Computational assistants for partially constrained writing tasks

Thesis proposal

Katy Ilonka Gero
Department of Computer Science
Columbia University
katy@cs.columbia.edu

August 22, 2022

Abstract

From journalism to research papers, speculative fiction to romance stories, writing underlies a wealth of cultural and political pursuits. As language technology improves, so does its potential to aid our writing – to make us write faster, clearer, even more creatively. According to psychology research on writing, the act of writing is propelled by goals, which are created by the writer and grow in number as the writing progresses. But typical writing assistants tend to support either highly constrained goals, like typing out a common email exchange, or highly unconstrained goals, like suggesting the next sentence in an open-ended story. They struggle to support the majority of our writing, where we have some constraints – like a very technical topic, or the complex content of what we have already written – but are not writing oft-repeated phrases. In this proposal, I define an under-explored research area of supporting *partially constrained writing goals* and propose designing writing assistants that focus on supporting these goals. In partially constrained situations, the writer must come up with new ideas within an existing, restrictive context. The writing assistants must then address the challenges of both computational creativity (generating novel and useful ideas) as well as natural language understanding (evaluating language meaning in context). I present two systems that address these challenges with a variety of techniques, and results from both quantitative and qualitative studies to demonstrate the effectiveness of these systems. In addition to these systems, I propose running a two month, longitudinal study of system use, as writing assistants are typically studied only in short-term, lab-based studies, limiting our ability to understand how they support writing in the long-term. This thesis opens up a new area for writing assistants, and through my work I demonstrate that writers make use of these writing assistants in varied, subtle, and powerful ways. Understanding this will further our ability to design creative, contextual writing assistants in the future.

Contents

1 Introduction

From journalism to research papers, speculative fiction to romance stories, writing underlies a wealth of cultural and political pursuits. As language technology improves, so does its potential to aid our writing – to make us write faster, clearer, even more creatively. According to psychology research on writing, the act of writing is propelled by goals, which are created by the writer and grow in number as the writing progresses [15]. Typical writing assistants tend to support either narrowly constrained goals, like typing out a common email exchange [28], or open-ended goals, like suggesting a sentence for a new story [10].

But much writing entails what I define as *partially constrained* goals. These goals are less constrained than something like writing a common email exchange, but more constrained than an open-ended story. We might want to make an existing description more visual, or figure out what a character will do next, or decide how to best explain some jargon. We might be wondering if our audience will know about a particular topic already, or how write an exciting first sentence that introduces our research topic. Partially constrained goals have many possible solutions – many possible sequences of words that would achieve the goal – but must still adhere to strict constraints.

Commonly used writing assistants of today focus on generating a continuation of already-written text. In gmail, if I type, 'Great to meet you. I'd love to have coffee. What time next week' then it will suggest 'works best for you?' because thats the most common sequence of words given the preceding ones [8]. In more creative contexts like storytelling, a writing assistant typically suggests the next phrase or sentence given what the writer has just written [51]. These systems struggle to support the majority of our writing because most of our writing is neither fully open-ended nor narrowly constrained.

In this thesis, I propose designing writing assistants that focus on supporting partially constrained writing goals, where the writer needs to come up with new ideas within an existing, restrictive context. These assistants must address the challenges of both computational creativity (generating novel and useful ideas) as well as natural language understanding (evaluating language meaning in context). I use a variety of techniques to address these challenges, and run user studies to demonstrate the effectiveness of these systems.

My work on writing assistants fits within a larger *design space* [65] of writing assistants, where areas of highly-constrained writing, like spelling correction or word completion, as well as lightly or even unconstrained writing, like open-ended storytelling, are already well-explored. This thesis opens up a new area for writing assistants, and through my work I demonstrate that writers make use of these writing assistants in varied, subtle, and powerful ways. Understanding this will further our ability to design creative, contextual writing assistants in the future.

1.1 Overview of Proposal

In section 2, I review relevant work on the psychology of writing and design space exploration. I present my work on how professional novelists make use of large language models, indicating the various ways text continuation as a writing assistant paradigm can fall short, and highlight my work on how people develop mental models of AI systems, discussing the implications of these mental models on the design of writing assistants. Given this, I present a design space for writing assistants, which introduces a 2-axis representation of goal magnitude v. level of constraint, and I review existing writing assistants using this design space, showing which areas of the design space

are currently under-explored and motivating my focus on partially constrained writing situations.

In section 3, I report on my own work designing writing assistants. In particular I report on a writing assistant for explaining abstract concepts with metaphors, where we generate metaphorical connections between any two concepts, and a writing assistant for explaining technical concepts, where we use a large-scale language model to generate ideas for the writer. For each system, I report on the results of a quantitative study – for metaphor generation we compare the system to existing metaphor generation systems, for explaining technical topics we compare to a competitive baseline and a human-written gold standard – as well as the results of a qualitative study – for metaphor generation we study professional poets, for explaining technical topics we study STEM Ph.D. students.

Finally, in section 4, I propose a final component of my thesis that builds upon my work on writing assistants for explaining technical concepts. I plan to run a two month, longitudinal study of how a writing assistant helps climate scientists write about their own work. Typical studies of writing assistants are one-hour lab studies, which lack the ability to discern the long-term impacts of the technology. What does the 'learning curve' of writing assistants look like? Do writers' find writing assistants more or less useful over time? This study will allow us to understand writing assistants in a more realistic context.

1.2 Contributions

To summarize, the completed and proposed contributions of this thesis are:

- Experimental and theoretical demonstrations of open research areas in the design of writing assistants:
  - An exploratory study showing that prompt completion with a large language model is inadequate for professional novel writers.
  - A framework for how people develop mental models of AI, and a discussion of how knowledge distribution is the main unknown for users interacting with large language models.
  - A design space for writing assistants that indicates partially constrained writing goals as an under-explored design area.
- Quantitative and qualitative results on how writing assistants can support partially constrained goals:
  - A system for supporting metaphor writing, and two studies – one quantitative, one qualitative – demonstrating how the system supports the partially constrained writing goal of describing a user-defined abstract concept.
  - A system for supporting the explanation of technical topics, and two studies – one quantitative, one qualitative – demonstrating how the system supports the partially constrained writing goal of explaining a user-defined technical topic.
  - A longitudinal study of the system for supporting the explanation of technical topics, reporting on how climate scientists use this system over the course of two months.

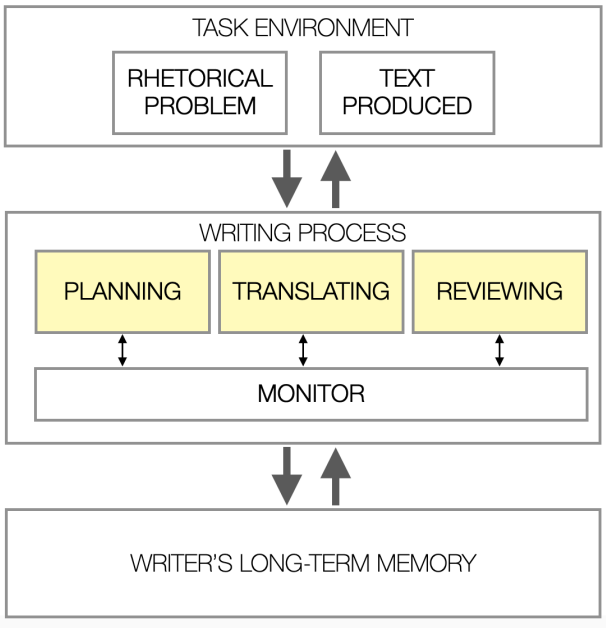


Figure 1: The cognitive process model for writing, as proposed by Flower and Hayes [15].

Together, these contributions provide both an understanding of how to design powerful new writing assistants, as well as a series of system-based user studies demonstrating how writers make use of novel writing assistants in a variety of partially constrained tasks. The hope is that this thesis furthers our understanding of how language technology can serve writers, and more generally contributes to the growing field of human-AI interaction.

2 A Design Space for Writing Assistants

2.1 Related Work

2.1.1 Cognitive Models of Writing

Flower and Hayes' theory of the cognitive processes involved in writing lay the groundwork for a plethora of research on the psychology of writing over the past four decades [15]. This process model, backed by empirical studies, proposed that writing is best understood as a set of distinct thinking processes which are hierarchical. Figure 1 shows a schematic of the model, with the three main writing processes – planning, translating, and reviewing – highlighted in yellow. When Flower and Hayes state that these processes are hierarchical, they mean that they can be called upon iteratively, being embedded within each other. For example, when a writer is constructing a sentence ('translating'), they may call in a compressed version of the entire writing process. Flower and Hayes' are also quick to note that these processes are not linear. While a common mantra is to plan, write, and review, in reality writers are making plans and reviewing what they have written all throughout the writing process.

Along with this process model, Flower and Hayes proposed that the act of writing is propelled by goals, which are created by the writer and grow in number as the writing progresses. These

goals, which span in complexity and level of abstraction from 'appeal to a broad audience' to 'don't use that cliche', are what direct the writer to different processes. Thus we can model the writing process by considering the writer's goals and what processes they enlist to achieve these goals.

While this model has since been updated with an increase in complexity – Hayes adds much more detail to the long-term memory component, and adds components for working memory and the motivation and affect of the writer [23] – considering how goals propel the writing process remains a useful model. Writing has long been considered a mode of learning, as it is both a process and a product, which allows near-constant reflection on the ideas the writer is trying to express [14]. By considering a writer's shifting goals, writing researchers have understood why mature writers are able to learn from their writing [54]. Immature writers are often bogged down with low-level goals like sorting out syntactical issues or ensuring topical cohesion, which does not allow cognitive effort to be directed to more high-level goals. In contrast, mature writers are able to, when appropriate, set goals that require new knowledge to be generated. For instance, a mature writer may realize, when writing down an argument, that there is a logical gap and set about bridging this gap, thus turning writing into a learning activity.

I make use of this theory to understand what existing writing assistants actually help with, and how we might design new writing assistants. To do that, I use this theory to structure a 'design space' of writing assistants.

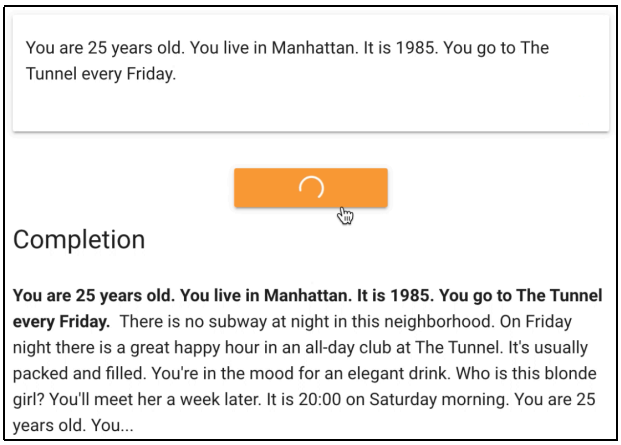
2.1.2 The Design of Design Spaces

MacLean et. al. describe *design space analysis* as an approach to representing design rationale [37]. Design space analysis places a design in a "space of possibilities" and uses this placement to explain why a design was chosen among all the various possibilities. They frame design spaces as a useful way of communicating with various stakeholders – not just designers but also salespeople, system maintainers, and various types of users. By explaining why a design was chosen, these various stakeholders can better sell, maintain, and otherwise interact with a product.

