



Impacts of Predictive Text on Writing Content

Citation

Arnold, Kenneth Charles. 2020. Impacts of Predictive Text on Writing Content. Doctoral dissertation, Harvard University, Graduate School of Arts & Sciences.

Permanent link

<https://nrs.harvard.edu/URN-3:HUL.INSTREPOS:37365774>

Terms of Use

This article was downloaded from Harvard University's DASH repository, and is made available under the terms and conditions applicable to Other Posted Material, as set forth at <http://nrs.harvard.edu/urn-3:HUL.InstRepos:dash.current.terms-of-use#LAA>

Share Your Story

The Harvard community has made this article openly available.
Please share how this access benefits you. [Submit a story](#).

[Accessibility](#)

©2020 – KENNETH C. ARNOLD

CREATIVE COMMONS ATTRIBUTION LICENSE 4.0.

YOU ARE FREE TO SHARE AND ADAPT THESE MATERIALS FOR ANY PURPOSE IF YOU
GIVE APPROPRIATE CREDIT AND INDICATE CHANGES.

Impacts of Predictive Text on Writing Content

ABSTRACT

People are increasingly communicating using digital technologies, especially using written language. Many modern text entry systems include features that offer predictive suggestions of words and phrases. These features have been designed and evaluated for the goal of making text entry more efficient. However, when these predictions are shown to writers to use, they also serve as suggestions of what content to write. This dissertation addresses two main questions: what effects do current predictive text system designs have on writing content, and what are some challenges and opportunities to designing the effects that predictive systems can have on writing content?

The human-subjects studies conducted found evidence that the text that people write using predictive text entry systems reflects bias of these systems. Specifically, the presence of word suggestions decreased the degree to which writers chose the sorts of words that the system would not have expected in an image captioning task. Phrase suggestions resulted in even more predictable writing than word suggestions. Finally, phrase suggestions that were generated

with a bias towards positive sentiment resulted in people writing more positive content, which implies that biases in training data could lead to biases in the content people create.

Several pilot studies towards designing suggestions with different effects failed because writers viewed the experimental suggestions as irrelevant, so the dissertation presents work along two approaches towards enabling alternative suggestion designs. First, it discusses how to design suggestion systems that can adjust content attributes without overly compromising suggestion acceptability by applying an approach from reinforcement learning. Then it discusses how to design suggestions to guide writers about what topics to include, including a study comparing the perceived relevance of two designs.

Contents

1	INTRODUCTION	1
1.1	Inherent Bias	4
1.2	Emergent Bias	6
1.3	Manipulated Bias	6
1.4	Opportunities for Guiding Content Creation	7
2	RELATED WORK	8
2.1	Text Entry	8
2.2	Bias in Predictive Systems	14
2.3	Cognition in Content Creation	15
2.4	Tools for Content Creators	17
2.5	Autocompletion	19
3	WORD SUGGESTIONS DISCOURAGE THE UNEXPECTED	21
3.1	Background	24
3.2	Research Questions	26
3.3	Study	29
3.4	Results	41
3.5	Supplemental Analyses	47
3.6	Discussion	52
4	EFFECTS OF SUGGESTION LENGTH ON WRITING CONTENT	59
4.1	System	63
4.2	Study	66
4.3	Results	68
4.4	Discussion	73
5	EFFECTS OF SENTIMENT BIAS OF PHRASE SUGGESTIONS	76
5.1	Phrase Suggestions are Biased Positive	80
5.2	Effects of Suggestion Sentiment on Writing Content	88
5.3	Results	101
5.4	Discussion	105
6	LEARNING TO MANIPULATE CONTENT	108
6.1	Applying Counterfactual Learning to Suggestion Generation	112
6.2	Simulation Experiment	116

6.3	Collecting Acceptance Data from Human Writers	120
6.4	Discussion	125
7	OPPORTUNITIES FOR PREDICTIVE GUIDANCE	126
7.1	Opportunities for Content Guidance	127
7.2	Experience Design for Content Suggestions	129
7.3	Feasibility of Generating Questions	137
7.4	Summary	140
8	CONCLUSION AND FUTURE DIRECTIONS	141
8.1	Implications	143
8.2	Future Work	150
8.3	Parting Thoughts	153
	REFERENCES	173

Listing of figures

3.1	Image captioning task with predictive text suggestions visible . . .	23
3.2	Estimated means of text content measures	42
3.3	Estimated pairwise differences of text content measures	43
3.4	Average speed (ratio to baseline) by suggestion visibility	46
3.5	Estimated pairwise differences in task load by suggestion Visibility	48
3.6	Use of nouns, adjectives, and color words by Visibility	50
4.1	Phrase preview keyboard	62
5.1	Example of contrasting suggestion sentiment	78
5.2	Positivity of generated suggestions vs original text	87
5.3	Comparison across experimental conditions of suggestion sentiment	94
6.1	Simulation results for suggestion policy optimization	119
6.2	Example reviews written in data-collection study	121
6.3	Estimates of acceptance rate of optimized suggestion policy	124
7.1	Length of text written by prompt type	134
7.2	Perceived relevance of prompt by type	135
7.3	Writer preference by prompt type	136
7.4	Examples of sentence clusters and hand-written questions	139

Acknowledgments

Prof. Krzysztof Gajos, my primary advisor, has been incredibly supportive of me throughout our long time together. He taught me what good research looks like and walked me through how to do it. I'm especially impressed by his patience: I have been very slow to learn, but he did not give up on me. Maybe I finally understand what "conceptual contribution" means now.

Dr. Adam Kalai started investing in me years before I interned with him at Microsoft Research. Our time together was invigorating. I learned a lot not just about machine learning but also research strategy, like how to be meticulously concerned about core details and ruthlessly pragmatic about all else. He often reached out to me well before I thought I was ready to ask to meet again, and went out of his way to buy me coffee more times than I remember.

Prof. Henry Lieberman was the first one to welcome me into the world of human-computer interaction by inviting me to the MIT Media Lab. While there, he helped me start to see how machine learning might be critical to empowering HCI, years before that once-radical idea became mainstream. He too was patient and supportive throughout my time at the Media Lab and beyond.

I have also been blessed by amazing collaborators and colleagues, who shaped half-baked ideas, reviewed drafts and practice talks, untangled technical challenges, inspired new ways of thinking, and more. At the great risk of missing some very important people, I'll name a few: Elena Agapie, Khalid Alharbi, Jason Alonso, Ofra Amir, Michelle Borkin, Alex Cabral, Kai-Wei Chang, Krysta Chauncey, Sebastian Gehrmann, Elena Glassman, Catherine Havasi, Cheng-Zhi (Anna) Huang, Bernd Huber, Maia Jacobs, Nam Wook Kim, Steve Komarov,

Andrew Mao, Katharina Reinecke, Pao Siangliulue, Zaria Smalls, Robyn Speer, Herb Sussman, Miraç Suzgun, Joseph Williams. Also, special thanks to our Harvard admins: Shannon Cardillo, Jess Jackson, David Lopez, and Anana Charles. I may never know all of the details that you took care of, skillfully and without complaint, so that I could focus on research.

Many more people have mentored and advised me, including: Ryan Adams, John Belina, Cullen Buie, Yiling Chen, Finale Doshi-Velez, Norden Huang, Crystal Huff, James Kelly, Nathan Matias, Sally McKee, Frank McNamara, David Parkes, Rosalind Picard, Stuart Shieber, Patrick Winston.

I would also like to thank (and each of these lists is very incomplete):

- *others in the Harvard CS community*, including: Pavlos Protopapas, Paul Tylkin, Harry Lewis, Uri Braun, . . .
- *the Harvard Aikido club*, including: Dolita C, Sally J, Alexis G, Kevin T, Peg H, Dya L, . . .
- *Citylife Church*: Bobby K, Andrew K, Dave C, Sam L, Nathan and Psalm H, Kaitlin G, Tim C, Bo Z, . . .
- *the Harvard and MIT Christian communities and the MIT Cross Products*, including: Martin S, Peter J, Daniel W, Jeff B, Kevin F, Tout W, Marcus G, Kunle A, . . .
- *my new colleagues at Calvin University*: Keith VL, Randy P, Joel A, David K, Derek S, Harry P, Stacy DR, Sharon G, Chris W, . . .

Finally, I would like to thank my family. Susan supported me with so much of her time and energy—and then somehow boosted that even more in busy

times. Naomi and Esther have a joy in life and learning that's been increasingly contagious in their second year of life. My parents (Harvey and Julia), brother (Austin), and in-laws (Clare and Sharon) have been a huge support, especially in the months after the twins' birth.

I remember Dustin Smith, with whom I worked on AI at the Media Lab, and Marvin Minsky, who inspired and partly advised that work. They introduced me to the quest for machine intelligence, and both died during my time at Harvard.

I also remember Sioux Hall and Terry Gibbs. Sioux led the Harvard Aikido club with compassionate strength; only later did I learn how much that had characterized other parts of her life also. Terry Gibbs was a patient and giving mentor on and off the mat.

This work was supported in part by a grant from Draper. Some of this work was done while I visited Microsoft Research, Cambridge. This dissertation was typeset in XeLaTeX, using <https://github.com/suchow/Dissertate>.

Thanks also to the hundreds of people who participated in my studies, recruited through Amazon Mechanical Turk and the Harvard Decision Science Lab.

1

Introduction

Intelligent text entry technologies have been developed to enable writers to express themselves with less effort and fewer mistakes. One of the most widely deployed mechanisms in intelligent text entry technology is predictive text, which is ubiquitous on touchscreen keyboards and increasingly deployed in desktop-based composition interfaces as well. For example, predictions are enabled by default on both Android and iOS smartphones, and Google’s Smart Compose (Chen et al., 2019) offers phrase predictions on both desktop and mobile.

Predictive text is unprecedented: for the first time in the history of writing,

our tablets do not only record our words, they offer words to us. But predictive text systems are currently designed and evaluated under the assumption that the way in which text is entered does not affect the content that people will choose to enter using it. This assumption may lead to missed opportunities for more effective design and unawareness of potential negative consequences such as bias.

The technology used to enter written content continues to be designed and evaluated primarily based on the task of transcribing text, not composing it. The ubiquitous QWERTY keyboard layout was developed and evaluated according to the needs of people writing down messages that they heard in Morse code (Yasuoka & Yasuoka, 2011). Even though taking dictation is very rare today compared with the 1800s, almost all studies of text entry systems, including all those presented at the most recent top-tier HCI conference (CHI 2019), have maintained the same evaluation approach: “type this.” By specifying exactly the text to type, a transcription study ignores even the possibility that the text entry system can affect the content of writing.

Data-driven predictive systems such as predictive text suggestions can have several kinds of biases. Some biases are inherent to the design of the system: for example, predictive text systems are typically designed to show only those

words that are most likely to occur in a given context. Other biases emerge from the interaction of training data and algorithms: for example, the training data may over-represent some kinds of writing, or even if the training data is superficially neutral with respect to some trait of the writing, the learning algorithm may perform better when modeling one type than another. Biases may also be created by intentional data-driven manipulation of system behavior by its designers, e.g., by shifting the objective function.

My thesis is that **the text that people write using predictive text entry systems reflects the biases of these systems.**

The evidence that supports this thesis is the following:

- When writing with predictive suggestions, participants in my first study wrote captions that were shorter and included fewer words that the predictive system did not expect (Chapter 3).
- The length of predictive suggestions (single words vs multi-word phrases) affected the number of unexpected words that participants wrote in restaurant reviews in my second study (Chapter 4).
- Human annotators found that phrase predictions generated by a conventional phrase prediction system are biased towards positive sentiment despite attempts to balance the sentiment of training data (Chapter 5).
- Participants wrote restaurant reviews that included more positive content when the phrase suggestions that they used were artificially slanted to-

wards positive content, but the reviews were more balanced when the suggestions given to writers were slanted negative. However, writers generally expressed a preference for the positively slanted system (also presented in Chapter 5).

Suggestion content that systematically differed from most-likely predictions was typically rejected by writers, who tended to view them as less relevant. However, I also present a simulation experiment that illustrates that system designers can intentionally manipulate content by explicitly managing the tradeoff between content manipulation and suggestion acceptability.

In a final exploratory study, I probed whether alternatives to the traditional next-word or next-phrase interaction might guide writers in generating novel and valuable content. I found that even high-quality example sentences were often viewed as irrelevant, but those same sentences re-expressed as questions were perceived as useful and relevant (Chapter 7).

The chapters are organized as follows:

1.1 INHERENT BIAS

Predictive text systems exhibit bias **inherent to their interaction design**: they are designed to offer the words that would be least surprising to the system given the preceding context. In chapter 3, I studied how writers respond to

this bias when writing image captions using predictive text systems. I show that people appropriate the system’s aversion to surprise: they write fewer words that the system would not have expected, leading to captions that are shorter overall.

Several recent systems (Quinn & Zhai, 2016; Chen et al., 2019) also hide suggestions when the system’s predictive confidence is low, causing there to be some words that the system would *expect* but chooses not to *show*. When participants in my study wrote with this type of thresholded suggestions, they still wrote shorter captions, but also used fewer of the expected words—perhaps because those words were not shown.

Several recent systems, such as Google’s Smart Compose, offer predictions of multi-word phrases (Chen et al., 2019). In chapter 4, I show that phrase predictions reduce unexpected words by an even greater extent than word predictions. Many participants expressed preference for the phrase suggestions (perhaps due to their novelty), but the effect on overall typing speed was neutral: timing data suggest that the benefit of being able to type multiple words with a single gesture was offset by the increased cost of evaluating each suggestion.

1.2 EMERGENT BIAS

Predictive text systems exhibit emergent bias due to interactions between algorithms and training data. In chapter 5, I show that phrase prediction systems readily exhibit sentiment biases, even when steps are taken to avoid biases in training data. I then show that people, when offered suggestions with biased sentiment, appropriate that sentiment in their writing—but only when they perceive the suggestions to be relevant. Together these findings indicate a *chain of bias*: bias in training data leads to bias in system behavior, which in turn leads to bias in what people write.

1.3 MANIPULATED BIAS

Predictive text systems can be manipulated to encourage specific kinds of desirable content, whether for benevolent ends (such as encouraging civil discourse on online forums) or malevolent (such as encouraging disinformation or click-bait). In chapter 6, I address a significant blocker to successful manipulation of suggestion content: how to maintain acceptability of the suggestions being offered. I apply a technique based on counterfactual learning to estimate the perceived relevance of a candidate system without requiring repeated human-subjects experiments. I demonstrate its effectiveness using simulated acceptance

data, then show an example of its results on suggestion acceptance data collected from human writers. In light of this potential for manipulation, I call on platforms to be transparent in the ways that they might attempt to manipulate writing and for further research on how third-party auditors might hold platforms accountable for their content nudges.

1.4 OPPORTUNITIES FOR GUIDING CONTENT CREATION

The results presented in the above chapters indicate that next-word prediction, the dominant interaction technique used for predictive text, is especially vulnerable to bias. Could alternative interaction techniques reduce the risk of bias while also enabling new kinds of support for content generation?

Drawing inspiration from writing interventions and creativity support tools, I propose next-topic prediction as a task for predictive text in chapter 7. An exploratory study comparing two ways of operationalizing it in an interactive interface resulted in a clear preference for a design that presents *questions*. I then present an initial feasibility study of applying predictive modeling techniques to generate relevant questions during a writing task.

2

Related Work

This dissertation builds on prior studies in intelligent text entry systems, predictive language modeling, and the cognitive processes involved in creative tasks such as writing.

2.1 TEXT ENTRY

Modern touchscreen keyboards use a flexible interface typically called the suggestion bar, which can be used to show completions, corrections (possibly automatically accepted) (Bi et al., 2014), alternative interpretations of ambiguous

stroke gestures (Reyal et al., 2015), and even emoji and other functionality.

The low-level process of text entry has been the subject of extensive study and innovation. Predictive language modeling in text input interfaces were first developed to assist those with motor impairments and poor typists (Darragh et al., 1990), but have seen much wider adoption today. They reduce motor effort and errors, but their overall effect on speed depends on system design choices (Quinn & Zhai, 2016) and individual characteristics (Koester & Levine, 1994).

2.1.1 INTERACTION TECHNIQUES

Many different interactions have been evaluated for intelligent technology for text entry. Broadly, these techniques can be divided as follows:

- Adaptations to the *processing* of the stream of input actions (e.g., key taps)
- Adaptations to the input interface *shown* to writers
- Interactive functions added to already-entered text (e.g., auto-correction suggestions); we will not discuss these

INPUT PROCESSING

Traditional text entry was characterized by a fixed mapping of physical inputs (presses on discrete mechanical keys) to character inputs. However, contemporary systems instead treat physical inputs as noisy signals of the desired characters. One such technique is word gesture typing, in which writers can enter words by tracing over their letters on a conventional keyboard layout (Zhai & Kristensson, 2012). Many study participants enjoyed gesture typing (Reyal et al., 2015), and it is available as a standard feature on nearly all current smartphones. Also, recent work by Vertanen and others found speed and accuracy benefits from systems that infer the desired input from sequences of imprecise or erroneous taps on touchscreen soft keyboards (Vertanen et al., 2015, 2018, 2019). By adjusting the logical target region of each key dynamically, these systems can function almost imperceptibly to writers (Findlater & Wobbrock, 2012; Baldwin & Chai, 2012). Similar approaches can be used to enable fundamentally ambiguous interfaces, such as the 1Line keyboard (Li et al., 2011), that map multiple characters to the same input region.

DYNAMIC INTERFACE ADAPTATION

Input processing systems are designed to process input provided by a human who has a fixed mapping from desired words to motor actions. In contrast, many systems have explored the potential of how users can respond to changing interfaces. For example, eye-tracking keyboards can dynamically adjust the dwell times required to enter a letter based on its likelihood in context (Mott et al., 2017). One of the most distinctive examples is a system called Dasher, which uses a spatial navigation metaphor to allow users to enter text (Ward et al., 2000; Rough et al., 2014; Vertanen & MacKay, 2010). However, the most common interaction of this type is *autocomplete*, discussed below.

AUTOCOMPLETION

Autocompletion systems adapt the text entry interface by offering completions of partially entered text that can be entered using shortcut gestures. These systems use language modeling of existing text to reduce the number of actions required to enter text (Koester & Levine, 1994; Stoop & van den Bosch, 2014). They can draw on both general past texts and the specific user's recent history (Fowler et al., 2015). Early systems required the prefix text to be correct (Darragh et al., 1990), but current systems integrate disambiguation to allow the

same interface to be used for both completion and correction (Bi et al., 2014).

Recent systems have explored alternative interfaces, such as offering complete-sentence replies (Kannan et al., 2016) and offering a single highly likely phrase continuation, like Google’s Smart Compose (Chen et al., 2019).

The predictions are usually generated by a language model that is trained on a wide range of data (Vertanen et al., 2018, 2015), though some implementations customize the predictions using the author’s past writing or suggestions from conversation partners (Fiannaca et al., 2017). Recent systems have explored alternative interfaces, such as offering complete-sentence replies (Kannan et al., 2016) and offering a single highly likely phrase continuation, like Google’s Smart Compose (Chen et al., 2019).

2.1.2 EFFECT ON PERFORMANCE

Almost all studies of text input have relied on transcription studies: participants are given phrases to enter as quickly and accurately as possible (Polacek et al., 2013).

Effective use of autocomplete interactions requires that writers read the predictions. Various researchers have pointed out that the costs of perceiving and evaluating the presented options detracts from the effort-saving benefit of reduc-

ing the number of error-prone actions, so predictions may not actually improve performance (Koester & Levine, 1994; Trnka et al., 2009; Quinn & Zhai, 2016).

Use of autocomplete suggestions varies widely between people (Buschek et al., 2018) and may vary depending on the emotional state of a single person (Ghosh et al., 2019).

2.1.3 EFFECT ON CONTENT

In 2014, Kristensson and Veranen advocated composition tasks in text entry studies (Kristensson & Vertanen, 2014), but only a few studies published since then (Buschek et al., 2018; Vertanen et al., 2018 and the work we present in this dissertation) have heeded their advice.

One of these studies presented a methodology to collect typing data in the wild, but the design consideration of participant privacy prevented collection of rich data about writing content (Buschek et al., 2018). Nevertheless, the study did find that people used suggestions extensively and that there were substantial differences in suggestion use between writers.

One text entry study (investigating key-target resizing) asked pairs of participants to chat with each other, so the text entered was highly open-ended and natural, but the researchers made no attempt to study how the text entry

system might affect content (Baldwin & Chai, 2012).

One study of an interactive translation prediction system found a weak trend towards translations being more conventional (as measured by higher BLEU scores when comparing with reference human translations) when written using a predictive text system that suggested phrases based on machine translation (Green et al., 2014).

2.2 BIAS IN PREDICTIVE SYSTEMS

Although machine learning algorithms typically do not contain discriminatory biases by themselves, recent work has demonstrated that systems based on these algorithms can make prejudiced decisions—in domains such as hiring, lending, or law enforcement—if the data sets used to train the algorithms are biased (Barocas & Selbst, 2016). Such biased data sets are more common than initially suspected: Recent work demonstrated that two popular text corpora, the Google News dataset and the Common Crawl database of website text, contain race and gender biases, and machine learning systems incorporate those biases into their internal representations (Caliskan et al., 2017) unless specific effort is made to remove a given bias (Bolukbasi et al., 2016).

2.3 COGNITION IN CONTENT CREATION

2.3.1 WRITING PROCESS

Writing integrates processes at many different levels of cognition (Deane et al., 2008; Torrance & Galbraith, 2006; Hayes & Chenoweth, 2006). Technology has been developed to aid in some of the higher-level processes of writing, such as collaboration (Teevan et al., 2018; Amir et al., 2019), ideation (Clark et al., 2018), information gathering (Babaian et al., 2002), and feedback (Hui et al., 2018).

Although many writers may think that they first plan out a specific phrase or sentence and then type it in, studies of writing process suggest that planning is interleaved in text production (Torrance & Galbraith, 2006). For example, writers plan sentences only a few words ahead of producing them (Martin et al., 2010), pause longer between larger semantic units (Torrance, 2015), and look back on previously written text while writing new text (Torrance et al., 2016).

Thus, decisions about writing content may be influenced by events that happen during text production, such as a predictive text system offering a suggestion.

2.3.2 SPEECH ACT THEORY

Communication (written or spoken) is more than description of the current state of the world; it is goal-directed. Speech Act theory accounts for speakers' (and writers') communicative choices in terms of the effect that the communication has on the hearer (or reader). In particular, Rational Speech Act (RSA) theory models speakers' choice of expressions by probabilistic reasoning about how listeners will update their beliefs about the speaker's intentions (Goodman & Frank, 2016). RSA has been used to model language choices made in lab settings (Goodman & Frank, 2016) and corpora (Orita et al., 2015). Objectives based on RSA have been used to automatically generate pragmatic image captions (Cohn-Gordon et al., 2018).

RSA theory weights the communicative utility of an utterance against the cost of producing it (Goodman & Frank, 2016; Orita et al., 2015). By offering some words as predictions, predictive text suggestions reduce one kind of cost—the number of taps necessary. However, they reduce the cost of certain utterances more than others, so people may therefore favor lower-cost utterances, i.e., those suggested as predictions.

