic, machine-generated thesaurus impacts a writer's process, in: Proceedings of the 2019 on Creativity and Cognition, C&C '19, Association for Computing Machinery, New York, NY, USA, 2019, pp. 597–603. URL: https://doi.org/10.1145/3325480.3326573. doi:10.1145/3325480.3326573.

[6] R. Louie, A. Coenen, C. Z. Huang, M. Terry, C. J. Cai, Novice-AI Music Co-Creation via AI-Steering Tools for Deep Generative Models, Association for Computing Machinery, New York, NY, USA, 2020, p. 1–13. URL: https://doi.org/10.1145/3313831.3376739.

[7] A. Fan, D. Grangier, M. Auli, Controllable abstractive summarization, in: Proceedings of the 2nd Workshop on Neural Machine Translation and Generation, 2018, pp. 45–54. URL: https://arxiv.org/abs/1711.05217.

[8] K. Gero, C. Kedzie, J. Reeve, L. Chilton, Low level linguistic controls for style transfer and content preservation, in: Proceedings of the 12th International Conference on Natural Language Generation, Association for Computational Linguistics, Tokyo, Japan, 2019, pp. 208–218. URL: https://www.aclweb.org/anthology/W19-8628. doi:10.18653/v1/W19-8628.

[9] D. Bau, H. Strobelt, W. Peebles, J. Wulff, B. Zhou, J. Zhu, A. Torralba, Semantic photo manipulation with a generative image prior, ACM Transactions on Graphics (Proceedings of ACM SIGGRAPH) 38 (2019).

[10] S. Gehrmann, H. Strobelt, R. Krüger, H. Pfister, A. M. Rush, Visual interaction with deep learning models through collaborative semantic inference, IEEE Transactions on Visualization and Computer Graphics 26 (2020) 884–894. doi:10.1109/TVCG.2019.2934595.

[11] T. Kane, The Oxford Essential Guide to Writing, Berkley reference, Berkley Books, 2000, p. 238. URL: https://books.google.com/books?id=tTgjAQAAIAAJ.

[12] J. Mallinson, R. Sennrich, M. Lapata, Paraphrasing revisited with neural machine translation, in: Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers, Association for Computational Linguistics, Valencia, Spain, 2017, pp. 881–893. URL: https://www.aclweb.org/anthology/E17-1083.

[13] C. Federmann, O. Elachqar, C. Quirk, Multilingual whispers: Generating paraphrases with translation, in: Proceedings of the 5th Workshop on Noisy User-generated Text (W-NUT 2019), Association for Computational Linguistics, Hong Kong, China, 2019, pp. 17–26. URL: https://www.aclweb.org/anthology/D19-5503. doi:10.18653/v1/D19-5503.

[14] M. Junczys-Dowmunt, R. Grundkiewicz, T. Dwojak, H. Hoang, K. Heafield, T. Neckermann, F. Seide, U. Germann, A. Fikri Aji, N. Bogoychev, A. F. T. Martins, A. Birch, Marian: Fast neural machine translation in C++, in: Proceedings of ACL 2018, System Demonstrations, Association for Computational Linguistics, Melbourne, Australia, 2018, pp. 116–121. URL: http://www.aclweb.org/anthology/P18-4020.

[15] J. Tiedemann, Parallel data, tools and interfaces in OPUS, in: N. Calzolari, K. Choukri, T. Declerck, M. U. Dogan, B. Maegaard, J. Mariani, J. Odijk, S. Piperidis (Eds.), Proceedings of the Eight International Conference on Language Resources and Evaluation (LREC'12), European Language Resources Association (ELRA), Istanbul, Turkey, 2012.

[16] A. Vijayakumar, M. Cogswell, R. Selvaraju, Q. Sun, S. Lee, D. Crandall, D. Batra, Diverse beam search for improved description of complex scenes, in: Thirty-Second AAAI Conference on Artificial Intelligence, 2018. URL: https://www.aaai.org/ocs/index.php/AAAI/AAAI18/paper/view/17329/16334.

[17] M. Honnibal, I. Montani, S. Van Landeghem, A. Boyd, spaCy: Industrial-strength Natural Language Processing in Python, 2020. URL: https://doi.org/10.5281/zenodo.1212303. doi:10.5281/zenodo.1212303.

[18] R. Bhagat, E. Hovy, Squibs: What is a paraphrase?, volume 39, 2013, pp. 463–472. URL: https://www.aclweb.org/anthology/J13-3001. doi:10.1162/COLI_a_00166.

[19] W. Kool, H. van Hoof, M. Welling, Stochastic beams and where to find them: The gumbel-top-k trick for sampling sequences without replacement, 2019. arXiv:1903.06059.

[20] J. E. Hu, H. Khayrallah, R. Culkin, P. Xia, T. Chen, M. Post, B. Van Durme, Improved lexically constrained decoding for translation and monolingual rewriting, in: Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers), Association for Computational Linguistics, Minneapolis, Minnesota, 2019, pp. 839–850. URL: https://www.aclweb.org/anthology/N19-1090.

doi:10.18653/v1/N19-1090.

[21] C. Hokamp, Q. Liu, Lexically constrained decoding for sequence generation using grid beam search, in: Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), Association for Computational Linguistics, Vancouver, Canada, 2017, pp. 1535–1546. URL: https://www.aclweb.org/anthology/P17-1141. doi:10.18653/v1/P17-1141.

[22] Y. Guo, Y. Liao, X. Jiang, Q. Zhang, Y. Zhang, Q. Liu, Zero-shot paraphrase generation with multilingual language models, 2019. arXiv:1911.03597.

[23] X. L. Li, P. Liang, Prefix-tuning: Optimizing continuous prompts for generation, 2021. arXiv:2101.00190.

[24] R. P. Ferretti, W. E. Lewis, S. Andrews-Weckerly, Do Goals Affect the Structure of Students' Argumentative Writing Strategies?, Journal of Educational Psychology 101 (2009) 577–589. doi:10.1037/a0014702.

[25] T. Limpo, R. A. Alves, Effects of planning strategies on writing dynamics and final texts, Acta Psychologica 188 (2018) 97–109. doi:10.1016/j.actpsy.2018.06.001.

[26] J. S. Hui, D. Gergle, E. M. Gerber, IntroAssist: A tool to support writing introductory help requests, Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems - CHI '18 (2018) 1–13. doi:10.1145/3173574.3173596.

[27] X. Du, J. Shao, C. Cardie, Learning to ask: Neural question generation for reading comprehension, in: Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), 2017, pp. 1342–1352.

[28] P. Rajpurkar, R. Jia, P. Liang, Know what you don't know: Unanswerable questions for SQuAD, in: Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers), 2018, pp. 784–789.