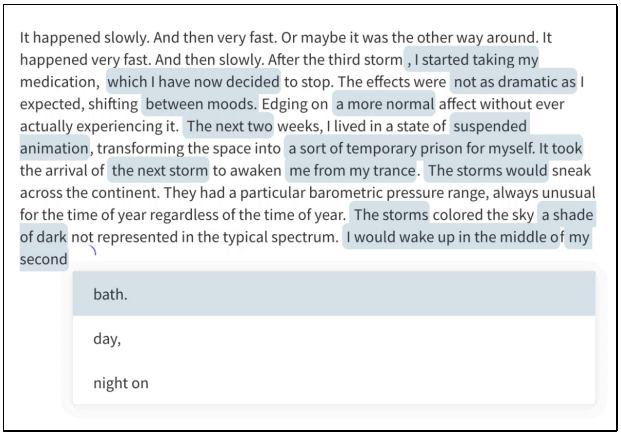
Woodbury and Burrow, addressing the growing popularity of design spaces in computational research, describe *design space exploration* as the idea that we can use exploring alternatives as a compelling model of design [65]. This involves representing designs in a meaningful way and arraying them in the resulting design space. This representation can be used by designers to explore the space, and can be used to build computer systems that can aid designers in the exploration.

A popular and highly-cited example of a design space comes from wireless sensor networks [53]. As the use of wireless sensor networks increased globally, it was found that it was very difficult to discuss specific application requirements, research directions, and challenges. The proposed solution was a sensor network design space: its various dimensions would be categorized in order to both understand the existing research as well as discover new designs and applications. The resulting space had 12 dimensions, and the initial paper categorized 15 systems. One conclusion of the paper was that a small set of platforms could cover the majority of the design space, rather than requiring numerous, application-specific platforms.

In this thesis I use design spaces both to think about what writing assistants currently do, and what we might want writing assistants in the future to do. In this sense I take both MacLean's and Woodbury's view: the design space is a way to talk about why existing writing assistants are the way they are, as well as a way to design new writing assistants.



(a) The 'Talk to' interface being used by a study participant.



(b) The 'Write with' interface being used by a study participant.

Figure 2: Comparison of the two interfaces used in the user study. While the 'Talk to' interface (a) gave longer suggestions, writers preferred 'Write with' (b) which allowed them to easily insert suggestions into the text document.

2.2 An Exploratory Study of Novelists Using GPT2

I report on an exploratory user study of four novelists writing in collaboration with a large language model [7]. Our goal was to understand what professional writers look for in generated suggestions, and in what ways these new language models do or do not meet this challenge.[1]

2.2.1 Methodology

We recruited four published novelists for our study, and observed them complete various tasks that had them interact with generative writing tools. Each novelist completed the study individually in an hour long session. Three of the writers had no previous exposure to the interfaces studied; one writer had been previously exposed but only briefly, and not for his professional writing. We first introduced the writing tools studied, and then described the study procedure.

The two interfaces chosen for the study were Talk To Transformer[2], and Write With Transformer[3], referred to as 'Talk to' and 'Write with' respectively. Both user interfaces rely on GPT-2 [46] to predict the most likely sequence of words following some input text. Both take into account at most the last 256 sub-word tokens available, though in many cases there is not that much preceding text. GPT-2 was trained on the WebText corpus, which contains 40GB of text from over 8 million articles linked to by Reddit from before 2017 that received at least 3 votes. Figure 2 shows screen captures from our study in which the writers are using the two different writing interfaces.

'Talk to' uses a text completion paradigm where the user writes into a small, centered text box and presses a button to have the system generate a completion which is shown as plain text. 'Write with' is a word processor-like interface, and requires that the user presses the tab key to trigger text

[1] I directed this study, forming the research questions, leading the analysis, and finalizing the paper writing. The lead author on the paper performed the experiments and did a first pass on the analysis and paper writing.

[2] https://talktotransformer.com/

[3] https://transformer.huggingface.co/doc/gpt2-large

generation. Doing so will show a drop down menu with three short suggestions, usually between 1 and 10 words. Selected suggestions appear directly in line with their previous writing, highlighted blue, and is editable.

The procedure for the study was as follows:

1. Following a very brief description of the user interfaces, the participant was given open ended experimentation with both interfaces. (2 - 10 minutes)

2. The participant was then asked to write 'the most interesting' or 'the best' original piece of fiction that they were able to with the assistance of the interfaces. They were allowed to switch between the interfaces at will, but were asked to use both. (10 - 20 minutes)

3. The participant was then asked to work on an in-progress piece of writing with the assistance of the interfaces. They were told to try and solve an 'issue' they'd been having with a piece of writing. (10 - 30 minutes)

4. The participant was again asked to write 'the best' thing they could with 'Write with', with the constraint that they had to use a suggestion at least once every other sentence. (10-20 minutes)

We recorded and transcribed each session. Additionally, we recorded all text written, including text written by the machine, and for each generated suggestion annotated if it was 'accepted' by the writer.

2.2.2 Results

To preserve anonymity, we refer to the four writers in our study as W1-W4. All four writers chose to use 'Write with' when asked to write 'the best' original piece that they could in the allotted time. To explain the preference, they generally cited the lack of control and the higher degree of randomness associated with the longer text generated from 'Talk to'. We noticed that writers often triggered 'Write with' multiple times at a single point in the text if the resulting suggestions were not what they wanted. We found that 25% of all triggers were a repeated trigger, suggesting that once a writer triggered the system, they were invested in finding a useful suggestion.

Unanimously, the writers pointed out that the tools appeared to deviate from the direction they were taking their writing, particularly referring to the 'Talk to' interface. All writers were quick to point out instances that the system changed point of view (it seemed to prefer 1st person even when they were in 2nd or 3rd). W3 said "it's like improv. You have to 'yes, and.' " Meaning that if the generated text does not incorporate the prior facts of the piece, it is not constructive.

W1 and W2 noted that the tools were much better at following them into 'genre' writing than into the more nuanced and stylized writing they were interested in. For example, W2 set up a fantasy scene and found the suggestions were more coherent than normal. They were more likely to take the suggestions during Tasks 2 and 4, when they weren't writing something they had pre-conceived.

The main use case we observed was *description creation*. All four participants experimented with using 'Write with' to generate mid-sentence descriptions for items, scenes, or characters. All four writers learned through the course of the session that they could get 'Write with' to focus on filling in descriptions such as colors or character details by requesting suggestions after

prepositions, and actions by requesting suggestions after a noun phrase. They rejected adjective descriptions like colors more often than any other type of suggestion, often dismissing them as "boring" and limited, though W4 and W1 noted that more than three suggestions given could be useful at those moments.

The writers didn't see the tool as a meaningful generator for plot or characters. W4 noted that he was not a "spiritualist" writer, meaning that rather than let the flow of ideas come to him during the writing process, he usually sat down with a set of "points to hit". The majority of writers mentioned they could see something like this being useful for generating plot outlines for writing exercises, where the writer does not yet have many ideas in mind.

Sometimes the unexpected was useful, though again mostly in very specific contexts. At one point, W1 set up 'Write with' to describe the color of the sky, and it suggested "dark blue", "yellow", and "a shade of dark"; he accepted the last suggestion. This is an example of the system steering from a direction that the writer clearly wanted to pursue (hue description) into a related, but separate concept, describing a shade instead, for stylistic effect. Both systems frequently introduced characters or dialogue, which for Tasks 1, 2, and 4 produced comments from the writers like "I wasn't going to go there, but that's interesting", especially when it brought into play family members (sister, wife, father) the writer had not been considering introducing.

Overall, the writers in our study found the language model lacking when it came to their own writing because it struggled to conform to the constraints of both what had already been written as well as their internal ideas of where they wanted the story to go.

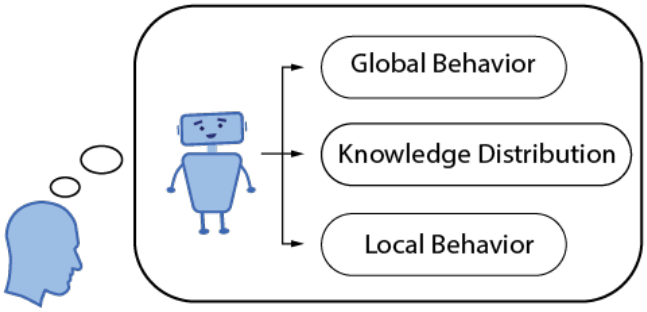
2.3 Mental Models of AI Agents

I ran two studies to investigate what appropriate conceptual models of AI systems look like and how users develop mental models of AI systems [19]. Norman [42] defines a *conceptual model* of a system as a model "invented to provide an appropriate representation of the target system" and notes that they tend to be developed methodically by experts. In contrast, he defines the *mental model* of a system that which is evolved by users through interaction with the system.

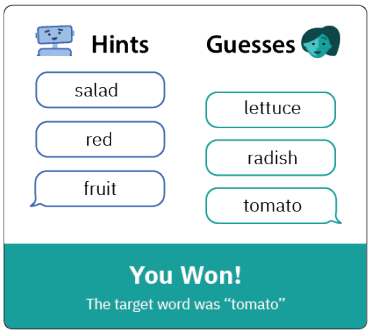
The first study was an in-person, think-aloud study, in which participants play a word guessing game with an AI agent while thinking out loud. Figure 3b shows an example round of gameplay. This study allowed me to identify the important aspects of a mental model and get a qualitative understanding of how people think about AI systems. The second was a large-scale online study, in which participants played 5 or 10 rounds of the game and then filled out a survey which probed their mental model of the AI agent. This study showed us who makes accurate estimations of the AI agent, and points us towards why these people do so.

The results of these studies allowed me to uncover three components participants used when modeling an AI agent. These components were developed iteratively through discussions by the entire team of researchers after the thematic analysis of the first study was conducted. These components are a framework for describing a conceptual model or a mental model.

Figure 3a shows the components. The components are: **Knowledge Distribution** which includes conceptions such as whether or not the AI agent knows about specific people or attributes, **Local Behavior** which includes conceptions of what kinds of hints the AI agent is likely to give or respond best to, and **Global Behavior** which includes conceptions of how the AI agent tends to play the game, such as what and how much the AI agent remembers from previous interactions.



(a) Framework for how people mentally model AI agents.



(b) Round of word guessing game.

It is hard to track the development of mental models, but in our studies we gained some insights into how users develop mental models of AI systems. The most common utterances in the think-aloud study had to do with anomalies, distress, and trust – people talked most about their mental model when something unexpected occurred. This is also where we saw the most revision; despite trying to explain an anomaly, when an anomaly persisted people did end up revising their model. Antonym-style hints in the word guessing game showed this clearly: most people were initially distressed by antonym hints that seemed to contradict the other hints presented, and some even thought that these hints were mistakes. However, after the game concluded they acknowledged that the behavior made sense, revising their mental model.

Miller [40], in his review of insights from social sciences for explainable AI, states two situations in which people desire explanation: 1) when a contradiction occurs, and 2) when shared meaning is desired. This dovetails nicely with our finding that people tend to revise their mental models in the face of anomalies. Considering how to design explanations for AI systems, our results confirm Miller's finding that we should provide explanations to people when anomalies occur, as this is when they are most open to revision and most desire an explanation.