2.4 TOOLS FOR CONTENT CREATORS

Technology has been developed to support content creators in a wide variety of ways, such as structuring collaboration in writing (Teevan et al., 2018), ideation (Clark et al., 2018), information gathering (Babaian et al., 2002), and feedback (Hui et al., 2018). In this dissertation we focus on tools that provide content suggestions to writers, rather than other kinds of support such as teamwork coordination or feedback. This section discusses tools that provide content suggestions in a variety of domains.

2.4.1 TECHNOLOGY-MEDIATED WRITING INTERVENTIONS

Writers can benefit from guidance about what makes for good writing in a particular domain. The IntroAssist system provided a checklist of elements to include in an effective introductory help request message, along with tagged examples (Hui et al., 2018). The Critique Style Guide assisted untrained workers writing design critiques by providing a “style guide” containing attributes of helpful feedback, paired with examples (Krause et al., 2017). Both the elements of IntroAssist and the attributes of Critique Style Guide were static and curated by experts (although the guidelines in Critique Style Guide were informed by using domain-general language processing methods to predict ratings of helpful-

ness); neither system provided automatic feedback.

Systems can scaffold novice writers in specific domains. When describing scientific images (graphs and charts), novices more often wrote descriptions containing desired aspects when an initial description was generated by filling templates with their answers to domain-specific questions (Morash et al., 2015). Story prompts can increase the quality of stories written by students, especially students with learning disabilities (Graves et al., 1994). The CritiqueKit system used formative assessment and examples to encourage writers to provide specific, actionable, and justified feedback on creative artifacts (Ngoon et al., 2018).

SUMMARIZATION

Summary Street is an automated writing evaluation system for summarization tasks that provides feedback on the degree to which a written summary covers various aspects of a source text (Wade-Stein & Kintsch, 2004). It uses LSA to measure similarity between the words used in sentences in the summary with words used in each section of the source text. Its feedback employs goal-setting and progress measurement to guide and motivate writers to make revisions.

CREATIVE WRITING

Systems have been developed to support writers in exploring alternatives in writing and related tasks. For example, fan-fiction writers have used VoiceBox by Botnik, a word prediction system that can be trained on a targeted corpus of writing, to produce entertaining fan fiction.

2.5 AUTOCOMPLETION

The “autocomplete” interaction has become widespread not only in text entry but also search queries, programming, and tagging.

In web search, it can be used to assist query formulation by discovering other possible search queries (Dehghani et al., 2017). In programming, it can be used to scaffold the entry of complex structures. In list selection, autocomplete can be used to help taggers converge on a set of consistent tags.

Autocomplete interactions can use various forms of context to generate completions. For example, “AutoComPaste” generates suggestions based on text visible in other windows, which enables copying any visible text by simply typing a prefix of it rather than an explicit copy and paste (Zhao et al., 2012). “Restorable Backspace” uses recently deleted text for suggesting phrases to insert, which helped users who delete and retype to fix mistakes (Arif et al.,

2016).

3

Word Suggestions Discourage the Unexpected

This chapter contains content published in the proceedings of IUI 2020 ([Arnold et al., 2020](#)). The pronouns “we”, “our”, and “us” in this chapter refer to the authors of that paper.

This chapter reports a study where we compared image captions written with different kinds of predictive text suggestions. Our key findings were that captions written with suggestions were shorter and that they included fewer words

that the system did not predict. Suggestions also boosted text entry speed, but with diminishing benefit for faster typists.

The contributions of this chapter are:

- a computable *measure* that is sensitive to the degree to which writing content reflects what a system expects
- results from a human-subjects writing study that captions written were shorter and contained fewer unexpected words when predictions were visible
- results from the same study that predictive suggestions still have a significant effect on content even when low-confidence predictions are hidden, and
- two conjectured mechanisms (*skip* and *substitution* nudges) to account for these results, supported by supplemental analysis

Predictive systems are designed to offer suggestions that reduce writers' typing effort by offering shortcuts to enter one of a small number of words that the system predicts are most likely to be typed next. As such, the suggestions are, by construction, the words that are the most predictable in their context. Thus, writers who follow these suggestions may create writing that is more predictable than they would create without such suggestions.

To study what effects predictive text suggestions might have on content, we conducted a within-subjects study (N=109 participants, $109 \times 12 = 1308$ texts)

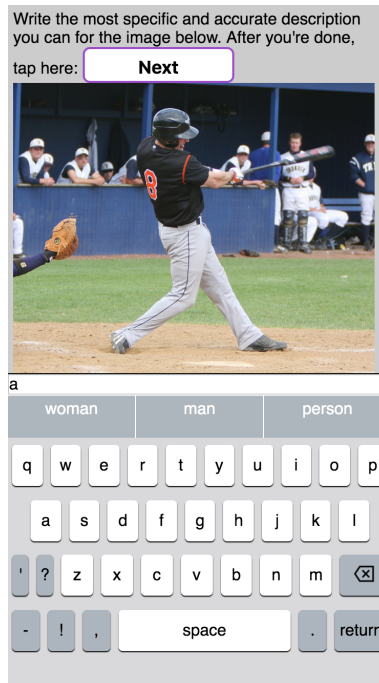


Figure 3.1: When writing with suggestions, people tended to choose shorter and more predictable wordings (such as 'man', in the screenshot above, instead of 'hitter' or 'baseball player'), than writing without such suggestions. (Image credit: <https://flic.kr/p/6Yxc61>)

where writers wrote captions for images while we varied characteristics of the predictive suggestions that a keyboard offered. Our study compared always-available predictive suggestions against two alternatives. To measure the overall effect of suggestions on writing content, we compared to a baseline where suggestions were never shown. (We were able to make this comparison because we used measures of *content* that were insensitive to differences in *process*.) And to study the content effects of an intervention previously studied only for efficiency (Quinn & Zhai, 2016), we also compared always-available suggestions to a condition in which suggestions were hidden when the predictive system had low confidence.

Our key findings were that captions written with suggestions were shorter and that they included fewer words that the system did not predict. Thus, predictive text led to more predictable writing. We also found that suggestions increased typing speed, but with diminishing returns for faster typists.

3.1 BACKGROUND

This section discusses background specific to the confidence gating intervention we study. General background and related work can be found in Chapter 2.

The decision of if and when to proactively offer assistance is one of the key

choices in the design of mixed-initiative user interfaces (Horvitz, 1999). Predictive text systems are no exception: some systems offer suggestions almost constantly (e.g., the “suggestion bar” in most touchscreen keyboards), while other systems, such as Smart Compose (Chen et al., 2019), offer suggestions only in situations where they are highly confident that the suggestion will be accepted. One study found that this confidence thresholding intervention made text entry faster when compared with always-visible suggestions (Quinn & Zhai, 2016), but the effect of this intervention on content is unknown.

The diagnostic automation literature has found that confidence thresholding can influence human attitudes and behaviors. Diagnostic automation systems such as alarms can use a confidence threshold to trade off misses and false alarms, as described by signal detection theory. Varying the threshold can result in different human responses to the system; frequently studied dimensions include reliance, compliance (Meyer, 2004), and trust (Chancey et al., 2017). These different responses can, in turn, affect how well people perform at detection tasks when using these systems (Wickens et al., 2006).

Although these studies typically focused on binary decisions in repetitive contexts, they may have implications for open-ended tasks like text composition. For example, the confidence threshold may affect the degree to which people

attend to predictive text suggestions, rely on them, or comply with them. If the system shows suggestions rarely but those suggestions are often useful (corresponding to a low false alarm rate), the writer may perceive system as being more useful, and thus pay more attention to it.

Perceptual considerations may also be relevant to how confidence thresholding affects writers. When the confidence level of the predictive system transitions from below-threshold to above-threshold, a suggestion will appear suddenly. Since the appearance of new content captures attention (Yantis & Jonides, 1984; McCormick, 1997; Rauschenberger, 2003), the writer may pay more attention to the new suggestion than if suggestions had been always available. If the new suggestion is irrelevant, it may interfere with the writer's working memory (Chenoweth & Hayes, 2003); if it is relevant, it risks out-competing the word that the writer would have generated (Raaijmakers & Jakab, 2013).

3.2 RESEARCH QUESTIONS

Our main research question is: how do predictive suggestions affect what is written? Since we expect that the primary mechanism of this effect might be that people *accept* suggestions that are offered, our more specific question is:

RQ1: To what degree do people choose the words that the system suggests?

We also wondered how suggestions might affect the length of text entered. Since suggestions reduce the physical effort (number of taps) required to enter texts, we wondered if writers would choose longer texts when suggestions were available. So our second research question is:

RQ2: How do suggestions affect text length?

Predictive systems using confidence thresholding have been widely deployed (Chen et al., 2019) and have been studied for speed effects (Quinn & Zhai, 2016), but their effects on content are unknown. Since applying a threshold reduces the frequency at which the system presents suggestions, its suggestions may have overall less effect on the content written than when suggestions are always available. But subjective and perceptual factors (discussed in the Related Work) may cause a relatively larger effect when the suggestions *do* appear. Since even the direction of any content effects is unclear, we ask:

RQ3: How does the effect of suggestions on writing content differ if only high-confidence suggestions are shown?

Finally, the literature is currently divided on the impact of intelligent text entry technology on speed. Some studies found speed and error rate benefits (Alharbi et al., 2019), especially for systems that permit ambiguous input gestures such as swipes (Reyal et al., 2015) and imprecise tapping (Vertanen et al.,

2015, 2018). But other studies failed to find speed benefits for predictive suggestions (Quinn & Zhai, 2016 and Chapter 4). The authors of those studies conjectured that the time required to attend to suggestions more than made up for the speed benefit of avoiding extra taps. However, the temporal costs of these two activities may have substantial individual differences (Koester & Levine, 1994), so there may be some people for whom predictive suggestions are indeed helpful (such as people with motor impairments) that may not be observed in small studies. Also, the speed impact of predictive suggestions depends on their accuracy, as pointed out by authors as early as Koester & Levine (1994); recent advances in language modeling technology have substantially improved predictive accuracy. Finally, transcription studies require the additional task load of attending to the text to transcribe, so study design may have a large impact on text entry performance (Polacek et al., 2013). Few studies have measured the speed of text entry outside of transcription tasks (exceptions include Vertanen et al. (2018) and our own work in Chapter 4). So our final question is:

RQ4: How does suggestion visibility affect text entry speed?

3.3 STUDY

To evaluate the effect of predictive text on writing content, we conducted a within-subjects experiment in which participants wrote captions for images while we varied the visibility of predictive suggestions that the keyboard offered.

3.3.1 TASK

An open-ended writing task allowed us to measure the effect of suggestions on content. We chose image captioning as our task because it was short, controlled, and repeatable: many different images can be captioned in a short time, so a within-subjects design was feasible. The range of possible captions for a single image was wide enough to observe differences in content but narrow enough that the variance in content characteristics between writers was not too large.

In each trial, participants were instructed to write a “specific and accurate” caption for a given image, by typing on a simplified touchscreen keyboard. Figure 3.1 shows an example of the task.

3.3.2 DESIGN

We manipulated a single factor, the VISIBILITY of suggestions presented by the touchscreen keyboard, with three levels:

Always The keyboard showed three predicted words above the keyboard, using the familiar “suggestion bar” interface.

Never No suggestions were shown (the suggestion bar was hidden)

OnlyConfident Like ALWAYS, except the keyboard only showed suggestions when the confidence of the predictive model exceeded a threshold.

The study was a within-subjects design: each participant wrote twelve captions, four with each level of VISIBILITY (NEVER, ALWAYS, and ONLYCONFIDENT). The order of conditions was counterbalanced across participants, but the images were presented in a fixed order, resulting in a counterbalanced assignment of images to VISIBILITY conditions.

3.3.3 MEASURES

CONTENT

Imagine typing a given sentence while a predictive text system offers suggestions. Sometimes a word to be entered will appear as one of the three suggestions before even its first letter is entered; we refer to such words as *predictable*. We refer to all other words as *unpredictable* (even if it is later suggested after more letters are entered). The predictability of a word is a property of the *text*,

not the manner in what that text was *entered*: we count a word as predictable even if the suggestions were *disabled* at the point that it was actually entered. This contrast is crucial to be able to compare the content written between different types of suggestion systems.

Texts differ in predictability: on one extreme are texts that use only suggested words, on the other would be texts that are generated by *avoiding* initially-suggested words. This observation motivates our primary measure.

Our primary measure is the number of *predictable* words. Since the length of text written could differ between conditions, we also measure the total length of captions in words and the number of words that were *not* predictable words (predictable + unpredictable = total length). To simplify analysis, we stripped all punctuation except for mid-word. (Almost all captions written were a single sentence, so punctuation was not needed to separate sentences.) These measures allow us to answer RQ1–RQ3.

We also measured the *uncorrected error rate* as the number of low-level errors (typos, misspelling, etc.) that were present in the caption that the writer submitted. (Since our primary interest was how system design affected *content*, we ignored errors that the writer corrected before submission.) Since most captions had zero or one typos, we simplified our analysis to consider only whether or

not a submitted caption included a typo.

To label typos, one of the authors inspected all writing, blind to condition, with the help of the Microsoft Word contextual spelling checker, and corrected typos and spelling mistakes including mismatched articles ('a' vs 'an'). Grammatical and factual errors, which occurred rarely, were left uncorrected. Any caption that was corrected in this way was labeled as having a typo.

Since typos would artificially reduce the number of predictable words, leading to inflated estimates of content effects, we computed that measure on typo-corrected text also.

PROCESS MEASURES

To answer RQ4, we used logged data to compute *typing speed*. We compute speed by dividing the final text length in characters (including spaces) by the interval between the first and last input action on the typing screen.

We used the participant's mean typing speed in the NEVER condition as a baseline to control for individual differences (which could stem from prior touch-screen typing experience, effort invested, device characteristics, and many other factors). Our main measure was the *ratio* of the participant's typing speed to this baseline speed.

SUBJECTIVE MEASURES

We collected both block-level and overall subjective measures. Surveys after each keyboard block collected task load data using all six NASA TLX items on a 7-point scale (Hart, 2006). We analyze the sum of these measures, but also individually examine the “physical” and “mental” load items, as has been done in prior work (Quinn & Zhai, 2016).

The final survey asked participants to pick which of the three keyboard designs they experienced were “most helpful” for three goals: accuracy, specificity, and speed. Keyboard designs were indicated by number, and participants could see all of their captions for reference. We analyzed the total number of times that participants picked each keyboard design.

3.3.4 ANALYSIS

We applied statistical estimation methods for our primary outcomes (Dragicevic, 2016). Except where indicated, we estimated means and confidence intervals by non-parametric bootstrapping. Since we expected substantial individual differences, we bootstrapped grouped by participant: Each of the 10,000 bootstrap iterations resampled participants with replacement; we used the complete data for each participant chosen.

Since we expected substantial variance across both participants and images for all measures, we used lme4 (Bates et al., 2015) to estimate linear mixed-effects models at each bootstrap iteration with both participant and image as random effects. (The random effects structure mitigates the pseudoreplication that would otherwise occur from analyzing trial-level data.) We report the bootstrapped estimates of the means and pairwise contrasts for the VISIBILITY fixed effect.*

3.3.5 PROCEDURE

IMAGES

We used 12 images selected from the Microsoft COCO (Common Objects in Context) dataset (Lin et al., 2014). Most images showed people doing outdoor activities (surfing, flying kites, etc.), or familiar scenes such as a train station or a bus on a street. Our selection process was motivated by potential use in a different (unpublished) experiment. We found the twelve pairs of images in the validation set of 2014 COCO release where the two images had the most similar captions. We defined similarity as the tf-idf similarity of unigrams in the concatenation of all five of the captions that crowd workers had originally

*The overall analysis approach was planned and clearly indicated content effects of predictive suggestions, but the analyses reported here reflect refinements and simplifications performed after seeing the initial results.

entered for each image. We randomly picked one image from each pair to be a prompt for caption writing.

PREDICTIVE KEYBOARD

We implemented a custom touchscreen keyboard modeled on commercial keyboards but where we could manipulate the content and visibility of the suggestions. Compared with commercial keyboards, our keyboard was simplified in several ways; the instructions explicitly pointed out the first three:

- the keyboard had a single layer (lowercase only, minimal symbols, and no numbers)
- no ability to edit past text except for backspacing and retyping (and delete key did not automatically repeat), so editing was more cumbersome than people may have been used to
- no auto-correct (the instructions encouraged participants to manually correct typos)
- no automatic insertion of suggestions or corrections; ignoring the suggestion bar produced the same results as if it were not present
- no key target resizing; the mapping from screen location to key was fixed

The UI showed word predictions in the familiar “suggestion bar” interface used in contemporary mobile phone keyboards (Bi et al., 2014; Quinn & Zhai,

2016). When the writer entered a partial word, the suggestions offered completions of that word, otherwise the suggestions showed likely next words. The writer could choose to tap a suggestion, tap a key, or tap the backspace key (which deleted a single letter at a time). The system updated the suggestions after each user action.

Figure 3.1 shows the task as it appeared on a participant’s device, including the image, caption written so far, and suggestions offered by the system. The screen layout ensured that the participant’s complete writing was always fully visible and visually close to the keyboard and suggestions (if applicable); participants may have had to scroll to see the complete image.

The keyboard showed the top three most likely predictions from the language model as suggestions, subject to the constraint that if the cursor was in the middle of a word, all predictions must have the characters typed so far as a prefix.

Our keyboard generated predictions using an LSTM language model using OpenNMT (Klein et al., 2017), trained on image captions. For this study we did *not* give the system access to visual features from the image being captioned (i.e., the system offered the same predictions regardless of image). Models ran on a cloud VM, providing predictions to the client with a typical latency of

under 300ms from tap to prediction visibility.

The language model was a single-layer LSTM, with hidden state dimension of 2048.[†] The model was trained on the COCO training set captions using the Adam optimizer with the “Noam” learning rate schedule (Vaswani et al., 2017), with a base learning rate of 2, 8000 warm-up steps, $\beta_2 = 0.998$, and parameters initialized using the Xavier uniform scheme (Glorot & Bengio, 2010). The batch size was 128. If the norm of the gradient for any batch exceeded 2, it was re-normalized to have a norm of 2. After 10 epochs, the model achieved a perplexity of 16.32 and a top-1 accuracy of 46.00%.

We constructed the ONLYCONFIDENT system by modifying the ALWAYS system to hide all three suggestions when the predicted likelihood of the words was less than a threshold. We chose the thresholding method and value by generating predictions at 1000 randomly chosen beginning-of-word locations in the COCO validation set and logging whether the word that followed was one of the three predicted. We considered thresholding based on the maximum, mean, or minimum likelihood of each of the three predictions, and chose to use the maximum because it obtained the highest AUC. We then chose the threshold value that would have resulted in suggestions being displayed 50% of the time. At

[†]For historical reasons, we actually used a “sequence-to-sequence” model but with the input set to a constant token; this does not affect our results.

this threshold value, the false positive rate was 25.7%. When the maximum confidence dropped below the threshold, the keyboard showed a blank suggestion bar.

PARTICIPANTS

The study was carried out remotely as a mobile web application that participants accessed using their own touchscreen devices.[‡] We recruited 111 participants (61 male, ages 19–61) from Amazon Mechanical Turk. Participants received \$5 for completing the study. Since the experiment required a low-latency connection to our US-based server, we limited participants to those in the US and Canada. We required participants to have a 99% approval rating on at least 1000 HITs. The study was conducted in English; all participants reported “native” or “fluent” English proficiency.

The landing page described the study as using various mobile phone keyboards to type descriptions of images, with an expected time of about 30 minutes. After a statement of informed consent, participants read a description of the task, which promised a \$0.50 bonus for the most specific and accurate captions. They then read a brief overview of the flow of the experiment, which emphasized that they would be using three different keyboard designs and they

[‡]The study procedure was approved by our institutional review board.

should attempt to remember their experiences with each.

Before any of the writing tasks, participants completed a task tutorial with the overall instruction to write the most specific and accurate caption they could for each image. The tutorial included examples of captions that differed in specificity and accuracy. Some pilot participants seemed to think that we simply meant for them to write *long* captions, so we revised the instructions to encourage writers to be concise. Examples were provided, based on different images than those used in the experiment. We did not prevent writers from writing multiple sentences, but all examples provided were a single sentence (as were most captions that participants wrote).

Each participant wrote captions for twelve images. The body of the experiment consisted of three blocks, one for each condition (which we referred to as “keyboard design”). Each block began with a page prominently displaying the number of the keyboard design they were about to use (e.g., “Keyboard Design 3”). Next, participants completed a “practice round” with that keyboard, in which they were given a sentence to transcribe (a caption written for an image, not shown, that was not one of the images to caption). If they did not use suggestions, they were encouraged to complete the transcription task again, in case they had been too fixated on the text to transcribe that they failed to no-

tice the suggestions. Then they typed captions for four images, followed by a survey about their experience with that keyboard. We chose to keep the same keyboard design within each block of trials so that participants could become accustomed to the behavior of each keyboard. The experiment closed with a survey asking for comparisons between their experiences with each of the three keyboard designs, as well as demographics (all questions optional).

The experiment enforced that participants typed at least one word before a caption could be marked as completed, but otherwise no restrictions were enforced on the length or time taken for writing captions. Notably, we did not require participants to use suggestions while writing their captions.

We excluded two participants who visibly violated our instructions to write captions that were specific and accurate. Both wrote captions that averaged less than five words, such as “there is teniss” and “people flying kites.” Other than those written by these participants, all captions seemed generally appropriate and grammatical.

All participants used a suggestion at least once when typing captions, and no participant accepted every suggestion, so we did not need to exclude participants based on those criteria.

3.4 RESULTS

We collected a total of 1308 captions (109 participants after exclusion; each wrote captions for 12 images).

3.4.1 CONTENT EFFECTS

PREDICTABILITY The figures show the estimated means (Figure 3.2) and pairwise differences (Figure 3.3) between suggestion **VISIBILITY** conditions for the main content measures. The strongest contrast that emerged was that an average of about one additional *unpredictable* word was used when suggestions were **NEVER** visible compared to the **ALWAYS** (CI: [0.68, 1.60]) or **ONLYCONFIDENT** (CI: [0.46, 1.27]) conditions.[§] The data also indicate (albeit less clearly) that captions written in **ALWAYS** had around 0.78 (CI: [0.32, 1.24]) more *predictable* words than **ONLYCONFIDENT**.

Figure 3.2 also shows two measures derived from the above measures, length and fraction predictable, which convey no new statistical information but may be useful for interpretation. Captions written with **NEVER**-visible suggestions were longer (14.6 words) than those written in the other two conditions (**ALWAYS**: 13.9 words, **ONLYCONFIDENT**: 13.4 words), with a clear difference of

[§]A pairwise difference that is statistically significant at the $\alpha=0.05$ level (in a null-hypothesis test setting) will have a 95% CI that does not contain 0.

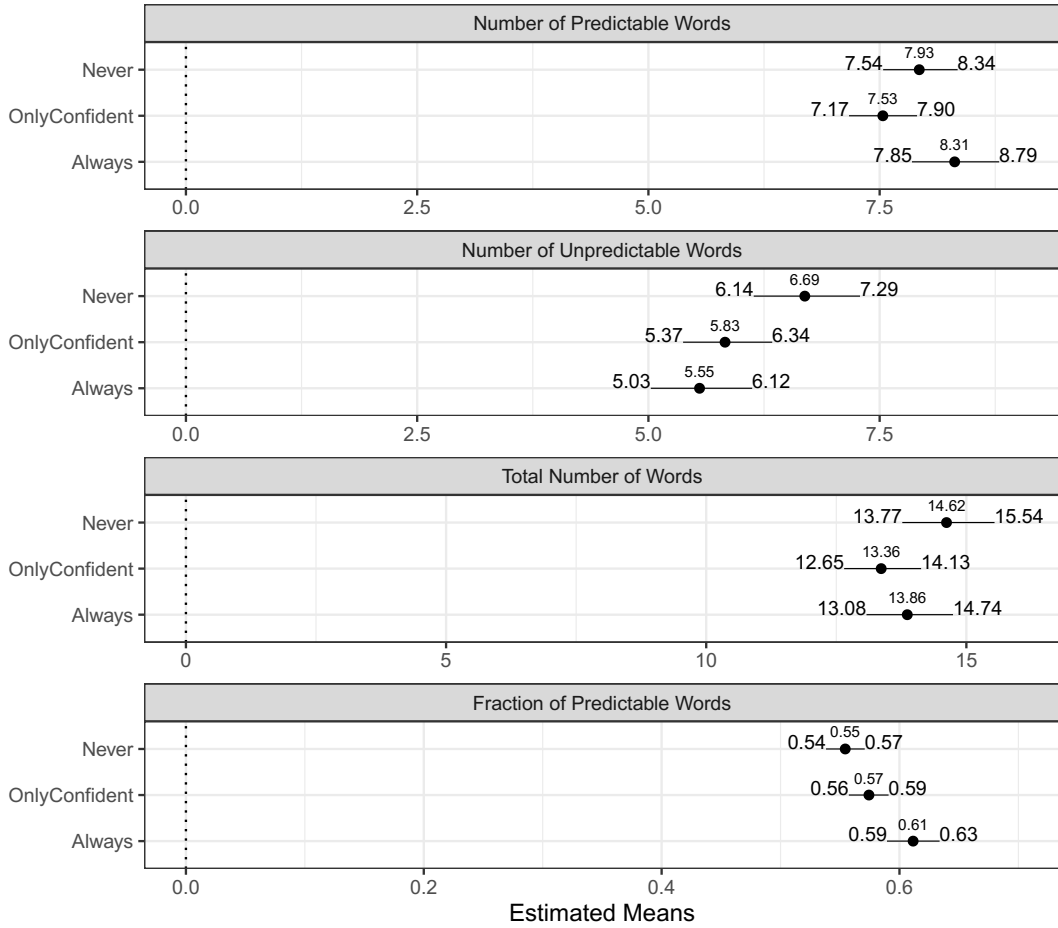


Figure 3.2: Estimated means of content measures for captions written. Error bars show 95% confidence intervals computed by non-parametric bootstrap by participant. Note that the visualization contains redundancy, e.g., the bottom two plots can be computed from the top two.

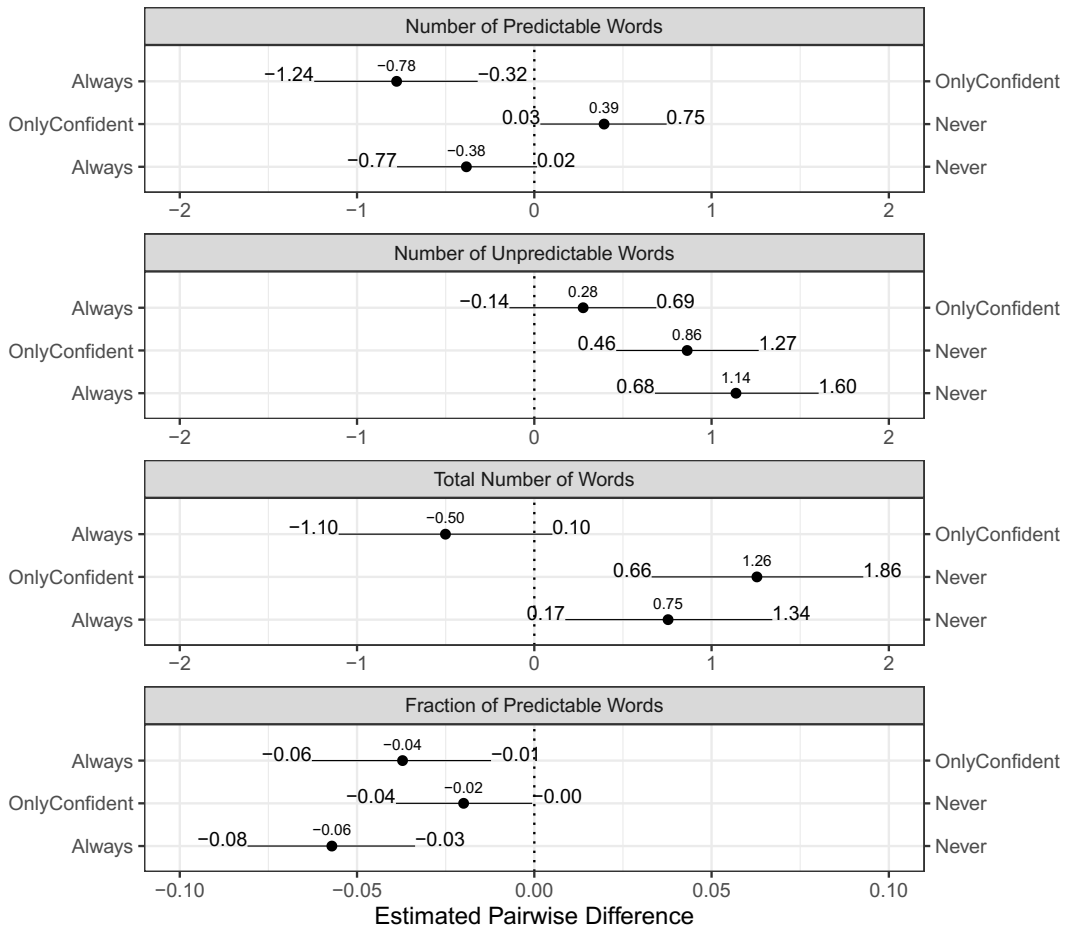


Figure 3.3: Estimated pairwise differences between VISIBILITY levels of content measures for captions written. See Figure 3.2 for notes.

about 1.26 (CI: [0.66, 1.86]) more words with NEVER than with ONLYCONFIDENT suggestions. The difference between NEVER and ALWAYS was in the same direction but did not seem to be as strong (.76, CI: [0.18, 1.36]), and there did not seem to be a substantial difference in caption length between ONLYCONFIDENT and ALWAYS. The fraction of predictable words was about 6% (CI: [3%, 8%]) higher for ALWAYS-visible than NEVER-visible suggestions and about 4% (CI: [1%, 6%]) for ONLYCONFIDENT than NEVER.

TYPOS Suggestions seemed to reduce the number of typos that participants left uncorrected in their captions. Of the 124 captions that had typos, 73 (59%) were written with NEVER suggestion visibility, 27 (22%) with ONLYCONFIDENT, and 24 (19%) with ALWAYS. Comparing the two conditions with suggestions visible (ALWAYS and ONLYCONFIDENT) jointly against the NEVER condition, Fisher’s Exact Test found that the odds ratio for a caption having a typo was 0.31 (CI: [.21, .45]) in favor of fewer typos for suggestion conditions.

3.4.2 PROCESS EFFECTS

We found that baseline typing rate was a strong predictor of the ratio between typing speed with suggestions (either ALWAYS or ONLYCONFIDENT)

and baseline typing speed.[¶] We used a linear mixed model to predict the block-wise mean ratio of speed to baseline speed: $\text{speed ratio to baseline} = a \times \text{baseline speed} + b + \epsilon_{\text{participant}}$, where ϵ represents the participant-level random effect. The 95% confidence interval for b was [1.35, 1.66], indicating that suggestions increased typing speed overall. But the 95% confidence interval for a was [-0.29, -0.14], indicating that as baseline speed increased, the benefit of suggestions decreased. As Figure 3.4 shows, some of the fastest typists in our experiment wrote slower when suggestions were visible, but since our participants included few such typists, we lack evidence to determine whether suggestions would slow down fast typists in general. The figure also shows that we did not observe a significant difference between the two levels of suggestion VISIBILITY (ALWAYS and ONLYCONFIDENT) in terms of speed. To quantify this observation, we fit a separate model including a term for VISIBILITY; the confidence intervals for both VISIBILITY ([-0.08, 0.08]) and its interaction with baseline speed ([-0.05, 0.03]) were nearly symmetric around 0.

[¶]Since NEVER forms the baseline, analyses in this paragraph consider only ALWAYS and ONLYCONFIDENT. Analyses in this paragraph use 1000 iterations of parametric bootstrapping.

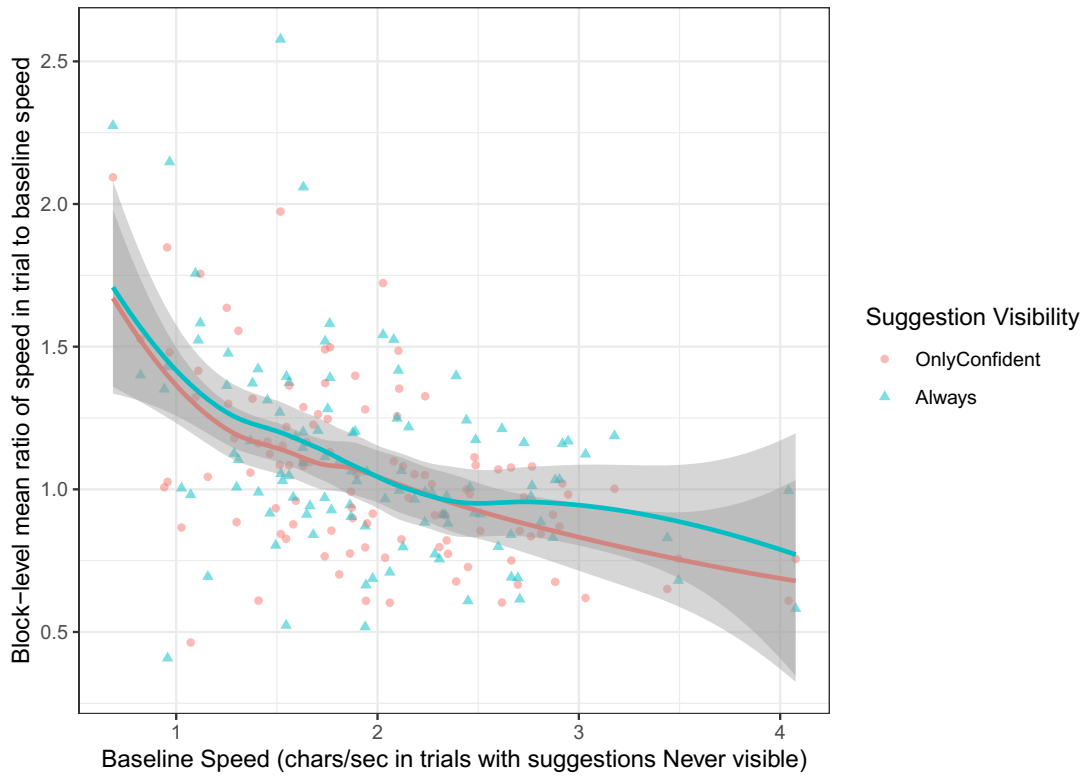


Figure 3.4: Predictive suggestions had a positive effect on typing speed overall, but with less benefit for faster typists. Scatterplot points show block-level average speed ratios, solid lines show loess curves for each VISIBILITY condition, and shaded areas show approximate confidence intervals for each curve.

3.4.3 SUBJECTIVE EXPERIENCE

Ranking results from the closing survey suggest that participants strongly preferred visible suggestions over NEVER and generally preferred ALWAYS over ONLYCONFIDENT visibility. Participants picked the ALWAYS condition as most helpful 206 times, ONLYCONFIDENT condition 101 times, and NEVER condition 20 times. A χ^2 goodness-of-fit test finds that this result would be highly unexpected under the null hypothesis that all three VISIBILITY conditions are equally helpful ($\chi^2_2 = 159.6, p < .0001$).

When suggestions were hidden (VISIBILITY=NEVER), participants reported higher task load overall as well as for both the physical and mental effort items individually. Figure 3.5 shows that the pairwise difference was approximately one point on a 7-point scale for both the physical and mental items, for a difference of about 5.5 points overall.

3.5 SUPPLEMENTAL ANALYSES

Since we observed that captions written with suggestions were *shorter* than those written without suggestions, we conducted supplemental analysis to explore potential explanations for this result.

Since the analyses in this section were conceptualized after seeing the data,

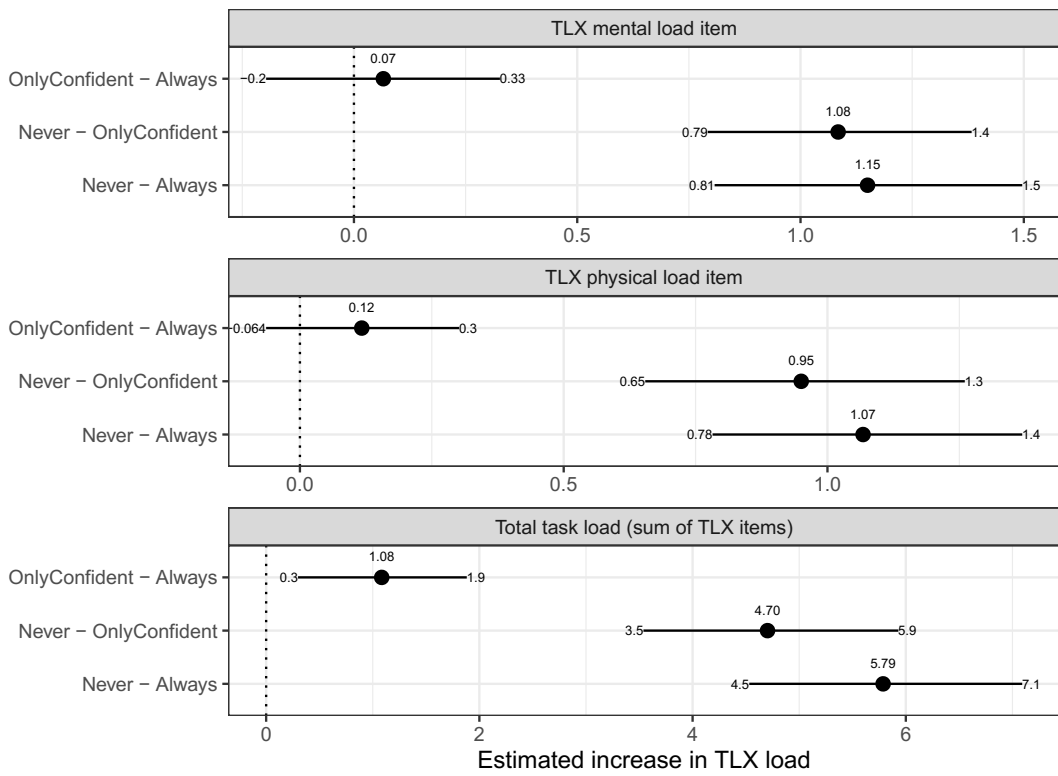


Figure 3.5: Bootstrap estimates of pairwise differences in task load (overall, physical, and mental)




image	P	U	%	corrected text
	11	11	50	people are on sand with five kites in the air as a man in a red shirt and two children hold kites
	7	3	70	a family of three on the beach flying kites together
	7	9	44	an old brown train pulling away from a small train station by a baby blue building
	5	3	62	a train pulling into a quaint train station
	11	12	48	a man in a black shirt with the number eight and grey pants swinging a baseball bat with many players in the background
	11	5	69	a baseball player with the number eight jersey has just hit the ball with the bat

Table 3.1: Examples of captions with varying percentages of predictable words (%). P = number of predictable words, U = number of unpredictable words. (The captions for each image were sorted by percentage predictable and an example was taken at the first quartile and third quartile for each image.) Image credits: <https://flic.kr/p/GyXLw>, <https://flic.kr/p/fkybX6>, <https://flic.kr/p/6Yxc61>

they should be treated as exploratory.

3.5.1 WHAT TYPES OF WORDS WERE PREDICTABLE?

Table 3.1 gives examples of captions at different levels of predictability. Inspection of examples like those shown suggested that the difference in the fraction of predictable words might express itself in terms of a difference in use of words of different parts of speech. Of particular interest seemed to be nouns and adjectives. Also, since we noticed that descriptions of *color* were sometimes missing in high-predictability image captions, we looked at the number of color adjectives used. Figure 3.6 shows that suggestions may have resulted in fewer adjectives used.

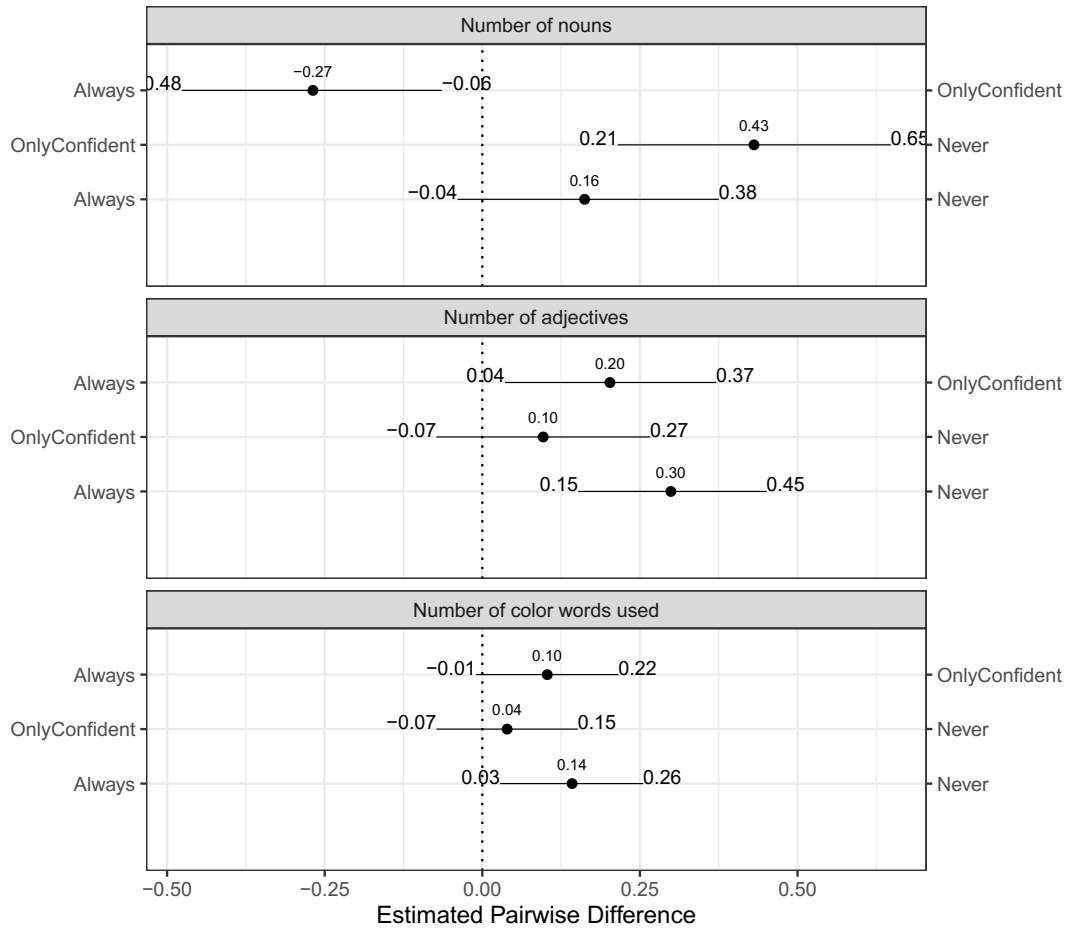


Figure 3.6: Exploratory analysis suggested that the ALWAYS-visible suggestions may lead to the use of fewer adjectives, especially color adjectives, and that ONLYCONFIDENT visibility resulted in fewer nouns.

tives.

3.5.2 WHY WERE CAPTIONS SHORTER WITH SUGGESTIONS?

We conjecture that the suggestions offered may nudge the writer to skip a word. For example, suppose someone is typing “a tennis player is swinging his racket on a green tennis court”. As they are about to type “green,” the system instead suggests “tennis,” encouraging the writer to skip “green.” To describe this scenario we will say that “green” was *skip-nudged*: one of the words suggested at the beginning of a word matched the *following* word.

We analyzed the 436 texts written in the NEVER condition, thus not affected by suggestions, to identify potential skip-nudges. Of these texts, 299 (69%) had at least one skip-nudge. There were a total of 488 skip-nudges, 202 (41%) of which were predictable (i.e., the skip-nudged word was *also* one of the predictions). (If we consider only those suggestions that would be still presented in ONLYCONFIDENT, there are only 228 skip-nudges, of which 120 (53%) are predictable.) The top 10 *predictable* skip-nudged words were: a, wedding, tennis, is, to, tree, at, train, of, baseball; the top 10 *unpredictable* skip-nudged words were: red, white, desktop, is, sits, sitting, computer, on, small, bathroom.

3.6 DISCUSSION

Captions that people wrote when presented with predictive suggestions differed from what they wrote without suggestions. The differences that were most clearly supported by our data are:

1. captions written with suggestions visible were *shorter* and used fewer words that were *unpredictable*, both by a magnitude of about 1 word, than when suggestions were not visible (RQ1, RQ2),
2. captions written with low-confidence suggestions hidden had fewer *predictable* words than those written with suggestions were always shown (RQ3), and
3. predictive suggestions had a positive effect on typing speed overall, but with decreasing benefit for faster typists (RQ4).

Supplemental analysis enables us to conjecture a two-part explanation for these observations. However, further study is needed to determine whether these explanations are accurate and sufficient.

First, suggestions may have sometimes encouraged people to *skip* a word that they would have entered. Since our analysis found that both predictable and unpredictable words could be skip-nudged at similar rates, this encouragement would lead to reduced numbers of both unpredictable and predictable words, resulting in shorter captions overall.

Second, perhaps people would have entered an unpredictable word but the appearance of a prediction caused them to *substitute* a predictable word instead (see, e.g., the caption of Figure 3.1). This substitution would increase the number of predictable words and reduce the number of unpredictable words by the same amount, so length would be unaffected.

Together, skipping and substitution imply that the number of unpredictable words would be reduced, which could account for the first observed difference.

Confidence thresholding reduced the number of times that predictable words were suggested, thus reducing the likelihood of substitution. This account could explain the difference in predictable word count between the two conditions where suggestions were shown.

Our speed findings agree with the AAC literature (surveyed in [Trnka et al. \(2009\)](#)) that predictions often improve communication rate but with substantial individual differences ([Koester & Levine, 1994](#)).

Writers overall preferred conditions where suggestions were always available (as indicated by lower task load and explicit preference rankings). However, the finding that captions entered using suggestions tended to be shorter suggests that minimizing physical effort does not fully account for the differences in word choice that we observed. If participants were simply minimizing their

physical effort, the captions entered with NEVER-visible suggestions would have been shortest, since that condition requires participants to type each character. Other participants typed shorter captions for the same images in conditions where suggestions were available, which indicates that an easier-to-enter utterance was available and acceptable. This finding underscores that the *content* of the suggestions influences text content.