There are important implications from this study for the design of writing assistants. Writers may have a poor understanding of the knowledge distribution of the tool they are using, overestimating some areas and underestimating others. It is also unclear how long or what kind of interaction would improve writers' understanding of a tool's knowledge distribution. In the case of language models, writers may also have a poor understanding of local behavior – why did a language model generate what it did given the inputs? Much research ink has been spilled on how to improve the outputs of language models through improved prompting alone [18, 32, 50], and it may be that writers need to develop some of their own intuition for this in order to best make use of this technology.

2.4 A Design Space for Writing Assistants

In this section, I propose a design space for writing assistants. This design space is based on my extensive work researching how writers can best make use of language technologies. According to Woodbury and Burrows, a design space is a tool for designers [65]. It's a lens through which to consider existing designs, and a way for designers to find new ones. The metaphor of a design "space" makes it natural to consider them graphically, where each design can be "plotted"

according to some meaningful axes. These axes are often called the design space representation. Good representations are both *vast*, in that they include a hyper-astronomical number of potential designs, and also *limited*, in that they allow for intentional and directed exploration [65]. I present a design space representation for writing assistants that has two axes: the magnitude of the goal being supported, and level of constraint of the goal.

2.4.1 Design Space Representation

In this subsection, I describe the two axes I define for the design space representation. They are 'magnitude' and 'amount of constraint'.

Magnitude, or size of writing goal. I used 'magnitude' to describe the amount of writing the goal is about. For instance, a large goal may be something like "write a story about a fairy" whereas a small one might be "correctly conjugate this verb". In writing psychology research, the largest goal is often called the 'rhetorical problem' – the impetus of the entire writing activity [15]. Flower and Hayes propose that writing goals are embedded within each other [15]. Embedded goals tend to be smaller than their parent: after considering how to "write a story about a fairy" a writer may then consider the smaller goal of "describe the fairy's home" and then perhaps "decide what color the fairy's house is" or even "choose a color word that's more specific than 'green"'.

We can imagine goals at the size of the word, the phrase, the sentence, the paragraph, and even the section or chapter (depending on the length of the writing). We might consider the computer science subfield of automatic story generation to be a writing assistant that aims to support the very large goal of writing an entire story, whereas certain autocomplete programs may aim to support the very small goal of typing out a long word.

Amount of constraint. Independent of magnitude, a highly constrained goal leaves little room for flexibility. A writer may be thinking about how to make a sentence grammatically correct, or how to tie up the plot of a story. These goals, though they may have more than one reasonable solution, are highly constrained. On the other end of the spectrum, a writer may sit down at a blank page to dream up a new character, or decide what will happen in the first chapter. While there may be some constraints in these goals, they are few and the space of reasonable solutions is large, perhaps even infinite.

Most goals sit somewhere between these extremes. Continuing the example of storytelling, most of the time spent writing a story is spent in the middle. Some aspects of the story are already in place, some characters are already introduced or the writer has defined them in their head, some settings and plot have been committed to. But the story is not finished. What might these characters do? Or what might happen to them? And at any point a writer may decide to revise past details, but they will not start from scratch. Instead, they revise only some details, and ensure the changes fit in within the existing writing.

2.4.2 Example Points in the Design Space

To better understand the *writing assistant design space*, I use the example of writing a fictional story. Figure 4 shows examples of writing goals at various points in the design space. We can imagine a writing assistant that attempts to support any one of these goals.

To aid in visually understanding this plot, I draw rectangles around goals of similar magnitude and use three 'levels' of magnitude: the word level, the paragraph level, and the story level. These

Figure 4: Example writing goals a writing assistant may support, plotted in the design space.

levels are (hopefully) quite intuitive to understand – they simply answer the question of how much writing a tool is expected to help with at any one moment.

So let's consider the 'level of constraint' axis in more detail. Very few, if any, writing goals are constrained to a single correct solution; perhaps the most constrained goal would be to correctly spell a long word. Even then, a writer may need to make some decisions between plausible solutions, like whether to use British or American spelling, or whether to hyphenate a word or not. However, I define such goals 'highly constrained'. Highly constrained goals have few solutions compared to the space of possibilities given the size of the goal. At the level of the sentence, a highly constrained goal may be typing out a common email exchange (a feature currently supported by Gmail [8]) or changing the name and pronouns of a character in a sentence already written. Such goals still have multiple solutions – variations on the common email exchange, decisions about when to use a name or pronoun or how to ensure correct coreferences – but the number is small compared to the number of possible sentences.

Compare those highly constrained, sentence-sized goals to this 'partially constrained' goal: describing an existing character. The writer is constrained by the characters attributes, which may have been previously defined in the story or may have been defined by the writer's other goals for the story. But there remain many options for the writer. Which attributes are relevant to describe now; how will the writer describe them; what is the structure of the sentence. There are many options available to the writer.

'Lightly constrained' goals have few constraints and the number of possible solutions is large, verging on infinite. What will the first sentence of a story be? Even with a prompt, the number of possibilities is gargantuan. What will the writer name a new character? Lightly constrained goals are easy to imagine in storytelling or poetry, where the writer may be unconstrained by

Figure 5: A plot of 13 writing assistants in the design space. My own work is highlighted in the center. Those system provide sentence-level assistant for partially constrained goals.

reality (though, again, as aspects of the story are introduced the reality of the story itself becomes constrained) however we may also consider non-fiction examples: what should we text to our friend, just to say 'hi'? A joke? A memory? A question? Such a writing goal is lightly constrained.

2.4.3 Putting Existing Writing Assistants in the Design Space

This design space can be considered in two ways: on one hand, what the designers' intention was for the assistant; on the other, how writers actually use the assistant. Because the evaluation for writing assistants can vary widely, and often does not focus on what kind of writerly goals the assistant supports (instead focusing on, for example, usability, likability, or how it impacts task completion) in this section I consider designers' intention for their assistant, as this is typically well-documented in research papers.

Looking at supporting large goals, in the upper left of Figure 5 there are systems that provide feedback on already-written non-fiction pieces, constrained both by the existing text and the external realities of the content [25, 8, 6, 36]. In the upper right, there are poetry generation systems that generate in response to very short prompts (typically noun phrases), lightly constrained by the prompt and the poetic form [41, 22].

Looking at supporting small goals, in the bottom left are phrase-level systems that support understanding or planning non-fiction phrase-level writing tasks [44, 26]. In the bottom right I have the GPT models [46, 5, 58] as they are intended to produce the next word (or token). The level of constraint of the GPT models, or really any language model, is potentially contentious. Language models are intended to produce a probability distribution over all possible next words, and how words are selected from this distribution is varied depending on the use-case. While they

are constrained by the input text, I argue this constraint is light because of the way these models are used – when considering beam search (a common decoding method) the space of possibilities being explored is huge.

Looking at supporting medium-sized goals, the closest work to my own is by Clark et al. [11] which presents two creative writing support systems, one that generates ideas for slogans (highly constrained by length, syntax, and the topic) and one that generates next sentences for stories (lightly constrained by previous sentence only).

My own work is highlighted in the center. These systems generate sentences that are less constrained than slogans, but more constrained than open-ended storytelling. I call this work *partially constrained* because it sits between highly constrained systems like those that generate feedback on already-composed pieces or those that work with common exchanges or well-defined genres, and those that are lightly constrained like generative poetry and storytelling systems.

3 Writing Assistants that Support Partially Constrained Goals

In this section I describe my work designing and evaluating writing assistants for partially constrained goals. For each assistant I report on the design of the system, a quantitative study where I evaluate system outputs compared to existing systems (independent of their usage), and a qualitative study where I evaluate the system being used by writers in a realistic writing situation.

3.1 Related Work

3.1.1 Language Technologies

These writing assistants make use of a variety of continually improving language technologies. All technologies have their strengths and weaknesses, which must be weighed when considering their application, and these technologies are constantly being developed further by the research community.

In Metaphoria I make use of word embeddings and knowledge graphs. Word embeddings [39] are learned vector representations of words, such that words with similar 'meaning' (which is defined by how the embeddings are learned) have similar vectors. This allows the distance between two word embeddings to be used as a proxy for semantic distance. Word embeddings are often trained using unlabeled text, such that words that occur in similar contexts – that is, have similar co-located words – have similar vectors.[4] Since they can be trained using unlabeled text, they can easily be created to reflect different language usage, like by training them on Wikipedia versus a corpus of tweets [45].

In contrast, knowledge graphs are discrete representations of word meanings, where each word (or concept) is represented by a node in the graph and labeled edges between nodes represent different relations. They often rely on hand-crafted data, where concepts are annotated by people as having specific relations to other concepts. ConceptNet is an extensive, open-source, and multilingual knowledge graph that makes use of several different data sources to provide extensive

[4]The selection of what counts as a co-located word, e.g. selecting words that are proximal in the sentence as written versus words that are proximal in the sentence's dependency graph, can change which words have similar vectors [30].

coverage of common concepts [35]. Newer work on knowledge graphs attempts to automatically learn these relations from unlabeled text, which allows them to be more easily updated [2].

In the past several years there has been extensive work on improving neural language models – models that assign a probability to a sequence of words. Language models can be used to generate text, and are the backbone of modern autocomplete systems [8, 28]. Language models can also be trained on unlabeled data, and the increase in size of corpora, amount of compute resources, and efficiency in the training process [58] has allowed neural language models to improve rapidly. Newer models are considered 'pre-trained' or 'multitask' models, as with minimal to no changes they perform well on a wide variety of tasks [46].[5]

As these language models improve, how to best use them in varied tasks has become an active area of research. Handcrafting [50] or automatically learning [18] prompts for the models has shown to improve outputs dramatically. Classification tasks especially benefit from learning continuous prompts for the task [55]. Chaining prompts [66], or using meta-prompts [50] has also been found to improve results. This continues to be an active and fast-moving area of research.

3.1.2 Creativity Support Tools

Creativity support tools have flourished for music and the visual arts, from the widespread adoption of software for generation and editing to the development of medium-specific programming languages [47, 33, 61]. These tools are beginning to tackle how to be compatible with existing manual practices [27], as well as how to be more compatible with current artificial intelligence frameworks [43, 13].

The way in which creativity support tools integrate with an artist's practice is at the heart of these issues. When a support tool provides more complete or conceptual contributions, or provides contributions without a request from the artist (as in mixed-initiative user interfaces [24]), the term co-creativity is often used. Critically, Davis defines human-computer co-creativity as when the "program is adapting to the input of the user" [12]. This distinguishes co-creative systems from more procedural contributions, in which an artist either has a high level of control over the outputs, as in a synthesizer, or little to no control over the outputs, as in a computer-generated poem based on a topic [22].

It is essential to think about tools as supporting artists in their desired practice, rather than replacing aspects deemed computationally tractable. Co-creativity emphasizes interaction in which all parties must feel control over the process and ownership of the result. Support for creative writing should align with the 'wide walls' design principle of creativity support tools, in which tools aim to "support and suggest a wide range of explorations" [49]. Unlike more specified writing tasks (such as writing an email to request help), creative writers do not want tools that will make their writing sound the same as others [56]. Thus, in co-creative domains, systems should be conducive to divergent outcomes.