3.6.1 LIMITATIONS

Several limitations of our study lead us to urge caution against overgeneralizing its results: we do not claim that commercially deployed predictive systems have the kind and degree of content effects that we found in our study. However, we conjecture that they do already influence content and that this influence will grow as prediction generation and interaction technology improves. We urge follow-up study of deployed systems to evaluate these content effects.

EXPERIMENTER DEMAND EFFECTS Even though the instructions and recruitment never mentioned predictions (or synonyms such as suggestions or recommendations), the design of this experiment was vulnerable to experimenter demand effects in other ways. For example, the opening survey asked about participants' ordinary use of the suggestion bar, the consent form indicated the

purpose of the research, and the suggestions constituted the main and salient difference between experiment blocks, which indicates to participants that their use is interesting to the researcher (Zizzo, 2010). Moreover, if the participant did not use any suggestions whatsoever, even completely unambiguous completions of a word, during a practice transcription task in which relevant suggestions were available, the system encouraged them to repeat the practice round and use the suggestions; this intervention may have created a carry-over demand effect in the captioning tasks. This happened for 48 participants, many of whom reported that they use the suggestion bar on their own phones “often” or “almost always”. So we suspect that participants did not use suggestions during practice rounds for more mundane reasons specific to the transcription task, such as having to switch attention between the text to transcribe, the text written so far, a potentially unfamiliar keyboard, and the suggestions offered.

Our findings are about the *effects* of suggestion use, not the *degree* to which they are used, so the presence of demand effects does not challenge the validity of our conclusions.

GENERALIZATION TO OTHER WRITING TASKS While the task we used was more representative of real-world writing tasks than transcription tasks used in most writing studies, captioning is still not a common task. We would ex-

pect our findings to generalize to other tasks where the main goal is describing concrete things (e.g., video description, reviewing of products and services, or describing real estate). But our findings may not generalize to other types of tasks, such as those involving conceptual exposition or persuasion, or even to writing descriptions longer than a sentence. Our findings may also be influenced by the specific prompt we provided, which asked participants to write captions that were “specific,” “accurate,” and “concise.” Finally, participants wrote as part of a paid task on MTurk; predictive text could have different effects on writing by other groups of people or for different objectives.

GENERALIZATION TO OTHER PREDICTIVE TEXT SYSTEMS The predictive keyboard that we used in our experiments differed from commonly deployed predictive keyboards in two ways that may affect the generalizability of our findings. First, the keyboard did not offer automatic corrections of mistyped words. The lack of corrections may have caused writers to increase their propensity to consider suggestions because entering a word without using completion suggestions incurs the greater cost of potentially having to backspace and correct a typo. (On the other hand, writers may have also needed to pay more attention to the text that they have just entered, rather than looking at suggestions, which would decrease their propensity to consider suggestions.) Second, our in-

terface did not allow participants to edit past words without backspacing over every character in between, so writers may have typed more carefully.

The suggestion generation system may also affect generalizability, since its suggestions were very strongly adapted to the domain of image captioning. As such, our findings could be viewed as a peek into the future: as predictive text systems gain access to more contextual data (e.g., Google’s Smart Compose, [Chen et al., 2019](#) uses context from the writer and current email thread), they will likely be able to make predictions that are even more strongly adapted to the task (and also to the writer) than ours were.

EXPERIENCE WITH SYSTEM Participants wrote only four captions (plus one practice) with each system. Writers may behave differently after more exposure to a predictive system; if that exposure leads them to trust the system more, the effects of the system on the content of their writing may be larger than what our short study observed.

3.6.2 CONCLUSION

Predictive text systems help many people write more efficiently, but by their nature these systems only make certain content efficient to enter. Our study found that writers are sensitive to these differences: when presented with predictive

text suggestions, people wrote shorter and more predictable language. In short, predictive text suggestions—even when presented as single words—are taken as suggestions of what to write.

ONLINE APPENDIX

Data and analysis code is available at <https://osf.io/w7zpa/>.

4

Effects of Suggestion Length on Writing

Content

This chapter contains content based on a paper presented at UIST 2016 ([Arnold et al., 2016](#)). The pronouns “we”, “our”, and “us” in this chapter refer to the authors of that paper.

This chapter presents the design and evaluation of a system that extends the contemporary “suggestion bar” design of mobile keyboards to present incrementally acceptable phrases.

The contributions of this chapter include:

- an extension of the mobile keyboard suggestion bar to offer phrases,
- a restaurant reviewing task for studying long open-ended text composition,
- behavioral and subjective evidence that phrase suggestions shape the resulting composition to a greater extent than single words, which are treated as predictions,

A system capable of suggesting multi-word phrases while someone is writing could supply ideas about content and phrasing and allow those ideas to be inserted efficiently. Meanwhile, statistical language modeling has provided various approaches to predicting phrases. This section introduces a simple extension to the familiar mobile keyboard suggestion interface that presents phrase suggestions that can be accepted by a repeated-tap gesture. In a composition study using this system, phrases were interpreted as suggestions that affected the content of what participants wrote more than conventional single-word suggestions, which were interpreted as predictions.

OPPORTUNITY

Most mobile keyboards include a *suggestion bar* that offers word completion, correction, and even prediction. While the suggestion bar plays a central role

in commercial products used daily by billions of people, it has received limited attention from academic research (Bi et al., 2014; Quinn & Zhai, 2016).

DESIGN GOALS

In designing our phrase prediction interface, we sought a design that would not interfere or conflict with the existing word prediction interface (e.g., by requiring writers to look in different parts of the interface for word vs phrase predictions) and use minimal extra screen real estate. We also sought a design where a writer who only wanted part of a phrase would be able to express that directly, rather than having to accept too much and delete.

Existing interaction techniques cannot fluently merge word and phrase prediction. For example, after typing “The rest o” the stock iPhone keyboard offers the two-word suggestion *of the* in addition to the single-word suggestion *of* to address the problem of accepting part of a multi-word suggestion. And Smart Compose and similar systems use a completely separate visual presentation and interaction technique to present and accept phrase suggestions.

STUDY

How do people use phrase predictions? How do phrase predictions affect what people write?

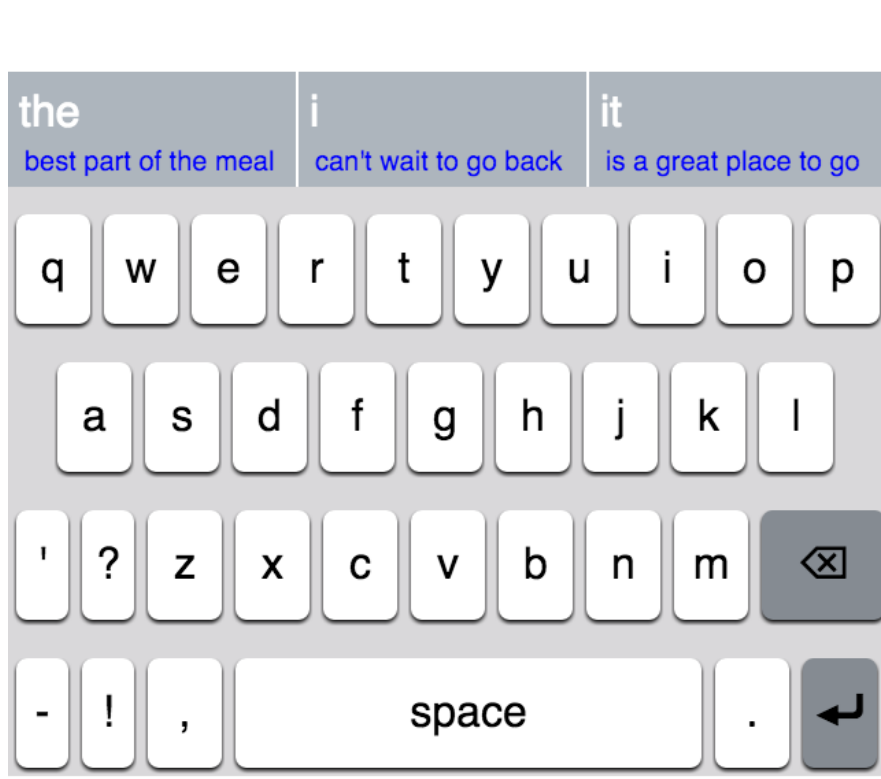


Figure 4.1: We augment the standard mobile keyboard suggestion bar (offering buttons containing individual words such as *The*, *I*, and *it*) by previewing the coming words in small blue text. A user inserts words by tapping repeatedly on the button, one tap per word. The suggestion advances after each button tap, e.g., if the rightmost button is quintuple-tapped, the text *it is a great place* is inserted, the word *to* is suggested, and the preview text advances to become *go for a romantic*.

To answer these questions, we conducted a within-subjects study comparing a simple phrase suggestion interface (Figure 4.1) with a nearly identical system in which only a single word was suggested in each of the three suggestion boxes. Participants were given the task of writing reviews for restaurants of their choice.

4.1 SYSTEM

Our system generates word predictions in a manner that closely matches existing predictive text systems. Given the text written so far, our system predicts three likely next words and shows them in equal-sized suggestion buttons, with the most likely suggestion in the center. If the user starts typing a word, the suggestions are instead completions of that word, but the behavior is otherwise the same.* Tapping a word inserts it with automatic spacing.

4.1.1 PHRASE PREVIEW DESIGN

We extend the next-word predictions into phrase suggestions by showing *phrase previews* within the existing suggestion bar. For each single-word suggestion, the system predicts the most likely 5-word continuation phrase that begins with that word and shows as much of the continuation phrase as fits (typically all

*We did not use autocorrect for this study.

five words on our devices, for a total of six visible words per slot) below the suggestion in smaller blue text (Figure 4.1).

Tapping repeatedly on that suggestion slot inserts that phrase, one word per tap. After each tap the system also shows suggestions in the two other prediction slots, each also including a word plus a continuation phrase.

This design fulfills our design goals. The phrase preview technique for presenting predictions only shows additional information, rather than changing the behavior of word predictions, so the basic tap-to-accept interaction is unchanged from word prediction. (The phrases generated may differ from those that would have been offered in a word-by-word prediction, since they are generated all at once.) Since the phrase previews are presented immediately below the corresponding word, reading them is as natural as reading the next line in an article. And since the suggestion bars on existing keyboards already have extra vertical space (perhaps in order to be easier to tap), we could add phrase predictions without taking up any additional screen real estate (unlike Smart Compose and similar approaches, which require room in the composition area to show the predicted phrase).

4.1.2 PHRASE SUGGESTION GENERATION

The system generates word predictions using an n-gram language model, then extends each word into a phrase prediction using beam search. The system first finds the three most likely next words, then for each word uses a beam search to find the 5-word phrase that is most likely to follow it. Beam search is commonly used in machine translation to find high-quality translations (Green et al., 2014; Rush et al., 2013).

We used KenLM (Heafield et al., 2013) for language model queries, which uses Kneser-Ney smoothing (Kneser & Ney, 1995), and we pruned n-grams that occur less than two times in the corpus. We mark the start-of-sentence token with additional flags indicating if the sentence begins a paragraph or document.

To increase the ability of the system to make useful long suggestions, we focus on a single domain—for this study, we choose restaurant reviews. This choice anticipates by perhaps only a few years the ability of mobile devices to run powerful language models (e.g., via model compression (Prabhavalkar et al., 2016; Geras et al., 2016)) such as contextual neural language models (Ghosh et al., 2016; Kannan et al., 2016; Chen et al., 2019) that can leverage a user’s complete activity context, such as what kind of artifact they are currently writing and to what audience, plus their location history, prior communications, and

other information. A remote server provides suggestions over WiFi; in practice, predictions in our study were shown in less than 100ms.

We built a word-level 5-gram model from the 213,670 restaurant reviews in the Yelp Academic Dataset.[†]

For simplicity of both the experiment interface and backend processing, we restricted the allowed set of characters to lowercase ASCII letters and a restricted set of punctuation (see Figure 4.1). This restriction allowed our experimental keyboard to only need a single layer. In our experiments we instructed participants to disregard capitalization.

4.2 STUDY

We wanted to learn how people use phrase predictions and how these predictions affect the content of what they write. We needed an open-ended writing task so that predictions could potentially affect content (or not). The task also needed to use a domain with enough training data to generate good phrase predictions. We chose restaurant reviewing as a task that fulfilled these goals. People often write reviews on mobile devices, and the task presents many opportunities for people to accept phrases that were not exactly what they might have written on their own but were still perfectly acceptable.

[†]https://www.yelp.com/dataset_challenge

4.2.1 DESIGN AND PROCEDURE

We compared two conditions for prediction length:

Phrase: phrase previews are shown in blue text beneath the single-word suggestions (as in Figure 4.1)

Word: identical behavior to Phrase except phrase previews are hidden

The only difference between the two conditions is whether or not the phrase preview is shown; identical one-word suggestions are shown in both conditions, and repeated taps on the same slot insert the same text that would have been inserted in the Phrase condition.

We used a within-subjects design: we asked participants to write four restaurant reviews, two for each condition (condition ordering was counter-balanced). To familiarize themselves with the keyboard and suggestion mechanism, participants first practiced with both conditions (order randomized). Then before writing reviews, participants wrote down names of four restaurants that they had visited recently. The system then guided them to complete each review in sequence (order randomized), alternating conditions between reviews. (This pre-commitment mechanism ensured that participants did not select restaurants based on, for example, the types of suggestions offered.) We instructed partic-

ipants to write reviews that were at least 70 words in length, and displayed a word counter. We offered a reward for high-quality reviews.

Twenty students (undergraduate and graduate) participated in a lab study lasting 45–60 minutes for monetary compensation. We used 5th-generation iPod Touch devices, which have a 4-inch 1136×640 display.

4.3 RESULTS

We report both behavioral data from system logs as well as subjective data from surveys done both after each review and at the conclusion of the session. All statistical analyses are mixed-effects models, with participant as a random effect and condition (Phrase or Word) as a fixed effect. Unless otherwise noted, we combine the logs of each participant’s two trials for each condition. We exclude from analysis 19 reviews where more than 95% of the review was written using suggestions, leaving 61 reviews from 16 participants. We only report on whole-word suggestions, i.e., those suggestions offered when the participant had not yet begun typing a word.

4.3.1 BEHAVIORAL MEASURES

Participants accepted more whole-word predictions in the Phrase condition ($F(1,15)=37.5, p < .0001$): 45% of words[‡] in Phrase condition compositions had been inserted by prediction, compared with 28% of words in Word condition reviews. This effect has two parts: (1) participants typed out a word when they could have used a suggestion more often in the Word condition (44% of times when a suggestion matching the final word chosen was offered) than in the Phrase condition (28%) ($F(1,15)=19.1, p < .001$), suggesting that participants paid more attention to suggestions in the Phrase condition, and (2) reviews written in the Phrase condition contain more words that had been offered as suggestions at the time they were entered: 63%, compared to 51% in the Word condition ($F(1,15)=42.1, p < .0001$). So showing phrases shaped the content that participants wrote more than showing the same suggestions one word at a time.

In both of our interfaces, repeated taps in the same suggestion slot insert successive words of a phrase. In the Phrase condition, where the participant saw a preview of upcoming words, participants accepted two suggestions in a row 1312 times, of which 85% were consecutively in the same slot, i.e., part of the

[‡]Here, a “word” is a contiguous sequence of non-space characters.

same phrase. In contrast, in the Word condition, of the 776 times that participants accepted two suggestions in a row, 56% were consecutively in the same slot ($F(1,15.3)=20.2$, $p < 0.001$; one participant had no consecutive suggestion acceptances in either condition). As expected, the average length of phrases accepted (defined as consecutive whole-word suggestion acceptances in the same slot) was longer in the Phrase condition (mean 2.8 words) than the Word condition (1.5 words; $F(1,15)=15.6$, $p = .0013$); 14% of phrase acceptances were the full 6 words initially shown.

We compute the error rate by dividing the number of backspace taps (each deleting a single character) by the total number of taps. We did not observe a significant difference between conditions (25% in Phrase, 19% in Word, $F(1,15)=3.2$, n.s.). Our keyboard did not support any assistive gestures for correction, such as tap-and-hold to delete whole words, which we suspect would reduce the difference between conditions.

We did not observe a significant difference in overall typing rate between the two conditions (20.0 wpm[§] for Phrase, 20.9 for Word, $F(1,15)=0.69$, n.s.). On the one hand, participants were able to insert a phrase faster when they could see the preview (for same-slot transition times, Phrase mean = 0.8 s, Word

[§]Here, a “word” is five consecutive characters, including spaces, the definition more common in text-entry studies.

	Transition time (s)	
	Phrase	Word
consecutive suggestions	1.0	1.3
... same slot	0.8	1.1
... diff slot	2.2	1.4
before suggestion	1.2	0.9
after suggestion	1.8	1.3

Table 4.1: Time (secs) between successive interactions involving suggestions. When phrases were shown, participants accepted consecutive suggestions in the same slot more quickly, showing that many participants successfully used a multi-tap gesture to enter a phrase. However, they delayed more before and after using a suggestion, so the overall text entry rate did not increase.

mean 1.1 s); overall, 24% of all suggestion-to-suggestion sequences in the Phrase condition took less than 300 ms, compared with 0.3% in the Word condition.

But on the other hand, participants spent more time before starting to accept a suggestion (Phrase mean 1.2 s, Word mean 0.9 s) and after finishing accepting a suggestion (Phrase mean 1.8 s, Word mean 1.3 s). See Table 4.1 for timing details.

Analyzing the two trials for each condition separately, we do not find any main effect of trial number on rate of suggestion usage ($F(1,41.9)=3.87$, n.s.) or error rate ($F(1,43.1)=2.01$, n.s.). Interaction of condition and trial number was also not significant for either analysis ($F(1,41.75)=.002$ for usage, $F(1,42.7)=.002$ for error rate, n.s.).

4.3.2 SUBJECTIVE MEASURES

Participants reported that suggestions helped them think of *how to say what they wanted to say* more in the Phrase condition (1=*strongly disagree*, 5=*strongly agree*, mean 2.8) than the Word condition (mean 2.1; $F(1,15)=6.4$, $p = .02$). Participants also rated whether suggestions helped them think of *what to say*; ratings were marginally higher in the Phrase condition (mean 3.0, vs mean 2.3 for Word; $F(1,15)=3.8$, $p = .07$). In a cumulative survey, they more often reported that Phrase suggestions gave them ideas. (Phrase mean 3, Word mean 2.2, $t(19)=2.3$, $p = .03$.)

Overall preference was split nearly evenly: 11 participants preferred the Word keyboard and 9 preferred the Phrase keyboard. Participants liked that the phrase keyboard gave them ideas of both what to say and how to say it, sometimes in ways that were better than what they had in mind or better matched the style of the genre (in this case, restaurant reviews). But they disliked that the phrases suggested were often generic or trite, and felt that the phrases forced them into certain less creative ways of thinking. In contrast, the “Word” suggestions helped people write in “my own words” and be more “thoughtful.” They also liked that text entry felt faster and easier in the Phrase condition, but some commented about spending a lot of time reading phrase suggestions

(though there was no significant difference in ratings on “I felt like I spent a lot of time reading the suggestions”: Phrase mean 3.0, Word mean 2.6, $F(1,15)=1.5$, n.s.). Participants commented that both Word and Phrase suggestions were often distracting, confirming the findings of a prior study (Quinn & Zhai, 2016).

4.4 DISCUSSION

Phrase suggestions affected both the *process* and *product* of composition. The short delays between successive suggestion insertions indicate that participants successfully inserted phrases as units, rather than re-evaluating the suggestions for each successive word. (Consistent with a previous study on word suggestions, the additional cost to evaluate suggestions counteracted the speed benefit of inserting a suggestion, so the overall speed did not improve (Quinn & Zhai, 2016).) The phrase suggestions also shaped the final review: when shown phrases, participants accepted a greater number of suggestions, and those suggestions were more often repeated taps in the same suggestion slot. Since the single word shown in a suggestion slot was identical in the Word condition (i.e., it was also part of a phrase), this evidence indicates that people made different choices about what to say when predictions were presented as phrases.

4.4.1 LIMITATIONS

Although our open-ended composition task is arguably much more representative of mobile typing in the wild than the transcription task used in almost all text-entry studies, our task was still a lab study with corresponding limitations: memories may have faded, and participants are not as invested in the resulting review as if it were to be actually published.

Since the phrase preview interface was novel, participants may have overused it in effort to explore how it worked or perhaps to please the experimenter. Therefore the effect sizes reported here are likely to be overestimates of the true effect sizes of phrase predictions.

A prominent limitation of multi-word suggestion systems is that high-quality suggestions are hard to generate unless something is known about what the user intends to write about. In this present study we partially avoided this limitation by focusing on the domain of restaurant reviews, where a large corpus of domain-specific text is available. Our approach is directly applicable to other task- or domain-specific entry tasks, such as other kinds of reviews (products, movies, etc.), customer relations management, or product support. In communications such as email, the recipients, subject, message thread, and prior sent messages can all serve as context with which to generate suggestions. In

a general context across applications (where the name of the currently active application can be a context feature), our interface could easily adapt by only showing phrase suggestions when there is sufficient context to merit suggestions. Gathering appropriate context across users, in an efficient and private manner, is left as an interesting open challenge. Since much of this data resides within specific applications, the keyboard software would need to interact heavily with applications so as to offer better suggestions.

5

Effects of Sentiment Bias of Phrase

Suggestions

This chapter contains content based on a paper presented at Graphics Interface 2018 ([Arnold et al., 2018](#)), including work done in collaboration with Krysta Chauncey. The pronouns “we”, “our”, and “us” in this chapter refer to the authors of that paper.

This chapter presents a study on whether text suggestions affect high-level writing decisions, such as the sentiment that a writer chooses to convey about

their personal experience at a restaurant. Motivated by the observation that the phrases offered tended to be positive, it presents a study that manipulated the sentiment of phrase suggestions and found that writers wrote more positive content when presented with positively slanted suggestions. Since online review corpora tend to be biased towards positive content, this finding has the concerning implication that biases in datasets not only cause biases in algorithm behavior (a fact well established by recent research) but also cause biases in the content produced by people interacting with those algorithms.

The contributions of this chapter include:

- evidence that phrase predictions exhibit a bias towards positive sentiment when trained on a restaurant review dataset, and
- evidence that the sentiment valence of predictions offered during writing affects the sentiment valence of what people write when presented with these predictions
- Evidence that naive text prediction systems for review-writing domains can produce suggestions that are biased towards positive sentiment
- A method for shaping the sentiment of contextual suggestions generated during real-time typing.
- A study demonstrating that writers generate restaurant reviews with more positive content when presented with positive suggestions.