[5] It is worth noting that the models themselves have also increased in size, making the financial and environmental costs associated with training them gargantuan [1]. It has also resulted in industry 'capture', since large corporations are the only organizations with enough data, compute, and financial resources to train the models [63].

3.2 Metaphoria: A Writing Assistant for Metaphor Creation

In this section I report on my work designing and evaluating a system to support writers with writing metaphors about abstract concepts [20].

Based on our literature review, **coherence to context** is the biggest barrier to use for creative writing support tools [11, 38, 52]. Secondarily, writers do not want tools that make their writing sound the same as others [56]. Thus, suggestions that result in **divergent outcomes** for writers is crucial. These goals map to previous methodology in HCI for the evaluation of generative drawing tools; Jacobs et. al. [27] evaluate their drawing tool on compatibility (coherence to context) and expressiveness (ability to express a divergent set of ideas) .

A system that is **coherent to context** provides suggestions that are relevant to the task at hand. If writers come to the system with an idea or intention, the system should generate metaphorical phrases coherent with this context, and should be flexible enough to be coherent for a wide range of writer ideas and intentions. A system that encourages **divergent outcomes** provides many compelling options and increases the variation in writers' work rather than propel all writers toward similar metaphors.

To address **coherence to context**, we focus on generating metaphorical connections for a given "seed metaphor". Seed metaphors are of the form *[source] is [vehicle]*, e.g. *envy is a bell*, where *envy* is the source and *bell* is the vehicle. By focusing on connections between the words, such as 'envy can sound the alarm like a bell', rather than the selection of the seed words, we leave open the possibility that the writer inputs one or both words of the seed metaphor.

To address **divergent outcomes**, we generate and present multiple, distinct suggestions for each seed metaphor. This approach allows writers to select a suggestion salient for them in particular.

3.2.1 System Design

Starting with a seed metaphor, our approach is to first generate many features of the vehicle (*bell*), and then rank these features by how related they are to the source (*envy*). This aligns with traditional metaphor usage, in which features of the vehicle are used to explain the source.

To find features of the vehicle we use ConceptNet [35], an open-source knowledge graph, as a source of structural and functional properties of words. Structural properties are elements that define or compose an object. For example, a *bell* has a *clapper* and a *mouth*. In ConceptNet, we select for structural features by querying the "HasA" relations of the vehicle. Functional properties focus on an object's actions and purpose. For example, a *bell* can *make noise* and be used for *alerting*. In ConceptNet, we select for functional features by querying the "UsedFor" and "CapableOf" relations. Together, structural and functional properties provide a large set of potential connections from the vehicle to the source.

Not all features of the vehicle (*bell*) will metaphorically map to the source (*envy*). To find the most relevant ones, we rank how related the vehicle features (e.g. *used for getting attention*) are to the source (*envy*). To rank suggestions we use GloVe word embeddings [45] trained on Wikipedia 2014 + Gigaword 5. Word embeddings are a common way to measure the semantic similarity between words [39]. Here, we use them to measure the semantic similarity between the vehicle property and source word. To find the semantic distance between vehicle features and the source word, we use a Word Mover's Distance (WMD) [29].

Figure 6: Screenshot of the Metaphoria tool being used by a professional poet.

In order to promote diverse outcomes, our systems presents writers with 10 coherent suggestions that are semantically distinct. For instance *get attention* and *getting people's attention* may both be coherent, yet they give effectively the same idea to the writer. For this reason, as we build our list of suggestions to show the writer, we throw out any feature that is too close to any of the features already ranked. This closeness is again calculated with the Word Mover's Distance, this time between two features.

The word embedding space is not sensitive to antonyms and thus some highly ranked features have a mismatched sentiment with the source concept. Pilot testing showed that people found mismatched sentiments to be jarring and caused them to lose faith in the system. However, people who are first shown more intuitive features were more likely to appreciate the antonym features. Thus, we first select the suggestions as shown above, and then re-rank them by how similar the valence of each one is to the source concept.

Valence is the positive or negative connotation of a word and we assign valence scores to all words based on Warriner, et. al's database [62]. We denote the valence of the source as V_{source} and the valence of word i in the feature V_i for words $1, ..., n$. Then we define the valence distance as

$$V_{dist} = |V_{source} - \text{avg}(V_1, ..., V_n)| \tag{1}$$

We can then reorder the suggestions from the smallest valence distance to the largest.

Finally, we rephrase all connections into a suggestion for the writer; given the source *envy*, the vehicle *bell* and the connecting feature *making noise*, the suggestion is presented as 'envy is used for making noise like a bell'.

Figure 6 shows a screenshot of the tool being used by a professional poet.

3.2.2 Quantitative Study

This study evaluates the quality of the suggestions Metaphoria generates. To achieve **coherence to context**, suggestions should make sense given their seed metaphor and enact principles of high quality writing.

To evaluate the suggestions, we compare them to two other state-of-the-art metaphor generation algorithms: Thesaurus Rex [59] and Intersecting Word Vectors [17]. As our system produces a ranked set of suggestions, we also compare both the highest ranked suggestions with the lowest to evaluate the effectiveness of the ranking algorithm. For each system we select the top three ranked suggestions. Ranking for Metaphoria is done using the WMD distance to the source concept (as explained in the Design section); both Thesaurus Rex and Intersecting generate ranked lists.

To compare the systems, we define three metrics for evaluating metaphor strength. The first is **aptness**, in which a metaphor accurately describes a connection between the concepts; this is the ground level of metaphors. The second is **specificity**, in which a metaphor describes a connection unlikely to be transferable other concepts. The third is **imageability**, in which a metaphor describes a connection the reader can visualize.

We expect that Intersecting will not be particularly apt as it relies solely on the embedding space to provide meaning and embedding spaces notoriously lack consistent discrete semantics [34]. Thesaurus Rex uses textual evidence, so we expect its connections to be apt, but because of this we also expect it to be less imageable and specific as it may only find higher level, and thus vaguer, attributes.

We have three hypotheses:

- H1: Metaphoria suggestions are **more apt** than Intersecting and **at least as apt** as Thesaurus Rex.
- H2: Metaphoria suggestions are **more specific** than Thesaurus Rex and Intersecting.
- H3: Metaphoria suggestions are **more imageable** than Thesaurus Rex and Intersecting.

Additionally, we want to know if top-ranked Metaphoria suggestions are more apt than bottom-ranked ones. For this, we compare the top three and bottom three ranked suggestions. Our hypothesis is:

- H4: Top-ranked Metaphoria suggestions are **more apt** than bottom ranked ones.

We have two professional writers with an MFA in Creative Writing act as annotators. We consider 12 different seed metaphors, e.g. *hope is a stream*, and for each generate the top 3 metaphor suggestions from each system. Additional we generate the bottom 3 metaphor suggestions for Metaphoria. This results in 144 suggestions total.

The annotators consider each metaphor suggestion and mark whether it is apt, specific, and imageable. They are told that all suggestions are generated by computers, but they are not told anything about how or the fact that they come from different systems. They are shown the suggestions for each seed metaphor in random order.

We report the percent agreement between the two annotators for apt, specific, and imageable (and the Cohen's Kappa correlation coefficients) to be 85% (0.63), 83% (0.67) and 88% (0.64),

	Apt	Specific	Imageable
Metaphoria (M)	97%	**82%**	**100%**
Thesaurus Rex (TR)	**100%**	47%	**100%**
Intersecting (I)	49%	43%	53%

Table 1: While both Metaphoria and Thesaurus Rex generate apt and imageable metaphors, only Metaphoria consistently produces specific metaphors.

	Apt	Specific	Imageable
Top-ranked	**97%**	82%	**100%**
Bottom-ranked	78%	**85%**	89%

Table 2: Top-ranked metaphors perform significantly better than bottom-ranked metaphors on aptness and imageability; there is no significant difference for specificity.

respectively. Given the natural ambiguity of metaphors and creative writing, this is a high level of agreement.

The following results are determined by combining the evaluations of the two annotators; the higher evaluation is used in cases of disagreement. Table 1 shows the percent of times a given systems' suggestions was marked as apt, specific, or imageable. While Metaphoria and Thesaurus Rex metaphors are both consistently apt and imageable, Metaphoria outperforms all systems on specificity.

To test H1-3, we perform paired t-tests (Bonferonni corrected) on the relevant pairs and disprove the null hypothesis for H1 and H2. However, it is clear that H3 does not hold as both Metaphoria and Thesaurus Rex were 100% imageable. The results of the statistical tests can be found in Table 3.

Surprisingly, Thesaurus Rex metaphors were as imageable as Metaphoria ones. In general the annotators found adjectives like *hard* more imageable than we expected. However, Metaphoria still outperforms other systems on specificity.

We also consider the difference between the top and bottom ranked Metaphoria suggestions.

Hypothesis	diff	t-value	p-value
H1a M *more apt* than I	0.48	5.83	2.8e-08
H1b TR *more apt* than I	0.51	6.16	4.8e-09
H2a M *more specific* than TR	0.34	3.36	2.7e-03
H2b M *more specific* than I	0.38	3.55	6.7e-04
H3a M *more imageable* than TR	0.00	n/a	n/a
H3b M *more imageable* than I	0.47	5.59	1.4e-09

Table 3: T-tests confirm that Metaphoria is as good or better across all metrics than state-of-the-art metaphor generation algorithms. P-values are Bonferonni corrected.

PO1's response	PO2's response	PO3's response
My island fills glasses like wine, its vines wrap around my new mouth like grapes. This new America is used to building things, anew, strange comfort like the rest of an air-bed at dusk. How new is new?	Garden Work with my mother, her tulips flaming blue and yellow, laboring to bloom beneath her palms, the soft lawn grating against early spring. We are wasting time, lingering under the porch light before dark, flirting with enemy weeds before my father returns home, drunk and swaying like a storm. She is used for currency and jewelry and lighting the pathway. She is for making flowers rise up to collide with her hands.	Metaphor for restoring quiet Use a gun to paint a room Addiction can clog a sink drain like hair History can win a war The garden of wasted time Fear to extinguish a fire like sand ice is for finding the source of light swimming is like snow. it is for children You can use caution to build fear in a movie You can use witchcraft to listen to music like an ear Corruption can outrun you like a horse

Table 4: Part of responses from three professional poets working with Metaphoria. Words highlighted in pink were input into Metaphoria by the poets, while words and phrases highlighted in green were suggestions that poets used.

Table 2 shows the percent of times a given systems' suggestions was marked as apt, specific, or imageable. Top ranked suggestions are more apt than bottom ranked ones (t = 2.49, p-value = 0.01) which confirms H4. There is no significant difference for specificity (t = -0.30, p-value = 0.76). However, top ranked suggestions are slightly more imageable than bottom ranked suggestions (t = 2.09, p-value = 0.04). It could be that aptness makes it easier visualize the suggestion.

This shows that Metaphoria creates high quality metaphors and can provide strong suggestions to writers.

3.2.3 Qualitative Study

This study evaluates if Metaphoria can adapt to a writer's own goals, and tests the system on inputs we did not expect. Our previous studies show Metaphoria is **coherent to context** and produces **divergent outcomes**; now we tackle whether these properties hold in real tasks which span a wide range of writer intentions.