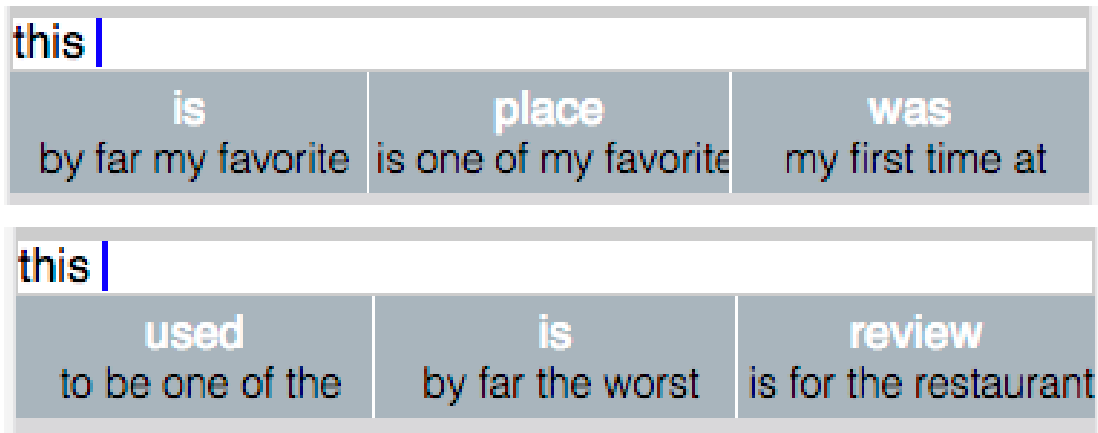


Figure 5.1: Biases in training data can cause intelligent systems to offer assistance that is unintentionally biased. In our experiment, we manipulate a predictive text system to exhibit a bias towards positive sentiment (top screenshot) or negative sentiment (bottom screenshot). Although the stated purpose of showing these predictions is efficiency, do biases in the prediction content affect the sentiment of what people write?

Prior research has demonstrated that intelligent systems make biased decisions because they are trained on biased data. As people increasingly leverage intelligent systems to enhance their productivity and creativity, could system biases affect what people create? We demonstrate that in at least one domain (writing restaurant reviews), biased system behavior leads to biased human behavior: People presented with phrasal text entry shortcuts that were skewed positive wrote more positive reviews than they did when presented with negative-skewed shortcuts. This result contributes to the pertinent debate about the role of intelligent systems in our society.

Motivated by the interest in understanding and mitigating bias in intelligent systems, we asked two questions: (1) can intelligent systems that support cre-

ative tasks exhibit unintentional biases in the content of the support that they offer, and (2) do these biases affect what people produce using these systems?

We investigated these questions in the context of a commonly used intelligent interactive system, namely, predictive text on mobile devices. We focused on the task of writing restaurant reviews, an everyday task that people often do on mobile devices. Writing can exhibit many kinds of biases, such as race, gender, or culture; we focused on one type of bias: the valence of the sentiment that is expressed in a review, i.e., is the review favorable or unfavorable toward the restaurant? In addressing the first question, we found that available review corpora are biased towards positive content and that a standard text-generation algorithm generated suggestions that humans perceived as biased positive, even after rebalancing the dataset using star ratings. Then, by manipulating the sentiment of suggestions in a controlled experiment, we found that positively biased suggestions bias people to write content that is more positive. Taken together, these two studies suggest a *chain of bias*: biases in training data cause biases in system behavior, which in turn cause biased human-generated products.

The sentiment of the suggestions could affect the sentiment of the result through several different mechanisms. First, the suggested phrases may provide specific positive (or negative) information that the writer uses, either because it

is easy to enter via accepting the suggestion verbatim, or because the suggestion reminds them of an aspect to discuss even if they do not use the exact words suggested. Second, the suggestions may “prime” the writer as they implicitly function as examples of what sentiment of writing is expected: if all examples are positive, a writer may feel like a negative phrase is out of place; in contrast, suggestions with a diversity of sentiments may convey that a variety of sentiments is expected.

5.1 PHRASE SUGGESTIONS ARE BIASED POSITIVE

Predictive text suggestions can vary in the sentiment that they express. For example, consider a writer composing a restaurant review. After the writer types ‘The’, the system could choose to suggest ‘best’ (positive sentiment valence), ‘worst’ (negative valence), or ‘food’ (neutral). The strength of sentiment expressed in suggestions can be even stronger when the system can suggest phrases (e.g., “staff are super friendly,” “prices are very reasonable,” or “only good thing about”).

Are contemporary predictive text systems equally likely to offer positive suggestions as negative suggestions, or do they exhibit biases in sentiment? We first study the biases present in existing approaches for suggestion generation in

Table 5.1: Examples of readily available datasets of reviews with star ratings: number of reviews (N) and statistics of their star ratings (1–5, 5 highest). Review datasets are biased positive. Datasets: Yelp (restaurants only, from <https://www.yelp.com/dataset>) Amazon product reviews (from <http://snap.stanford.edu/data/amazon/productGraph/>), TripAdvisor (from <http://sifaka.cs.uiuc.edu/~wang296/Data/index.html>)

Dataset	N	Median	Mean \pm stdev
Yelp	196,858	4	3.59 \pm 1.17
Amazon	82,456,877	5	4.16 \pm 1.26
TripAdvisor	1,621,956	4	3.93 \pm 1.21

writing.

5.1.1 ONLINE REVIEW CORPORA HAVE BIASED SENTIMENT

Suppose a practitioner wants to develop an intelligent system for supporting review-writing. This system likely needs a set of existing reviews so that it can learn typical characteristics of reviews. A reasonable strategy would be to search the Internet for datasets of user-generated reviews and pick one with an appropriate domain and size. Unfortunately, this reasonable strategy is likely to give the practitioner a biased dataset. Table 5.1 shows the distribution of star ratings in several corpora of online reviews. In none of these readily available review datasets is the mean or median star rating below a 3 on a 1–5 scale. Though these are far from the only collections of review texts on which a practitioner may train a text generation system, their bias is clear and large, even when only considering star rating as a coarse proxy for sentiment.

5.1.2 SYSTEMS TRAINED ON REVIEWS MAKE BIASED SUGGESTIONS

Predictive text systems use statistical models of language to estimate the probability that a given word or phrase will follow in a particular context, then show the phrases with the highest probability. Consider a corpus composed of several different groups, for example, positive, neutral, and negative restaurant reviews. If the training data contain more examples of one group than the others, then the predictions will favor a relevant word or phrase from the more common group over an equally relevant word or phrase from a less common group simply because that phrase occurs more frequently.*

In this section, we demonstrate that phrase prediction systems do present biased suggestions to writers.

SUGGESTION GENERATION SYSTEM We used a phrase suggestion generation system similar to that of Chapter 4. To try to correct for the overabundance of positive reviews that we noted above, we constructed a training corpus by randomly under-sampling reviews from the Yelp restaurant review corpus so that there was an equal number of reviews with each of the five available star ratings. We also held out a 10% sample of reviews for validation experiments

*Another cause of stereotyped predictions is more subtle: even if the probabilities could be corrected for the differences in base rates between groups, the *accuracy* of the model will be lower in underrepresented groups because of the reduced amount of training data.

described below. Despite the smaller training set size, the relevance of the suggestions seemed qualitatively sufficient for our purposes. We will refer to this text generation system as `BALANCED` in this paper, though as the results below show, the system’s output is not actually balanced.

RE-TYPING PARADIGM We simulated re-typing existing reviews and generated suggestions using `BALANCED`, then compared the suggestions with the text that was originally typed. We constructed samples to evaluate in the following way. First, we subsampled held-out reviews evenly from the five star rating classes. For each review, we picked a word boundary such that at least five words remained before the end of the sentence. We then simulated re-typing the review until that word boundary and extracted the set of suggestions that the system would present. We picked one of the three suggestions uniformly at random and presented it in comparison with the five words that actually followed in the original review. If the suggestion happened to match the original text exactly, we drew a new location.

Writing process theories posit that writers pause longer at sentence boundaries than other word boundaries because they are planning their next sentence (Torrance, 2015). While doing so, they often read what they have already written (Torrance et al., 2016). Thus, a suggestion displayed at the beginning of

a sentence has a larger potential impact on the writer’s plan for the sentence that follows. Since the retyping process described above would otherwise sample sentence beginnings rarely, we oversampled sentence beginnings by deciding uniformly whether to pick the beginning of a sentence or a different word.

SUGGESTIONS TREND POSITIVE, ESPECIALLY PHRASES We compared the sentiment of the text of the suggestions offered by the system with the text that was actually written in the original review. To do so, we presented pairs of texts (with their original context of five prior words) to MTurk workers and asked which they perceived as more positive. Workers could also choose a “neither” option if the sentiment valence was indistinguishable. The interface randomized whether the suggestion or the original text was shown first. We showed each pair to three different workers and took the majority vote (breaking three-way ties in favor of “neither”). We coded the result as an ordinal variable taking values -1 (original word/phrase selected as more positive), 0 (neither), or 1 (suggested phrase selected as more positive).

We collected rankings for 500 suggestions. A binomial test showed that at sentence beginnings, the suggestions were picked as more positive significantly more often (164 out of 263 total suggestions, $p < .0001$) than the original review text, across all star ratings of original reviews. For mid-sentence sugges-

tions, the difference was less pronounced, but in comparisons where there was a winner (rather than “neither”), the generated text was more positive than the original text significantly more often (112 out of 184 decided comparisons, $p = .003$).

Since the original star rating of the review should predict how positive the original text is, we expected it to influence how its sentiment compares with the generated text. If the generated text were always consistently like that of a 5-star review, we would expect a strong influence of star rating on the binary comparison: the original text would always be more positive than text from 1-star reviews, but compared with text from 5-star reviews it would be a toss-up. On the other hand, if the generated text tended to follow the sentiment of the original text (because the context of the suggestion leads in a particular direction), the star rating would have a relatively minor effect on the binary comparison.

Figure 5.2 shows that generated phrases were rated on average more positive than the original text, but less so for higher star ratings and for mid-sentence suggestions. To quantify this effect, we fit two separate ordinal logistic models predicting the more positive option, one for beginning-of-sentence suggestions and one for mid-sentence suggestions, with the star rating of the original review

as an ordinal fixed effect. For beginning-of-sentence suggestions, we observed a strong effect of review star rating: the likelihood ratio was 31.7, $p < .0001$. For mid-sentence suggestions, we observed a much weaker effect (likelihood ratio of 9.79, $p=0.044$).[†]

These findings indicate that generated phrases at the beginning of sentences were more strongly positive than what people wrote without those suggestions. In the middle of a sentence, suggestion sentiment stayed closer to the sentiment in the original text but still leaned positive.

These findings were in the context of *phrase* suggestions. To determine if single-word suggestions are also perceived as biased, we repeated this entire process of suggestion sampling (with a different random seed) and annotation to generate another 500 suggestion pairs, except that this time we limited both the original and generated text to a single word. We found that single-word suggestions did not have a clear difference in sentiment compared with the corresponding original words: in most beginning-of-sentence pairs, participants indicated that neither text was more positive; mid-sentence votes were split evenly among the three options.

[†]Since the effect size for ordinal effects in ordinal regressions is unintuitive, we repeated the analysis with the original star rating as a continuous effect. For beginning-of-sentence suggestions, the log-odds was 0.43 per star (95% CI 0.304–0.691), and for mid-sentence suggestions, the log-odds was 0.18 (CI 0.019–0.349).

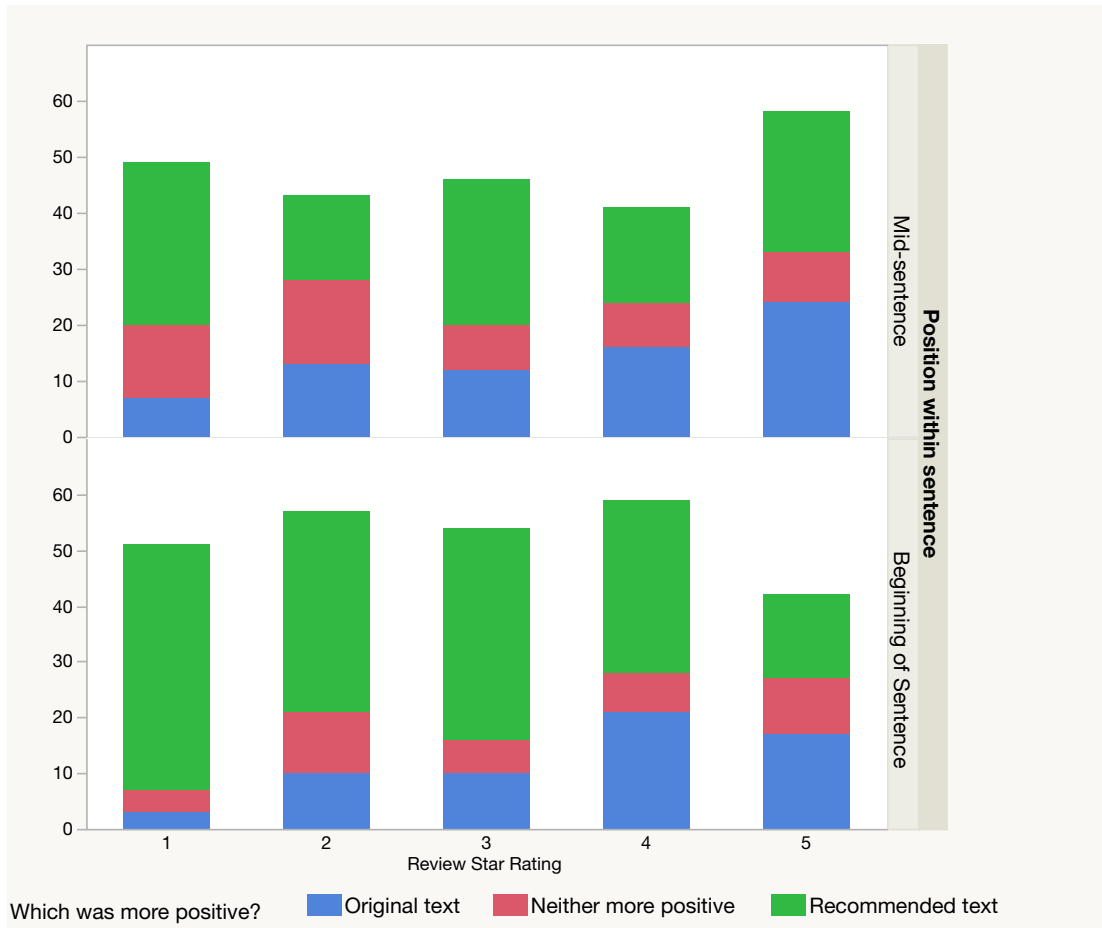


Figure 5.2: Stacked histograms of the number of times that phrases generated by the BALANCED system were chosen by crowd workers as more positive than naturally occurring text, in pairwise comparisons. The generated text was usually perceived as more positive, reflecting a bias towards positive content in the training data. The effect was strongest for suggestions at the beginning of a sentence (lower panel).

5.1.3 DISCUSSION

The above findings indicate that phrase suggestions reflect the positive bias observed in the training data: generated phrases are usually perceived as more positive than the text actually written in prior reviews.

It is not clear why a system trained on a dataset with equal counts in each star rating would be biased positive. If positive reviews tended to be longer than negative reviews, that could explain the bias, but in fact negative reviews tend to be longer (146.6 words for 1-star reviews vs 115.4 words for 5-star reviews). A possible explanation is that even 1-star reviews often have some characteristics of positive content, such as phrases like “a friend recommended this to me.” Also, some negative reviews begin with positive aspects of the product or experience before their complaints.

5.2 EFFECTS OF SUGGESTION SENTIMENT ON WRITING CONTENT

We have found that biased training data results in biased suggestions; we now study whether biased suggestions lead to biased writing output. In this section, we describe an experiment in which we present writers with suggestions manipulated to vary in sentiment valence, and measure the effects that these suggestions have on the sentiment of the resulting writing. We first introduce

the interactive system that we build on, then discuss the conceptual design of the experiment, and the modifications needed to manipulate the behavior of the system. We then describe the details of the experimental task, measures, and procedure.

5.2.1 INTERACTIVE SYSTEM FOR TEXT SUGGESTION

We use the same phrase-shortcut keyboard as described and studied in Chapter 4.

5.2.2 EXPERIMENT DESIGN

We hypothesize that when writers are given positive suggestions, their writing will include more positive content than when they are given negative suggestions. To test this hypothesis, we needed to manipulate the sentiment of suggestions that a system provides to participants and measure the sentiment valence of their writing. It is not possible to offer suggestions that are uniformly “positive” or “negative”; in the middle of a glowingly positive sentence, a negative suggestion would be seen as irrelevant; in a purely factual sentence, it may not be possible to offer text that has any perceived sentiment at all. Instead, we *skew* the distribution of sentiment of the generated text: in the condition we call SKEW-POS, we increase the likelihood that the system generates a pos-

itive suggestion instead of a negative one, and in SKEW-NEG, we increase the corresponding likelihood of negative suggestions. As the results of Study 1 suggest, the differences between these two systems will be most apparent at the beginning of a sentence. The system must manipulate the sentiment of the suggestions without being irrelevant, ungrammatical, or unreasonably extreme.

Since we expected that participants may take some time to react to changes in suggestions, we chose to keep the suggestion strategy constant for each writing artifact (in this case, a restaurant review), changing only between artifacts. Since we expected individual differences in behavior and artifact, we used a within-subjects design and mixed-effects analysis. We had participants write about both positive and negative experiences, for a total of 2 (prior sentiment valence) x 2 (suggestion valence) = 4 trials for each participant. In our analyses, we fit random intercepts for each participant and include block and prior sentiment as ordinal control variables, unless otherwise noted.

5.2.3 MANIPULATING SENTIMENT OF SUGGESTIONS

Controlling the sentiment of text generation is an active area of research (Lipton et al., 2015; Radford et al., 2017; Hu et al., 2017). However, we were not yet able to get these new techniques to run at interactive speed on commodity

hardware. On the other hand, training on only reviews of a certain star rating unduly compromised the relevance of the language model. So for the present experiment, we used a simple “reranking” approach in which a contemporary language generation system generates a large number of candidate phrases, then a second process re-orders these candidates to pick the most positive (for SKEW-POS) or negative (for SKEW-NEG).

The system generates the set of candidate phrases using a modification of the beam-search process used in the system of study 1. We used the same base language model as for that study, BALANCED, based on subsampling the Yelp review corpus so that it had an equal number of reviews with each star rating (1 through 5). However, we modified the beam search process so that it would generate a range of possible phrases. To generate candidate phrases, the system first identified the 20 most likely next words under the language model, then for each word generated the 20 most likely phrases starting with that word using beam search with width 20, resulting in 400 candidate phrases.

We then used a classifier to select a set of phrases from among the candidate set according to the desired sentiment skew. We trained a Naive Bayes classifier to predict review star rating using bigrams as features.[‡] For each candidate

[‡]For short snippets, such as the phrases that we evaluate, such a simple approach can outperform more complex models (Wang & Manning, 2012).

phrase, the system computed the probability that each phrase came from a 5-star review (for SKEW-POS), or a 1-star review (for SKEW-NEG). A simplistic approach would be to then suggest the phrases with the most extreme positive (or negative) sentiment. However, the phrases with the most extreme sentiment were sometimes ungrammatical, awkward, or simply irrelevant. We found that pilot study participants tended to ignore suggestions that they perceived as irrelevant, so we added a likelihood constraint to the generation process: the system first picked the three phrases with highest likelihood under BALANCED that start with distinct first words, then iteratively replaced each phrase with one of the candidate phrases that was more positive (or more negative), so long as (1) the set of suggestions would still start with distinct first words and (2) the contextual likelihood of the replacement phrase was no less than β times the likelihood of the phrase it replaced. We chose $\beta = e^{-1} \approx 0.36$ (one nat) because the resulting phrases tended to be grammatically acceptable and still skewed in sentiment. Although likelihood can be a poor proxy for grammatical acceptability (Lau et al., 2016), the approach seemed reasonably successful in pilot studies. Figure 5.1 shows an example of the output of this approach. We parallelized language model evaluations in the beam search to increase speed. The overall latency from tapping a key to seeing a suggestion was typically

less than 100ms with commodity hardware, which is similar to the latency of deployed smartphone keyboards.

5.2.4 VALIDATION OF THE SENTIMENT MANIPULATION APPROACH

We validated our sentiment manipulation approach using the sentiment analysis functionality of Google Cloud NLP. Using a methodology identical to that used in Study 1, we generated 500 sample contexts. We took the last six words of each context and appended each of four phrases: the text that had followed that context in the original review (TRUE), and the suggestions generated by each of the three systems we studied: the baseline BALANCED system (used in Study 1), the SKEW-POS system, and the SKEW-NEG system. We then submitted each of the $500 \times 4 = 2000$ resulting phrases to the Google Cloud NLP sentiment analysis service and recorded the sentiment score (which ranges from -1.0 for maximally negative to +1.0 for maximally positive).

Figure 5.3 shows that the sentiment manipulation approach successfully created a difference in sentiment valence between the SKEW-POS and SKEW-NEG systems. An all-pairs Tukey's HSD test confirms that the difference in sentiment means between SKEW-POS, SKEW-NEG, BALANCED, and TRUE was significant at the $p=0.05$ level for all pairs except for SKEW-POS and BALANCED. Note,

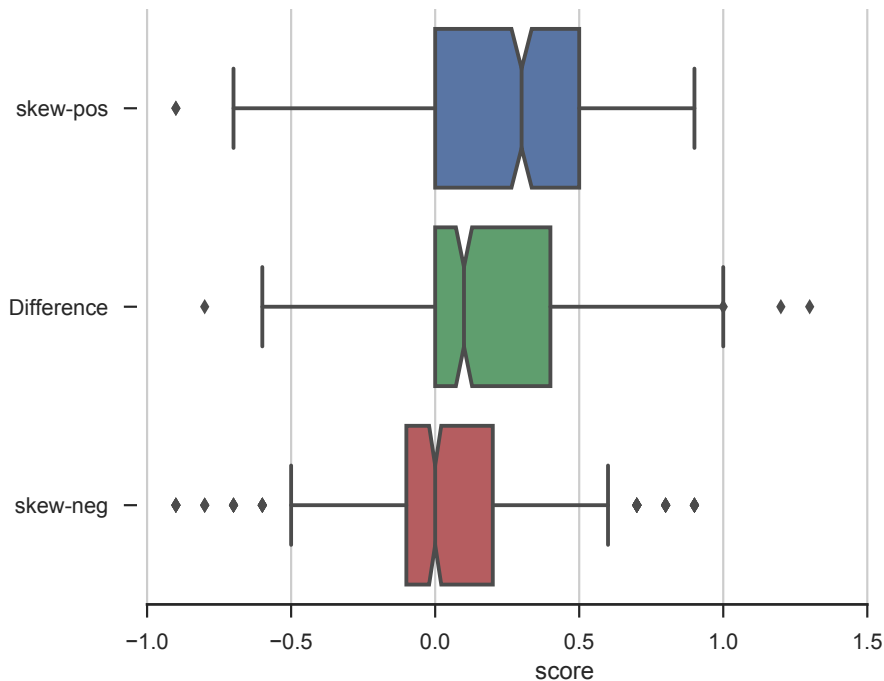


Figure 5.3: Box plots of sentiment scores computed by the Google Cloud NLP sentiment API for suggestions generated by SKEW-POS (top; mean score 0.23) and SKEW-NEG (bottom; mean score 0.07), for the same 500 phrase contexts drawn from existing reviews; middle plot shows the difference (SKEW-POS - SKEW-NEG) for each context. Notches show confidence intervals for the median. The sentiment manipulation method successfully creates a difference in sentiment valence. Neither system was always positive or always negative, however.

though, that all means are above 0.0, indicating that in no condition are the suggestions more negative than positive on average.