We gave three professional poets a 15 minute tutorial of Metaphoria and then asked them to write a poem on a subject of their own choosing using Metaphoria in some way. The poets wrote for around 30 minutes each. We then conducted a semi-structured interview, and utilized having Metaphoria available to discuss their process and response. The poets were recruited through a mailing list for current and past MFA in Creative Writing students at a local university. All had a regular writing practice, were published poets, and one also held a teaching position in which they taught poetry writing workshops to undergraduates.

Coherence to context. All poets used several of the suggestions in their poem. Part of each poem is reproduced in Table 4, where words they input into Metaphoria are highlighted in pink and phrases from the suggestions they used are highlighted in green.

The context each poet brought to Metaphoria was very different. PO1 initially entered the word *island*; the first line of their poem was inspired by the suggestion 'island can fill a glass like wine', though they first spent several minutes with other suggestions like 'island can travel over water like a ship' and 'island can age over time like wine'. PO2 was initially inspired by suggestions for the seed metaphor *work is a garden*, where *work* was input during the tutorial; several words in the

first stanza came from the suggestions for this seed. Later they input the words *swaying* and *she*.

PO3 brought a very different type of context. They input many more words than the other two poets, more interested in finding interesting suggestions than crafting a poem with a particular direction; almost every line derives from some part of Metaphoria. They first input *sales*, then *marketing*, before exploring the word *metaphor*. Their first line is inspired by the suggestion 'metaphor is for restoring quiet like a bell'. Later they input words like *time*, *guns*, *history*, *elections*, *laughter*, and *stone*, to mention only a small number.

All poets found suggestions that resonated with them, though they were discriminate and often searched through several seeds before finding something they used. However, there were clearly different styles of use: PO1 and PO2 composed poems with some kind of linear narrative or thought, and used Metaphoria on words they had already written, often finding a suggestion that would finish the line they were working on. In contrast, PO3 input words they thought might be make for interesting metaphors, or words they simply overheard (we met in a coffee shop), many of which never made it into the poem. PO3's use was more like collecting interesting phrases, which they then arranged and edited.

Divergent outcomes. The resulting poems were of dramatically different styles, both due to each poet's differing usage of Metaphoria and their different writing styles. When explicitly asked about the expressiveness of the system, all poets noted that established writers have their own style and the system was unlikely to dramatically change it. Both PO2 and PO3 thought Metaphoria would increase the creativity of amateur poets, who tend to get stuck in cliche language; they thought the unexpectedness of the word combinations was likely to help.

However, PO2 did bring up concerns of ownership. While they did not think that Metaphoria limited them, they were concerned about using suggestions from Metaphoria that were too different from their intention, even if these suggestions were very good. PO3 used Metaphoria most liberally, yet had no such concerns. They drew a comparison between Metaphoria and Instagram, noting that while Instagram has produced a genre of photography that is very recognizable and thus the photos are somewhat similar, it has also produced unexpected and creative artworks. They speculated that Metaphoria might create a genre of Metaphoria-style poems, but would also allow poets to move in new and exciting directions.

3.3 Sparks: A Writing Assistant for Explaining Technical Topics

In this project, we study how language models can be applied to a real-world, high-impact writing task. In particular, we use a science writing form called tweetorials which explain technical concepts on Twitter for a general audience [4]. Tweetorials are short explanations of around 500 words which have a low-barrier to entry and are gaining popularity as a science writing medium [57]. Working on science writing requires a system to demonstrate proficiency within an area of technical expertise. This is much more difficult than our study of metaphors, which tended to deal with common concepts, objects and relations.

Given that language models have no model of truth, we design our system to come up with "sparks", intended to spark ideas in the writer, rather than having the system provide the ideas themselves. This aligns with prior work on metaphor creation, where users make use of system outputs as initial directions that are then interpreted and diverged from in the users' actual creation [20]. Additionally, this also encourages the writer to feel more ownership over their final product, which has shown to be a concern in past work [44].

3.3.1 System Design

To generate sparks we use GPT-2, an open source, mid-sized (1.5 billion parameters), transformer language model trained on 40GB of text from the web [46]. We use the huggingface implementation [64].

In addition to selecting a model, we had to design a decoding method – how to select the next token given the probability distribution the model outputs. We designed a method that attempts to further increase the coherence of beam search while also increasing its diversity. [21] explains our method in detail. As an overview, we increase the likelihood of infrequent words (to improve specificity) while limiting the selection to the top 50 tokens (to retain coherence). To increase the diversity of outputs, we force the first token of each output to be unique, but attempt to retain coherence by generating the rest of the tokens with beam search. While several more sophisticated methods have been proposed to increase diversity while retaining the coherence of beam search (e.g. [60]), in testing we found none were as effective as simply enforcing the first token to be unique.

Designing prompts for language models has become an active area of research, with many automatic methods being proposed [18, 32]. However, any automatic method requires at least some training data, and it's yet to be seen that automatically developed prompts can outperform hand-crafted prompts [18]. For these reasons, we hand-craft our prompts.

First we craft a 'prefix' prompt to prepend to any prompt used by a writer. Prefix prompts have been shown to greatly improve performance by providing the language model with appropriate context [50]. We found early on in development that simply providing the model with a technical topic was not enough – also providing a context area was necessary for it to appropriately interpret technical terms. For instance, if you use a prompt like "Natural language generation is used for", the model is likely to talk about linguistic research on languages, rather than computational methods. If instead you use the prompt, "Natural language generation, a topic in computer science, is used by" the results are much more likely to refer to computational language generation. Given this, we prepend all prompts with the following: "{topic} is an important topic in {context area}" where {topic} and {context area} are provided by the writer.

In hand-crafting our prompts, we wanted to make sure our prompts captured a range of relevant angles, so our system could flexibly work with any technical discipline. To do so, we synthesized work from expository and narrative theory into prompts capturing five categories: expository, instantiation, goal, causal, and role. Each category represented an angle that a writer might want to explore. Figure 9 shows five of the ten prompts used. Figure 7 shows a screenshot of the system with its important features marked.

3.3.2 Quantitative Study

We wanted to evaluate the quality of ideas for a variety of topics. We selected three disciplines – computer science, environmental science, and biology – that have a glossary of terms page on Wikipedia, and that have been demonstrated to be a rich discipline for science writing on social media.[6] For each discipline we randomly sampled 10 topics from their glossary of terms page.

[6] e.g. https://twitter.com/dannydiekroeger/status/1281100866871648256, https://twitter.com/GeneticJen/status/897153589193441281, and https://twitter.com/meehancrist/status/1197527975379505152

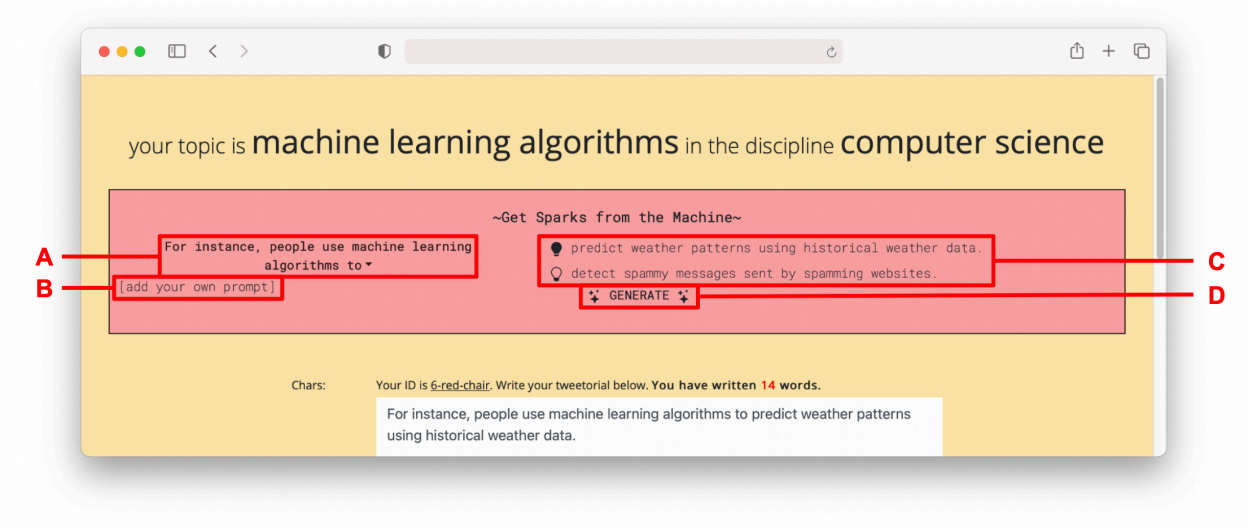


Figure 7: Example screenshot of our system generates sparks. A: writers can select from 10 template of prompts in a drop-down menu. B: writers can add their own prompt to the drop-down menu. C: sparks are generated with a lightbulb icon to the left, if writers click the lightbulb it will highlight and the spark is copied into the text area. D: writers can hit the generate button in order to generate a new spark.

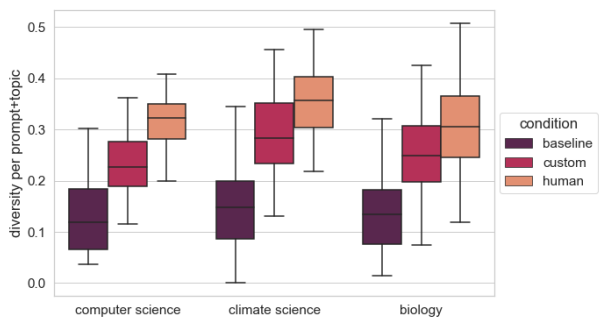
We wanted to collect human responses to our prompts to represent a gold standard or upper limit on the quality of ideas these prompts can generate. To do this, we recruited 2-3 PhD or senior undergraduate students in each discipline and had them complete the same prompts the language model did. Each student was paid $20/hour for as long as it took them to finish the task.

We compare the custom decoding to a competitive alternative: group beam search with hamming diversity penalty. This is a strong baseline that encourages diversity in the way [60] recommends, and can be implemented using arguments in the generate function in the huggingface transformer library. Both the custom decoding and baseline model use the same underlying language model.

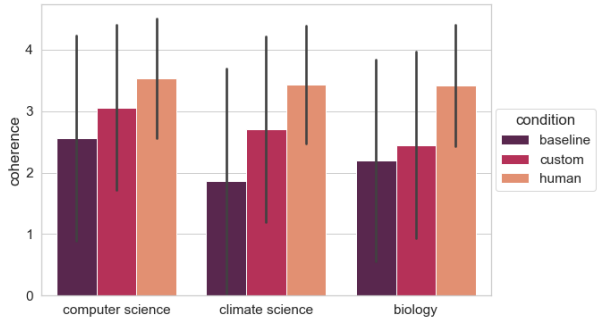
Coherence is notoriously difficult to measure automatically, especially without training data – measures like perplexity merely measure an output's likelihood under the model itself. For this reason we recruited 10 domain experts to annotate outputs for coherence on a 0 - 4 scale, in line with knowledge graph evaluations [31]. For biology we had 3 senior undergraduate students majoring in biology; for environmental science we had 2 senior undergraduate students majoring in environmental science; for computer science we had 2 PhD students from the computer science department.[7] Each discipline had 900 sentences to annotate (300 human generated, 300 from the baseline model, and 300 from the custom decoding). 250 randomly selected outputs from each discipline were annotated by two different domain experts, and the Cohen's weighted kappa was calculated as: $\kappa = .54$ for biology, $\kappa = .51$ for environmental science, and $\kappa = .34$ for computer science. Given that the agreement was moderate, we had a single annotation for the remaining sentences.

We measure diversity with sentence embeddings [48], in particular we report the average dis-

[7]The students could not have also participated in the generation portion.



(a) Distribution of diversity, split by discipline. Diversity is measured as the average sentence embedding distance per prompt+topic combination.



(b) Mean coherence per prompt+topic combination, split by discipline. Each prompt completion was scored by a domain expert on a scale of 0 to 4.

Figure 8: Diversity and coherence measures across three test disciplines for three conditions: a baseline language model, a language model with the custom decoding, and a human-created gold standard. The custom decoding improves upon the baseline and approaches the human gold standard.

tance between outputs within a given prompt. A higher average distance means that outputs are more dissimilar, and therefore more diverse.

Overall, the baseline had low diversity and coherence across all disciplines, while the human-created outputs perform much better. Figure 8a and Figure 8b show that the custom decoding method outperforms the baseline, but does not reach the performance of the human-created outputs. For diversity, two-tailed t-tests show this to be a significant difference for all disciplines (computer science: $p < .001$, climate science $p < .001$, biology: $p < .001$); for coherence, mann-whitney U tests show this to be a significant difference for all disciplines (computer science: $p < .001$, climate science $p < .001$, biology: $p < .001$).

Figure 9 shows the average coherence per topic for the custom decoding method and the human-created outputs. It plots the average coherence for each topic with the black dots, and the coherence for each prompt+topic combination in the colored dots. From this we can see the variation in quality over the topics for the custom decoding method. For instance, the ”computer security” outputs score an average of 3.7 in coherence, while ”automata theory” outputs score 2.1. When looking at the human-created outputs, the quality is far more consistent, with no topics dropping below an average of 3 in coherence.

This demonstrates that our system works well for some topics and less well for others. While we expected that our system would not perform as well as a human would, we did expect that the system would perform more consistently across topics. It is unclear why the language model performs significantly better on some topics, and given the way that these language models are trained it is difficult to inspect or even predict how well the model will perform on a given topic.

Figure 9 also shows that some prompt templates work better for some topics than others. In our system, the quality of outputs vary significantly with the prompt template. In the human generated outputs, the variation is smaller, but still we see some range.

Figure 9: This graph shows the coherence per topic for the custom decoding and the human-created gold standard, where 0 is nonsensical or untrue and 4 is generally true. The black dot shows the average coherence of all responses for a given topic, while the colored dots show the average coherence for a given topic per prompt. Topics are ordered by average coherence in the custom decoding. This graph shows that some topics perform much better than others with custom decoding, while the human outputs are generally high quality regardless of topic. It also shows that within a topic there can be a large variation between prompt templates.

3.3.3 Qualitative Study

We recruited 13 participants, all graduate students from five different STEM disciplines, to write tweetorials on a topic of their own choice, related to their area of study. By letting the participants pick their own topic, we ensured that they were writing within their area of expertise, and we were able to test our system on unseen topics.

Participants were given 15 - 20 minutes to interact with the system and write approximately the first 100 words of their tweetorial. Mouse clicks and key presses while the participant interacted with the system were collected, as well as all sparks generated. After this, the participant filled out a short survey and partook in a semi-structured interview with the facilitator. The survey questions and the questions that structured the interview can be found in the appendix. The study took about an hour and participants were compensated $40 USD for their time.

Participant interviews were transcribed and the authors performed a thematic analysis [3] on

the interview transcripts. The analysis centered on three areas: how sparks were helpful, how sparks were unhelpful, and ownership concerns in response to writing with a machine. Relevant quotes were selected from the transcripts and collated in a shared document, where the researchers discussed and collected the quotes into emergent themes.

The 13 participants came from five STEM disciplines, with the most common disciplines being Climate Science and Public Health. All but one were doing PhD (the remaining doing a research Master's) and varied from their 2nd year to their 7th year in their program. Participants were asked how often they wrote about technical topics for a public audience, and how often they did so on Twitter. Most participants rarely or never did so, though a few did so on a monthly or even weekly basis.

Participants were asked to select a topic they understood well that was related to their research. The facilitator attempted to aid participants in selecting a topic that wasn't too broad, but also not too specific, but as the facilitator did not necessarily have the same expertise as the participant this was at times difficult. Participants selected a wide range of topics, with no overlap.

We report on the prominent themes that emerged through our analysis in Table 5. Given the diversity in the participants' topics, how well the system generated sparks on their topics, and how they articulated or responded to questions in the interview, there was a high variability in how participants felt about the system. For this reason, a prevalence of 50% or above is considered very high. This would mean that over 50% of participants independently responded in the same way to an open ended question, despite writing about a unique topic and seeing a unique response from the system.

Sparks helped participants craft concise and detailed sentences quickly. Although we intended the system to inspire participants with new ideas, the most prevalent reason the participants cited for the spark being helpful was for crafting sentences. Many participants remarked that although the sparks were showing them information that they already knew well, it was much faster and easier to draw on language from the sparks than to write a sentence from scratch.

For instance, P12 said:

> Most of the time [the system] was articulating the ideas that were already in my head in a way that's short and concise, which is useful. Like 'deprivation index measures the relative deprivation experienced by an individual relative to others,' that would have probably taken me like three sentences to write, then I'd have to spend time editing it down. And then yeah... this is a lot quicker.

Several participants noted that they often go to Google or Wikipedia simply to get a well-written definition of a topic they understand well. This is something that the system was able to do for them without requiring a click away from the writing interface and incurring a change in context. P7 noted that it did a good job compressing what he would have looked for or found a Google search. P8 noted that all the sparks were similar to what she would have found on Wikipedia or via a Google search, but that they were "bite-sized" or "sound bite ready".

Sparks reminded participants of other ideas or angles about their topic. Several participants noted that the sparks provided good ideas or angles for discussing or introducing their topic. P2 noted that 'weather prediction models' – something a spark suggested – was a useful entry point to their research. They said, "that's something within my field that the general public might be more familiar with than what I actually do."

Code	Prevalence	Example Quote
reasons sparks were helpful		
Crafted concise sentences.	54%	Most of the time it [the system] was articulating the ideas that were already in my head in a way that's short and concise.
Came up with ideas / angles.	46%	It [the system] reminded me, Oh, it's not just my application, there's these other people using the same technology, but working on other problems.
Showed reader perspectives.	31%	It [the system] reminded me that there might be a more common understanding of this thing that I'm writing about, that's different from the highly specific one I've been living in.
reasons sparks were unhelpful		
Incorrect topic interpretation.	38%	It [the spark] just wasn't helpful, but only because it was using the different sense of 'embedded'.
Inaccuracies.	23%	Some of the sparks said, like, logistic regressions are used to estimate relative risks, which is completely not true.
Not desired angle.	23%	Someone probably does really care about measuring sexist attitudes ... but it just isn't my focus.
Vagueness.	23%	I would say about 20% of them were just not specific enough to warrant talking about.

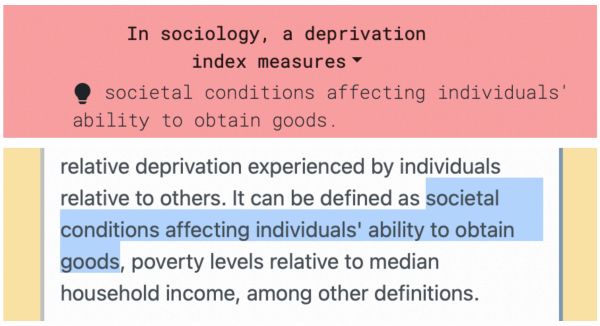
Table 5: Results of thematic analysis

Figure 10b shows how P4 drew on a spark about the 'sea level rise' in order to make their topic of 'glacier retreat over the holocene' interesting to the reader. P4 said in their interview, "It's often hard to figure out how to spin things in ways that feel relevant to people who don't study this," and that the sparks helped her find ways to make her research relevant. P7 said, "[the system] definitely generated multiple [ideas] that I could have written different tweetorials about."

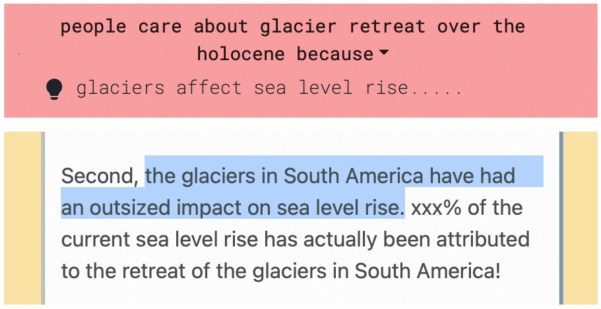
When asked if the sparks were giving them new ideas, many participants said that the system was helping them get to ideas they likely would have come up with themselves, but faster. For instance P4 said, "It was definitely faster. I think I would have gotten there, but it would have taken me longer."

Sparks encouraged participants to think about common reader perspectives. Several participants noted that the sparks reminded them of how people reading their tweetorial might be interpreting their topic. For instance P10, who was writing about measuring sexism through an economic lens, noted that many of the sparks talked about sexist attitudes. She said the while that certainly is an aspect of measuring sexism, it isn't the aspect that she actually studies. And so the sparks reminded her that people's main assumptions when thinking about sexism is probably about attitudes and therefore that might be an assumption that she will have to address in her tweetorial.

P5, writing about computationally hard problems, noted that one of the sparks talked about NP-completeness. He said that while at first he thought this might be too technical, it then made him wonder if someone who was reading a tweetorial about computational hard problems might



(a) An example of how a participant in our study used a spark for crafting a detailed sentence. Highlighted text was inspired by the spark.



(b) An example of how a participant in our study used a spark to make their tweetorial more engaging to a general audience. Highlighted text was inspired by the spark.

already know at least some of the keywords about this topic. In this way the sparks made him reflect on what knowledge his readers might already have.

Sparks failed in different ways for different participants. The ways in which participants found sparks to be unhelpful varied highly. The most common reason participants said sparks were unhelpful was that they incorrectly interpreted their topic. In this case, the sparks were not necessarily incorrect, but rather they reflected some alternate interpretation of their topic. For instance, P12, who was writing about deprivation indices, said that some of the sparks were about obesity. Obesity has little to do with deprivation indices, but they thought the algorithm may have been associating 'deprivation' with 'nutrition' (rather than, e.g., economic deprivation). Similarly P8, who was writing about regulatory fit, commented on several sparks about government regulation, which is unrelated to her psychology topic, but she assumed the algorithm was simply free associating with the word 'regulatory'.

Other reasons participants found the sparks to be unhelpful were factual inaccuracies, dealing with aspects of their topic that they were not trying to explain or that they did not study, and vague outputs. Participants also mentioned that some sparks were nonsensical, tautological, had too much jargon, or were simply "bizarre".

Overall participants varied highly in how useful they found the sparks. Some participants found that the sparks were so low quality that they found the system completely unhelpful. Others said that even though some of the sparks were not helpful the ones that were helpful were so helpful they were unconcerned with a few that didn't make sense or were off-topic.