5.2.5 TASK

We asked participants to write restaurant reviews using touchscreen keyboards that offered word and phrase shortcut entry functions. We modeled our task

design on the design used in Chapter 4. The reviews that we asked participants to write were about specific experiences at actual restaurants that they committed to write about before the experiment began. Our instructions motivated accuracy and quality using quotes from the reviewing guidelines on Yelp and promised a bonus for high-quality reviews. We also encouraged participants to avoid typos, since the contextual suggestion system relied on accurate spelling of the context words.

5.2.6 PROCEDURE

We implemented a simplified touchscreen keyboard as a mobile web application using React, using WebSockets to connect to a server that generated suggestions and managed the experiment state. After tapping on a suggested word, the web application opportunistically advanced the suggestion to the next word in the corresponding phrase, so multiple words from a phrase could be entered quickly, without waiting for a response from the server. The keyboard was designed to mimic a standard mobile phone keyboard in look and feel but was simplified to be modeless. As such, it only supported lowercase letters and a selected set of punctuation. For simplicity and to focus attention on the prediction (rather than correction) aspect of the typing interface, the keyboard did not support

autocorrect.

We recruited 38 participants from a university participant pool to participate in our web-based study. Participants were compensated with a gift card for \$12 for an estimated duration of 45–70 minutes. Study procedures were approved by the university IRB. Participants were instructed to use their own mobile devices, so screen size and device performance varied between participants.

At the start of the experiment, we asked participants to list four establishments that they would like to write about, two above-average experiences and two below-average experiences. For each one, we also asked for their overall opinion about the establishment in terms of a star rating. We chose this procedure so that participants would be strongly encouraged to report faithfully about their experiences with accuracy and detail, rather than making up an imaginary review to play with the suggestions or get through the experiment quickly.

Participants then completed a tutorial to familiarize themselves with the keyboard and suggestions. Participants were instructed to write a sentence about the interior of a residence they know well, as if writing a description for a site like Airbnb. During the tutorial, the keyboard presented suggestions using the same algorithms as the main experiment, but with training data drawn from

Airbnb postings from 16 cities in the US and without sentiment manipulation.

The system then instructed participants to write about the four establishments they listed, one at a time. To ensure that each participant experienced all four combinations of writer sentiment and suggestion sentiment, the order of establishments was randomized in a specific way. The system chose whether to have the participant write about the above-average experiences or below-average experiences first, then it shuffled the restaurants within each category. The order of conditions was also randomized: the first condition is randomly chosen as one of SKEW-POS or SKEW-NEG, then subsequent conditions alternated.

The framing used to describe the suggestions is important to the validity of our experiment. A term such as “suggestion” or “suggestion” implies that the content of the suggestions reflect what the experimenter desires. If participants simply viewed the suggestions as telling them what they should write, then the effect of suggestions on writing content would be trivial. Even with more neutral language such as “words or phrases offered by the keyboard,” participants may still guess the intent of the researchers. Instead, we needed to actively focus participants on a different aspect of the suggestions. Since the selling point of these systems is usually efficiency, we chose to emphasize that aspect. We did this in two ways: first, we referred to the suggestions as “shortcuts.” Second, we

added a feedback mechanism to help participants gauge whether the suggestions would help them write more efficiently. Since the suggestions offered by our system were generally much more relevant to the task than the domain-general suggestions that participants may have been accustomed to from their experience with predictive text keyboards, we added a feedback element to the interface: whenever a participant typed a character that caused the current word to be a prefix of one of the words that are currently being presented as suggestions, the interface highlighted that fact: that word remained in its corresponding suggestion slot (even if the suggestions generated after entering new character would have otherwise caused it to be reordered), the suggestion flashed, and the prefix was highlighted.

After each of the four trials, participants completed a short survey asking what star rating they would now give to the establishment, and how the sentiment of the “shortcuts” compared with the experience they were writing about.

5.2.7 MEASURING SENTIMENT OF WRITING

For evaluating the sentiment of complete writings, we chose the unit of analysis to be the sentence: smaller units would be tedious and potentially ambiguous (e.g., for “the only good thing about this place,” what is the sentiment of

“good”?); larger units such as paragraphs or complete writings are overly coarse. Since each sentence can contain both positive and negative content (e.g., “the service was atrocious but the food made up for it”) or neither (e.g., “each entree comes with two sides”) (Berrios et al., 2015), we asked annotators to rate, for each sentence, how much positive content it had and how much negative content it had. Pilot rating studies showed that raters could only reliably distinguish three levels of sentiment content, so we had annotators rate the positive content of each sentence on a scale of 0 (no positive content), 1 (vaguely positive content), or 2 (clearly positive content), and the negative content of each sentence on a corresponding scale. (Pang & Lee (2005) reports similar limitations of annotation granularity.) We computed the mean across raters for each sentence. We used the `sent_tokenize` routine from NLTK (Bird et al., 2009) to split reviews into sentences. We summarized the sentiment of a review by two quantities: the mean amount of positive sentiment and mean amount of negative sentiment, taken across sentences.

5.2.8 ADJUSTMENTS TO DATA

Despite instructions to avoid typos, most reviews included one or two clear typos. (Recall that the keyboard did not employ autocorrect.) Since interpre-

tation of typos can be difficult and sometimes ambiguous for annotators, we added a typo correction step before sentiment annotation. The typo correction was done by one of the authors, blind to all metadata, with the assistance of the Microsoft Word contextual spelling checker. In almost all cases the intended text was unambiguous; the few ambiguous cases were left as-is. We did not exclude participants based on excessive typos.

Despite our instructions to list two positive and two negative experiences to write about, and separate labeled areas to enter positive and negative experiences, some participants did not list any experience with less than 4 stars out of 5. So while we used the participant's labeling of experiences as positive and negative to determine the trial and condition order (as described in the Procedure section above), and had planned to use that label as a control variable in our analysis, because of this mismatch we decided to use their star rating for analysis instead. Nevertheless, we have reason to believe that the counterbalancing was successful: a regression of star rating on condition has an r^2 of less than 0.01.

5.3 RESULTS

5.3.1 EFFECTS ON WRITING CONTENT

We recruited annotators from MTurk to perform the task described in subsection 5.2.7 to measure the sentiment of complete writing artifacts. Each annotator rated one or more batches of four writing samples, randomly chosen from among the entire set of writing samples. Each of the $38 \times 4 = 152$ writing samples was rated by three annotators. Krippendorff’s alpha agreement was 0.84 on positive sentiment and 0.85 on negative sentiment, indicating acceptable agreement.

We observed a significantly larger amount of positive sentiment in the reviews written with positive-skewed suggestions (SKEW-POS condition $M=1.22$, $\sigma=0.67$) compared with negative-skewed suggestions (SKEW-NEG condition $M=0.98$, $\sigma=0.61$) ($F_{1,106.8} = 12.3$, $p=.0007$). For comparison, the magnitude of the effect of switching from SKEW-NEG to SKEW-POS is 77% of the estimated magnitude of having given one additional star.

We did not observe a significant difference between conditions in the amount of negative sentiment in the reviews written ($F_{1,107.4} = 0.85$, n.s.). Since the validation showed that the SKEW-NEG condition was only relatively negative,

not negative in an absolute sense, this result is not surprising.

Compared to the star rating that participants gave their experiences when listing them at the start of the experiment, participants gave an average of 0.27 more stars to their experience after writing about it in the SKEW-POS condition, and 0.1 more stars after the SKEW-NEG condition. However, a mixed ANOVA did not show a statistically significant effect of condition ($F_{1,98.13} = 1.55$, n.s.).

These results reflect analyses that included all participants. We observed that a few participants typed their reviews almost exclusively by tapping suggestions. Though this may have been honest behavior, it seems more likely that these participants attempted to complete the experiment with minimal effort or misinterpreted the instructions. We re-ran the analyses with various exclusion criteria, such as excluding participants who tapped suggestions more than 90% of the time in a single trial. However, none of these exclusions changed the overall results, so we chose not to exclude any data in the final analysis.

5.3.2 PARTICIPANT EXPERIENCE

Participants often remarked on whether the “shortcuts” were accurate or if they saved them time or effort. Many comments were favorable: “Helped me save a lot of time.” However, some participants noted that the benefit of the

shortcuts came at the cost of distraction: “It was very pleasant as I did not have to write out all the words. But I think I didn’t save much time using it, as I was constantly only looking whether the word I was wanting to write appeared in the box.” and “It was nice to have them, but not worth the trouble.”

Several participants commented about a mismatch between the sentiment of the suggestions and what they were trying to write, and one participant said, “At times I felt like the predictions were guiding my writing.”

Some participants noted that the suggestions tended to be generic: “the responses lacked specificity and were difficult to incorporate”; “They definitely make my writing more generic, but I don’t mind that.” Since the suggestions were chosen to be those that were the most likely, it is unsurprising that they should be perceived as generic. Future work could investigate how to offer suggestions that help writers be more specific.

An error caused our Likert-scale surveys not to be administered, so we quantified participant experiences with the suggestions by coding the open-ended responses that most participants gave after each trial. For each response, blind to condition, one of the authors rated whether it included any favorable remarks about the suggestions (or “predictions” or “recommendations”) (on a scale of 0=none, 1=possible, 2=clear positive) and separately whether it included any

unfavorable remarks (same 0–2 scale). For this rating process, only comments about the content of the suggestions were considered; other kinds of comments (e.g., responsiveness, lack of autocorrect, or the word count target) were ignored. We excluded the five participants who gave no intelligible comments for one or more trials, leaving 33 participants. Each participant used each condition twice, so we summed the participant’s ratings of favorable comments and of unfavorable comments for each condition. This procedure resulted in four numbers for each participant: SKEW-POS-favorable, SKEW-POS-unfavorable, SKEW-NEG-favorable, and SKEW-NEG-unfavorable.

A Wilcoxon signed-rank test showed that participants left more favorable comments after writing in the SKEW-POS condition than in the SKEW-NEG condition ($Z=95$, $p=.0008$). The average difference in ratings between favorable comments about SKEW-POS and favorable comments about SKEW-NEG (SKEW-POS-favorable - SKEW-NEG-favorable) was 0.34. However, the difference in negative comments was much less pronounced for unfavorable comments: participants left marginally more unfavorable comments after writing in SKEW-NEG than in SKEW-POS (mean of SKEW-POS-unfavorable - SKEW-POS-unfavorable was -0.14). The difference fails to reach statistical significance after accounting for multiple comparisons ($Z=159$, $p=0.029$).

5.4 DISCUSSION

Our results supported our primary hypothesis: writers given positive suggestions included more positive content. This finding suggests that positively skewed suggestions cause writers to intensify their positive sentiment: if they would have written a mildly positive sentence without suggestions, they instead write a strongly positive sentence when given positive suggestions.

We did not find a corresponding effect of negatively skewed suggestions, but this could be due to the very bias we are studying: since the suggestion system we were manipulating was biased positive, our manipulations in the SKEW-NEG condition successfully reduced the positive bias, but the system still tended to present positive suggestions more often than negative ones. Reaching a definitive conclusion about the nature of truly negative suggestions requires additional study with a more sophisticated text generation approach.

We find it particularly concerning that participants gave more favorable comments to the SKEW-POS system. While some participants were able to critically reflect on the system's behavior and realize that it could be biasing their writing, many participants seemed to prefer to write with the system that biased their writing to be more positive.

5.4.1 LIMITATIONS AND THREATS TO VALIDITY

Since this experiment had participants write in artificial circumstances, generalizations to natural conditions must be drawn carefully. The strongest threat to the external validity of our findings is that participants behaved in a way that would “please the experimenter,” a kind of participant response bias (Dell et al., 2012). Although we used instructions and system features to attempt to focus participants’ attention on using the suggestions only for efficiency, some participants may have felt pressured to use the suggestions more overall. For example, some participants may have felt that the experimenters wanted them to write in a way that allowed them to use the suggested phrases more.

Two aspects of the experiment design may have given participants clues that sentiment valence was important to us. First, we asked for experiences that differed in sentiment valence (though we used the language “above-average experience” and “below-average experience”). Second, we asked for the perceived sentiment of the suggestions after each trial (though among other survey questions). Comments in the closing survey suggest that at least one participant realized that sentiment was interesting to us. Future work should confirm if the results we present still hold in an experimental setting where sentiment is less salient.

Code to replicate these experiments is available at <https://github.com/kcarnold/sentiment-slant-gi18/>.

6

Learning to Manipulate Content

This chapter contains content based on a paper presented at IJCNLP 2017 (Arnold et al., 2017). The pronouns “we”, “our”, and “us” in this chapter refer to the authors of that paper.*

Platforms like Facebook, Twitter, and Reddit struggle to cultivate healthy speech by their participants. Since predictive text can sway content (as the results of Chapters 3 and 5 show), it is natural to wonder if platforms might be able to design predictions that encourage the kind of speech that they desire.

*Data and code are available at <https://github.com/kcarnold/counterfactual-lm>.

The ability to meaningfully modulate predictions may also be a helpful creative tool for writers. For example, predictive suggestions tend to nudge writers towards words that are more common; would it be possible to nudge writers towards words that are *less* common—without the predictions seeming irrelevant?

At the same time, the potential for intentional content manipulation poses ethical challenges and would require transparency and accountability on the part of these platforms. In particular, it would be concerning if platforms were able to manipulate the content that contributors wrote without the knowledge or intent of those contributors.

Using predictive text to manipulate written content faces challenges that include:

- **Relevance:** Although the system of Chapter 5 was able to manipulate the sentiment of the content, only the positive slant—the bias that the training data already had—was perceived as relevant by writers. How can we manipulate suggestion content towards characteristics that may be a minority in the training set, without overly compromising perceived relevance?
- **Training data:** while text to train high-quality language models is abundant (Radford et al., 2018), a much smaller corpus of “desirable” data may be available; how can this be smaller corpus be leveraged?
- **Sensitivity:** some users may be particularly sensitive to any attempts to

manipulate, and such efforts might backfire.

- **Adaptation:** given the substantial individual differences between writers regarding the use and influence of predictive text, it would be desirable to be able to adapt the approach to individual users, or at least detect the users for which the intervention is likely to be successful
- **Transparency:** what information can a platform provide about the way that it is attempting to manipulate content? Can its approach be summarized succinctly and accurately?
- **Accountability:** What evidence could platforms provide that could convince an auditor that any content manipulation being attempted was in line with what the platform reports and not being used for, e.g., increasing engagement?

This chapter addresses the first two challenges—relevance and training data.

APPROACH Instead of predicting *which words a writer will type*, is it possible to predict *which suggestions a writer will accept*? Such modeling would enable us to make suggestions that have different characteristics, making intentional trade-offs, if necessary, between suggestion acceptability and writing characteristics. Unlike the language modeling task of next-word prediction, a model of suggestion acceptance must depend on the suggestions actually offered, not just the linguistic context as in language modeling.

Directly estimating a model of suggestion acceptance behavior would require an unrealistically large amount of data. To circumvent this issue, we propose adapting an existing language model trained on next-word prediction. We can collect data with the targeted goal of learning the difference between prediction and suggestion.

Counterfactual learning allows us to evaluate and ultimately learn models that differ from those that were deployed to collect the data, so we can deploy a single model and improve it based on the data collected (Swaminathan & Joachims, 2015). Intuitively, if we deploy a stochastic suggestion system and observe the actions it takes, the *propensity* of the system to take that action in that context, and what feedback that action gets, we could improve the system by making it more likely to suggest the phrases that got desirable feedback.

CONTRIBUTIONS

- We show how to use counterfactual learning for goal-directed training of language models from interaction data.
- We show in simulation that a simple language model can be adapted to manipulate the content of writing, given a simplified model of suggestion acceptance.
- We demonstrate how a simple discriminative language model can be trained with offline interaction data to generate more frequently accepted

suggestions in unseen contexts.

6.1 APPLYING COUNTERFACTUAL LEARNING TO SUGGESTION GENERATION

Let h denote a suggestion system, characterized by $h(y|x)$, the probability that h will suggest the word or phrase y when in context x (e.g., words typed so far).[†] We consider deploying h in an interactive interface such as the phrase suggestion interface of Figure 4.1. Let δ denote a reward that a system receives from that interaction; in our case, the number of words accepted.[‡] We define the overall quality of a suggestion system by its expected reward $E[\delta]$ over all contexts.

Suppose we deploy a *reference model*[§] h_0 and log a dataset

$$\mathcal{D} = \{(x_1, y_1, \delta_1, p_1), \dots, (x_n, y_n, \delta_n, p_n)\}$$

of contexts (words typed so far), actions (phrases suggested), rewards, and

[†]Our notation follows Swaminathan & Joachims (2015) but uses “reward” rather than “loss.” Since $h(y|x)$ has the form of a contextual language model, we will refer to it as a “model.”

[‡]Our setting admits alternative rewards, such as the speed that a sentence was written, or an annotator’s rating of quality.

[§]Some literature calls h_0 a *logging policy*.

propensities respectively, where $p_i \equiv h_0(y_i|x_i)$. Now consider deploying an alternative model h_θ (we will show an example as Eq. 6.1 below). We can obtain an unbiased estimate of the reward that h_θ would incur using importance sampling:

$$\hat{R}(h_\theta) = \frac{1}{n} \sum_{i=1}^n \delta_i h_\theta(y_i|x_i)/p_i.$$

However, the variance of this estimate can be unbounded because the importance weights $h_\theta(y_i|x_i)/p_i$ can be arbitrarily large for small p_i . Like [Ionides \(2008\)](#), we clip the importance weights to a maximum M :

$$\hat{R}^M(h) = \frac{1}{n} \sum_{i=1}^n \delta_i \min \{M, h_\theta(y_i|x_i)/p_i\}.$$

The improved model can be learned by optimizing

$$\hat{h}_\theta = \operatorname{argmax}_h \hat{R}^M(h).$$

This optimization problem is convex and differentiable; we solve it with BFGS.

6.1.1 ADAPTING A PREDICTIVE MODEL TO GENERATE SUGGESTIONS

We now demonstrate how counterfactual learning can be used to evaluate and optimize the acceptability of suggestions made by a language model. We start with a traditional predictive language model h_0 of any form, trained by maximum likelihood on a given corpus. This model can be used for generation: sampling from the model yields words or phrases that match the frequency statistics of the corpus. However, rather than offering samples from h_0 directly, many interactive language generation systems instead sample from $p(w_i) \propto h_0(w_i)^{1/\tau}$, where τ is a “temperature” parameter; $\tau = 1$ corresponds to sampling based on p_0 (soft-max), while $\tau \rightarrow 0$ corresponds to greedy maximum likelihood generation (hard-max), which many deployed keyboards use (Quinn & Zhai, 2016). The effect is to skew the sampling distribution towards more probable words.

To expand the range of suggestion strategies that the system is able to take, we add features that can emphasize various characteristics of the generated text, then use counterfactual learning to assign weights to those features that result in suggestions that writers accept.

We consider locally-normalized log-linear language models of the form

$$h_\theta(y|x) = \prod_{i=1}^{|y|} \frac{\exp \theta \cdot f(w_i|c, w_{[:i-1]})}{\sum_{w'} \exp \theta \cdot f(w'|c, w_{[:i-1]})}, \quad (6.1)$$

where y is a phrase and $f(w_i|x, w_{[i-1]})$ is a feature vector for a candidate word w_i given its context x . ($w_{[i-1]}$ is a shorthand for $\{w_1, w_2, \dots, w_{i-1}\}$.) Models of this form are commonly used in sequence labeling tasks, where they are called Max-Entropy Markov Models (McCallum et al., 2000).

The feature vector can include a variety of features. By changing feature weights, we obtain language models with different characteristics. To illustrate, we describe a model with three features below. The first feature (LM) is the log likelihood under a base 5-gram language model $p_0(w_i|c, w_{[i-1]})$ trained on the Yelp Dataset[¶] with Kneser-Ney smoothing (Heafield et al., 2013). The second and third features “bonus” two characteristics of w_i : **long-word** is a binary indicator of long word length (we arbitrarily choose ≥ 6 letters), and **POS** is a one-hot encoding of its most common POS tag. Table 6.1 shows examples of phrases generated with different feature weights.

Note that if we set the weight vector to zero except for a weight of $1/\tau$ on **LM**, the model reduces to sampling from the base language model with “temperature” τ .

REFERENCE MODEL h_0 . In counterfactual estimation, we deploy one reference model h_0 to learn another \hat{h} —but weight truncation will prevent \hat{h} from deviat-

[¶]https://www.yelp.com/dataset_challenge; we used only restaurant reviews

LM weight = 1, all other weights zero:
 i didn't see a sign for; i am a huge sucker for
LM weight = 1, long-word bonus = 1.0:
 another restaurant especially during sporting events
LM weight = 1, POS adjective bonus = 3.0:
 great local bar and traditional southern

Table 6.1: Example phrases generated by the log-linear language model under various parameters. The context is the beginning-of-review token; all text is lowercased. Some phrases are not fully grammatical, but writers can accept a prefix.

ing too far from h_0 . So h_0 must offer a broad range of types of suggestions, but they must be of sufficiently quality that some are ultimately chosen. To balance these concerns, we use temperature sampling with a temperature $\tau = 0.5$):

$$\frac{p_0(w_i|c, w_{[:i-1]})^{1/\tau}}{\sum_w p_0(w|c, w_{[:i-1]})^{1/\tau}}.$$

We use our reference model h_0 to generate 6-word suggestions one word at a time, so p_i is the product of the conditional probabilities of each word.

6.2 SIMULATION EXPERIMENT

While large datasets of pre-written text are readily available, data on *suggestion acceptance* is scarce outside of industry. Moreover, even if we had suggestion acceptance data from a deployed system, we need an empirical demonstration that applying the methodology described above would correctly estimate the effect of deploying a different system in that same context.

So our first experiments use a *simulated* writer that accepts suggestions according to a word-level *desirability model*. This setting will allow us to evaluate the effect of different suggestion policies in the same simulated writing situation.

We begin by describing our simulated writer, then report the methodology and results of applying counterfactual language model optimization for this simulated writer.

DESIRABILITY MODEL Suppose a writer is using the interface in Figure 4.1, which displays three suggestions at a time. At each time step i they can choose to accept one of the three suggestions $\{s_j^i\}_{j=1}^3$, or reject the suggestions by tapping a key. Let $\{p_j^i\}_{j=1}^3$ denote the likelihood of suggestion s_j^i under a predictive model, and let $p_\emptyset^i = 1 - \sum_{j=1}^3 p_j^i$ denote the probability of any other word. Let a_j^i denote the writer’s probability of choosing the corresponding suggestion, and a_\emptyset^i denote the probability of rejecting the suggestions offered. If the writer decided exactly what to write before interacting with the system and used suggestions for optimal efficiency, then a_j^i would equal p_j^i (assuming a perfectly accurate language model p). But suppose the writer finds certain suggestions *desirable*. Let D_j^i give the desirability of a suggestion, e.g., D_j^i could be the number of long words in suggestion s_j^i . We model their behavior by adding the desirabilities to

the log probabilities of each suggestion:

$$a_j^{(i)} = p_j^{(i)} \exp(D_j^{(i)}) / Z^{(i)},$$

and making the corresponding correction to the reject action,

$$a_\emptyset^{(i)} = p_\emptyset^{(i)} / Z^{(i)},$$

where $Z^{(i)} = 1 - \sum_j p_j^{(i)} (1 - \exp(D_j^{(i)}))$. The net effect is to move probability mass from the “reject” action a_\emptyset^i to suggestions that are close enough to what the writer wanted to say but desirable.

EXPERIMENT SETTINGS AND RESULTS. We sample 10% of the reviews in the Yelp Dataset, hold them out from training h_0 , and split them into an equal-sized training set and test set. We randomly sample suggestion locations from the training set. We cut off that phrase and pretend to retype it. We generate three phrases from the reference model h_0 , then allow the simulated author to pick one phrase, subject to their preference as modeled by the desirability model. We learn a customized language model and then evaluate it on an additional 500 sentences from the test set.

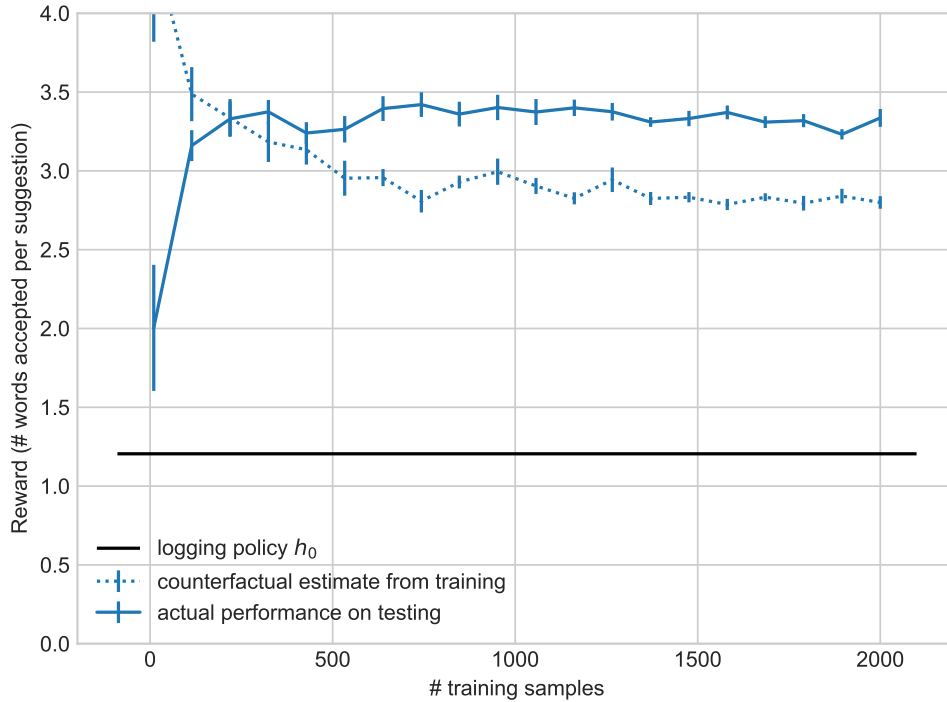


Figure 6.1: We simulated learning a model based on the behavior of a writer who prefers long words, then presented suggestions from that learned model to the simulated writer. The model learned to make desirable predictions by optimizing the counterfactual estimated reward. Regularization causes that estimate to generally be conservative: once the amount of training data is sufficiently large, the reward actually achieved by the model exceeded the estimate. Plot shows mean and standard error of the mean across different randomly selected sets of suggestion acceptances used for training. The cutoff is set at $M = 10$ for all trials.

For an illustrative example, we set the desirability D to the number of long words (≥ 6 characters) in the suggestion, multiplied by 10. Figure 6.1 shows that counterfactual learning quickly finds model parameters that make suggestions that are more likely to be accepted, and the counterfactual estimates are not only useful for learning but also correlate well with the actual improvement. In fact, since weight truncation (controlled by M) acts as regularization, the

counterfactual estimate typically *underestimates* the actual reward.

6.3 COLLECTING ACCEPTANCE DATA FROM HUMAN WRITERS

We then sought to measure the degree to which our method might capture the difference between human acceptance behavior and what a conventional language model would predict. The goal of this experiment was *not* to manipulate the suggestions that writers actually received, but to collect data to evaluate models offline. We first collect data from human writers using a baseline suggestion policy, then use that data to train an improved suggestion policy, which we evaluate on held-out acceptance data.

6.3.1 DATA COLLECTION

We recruited U.S.-based 74 workers through MTurk to write reviews of *Chipotle Mexican Grill* using the interface in Figure 4.1. Based on pilot experiments, Chipotle was chosen as a restaurant that many crowd workers had dined at. Participants could elect to use the interface on either a smartphone or on a personal computer. In the former case, the interaction was natural as it mimicked a standard keyboard. In the latter case, users clicked with their mouses on the screen to simulate taps.

i love this place. the food is good. it's a little expensive, but the food is so much more than that. and i love the people that work there. the burritos are huge and packed with flavor. i got one with chicken and beef and extra quac. it tasted fresh and i couldn't even finish it all as it was huge! i love the location and the atmosphere was great. i will definitely come back to try something different!

i hate spicy food but for some reason i love the flavor of chipotle chiles in any form, so i loooove chipotle. i have been nervous about eating here lately because of the food poisoning scandals but thankfully i have not had any problems! i always order the burrito bowls and the portions are huge! service is just mediocre but not bad and you cant expect too much from a chain restaurant. overall i would give chipotle four stars.

Figure 6.2: Example reviews. A colored background indicates that the word was inserted by accepting a suggestion. Consecutive words with the same color were inserted as part of a phrase.

User feedback was largely positive, and users generally understood the suggestions' intent. The users' engagement with the suggestions varied greatly—some loved the suggestions and their entire review consisted of nearly only words entered with suggestions while others used very few suggestions. Several users reported that the suggestions helped them select words to write down an idea or also gave them ideas of what to write. We did not systematically enforce quality, but informally we find that most reviews written were grammatical and sensible, which indicates that participants evaluated suggestions before taking them. Figure 6.2 shows two examples of reviews that were written.

The dataset contains 74 restaurant reviews typed with phrase suggestions. The mean word count is 69.3, std=25.70. In total, this data comprises 5125 words, along with almost 30k suggestions made (including mid-word).

# accepted		0	1	2	3	4	5	6
count		27,859	1,397	306	130	91	68	107

Table 6.2: Our dataset includes 29,958 suggestions made by the system during typing. Authors accepted at least one word of 2099 suggestions (7%), and at least 2 words in 702 suggestions (2.3%). In total, 3745 out of 5125 words in the corpus were entered using suggestions. These acceptance rates are comparable with those observed in other work.

6.3.2 ACCEPTANCE POLICY ESTIMATION

We used the data we collected to learn an improved suggestion generation policy. The logs we collected included both data from the *suggestion system* about suggestion generation propensities and data from *writers* about which suggestions they accepted. Having both types of data enabled us to apply our counterfactual policy optimization strategy to learn a new policy that would achieve a higher estimated suggestion acceptance rate.

Our measure is the acceptance rate predicted for the new generation policy, evaluated on held-out data. Intuitively, we consider a policy to be successful if it increased the overall likelihood that the system would have generated suggestions that writers responded positively to. Although it is evaluated on held-out data, this quantity is still an *estimate*, since we do not actually deploy this revised policy. As such, the value of this estimate depends on the threshold parameter M , the maximum allowed importance weight.

We learn an improved suggestion policy by the estimated expected reward

(\hat{R}^M). We fix $M = 10$ and evaluate the performance of the learned parameters on held-out data using 5-fold cross-validation.

We compare this policy against an ablated policy that can only adjust the temperature parameter τ . We also compare against the logging policy (which had a fixed temperature, $\tau = .5$).

Although the results of our earlier chapters suggest that there are large individual differences in responses to suggestions, for this experiment we used the same policy for all human writers for the sake of simplicity.

6.3.3 RESULTS

Figure 6.3 shows that while the estimated performance of the new policy does vary with the M used when estimating the expected reward, the relationships are consistent: the fitted policy consistently receives the highest expected reward, followed by the ablated policy that can only adjust the temperature parameter τ , and both outperform the reference policy (with $\tau = .5$).

The fitted model weights (shown in Table 6.3) suggest that the workers seemed to prefer long words and pronouns but eschewed suggestions of punctuation.

	mean	std
base LM	2.04	0.16
is.long	0.92	0.14
PUNCT	-1.16	0.26
ADJ	1.03	0.61
ADP	1.45	0.38
ADV	0.45	0.55
CONJ	0.91	0.26
DET	0.36	0.22
NOUN	0.96	0.14
NUM	0.87	0.27
PRON	1.68	0.20
PRT	0.23	1.00
VERB	0.79	0.32

Table 6.3: Fitted weights for each feature in the log-linear model, averaged across dataset folds

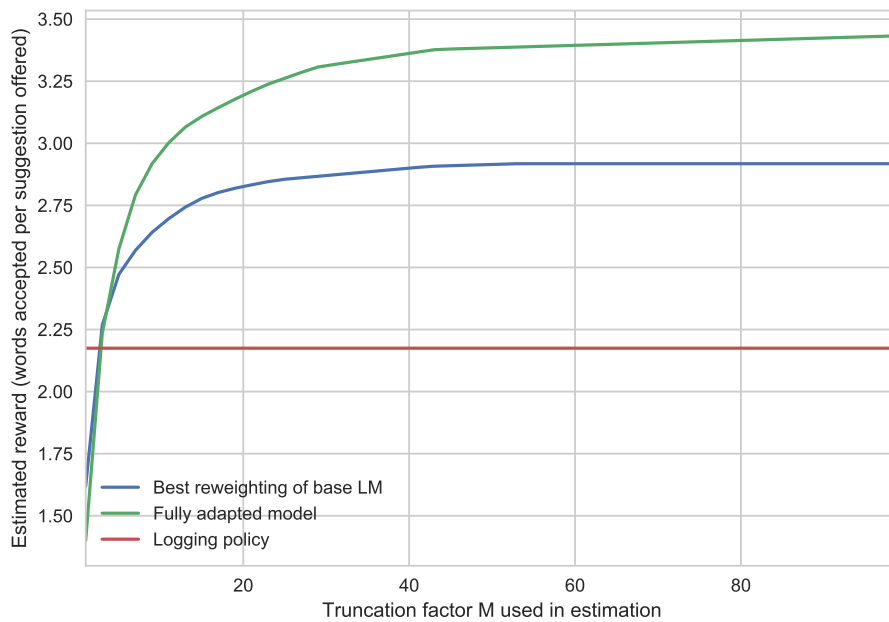


Figure 6.3: Estimates of acceptance rate of the optimized suggestion generation policy, averaged over 5 validation folds, plotted against the importance weight truncation parameter M . The customized model consistently improves expected reward over baselines (reference LM, and the best “temperature” reweighting LM) in held-out data. Although the result is an estimated using weight truncation at M , the improvement holds for all reasonable M .

6.4 DISCUSSION

Our model assumed all writers have the same preferences. Modeling variations between writers, such as in style or vocabulary, could improve performance, as has been done in other domains (e.g., [Chen et al. \(2019\)](#)). Each review in our dataset was written by a different writer, so our dataset could be used to evaluate online personalization approaches.

Our task of crowdsourced reviews of a single restaurant may not be representative of other tasks or populations of users. However, the predictive language model is a replaceable component, and a stronger model that incorporates more context (e.g., [Sordoni et al. \(2015\)](#)) could improve our baselines and extend our approach to other domains.

Future work can improve on the simple discriminative language model presented here to increase grammaticality and relevance, and thus acceptability, of the suggestions that the customized language models generate.

7

Opportunities for Predictive Guidance

The results thus far in this dissertation show that existing designs for predictive text systems exhibit various types of biases—designed and emergent, incidental and intentional. In this chapter, I consider alternative designs for systems that predict text but avoid offering writers specific words to easily enter. Since all of the biases studied in this dissertation take the form of writers accepting specific words that originate from the system, such alternative designs could avoid these biases.

Could alternative designs avoid some of these biases while still supporting

content generation? In this chapter I ask three questions:

1. **What opportunities might exist for technology to support of writing by offering guidance for content creation without the bias inherent in next-word prediction?** I consider opportunities around technology support for *goal-setting*, especially around providing *inspiration* for content goals.
2. **How might such guidance be presented to writers? How might writers respond to these suggestions?** An exploratory user study comparing two suggestion methodologies showed a clear preference for using *questions* as a mechanism for communicating content inspiration.
3. **What advances beyond current technology are necessary to make the proposed interventions technically feasible?** I show both the opportunities and limitations of current technology.

7.1 OPPORTUNITIES FOR CONTENT GUIDANCE

What kinds of assistance could a predictive system offer, other than making certain words easy to enter? Based on studies of educational writing interventions, I argue that interventions that suggest *content goals* are promising, but new technical approaches are needed to scale those interventions in breadth and depth.

Writers often struggle with generating effective structure for their documents. Interventions that provide structural guidance to writers have shown benefits to the quality of the final result (Ferretti et al., 2009; Hui et al., 2018). For example, fourth- and sixth-grade students produced more effective essays when provided with a list of subgoals appropriate for argumentative writing, such as “You need to explain why those reasons are good reasons for your opinion” (Ferretti et al., 2009). Structure-based planning, in which writers are given high-level goals to organize their outlining, also resulted in higher quality texts than planning using unstructured lists or not planning, although the effect was weak and the no-planning condition had lower overall time-on-task (Limpo & Alves, 2018).

These approaches have been operationalized into technological interventions, but past work has been domain-specific and minimally adaptive. For example, the IntroAssist system (Hui et al., 2018) uses checklists paired with annotated examples, both generated by experts, to scaffold writers in an uncommon but high-impact writing task. Questions are commonly used as prompts to scaffold contributions. For example, most of the micro-activities in the CommunityCrit system involved answering an expert-curated question (Mahyar et al., 2018).

Examples of prior writing are often used to help writers produce new texts.

For example, writing teachers use “Mentor Texts” to guide students to understand the structures used in high-quality writing. Also, sentences that could be included in a story helped students with learning disabilities generate better stories (Graves et al., 1994), and researchers have suggested incorporating usage examples from web search into writing support tools (Fourney et al., 2017).

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

7.2 EXPERIENCE DESIGN FOR CONTENT SUGGESTIONS

The purpose of this study is to determine what form a predictive suggestion could take that would (a) provide actionable guidance about a sentence-level topic to discuss and (b) be perceived as relevant by writers.

Like Yang et al. (2019), my methodology for this section relied on simulated system behavior in way that anticipated likely constraints and errors of real systems but did not depend on building a specific system. However, rather than prototyping a complete interaction in detail, I focused this study more narrowly

on comparing two different kinds of prediction *content*, delaying interface considerations.

7.2.1 TASK

The task was writing sentences that would belong in Wikipedia-style encyclopedia articles about subjects that people were familiar with. I chose this task because:

- the Wikipedia community has identified high-quality articles from which structure and content exemplars could be drawn,
- those examples are permissively licensed,
- these high-quality articles often shared some topical structure,
- the task is accessible to many crowd workers, and
- the basic task can readily be repeated for different domains.

The writing tasks needed to be ones that people could meaningfully write based on their own knowledge without seeking external information. I chose three types of topics as ones that participants were likely to have sufficient background knowledge in: films, books, and travel destinations.

7.2.2 CONDITIONS

Prior approaches to providing structural guidance in writing leveraged two main types of interventions: Examples and Expert Guidance.

Although these past interventions generally require expert curation and input, I wondered if designs inspired by them could be used in an automated setting. In particular, I consider designs in which a predictive system could identify *situations in which to present* expert-curated scaffolding content. My main question is, what kind of curated content should be presented?

Based on the categories of past interventions that I identified, I considered two alternatives for the kind of content to present: Snippets and Questions. Suppose a system predicts that a section of past writing would be a relevant example for the writer in their current context. **Snippets** are those short sections presented as-is; **Questions** are those sections presented in the form of *questions that the snippet would answer*.

I expected that Snippets would be useful to writers because they include not just information that writers may be able to adapt to their present task but also structures that they could use to express that information. However, Questions could be useful to writers in a different way: they draw attention to the information needs of the reader.

7.2.3 DESIGN AND PROCEDURE

I recruited 30 participants from MTurk. Participants pre-committed to three items of their own choice, e.g., one film, one book, one travel destination. Each participant was then asked to write 10 sentences for an article on each of those items.

For each sentence, participants were given 10 prompts in a fixed order. The presentation of the prompts varied by condition: was it presented as the original sentence (Snippet) or as an abstracted set of Questions? I also included a NoPrompt condition in which no prompts were given.

For each prompt, the writer was first asked whether the prompt gave them an idea about what to write for their article. If they answered Yes, they were then asked to write a sentence.

7.2.4 CONTENT GENERATION

I started by selecting one or two Wikipedia Featured Articles in each of the categories, since these were judged by the community to be of exceptional quality. (For WikiVoyage, I used “star city” articles, which have similar roles.) I chose a simple approach to extract content: for each sentence in the original article, I attempted to identify a clear question that it answered. I omitted History sec-

tions of Wikivoyage articles and plot summaries in book and film articles since these are much more highly idiosyncratic. (Future work could study ways of supporting writing in these cases.) For most encyclopedic texts, each sentence answered a clear question, so this approach worked well.

I then picked the 10 sentences for which the identified questions would most straightforwardly apply to other articles.

7.2.5 MEASURES

I measured the *relevance* of a suggestion as a binary variable: whether the writer reported that the suggestion gave them an idea about what to write.

I measured the *usefulness* of a relevant suggestion by the amount of text that the participant wrote in response to that suggestion.

Finally, I measured writers' *preference* by asking, at the close of the experiment, which of the systems (described as “bots”) they would prefer to have for writing in the future.

7.2.6 RESULTS

Participants wrote longer texts when given prompts. A mixed ANOVA predicting the block-level mean of log-transformed text length from Presentation and prior confidence showed a main effect of Presentation ($F(2,43.8)=9.43$, $p=.0004$)

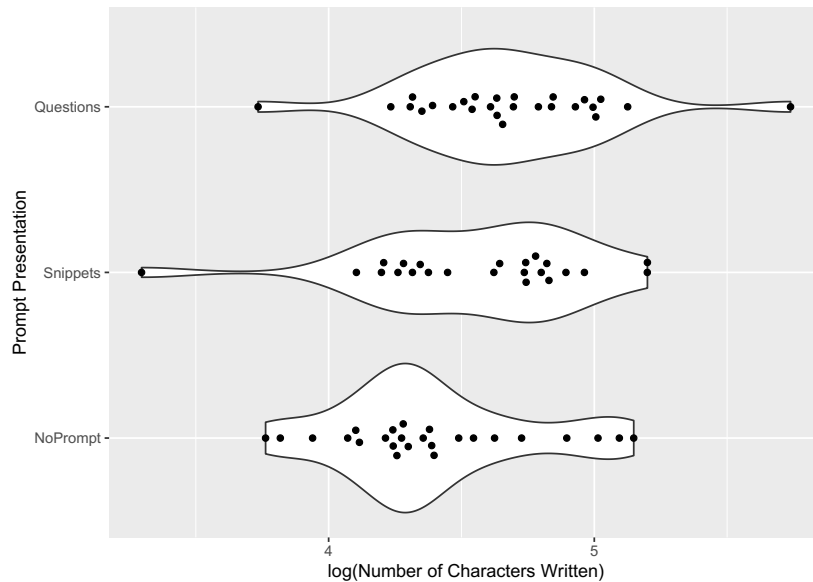


Figure 7.1: Participants wrote longer texts when given prompts (both Questions and Snippets).

but not of prior confidence ($F(1,58.94)=.84$, n.s.). Post-hoc comparisons showed a significant contrast between NoPrompt and the two prompt conditions, with the strongest contrast between Questions and NoPrompt, but the prompt conditions (Questions and Snippets) were not significantly different from each other. In this analysis, both Participant and Task were treated as random effects.

Questions gave usable ideas more often than Snippets Figure 7.2. Likelihood ratio tests in a binomial mixed model predicting number of prompts marked as “relevant” found a significant effect of Presentation ($\chi^2 = 48.99$, $p<.0001$) and category relevance ($\chi^2 = 7.35$, $p=.007$), but no interaction between the two ($\chi^2 = 3.75$, $p=.05$). In this analysis, both Participant and Task were treated as

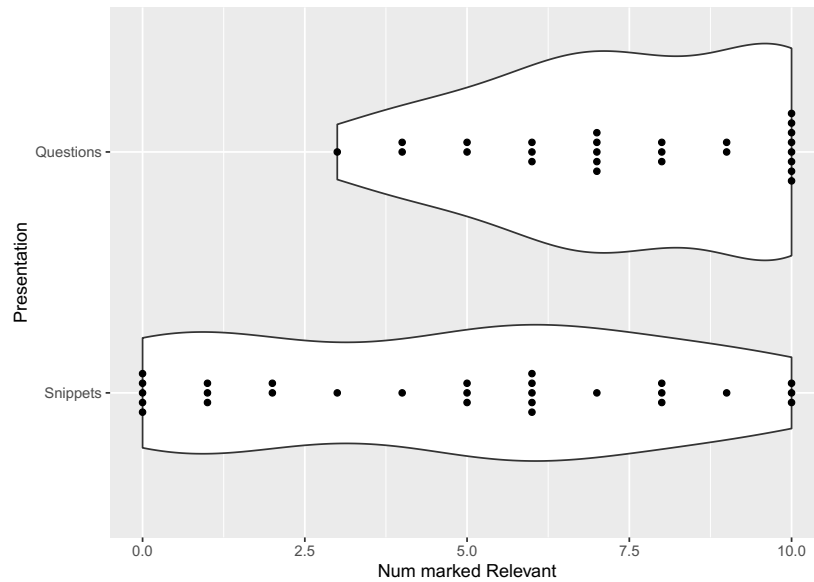


Figure 7.2: Prompts presented as Questions were more often marked as relevant by participants.

random effects.

Writers expressed strong preference for Questions over Snippets presentation (Figure 7.3).

7.2.7 DISCUSSION

Results of the study suggest that suggestive topic prompts can be useful to writers in the sense that they help writers generate more content. Since the topic suggestions were static, not adapted to the specific document that the participant was writing or what they had already discussed, it is unsurprising that many suggestions were deemed irrelevant. However, it is instructive to note

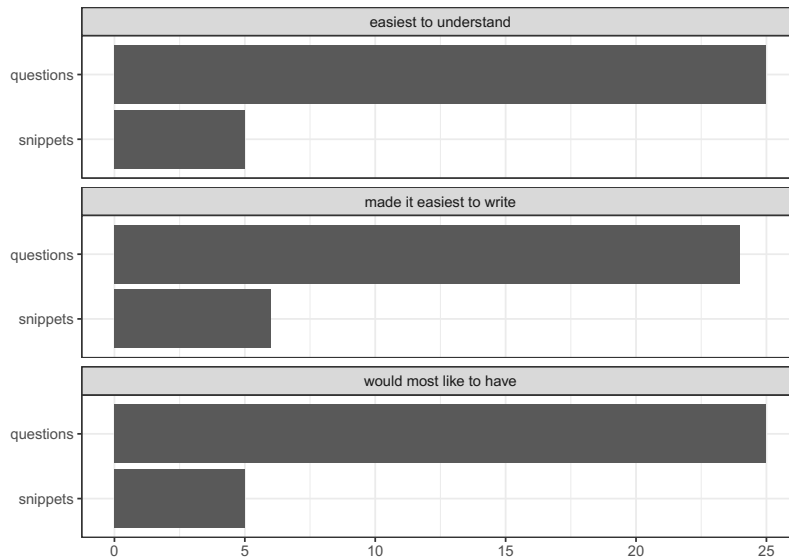


Figure 7.3: Participants chose the “bot” that offered Questions as most preferable along all three measures asked.

that Questions were deemed relevant significantly more often than Snippets, despite the fact that they represented two different views of the same topic. In fact, Questions contained *less* information than Snippets, since they were constructed by removing information. But the study results suggest that the information that remained was more generalizable.