4 Research Plan

In this section I outline the work needed to complete my thesis. I intend to perform a formal literature review structured by my proposed design space. This work should be quite fast to complete. I also intend to perform a longitudinal study on my science writing assistant. This work will take longer, and I intend to submit it to TOCHI, the leading journal in HCI research.

4.1 Formal Literature Review Structured by Design Space

I plan to use my design space to perform a formal literature review of writing assistants. I intend to use the ACM Digital Library advanced search feature to collect writing assistant papers from the past five years, which will allow the collection to be easily replicated by others. I will then annotate these papers in a variety of ways.

I will the following measures from a literature review of creativity support tools [16]:

- complexity (low, medium, high)
- user group (novice, casual, expert, unspecified)

Additionally I will annotate for the following measures:

- level of constraint (1 - minimal constraint, 5 - highly constrained)
- size of task being supported (word, sentence, paragraph, multi-paragraph)
- length of evaluation (none, $<$30 min, 30-60 min, $>$60 min)
- technology used (examples, templates, retrieval, language model)

These measures will allow us to understand the design space of writing assistants, where they are moving and what is lacking in the field. We will write this up as a short paper and submit it to Designing Interactive Systems (DIS) in February.

4.2 A Longitudinal Study of Writing Assistants

Writing assistants tend to be evaluated with hour long, lab-based studies. Participants may be given multiple writing tasks, but each writing task is externally defined and quite short, typically 10 or 20 minutes long. Of all the writing assistants referenced in this thesis proposal, none perform studies that include more than half an hour of writing (including my own). However, many real-life writing tasks are not completed within a single half-hour period. A fiction writer may return to their short story many times before it is finished – even amateur writers understand the power of revision, and all writers experience 'writer's block' and may come back to their writing when they feel more inspired. Science writing is rarely completed within a single period and scientists, too, understand the importance of returning to a paper draft again and again. They are not expected even to write a complete abstract draft in a single sitting.

In addition to the realities of working on a single piece of writing, short lab studies lack the ability to discern how users learn to use the system over time. A short tutorial is rarely enough for a user to learn the intricacies of a sufficiently complex system [27], especially a system that makes use of large, neural models that may exhibit idiosyncrasies. In my work on mental models of AI agents, we found that participants came into the studies with strong priors about the AI agent, and many were not able to revise those priors in a short lab study. Additionally, sufficiently complex systems cannot be learned with a short tutorial and require repeated interactions for user to make use of it in their own lives [27].

For these reasons, I propose running a two-month longitudinal study of a writing assistant in order to understand how the results from short lab studies hold up to repeated interactions and user-driven usage. I propose improving the 'Sparks' system using the results from our short lab study, and recruiting climate scientists to use it repeatedly over the course of the study. I ask the following research questions:

- **What is the 'learning curve' for interacting with the language model?** Is there an amount or type of usage after which participants' perception of its utility greatly increase? If so, what do the participants report having learned? And how have their interaction patterns changed?
- **How does how the system supports writers change over time?** Do participants' find that the system supports the same kinds of writing tools throughout continued usage? Or do they learn which kinds of goals it best supports? Do they 'grow out' of some kinds of assistance?
- **Do participants' mental models of the language model change over time?** If so, in what ways? Do their mental models become more accurate? Is an increase in mental model accuracy correlated with a participants' perceived utility of the system?

4.2.1 Proposed Methodology

System design. We already have a prototype of a writing assistant for explaining technical concepts, and have run a short, lab-based user study. Based on those results, we intend to allow more customization of the templated prompts and update the prompts to better serve common usages. This includes allowing users to edit the 'prepended' text and store their own templates for easy usage with different topics. We will run pilot studies to iterate on these features such that they are transparent and understandable for writers.

Our initial system was designed for one-off usage – no user data or preferences were stored for later access by the users. We will need to implement user accounts and storage of user data and preferences, such that customizations and drafts can be stored between writing sessions. We will also implement functionality to store data useful for the longitudinal study. This includes built-in weekly surveys, recording the frequency and length of writing sessions, and storing pertinent information about how users are interacting with the system, for example mouse clicks and key presses.

Participants. We will recruit 5-10 climate science professionals who are interested in improving their science writing or increasing the amount of science writing they do. We choose climate scientists as our participants because climate scientists often have a high motivation to participate in science communication, and their research is highly technical. We will recruit at least one graduate student, one professor, and one industry professional in order to diversify our participant population. We will appropriately compensate the participants for their time. We will get approval from the IRB of our institution.

Participant instructions. Participants will be asked to spend at least 30 minutes with the tool each week. They could be writing a tweetorial, essay, blog post, op-ed, or any other kind of science writing intended for a general audience, and they could spend this time on a new peice or editing one they have already started. They will be asked to 'complete' at least two pieces over the course of the two month period, and can produce as many drafts as they like. They will be instructed that the main goal of the study is to understand how writers make use of writing assistants in a

real-world context, and thus they should use the system as much as possible and for tasks they are most invested in.

Following previous methodology on longitudinal support tool studies [27], we will perform regular interviews with the participants of at least 15 minutes each week, asking participants about their experience with the tool, and making small changes to the tool when necessary. Additionally participants will do weekly surveys that measure their perception of the system. Compensation will be tied to continual weekly participation.

4.2.2 Proposed Analysis

In order to answer the proposed research questions, we need to measure the following aspects of the participants and their usage of the system over time:

- **Their mental model of the system.** We will repeatedly administer a quick test of their self-reported mental model. Participants will be asked to type in a prompt for the system, think about what they expect the system to produce, and then look at the outputs. They then report on a Likert-scale how much the actual outputs matched their expectations. We will also ask them a series of questions about the system's abilities before the study begins, mid-way through the study, and after the study.

- **Their perceived utility of the system.** We intend to measure perceived utility in two ways. First is a self-reported measure. We will administer a weekly survey that includes the creativity support index survey [9] and some task-specific questions. Second is a behavioral measure. We will also measure how often participants make use of the language model outputs in their writing, both through how often they 'star' suggestions and how much of their writing contains text that was generated by the language model.

- **Their interaction patterns with the system.** We will collect a broad range of data about their interaction with the system: what prompts they put into the system, what the current state of their writing is when interacting with the system, how and how often system outputs are incorporated into their writing.

This data will allow us to perform the following analyses:

- Test if participants' perceived utility (measured with survey questions as well as system output usage) increases over time.

- Test if participants' self-reported mental model increases in accuracy over time.

- Test for a positive correlation between mental model accuracy and perceived utility.

- Perform a hypothesis-driven thematic analysis on participants' answers to open-ended questions about system utility (what kinds of goals it supports) and how this changed over time.

We will write up the results of this study and submit it to the Transactions on Computer-Human Interaction (TOCHI) journal.

Timeline	Work	Progress
Dec. ‘21 - Feb. ‘22.	Complete formal literature review; submit to DIS	ongoing
	Implement user accounts and data logging	ongoing
	Iterate on interface; pilot study	ongoing
	Develop survey questions and mental model probe	
	Write and submit IRB; recruit participants	
Mar. - Apr. ‘22	Run study! Make updates to systems as necessary	
	Start analyzing data and writing paper (methodology, etc.)	
	Start writing dissertation	
May ‘22	Finish analysis and paper writing; submit to TOCHI	
Jun. - Aug ‘22	— summer internship —	
Sep. - Oct. ‘22	Finish writing dissertation	
Nov. ‘22	Thesis defense!	

Table 6: Plan for completion of my research

4.3 Timeline for Completion

My timeline for completion can be found in Table 6. I intend to finish preparation for the longitudinal study by March. At this point, the study can begin and the remaining work is analysis and writing. Since the study is two months long, I expect to be able to start work on the dissertation during the study, while I am waiting for results to come in. I also expect to write some of the paper (like the related works and methodology) during the study.

I hope to do a summer internship with either IBM or Microsoft Research on what makes language model outputs offensive to users. This work will not be part of my thesis, but will open up a new line of research I hope to continue after graduation. During the internship I hope to hear back from TOCHI – in the case of rejection, I can submit the paper to CHI in September.

After the internship I plan to take two months to finish writing my dissertation. Thus, I intend to defend my thesis in November, 2022.

References

[1] Emily M. Bender, Timnit Gebru, Angelina McMillan-Major, and Shmargaret Shmitchell. On the Dangers of Stochastic Parrots: Can Language Models Be Too Big? In *Proceedings of the 2021 ACM Conference on Fairness, Accountability, and Transparency*, pages 610–623, Virtual Event Canada, March 2021. ACM.

[2] Antoine Bosselut, Hannah Rashkin, Maarten Sap, Chaitanya Malaviya, Asli Celikyilmaz, and Yejin Choi. COMET: Commonsense Transformers for Automatic Knowledge Graph Construction. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4762–4779, Florence, Italy, 2019. Association for Computational Linguistics.

[3] Virginia Braun and Victoria Clarke. Thematic analysis. In Harris Cooper, Paul M. Camic, Debra L. Long, A. T. Panter, David Rindskopf, and Kenneth J. Sher, editors, *APA handbook of research methods in psychology, Vol 2: Research designs: Quantitative, qualitative, neuropsychological, and biological.*, pages 57–71. American Psychological Association, Washington, 2012.

[4] Anthony C. Breu. From Tweetstorm to Tweetorials: Threaded Tweets as a Tool for Medical Education and Knowledge Dissemination. *Seminars in Nephrology*, 40(3):273–278, May 2020.

[5] Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared 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 M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. Language Models are Few-Shot Learners. *arXiv:2005.14165 [cs]*, July 2020. arXiv: 2005.14165.

[6] Jill Burstein, Norbert Elliot, Beata Beigman Klebanov, Nitin Madnani, Diane Napolitano, Maxwell Schwartz, Patrick Houghton, and Hillary Molloy. Writing MentorTM: Writing Progress Using Self-Regulated Writing Support. *The Journal of Writing Analytics*, 2(1):285–313, 2018.

[7] Alex Calderwood, Vivian Qiu, Katy Ilonka Gero, and Lydia B Chilton. How Novelists Use Generative Language Models: An Exploratory User Study. In *Human-AI Co-Creation with Generative Models*, Tokyo Japan, March 2018. ACM.

[8] Mia Xu Chen, Benjamin N. Lee, Gagan Bansal, Yuan Cao, Shuyuan Zhang, Justin Lu, Jackie Tsay, Yinan Wang, Andrew M. Dai, Zhifeng Chen, Timothy Sohn, and Yonghui Wu. Gmail Smart Compose: Real-Time Assisted Writing. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 2287–2295, Anchorage AK USA, July 2019. ACM.

[9] Erin Cherry and Celine Latulipe. Quantifying the Creativity Support of Digital Tools through the Creativity Support Index. *ACM Transactions on Computer-Human Interaction*, 21(4):1–25, August 2014.

[10] Elizabeth Clark, Anne Spencer Ross, Chenhao Tan, Yangfeng Ji, and Noah A. Smith. Creative Writing with a Machine in the Loop: Case Studies on Slogans and Stories. In *23rd International Conference on Intelligent User Interfaces*, pages 329–340, Tokyo Japan, March 2018. ACM.

[11] Elizabeth Clark, Anne Spencer Ross, Chenhao Tan, Yangfeng Ji, and Noah A. Smith. Creative writing with a machine in the loop: Case studies on slogans and stories. In *23rd International Conference on Intelligent User Interfaces*, IUI '18, pages 329–340, New York, NY, USA, 2018. ACM.

[12] Nicholas Davis. Human-computer co-creativity: Blending human and computational creativity. In *Ninth Artificial Intelligence and Interactive Digital Entertainment Conference*, 2013.

[13] Nicholas Davis, Chih-Pin Hsiao, Kunwar Yashraj Singh, and Brian Magerko. Co-creative drawing agent with object recognition. In *Twelfth Artificial Intelligence and Interactive Digital Entertainment Conference*, 2016.

[14] Janet Emig. Writing as a Mode of Learning. page 8.

[15] Linda Flower and John R. Hayes. A Cognitive Process Theory of Writing. *College Composition and Communication*, 32(4):365, December 1981.

[16] Jonas Frich, Lindsay MacDonald Vermeulen, Christian Remy, Michael Mose Biskjaer, and Peter Dalsgaard. Mapping the Landscape of Creativity Support Tools in HCI. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*, pages 1–18, Glasgow Scotland Uk, May 2019. ACM.

[17] Andrea Gagliano, Emily Paul, Kyle Booten, and Marti A Hearst. Intersecting word vectors to take figurative language to new heights. In *Proceedings of the Fifth Workshop on Computational Linguistics for Literature*, pages 20–31, 2016.

[18] Tianyu Gao, Adam Fisch, and Danqi Chen. Making Pre-trained Language Models Better Few-shot Learners. *arXiv:2012.15723 [cs]*, June 2021. arXiv: 2012.15723.

[19] Katy Ilonka Gero, Zahra Ashktorab, Casey Dugan, Qian Pan, James Johnson, Werner Geyer, Maria Ruiz, Sarah Miller, David R. Millen, Murray Campbell, Sadhana Kumaravel, and Wei Zhang. Mental Models of AI Agents in a Cooperative Game Setting. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*, pages 1–12, Honolulu HI USA, April 2020. ACM.

[20] Katy Ilonka Gero and Lydia B. Chilton. Metaphoria: An Algorithmic Companion for Metaphor Creation. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*, pages 1–12, Glasgow Scotland Uk, May 2019. ACM.

[21] Katy Ilonka Gero, Chris Kedzie, Savvas Petridis, and Lydia Chilton. Lightweight Decoding Strategies for Increasing Specificity. *arXiv:2110.11850 [cs]*, October 2021. arXiv: 2110.11850.

[22] Marjan Ghazvininejad, Xing Shi, Yejin Choi, and Kevin Knight. Generating topical poetry. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1183–1191, 2016.

[23] John R. Hayes. A new framework for understanding cognition and affect in writing. In *The Science of Writing: Theories, Methods, Individual Differences, and Applications*. Lawrence Erbaum Associates, 1996.

[24] Eric Horvitz. Principles of mixed-initiative user interfaces. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '99, pages 159–166, New York, NY, USA, 1999. ACM.

[25] Julie S. Hui, Darren Gergle, and Elizabeth M. Gerber. Introassist: A tool to support writing introductory help requests. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*, CHI '18, pages 22:1–22:13, New York, NY, USA, 2018. ACM.

[26] Shamsi T. Iqbal, Jaime Teevan, Dan Liebling, and Anne Loomis Thompson. Multitasking with Play Write, a Mobile Microproductivity Writing Tool. In *Proceedings of the 31st Annual ACM Symposium on User Interface Software and Technology*, pages 411–422, Berlin Germany, October 2018. ACM.

[27] Jennifer Jacobs, Joel Brandt, Radomír Mech, and Mitchel Resnick. Extending manual drawing practices with artist-centric programming tools. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*, CHI '18, pages 590:1–590:13, New York, NY, USA, 2018. ACM.

[28] Anjuli Kannan, Karol Kurach, Sujith Ravi, Tobias Kaufmann, Andrew Tomkins, Balint Miklos, Greg Corrado, Laszlo Lukacs, Marina Ganea, Peter Young, and Vivek Ramavajjala. Smart Reply: Automated Response Suggestion for Email. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 955–964, San Francisco California USA, August 2016. ACM.

[29] Matt Kusner, Yu Sun, Nicholas Kolkin, and Kilian Weinberger. From word embeddings to document distances. In *International Conference on Machine Learning*, pages 957–966, 2015.

[30] Omer Levy and Yoav Goldberg. Dependency-based word embeddings. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, volume 2, pages 302–308, 2014.

[31] Xiang Li, Aynaz Taheri, Lifu Tu, and Kevin Gimpel. Commonsense Knowledge Base Completion. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1445–1455, Berlin, Germany, 2016. Association for Computational Linguistics.

[32] Xiang Lisa Li and Percy Liang. Prefix-Tuning: Optimizing Continuous Prompts for Generation. *arXiv:2101.00190 [cs]*, January 2021. arXiv: 2101.00190.

[33] Zach Lieberman, T. Watson, and A. Castro. openframeworks. http://openframeworks.cc/about, 2015. Accessed: 2018-09-19.

[34] Tal Linzen. Issues in evaluating semantic spaces using word analogies. *CoRR*, abs/1606.07736, 2016.

[35] H. Liu and P. Singh. Conceptnet — a practical commonsense reasoning tool-kit. *BT Technology Journal*, 22(4):211–226, Oct 2004.

[36] Ming Liu, Rafael A Calvo, and Vasile Rus. Automatic Generation and Ranking of Questions for Critical Review. page 15, 2020.

[37] Allan MacLean, Richard M Young, Victoria M E Bellotti, and Thomas P Moran. Questions, Options, and Criteria: Elements of Design Space Analysis. page 51.

[38] Enrique Manjavacas, Folgert Karsdorp, Ben Burtenshaw, and Mike Kestemont. Synthetic literature: Writing science fiction in a co-creative process. In *Proceedings of the Workshop on Computational Creativity in Natural Language Generation (CC-NLG 2017)*, pages 29–37, 2017.

[39] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. Distributed representations of words and phrases and their compositionality. In C. J. C. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 26*, pages 3111–3119. Curran Associates, Inc., 2013.

[40] Tim Miller. Explanation in artificial intelligence: Insights from the social sciences. *Artificial Intelligence*, 2018.

[41] Yael Netzer, David Gabay, Yoav Goldberg, and Michael Elhadad. Gaiku: generating Haiku with word associations norms. In *Proceedings of the Workshop on Computational Approaches to Linguistic Creativity - CALC '09*, pages 32–39, Boulder, Colorado, 2009. Association for Computational Linguistics.

[42] Donald A Norman. Some observations on mental models. In *Mental models*, pages 15–22. Psychology Press, 2014.

[43] Changhoon Oh, Jungwoo Song, Jinhan Choi, Seonghyeon Kim, Sungwoo Lee, and Bongwon Suh. I lead, you help but only with enough details: Understanding user experience of co-creation with artificial intelligence. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*, CHI '18, pages 649:1–649:13, New York, NY, USA, 2018. ACM.

[44] Zhenhui Peng, Qingyu Guo, Ka Wing Tsang, and Xiaojuan Ma. Exploring the Effects of Technological Writing Assistance for Support Providers in Online Mental Health Community. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*, pages 1–15, Honolulu HI USA, April 2020. ACM.

[45] Jeffrey Pennington, Richard Socher, and Christopher Manning. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543, 2014.

[46] Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9, 2019.

[47] Casey Reas and Ben Fry. Processing. http://processing.org, 2004. Accessed: 2018-09-19.

[48] Nils Reimers and Iryna Gurevych. Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks. *arXiv:1908.10084 [cs]*, August 2019. arXiv: 1908.10084.

[49] Mitchel Resnick, Brad Myers, Kumiyo Nakakoji, Ben Shneiderman, Randy Pausch, Ted Selker, and Mike Eisenberg. Design principles for tools to support creative thinking. In *NSF Workshop Report on Creativity Support Tools*, pages 25–36. Citeseer, 2005.

[50] Laria Reynolds and Kyle McDonell. Prompt Programming for Large Language Models: Beyond the Few-Shot Paradigm. *arXiv:2102.07350 [cs]*, February 2021. arXiv: 2102.07350.

[51] Melissa Roemmele and Andrew S. Gordon. Automated Assistance for Creative Writing with an RNN Language Model. In *Proceedings of the 23rd International Conference on Intelligent User Interfaces Companion*, pages 1–2, Tokyo Japan, March 2018. ACM.

[52] Melissa Roemmele and Andrew S. Gordon. Automated assistance for creative writing with an rnn language model. In *Proceedings of the 23rd International Conference on Intelligent User Interfaces Companion*, IUI '18 Companion, pages 21:1–21:2, New York, NY, USA, 2018. ACM.

[53] K. Romer and F. Mattern. The design space of wireless sensor networks. *IEEE Wireless Communications*, 11(6):54–61, December 2004.

[54] Marlene Scardamalia and Carl Bereiter. Knowledge telling and knowledge transforming in written composition. In *Advances in applied psycholinguistics*. Cambridge University Press, 1987.

[55] Taylor Shin, Yasaman Razeghi, Robert L. Logan IV, Eric Wallace, and Sameer Singh. AutoPrompt: Eliciting Knowledge from Language Models with Automatically Generated Prompts. *arXiv:2010.15980 [cs]*, November 2020. arXiv: 2010.15980.

[56] Robin Sloan. Writing with the machine. `https://www.robinsloan.com/notes/writing-with-the-machine/`, 2016. Accessed: 2018-09-19.

[57] Alice Soragni and Anirban Maitra. Of scientists and tweets. *Nature Reviews Cancer*, 19(9):479–480, September 2019.

[58] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, ukasz Kaiser, and Illia Polosukhin. Attention is All you Need. page 11.

[59] Tony Veale and Yanfen Hao. Comprehending and generating apt metaphors: a web-driven, case-based approach to figurative language. In *AAAI*, volume 2007, pages 1471–1476, 2007.

[60] Ashwin K. Vijayakumar, Michael Cogswell, Ramprasath R. Selvaraju, Qing Sun, Stefan Lee, David Crandall, and Dhruv Batra. Diverse Beam Search: Decoding Diverse Solutions from Neural Sequence Models. *arXiv:1610.02424 [cs]*, October 2018. arXiv: 1610.02424.

[61] Ge Wang. *The ChucK Audio Programming Language: An Strongly-timed and On-the-fly Environ/mentality*. PhD thesis, Princeton University, 2008.

[62] Amy Beth Warriner, Victor Kuperman, and Marc Brysbaert. Norms of valence, arousal, and dominance for 13,915 english lemmas. *Behavior Research Methods*, 45(4):1191–1207, Dec 2013.

[63] Meredith Whittaker. The steep cost of capture. *Interactions*, 28(6):50–55, 2021.

[64] Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander Rush. Transformers: State-of-the-Art Natural Language Processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online, 2020. Association for Computational Linguistics.

[65] Robert F. Woodbury and Andrew L. Burrow. Whither design space? *Artificial Intelligence for Engineering Design, Analysis and Manufacturing*, 20(2):63–82, May 2006.

[66] Tongshuang Wu, Michael Terry, and Carrie J Cai. Ai chains: Transparent and controllable human-ai interaction by chaining large language model prompts. *arXiv preprint arXiv:2110.01691*, 2021.