Thus I conclude that while both Snippets and Questions were often useful to writers, writers find it easier to use Questions and are not significantly hurt by the loss of utility presented by not seeing a complete target example. However, I note that overall relevance of static suggestions was low, which presents an opportunity for using predictive text technology to adapt the suggestions to the

writer’s global and local context.

7.3 FEASIBILITY OF GENERATING QUESTIONS

Predictive models of language almost always predict content in the form of words and phrases, but the suggestions studied above—exemplar sentences and questions—are of a very different form. How might we generate topically relevant questions using statistical language understanding techniques?

In this section I present a brief example of how to adapt language understanding and generation approaches to the new task of identifying relevant questions.

The conceptual approach is to use language understanding techniques to group snippets from past writing into structural elements, then use language modeling techniques to predict which group is relevant.

Automated approaches based on language modeling technology can predict what groups are *relevant* given what the writer has generated so far. Since the number of automatically identified example groups is far smaller than the total size of the corpus, manual generation of the particular abstracted content is tractable, though future work may investigate automating even this step.

For this example, I took all Wikipedia articles that have the “film” infobox as the source data. I then extracted the plain text of the introduction section

of the article. I split those documents into sentences, and considered only documents that have ≥ 5 sentences, truncated documents at 50 sentences, and picked documents in random order to get a total of 1.25k sentences. I then filter those sentences to just those that are within the 25th to 75th percentile of word count.

I then computed a vector representation for each sentence as the mean of the ConceptNet Numberbatch vector for each word in the sentence, after removing stop words. I then use k-means to group these sentences into 128 clusters. I then filter the clusters to only the 80 clusters that occurred in at least 1% of documents.

To evaluate whether these groups were understandable and relevant, I attempted to generate questions that would be answered by sentences that were closest to the center of the identified cluster. Figure 7.4 shows some examples of the clusters identified, showing that at least some of the clusters identified even using this very simple pipeline group together sentences that answer similar questions, and those questions are meaningful for the target domain.

I then evaluated the ability of language modeling approaches to predict relevant clusters. For each of sentences that were assigned to one of the clusters, I trained a classifier to predict the cluster it belongs to, given the content of the

Questions: Did the film premiere at a film festival? How did it perform?

- The film premiered at the 2012 Sundance Film Festival and was screened out of competition at the 62nd Berlin International Film Festival in February 2012.
- The film premiered on January 21, 2013 at the 2013 Sundance Film Festival and was screened in competition at the 63rd Berlin International Film Festival.
- The film had its world premiere at the Sundance Film Festival on January 22, 2017, and later screened at the Berlin International Film Festival.
- The film premiered at the International Documentary Film Festival Amsterdam in November 2010 and has screened at several international festivals.

Questions: Is there live music in the film?

- A scene in which Susanna performs at a concert was filmed at Webster Hall using a pre-recorded vocal track, a backing band and a small audience.
- It shows her seeking out old musicians and asking them to sing or strum the songs they knew.
- In a mid-credits scene, Steve and his friends dance onstage with the Kids of Widney High as they perform the song "Respect".

Questions: What awards was it nominated for?

- The film was selected as entry for the Best Foreign Language Film at the 88th Academy Awards, but it was not nominated.
- It was selected as the Nigerien entry for the Best Foreign Language Film at the 91st Academy Awards, but it was not nominated.

Questions: What did reviewers say about it? What aspects impressed them?

- Stephen Holden, writing for the New York Times, calls the cinematography "spectacularly beautiful," and calls the film "a fascinating but rambling documentary."
- Janet Maslin of The New York Times called it "an absorbing, frightening, entirely believable movie, which is particularly amazing in view of its subject matter."
- Kinemacolor is wonderfully interesting and very beautiful and gives one the impression of having seen it all in reality".
- "TV Guide" rated the film 3 of 4 stars: "surprisingly exciting", "fascinating" and "sharp looking" with a good soundtrack.

Figure 7.4: Examples of sentences clustered by word vector k-means, and questions that could be abstracted from those sentences.

document that proceeds it. The baseline of always predicting the most common cluster would achieve a top-1 accuracy of 18.8% on this task. A simple Multinomial Naive Bayes classifier on tf-idf transformed unigram and bigram counts in the context document achieves a top-1 accuracy of 25.1%.

Thus, existing methods are applicable to this problem, though could be substantially improved.

7.4 SUMMARY

I reported on interventions that help writers set *content goals* as a promising area of opportunity for technological support of writing. I identified two types of ways that guidance about content goals could be presented to writers during writing: as verbatim *snippets* or as *questions*. In an exploratory user study, questions were preferred by a large margin. Finally, I found that Identifying clusters and predicting which are relevant is generally feasible using current technology, but more data (and perhaps new methods) will be necessary to generate question text.



Conclusion and Future Directions

Predictive text promises to help people write more efficiently and with fewer errors. However, when these predictions are shown to writers to use, they also function as suggestions of what content should be written. Past studies of these systems have neglected how suggestions might affect content, which has led them to not notice the unintentional and potentially harmful effects of current system designs and also to miss opportunities to design text predictions to achieve desired content effects.

In this dissertation I presented human-subjects studies of three attributes

of current predictive text systems: *visibility*, *length*, and *sentiment*. The first study found that word-level predictive suggestions reduced the extent to which writers used words that the predictive system did not expect, leading in some cases to shorter writing. The second study found that the multi-word phrase suggestions had a stronger effect on the predictability of writing than single-word suggestions. The third study found that phrase predictions are particularly vulnerable to propagating a present in training data. Since that bias was about the *sentiment* of the writing, this study also implied that suggestions can affect meaning, not just wording.

I then presented two enabling works in the direction of designing the content effects of predictive suggestions. First, I presented an approach to modeling how writers will respond to suggestions. This technique could enable designers to create systems that offer suggestions with characteristics that differ from those of the training data but nevertheless with some assurance that writers will find them relevant. Then, towards the goal of designing systems that guide writers about the topical structure of their documents, I presented a study of how writers respond to two different forms of topic prompts. This study found that writers found topic prompts more useful when they were presented as questions rather than example sentences, even when those sentences were drawn from

high-quality writing. Although questions are atypical outputs for natural language generation systems, I provided an illustrative example of how mainstream natural language modeling and generation approaches could be appropriated for this task.

The systems I used for the experiments in this dissertation used domain-specific training data to be able to offer predictive suggestions that were more relevant than is currently typical for deployed predictive text systems. However, language modeling technology has been advancing rapidly in recent years*, the results I presented here suggest that content effects will become more prevalent in the wild.

8.1 IMPLICATIONS

My findings underscore a general call that evaluations of intelligent interactive systems be based on authentic tasks (Buçinca et al., 2020), and specifically the call of Kristensson & Vertanen (2014) that text entry studies should include composition tasks. This is especially important since the effects of predictive text suggestions on writing may be subconscious.

Specifically, future studies of text entry technologies, especially predictive

*A recent language model by Microsoft uses 17 billion parameters: <https://www.microsoft.com/en-us/research/blog/turing-nlg-a-17-billion>

text, should measure the *content effects* of these systems. The main measure I used—number of predictable or unpredictable words—is simple to interpret and apply, even in large-scale deployed systems. Other measures are possible; for example, earlier versions of this work used a measure of the ideal number of taps needed to enter a text. Studies of content effects need to consider effect sizes: my studies found effect sizes of about one word per sentence, so future studies of content effects must be designed with sufficient statistical power to notice effects of comparable size.

In contrast, traditional text entry studies that use exclusively transcription tasks and measure speed and errors treat humans like transcribing machines, ignoring human thoughtfulness. Designers need to evaluate the systems that they make in a way that treats users more like whole people.

Overall, researchers and platforms should measure and document how intelligent technologies may manipulate what people create.

8.1.1 ETHICAL QUESTIONS

The evidence presented in this dissertation raises concerns regarding the ethics of current widely deployed predictive text systems and systems of the near future. I found that the content that the system shows affects the content that

writers create. The nature of this effect on content may be to unintentionally amplify biases in the data used to train the predictive system, or it might be in the form of intentional manipulation to the content based on models of suggestion acceptability.

Ethical issues arise in the design, implementation, and deployment of predictive text systems. Considerations include:

- How can privacy be maintained in the training, tuning, evaluation, and feedback processes?
- How can users (whose data is used for training the system) be informed about the degree to which their text is (or is not) being shown to other people?
- How can writer autonomy be respected in the design and evaluation of the user experience of the predictive text system? For example, what kinds of awareness of system characteristics and their own use of them would enable writers to make informed choices about whether and when to enable which predictive text system?
- How can potential risks involved be considered before deploying the system? For example, since phrases have greater impact on writing content than word suggestions (chapter 4) and risk perpetuating biases (chapter 5), the potential effects on writing content should be considered when making choices about how long suggestions should be and which ones should be offered.

- How can potentially harmful content recommendations be dealt with in a proactive way?
- Suggestions privilege a particular kind of “standard” English usage and communication norms. Although some systems use personalized data from individual writers, the vast majority of the training data—and thus suggestion quality and (probably) frequency—will still be biased towards “average” language. So: what effect do predictive systems have on linguistic and cultural diversity, especially for minority languages and cultures? How can these effects be considered in the design process?
- Since predictive systems function poorly in low-resource languages, might predictive text accelerate the demise of minority languages?
- While suggestion systems have been extensively studied in the context of accessible technology, motor disabilities have been studied much more than perceptual and cognitive disabilities. The suggestion interfaces commonly studied and deployed seem to privilege the sighted, physically able, and neurotypical. How can potential opportunities and harms to those with diverse physical and cognitive abilities be considered in the design process? For example, might people who struggle with impulsivity or who tend to over-trust systems accept suggestions that may in fact have been inappropriate?
- How might we be able to audit the effects that predictive systems are having on human communication at scale? Specifically:
How might we notice effects that we didn’t think to look for? and How might *external* observers be able to audit at least some of these potential

impacts while respecting user privacy?

- Platforms providing predictive text systems bear responsibility for the suggestions that their systems generate, to at least the degree that search engines are responsible for search results and autocomplete suggestions (Karapapa & Borghi, 2015). What technical and organizational tools and practices would enable platforms to be aware of the impact, potentially unintentional, of their systems on the people using them and the content that they produce? How can they do this in a way that respects the privacy and confidentiality of writers?
- How can communities govern the use of predictive text systems? What kinds of transparency about prediction methods, content, and use can help governments and online communities decide what limitations to place on predictive text systems? How can communities hold platforms accountable to the effects of their predictions on members of the community?
- If systems make certain kinds of speech easier than others (e.g., they are trained on a corpus of pro-government propaganda), do they limit freedom of speech in a practical sense?
- If a document is created largely by accepting suggestions, who is rightly the author of that document? Do the authors of the training data on which those predictions were based have any share in the resulting document?

A full discussion of all of these considerations is beyond the scope of this dissertation. However, I think the following two implications are straightforward:

First, predictive text systems should make it possible to disable suggestions of *full words* while continuing to suggest completions and corrections, even after a single letter. This change would greatly reduce the risk of content manipulation since writers likely already have a specific word in mind by the time they start typing the first letter, at a relatively small cost of at most one tap per word. At time of writing, neither Google’s GBoard or Apple’s iOS keyboard has this option.

Until such customization is available, writers can choose to use gesture typing: after a gesture, the suggestion bar is repurposed to show alternative interpretations for the previously entered word rather than next-word predictions. Speech recognition functionality may also be appropriate in some settings.

Second, platforms should share data about the use and impact of predictive suggestions made by their products, both individually and in aggregate. Individual writers should have the right to the data on how they used predictions. For example, the SwiftKey keyboard provides overall statistics about how efficiently and accurately a writer is using the keyboard, but these overall statistics do not allow writers insight into the content of what the keyboard suggested to them or how they used those suggestions. Google’s GBoard sends usage statistics to Google, including suggestion use (Hard et al., 2018), but does not provide a way

for writers to access their own information. In aggregate, platforms should share measures of the impacts of their predictions on content.

Other implications are less immediate but still important:

- The content that people write using predictive systems will become part of the corpora used to train language models used by future predictive systems. The resulting feedback effect could amplify biases. Those who collect data to train text prediction systems should be mindful of the ways in which the data they use was likely written.
- The language models that platforms use to generate predictions should be released openly as much as possible. Not only would this practice help defend against fake fully-automated text as [Gehrmann et al. \(2019\)](#) point out, but it would also assist in accountability for predictive text systems, and help flag training data that may have been computer-generated.
- Assistance through error avoidance ([Baldwin & Chai, 2012](#); [Mott et al., 2017](#)), correction ([Bi et al., 2014](#); [Vertanen et al., 2018](#)), and disambiguation ([Reyal et al., 2015](#)) may better preserve writer autonomy than word or phrase suggestion. These systems do still make certain texts easier to enter than others (e.g., it becomes more difficult to enter creative misspellings or made-up words), but the system's biases are less salient, so we expect that they would impact writing content less.

Predictive text has the power to impact human correspondence. Those who develop and deploy such systems must consider deeply how to use that power responsibly.

8.2 FUTURE WORK

Future work could further characterize how current predictive text affects writing content, such as by using even more sensitive measures to study other tasks (such as computer-aided translation or persuasive writing), languages, and suggestion interaction designs. Future work should also explore individual differences in how suggestions affect people: both situational affect (Ghosh et al., 2019) and stable traits (Garver et al., 2017) have been shown to modulate how people use predictive systems.

Future work may investigate how systems may be designed to have useful intentional biases. For example, biases towards a kind of language that is stereotypical in a domain can help those unfamiliar with that domain (or second-language learners) write in a more stylistically appropriate way. Systems could make recommendations that support members of minority groups in their goals of how much and to whom they reveal markers of their group membership (Reddy & Knight, 2016). Biases towards neutral, negative, or more factual review text, if implemented in a socially and technically thoughtful way, may help *reduce* the positive bias of online review data. And perhaps writers on opposing sides in a debate could receive writing assistance that helps them engage with the opposing side or ground their arguments in generally accepted facts.

Future work could also explore ways in which suggestions may be designed to have desirable effects on content. For example, predictive scaffolding could be used to help second-language writers write more fluently and naturally. Could suggestions be designed to help writers come up with ideas or express those ideas creatively? Initial studies have yielded mixed results (Clark et al., 2018), but the challenge is promising.

I studied descriptive writing, but writing has many interrelated purposes—to inform, to persuade, to explain, to entertain, etc. While many of the findings here may generalize to writing for other purposes, some different content effects are possible, and these other purposes present different kinds of design opportunities for alternative prediction designs.

The studies presented in this dissertation used writing that is relatively short, rarely exceeding a single paragraph. However, much of writing education is focused on making coherent writing beyond the paragraph level. Longer writing presents more opportunities for completion systems to take advantage of a greater degree of context in order to make even more relevant predictions, but predictions even more strongly informed by the way that other writers have made similar arguments. These predictions could have more nuanced effects on the content that a writer produces using a completion interaction. However,

longer writing also expands the design space of potential predictive augmentations, such as making use of prior drafts, helping writers move back and forth between surface text and abstracted representations such as outlines, “refactoring” the structure of documents to improve structure and clarify arguments, and adapting system behavior over the course of writing a single document.

I also only studied isolated writers, but writing is an increasingly collaborative process. Many opportunities exist for using text predictions in the service of collaborative writing. For example, a team of graders for a course could iteratively build on each others’ comments to students to efficiently formulate a library of high-quality comments that respond to issues common to many students’ work. Customer support representatives and help-desk workers already leverage common libraries of stock responses, but customization to specifics of the current conversation is difficult. A deeper integration of predictive text systems might enable these representatives to be both personable and informative.

These studies focused exclusively on prediction interactions, so the experimental apparatus was kept simple in other respects. However, modern text entry systems consist of closely interleaved prediction, correction, feedback, and editing mechanisms. These other mechanisms could modulate the effect of predictions.

8.3 PARTING THOUGHTS

This dissertation tells the story of an interplay between advances in system capabilities and development in interaction design. The growing digitization of communication has enabled access to language data at an unprecedented scale, which in turn has led to rapid advances in data-driven models of language. That language modeling technology has, in turn, powered new kinds of interactions with writers, such as autocomplete-style predictive text. More than simply making existing interactions better, improved language understanding technology has enabled an explosion of creativity in how interactive systems can be used to support writers. But my studies of how predictive text affects writers then suggested new kinds of interaction designs, which in turn suggested a new kind of modeling. I expect that this story is barely the opening pages of how pervasive connectivity, interactivity, and intelligent systems will support our connected, collaborative society of creative thinkers building on each others' thoughts using advanced versions of the technology we once called reading and writing.

References

- Alharbi, O., Arif, A. S., Stuerzlinger, W., Dunlop, M. D., & Komninos, A. (2019). Wisetype: A tablet keyboard with color-coded visualization and various editing options for error correction. In *Proceedings of Graphics Interface 2019*, GI 2019: Canadian Information Processing Society.
- Amir, O., Grosz, B. J., Gajos, K. Z., & Gultchin, L. (2019). Personalized change awareness: Reducing information overload in loosely-coupled teamwork. *Artificial Intelligence*, 275, 204–233.
- Arif, A. S., Kim, S., Stuerzlinger, W., Lee, G., & Mazalek, A. (2016). Evaluation of a Smart-Restorable Backspace Technique to Facilitate Text Entry Error Correction. *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*, (pp. 5151–5162).
- Arnold, K. C., Chang, K.-W., & Kalai, A. T. (2017). Counterfactual language model adaptation for suggesting phrases. In *Proceedings of the 8th International Joint Conference on Natural Language Processing (IJCNLP 2017)*, IJCNLP 2017.

Arnold, K. C., Chauncey, K., & Gajos, K. Z. (2018). Sentiment Bias in Predictive Text Recommendations Results in Biased Writing. In *Graphics Interface 2018* (pp. 8–11). Toronto, Ontario, Canada.

Arnold, K. C., Chauncey, K., & Gajos, K. Z. (2020). Predictive text encourages predictable writing. In *Proceedings of the 25th International Conference on Intelligent User Interfaces, IUI '20* (pp. 128–138). New York, NY, USA: Association for Computing Machinery.

Arnold, K. C., Gajos, K. Z., & Kalai, A. T. (2016). On suggesting phrases vs. predicting words for mobile text composition. In *Proceedings of the 29th Annual Symposium on User Interface Software and Technology, UIST '16* (pp. 603–608). New York, NY, USA: Association for Computing Machinery.

Babaian, T., Grosz, B. J., & Shieber, S. M. (2002). A writer's collaborative assistant. In *Proceedings of the 7th international conference on Intelligent user interfaces - IUI '02* (pp.7). New York, New York, USA: ACM Press.

Baldwin, T. & Chai, J. (2012). Towards online adaptation and personalization of key-target resizing for mobile devices. In *Proceedings of the 2012 ACM International Conference on Intelligent User Interfaces, IUI '12* (pp. 11–20). New York, NY, USA: Association for Computing Machinery.

- Barocas, S. & Selbst, A. (2016). Big Data's Disparate Impact. *California law review*, 104(1), 671–729.
- Bates, D., Mächler, M., Bolker, B., & Walker, S. (2015). Fitting Linear Mixed-Effects Models Using lme4. *Journal of Statistical Software*, 67(1).
- Berrios, R., Totterdell, P., & Kellett, S. (2015). Eliciting mixed emotions: A meta-analysis comparing models, types and measures. *Frontiers in Psychology*, 6(MAR), 1–15.
- Bi, X., Ouyang, T., & Zhai, S. (2014). Both complete and correct? Multi-Objective Optimization of Touchscreen Keyboard. In *Proceedings of the 32nd annual ACM conference on Human factors in computing systems - CHI '14* (pp. 2297–2306). New York, New York, USA: ACM Press.
- Bird, S., Klein, E., & Loper, E. (2009). *Natural Language Processing with Python*. O'Reilly Media, Inc.
- Bolukbasi, T., Chang, K.-W., Zou, J., Saligrama, V., & Kalai, A. (2016). Man is to computer programmer as woman is to homemaker? debiasing word embeddings. In *Proceedings of the 30th International Conference on Neural Information Processing Systems, NIPS'16* (pp. 4356–4364). Red Hook, NY, USA: Curran Associates Inc.

Buçinca, Z., Lin, P., Gajos, K. Z., & Glassman, E. L. (2020). Proxy tasks and subjective measures can be misleading in evaluating explainable ai systems. In *Proceedings of the 25th International Conference on Intelligent User Interfaces, IUI '20* New York, NY, USA: ACM.

Buschek, D., Bisinger, B., & Alt, F. (2018). Researchime: A mobile keyboard application for studying free typing behaviour in the wild. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems, CHI '18* New York, NY, USA: Association for Computing Machinery.

Caliskan, A., Bryson, J. J., & Narayanan, A. (2017). Semantics derived automatically from language corpora contain human-like biases. *Science*, 356(6334), 183–186.

Chancey, E. T., Bliss, J. P., Yamani, Y., & Handley, H. A. (2017). Trust and the Compliance-Reliance Paradigm: The Effects of Risk, Error Bias, and Reliability on Trust and Dependence. *Human Factors*, 59(3), 333–345.

Chen, M. X., Lee, B. N., Bansal, G., Cao, Y., Zhang, S., Lu, J., Tsay, J., Wang, Y., Dai, A. M., Chen, Z., & et al. (2019). Gmail smart compose: Real-time assisted writing. In *Proceedings of the 25th ACM SIGKDD International*

Conference on Knowledge Discovery & Data Mining, KDD '19 (pp. 2287–2295).

New York, NY, USA: Association for Computing Machinery.

Chenoweth, N. A. & Hayes, J. R. (2003). The Inner Voice in Writing. *Written Communication*, 20(1), 99–118.

Clark, E., Ross, A. S., Tan, C., Ji, Y., & Smith, N. A. (2018). Creative writing with a machine in the loop: Case studies on slogans and stories. In *23rd International Conference on Intelligent User Interfaces, IUI '18* (pp. 329–340). New York, NY, USA: Association for Computing Machinery.

Cohn-Gordon, R., Goodman, N., & Potts, C. (2018). Pragmatically informative image captioning with character-level inference. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)* (pp. 439–443).

Darragh, J. J., Witten, I. H., & James, M. L. (1990). The Reactive Keyboard: A Predictive Typing Aid. *Computer*, 23(11), 41–49.

Deane, P., Odendahl, N., Quinlan, T., Fowles, M., Welsh, C., & Bivens-Tatum, J. (2008). Cognitive models of writing: Writing proficiency as a complex integrated skill. *ETS Research Report Series*, 2008(2), i–36.

Dehghani, M., Rothe, S., Alfonseca, E., & Fleury, P. (2017). Learning to Attend, Copy, and Generate for Session-Based Query Suggestion. In *CIKM*.

Dell, N., Vaidyanathan, V., Medhi, I., Cutrell, E., & Thies, W. (2012). "Yours is better!": Participant response bias in HCI. *Proceedings of the 2012 ACM annual conference on Human Factors in Computing Systems - CHI '12*, (pp. 1321–1330).

Dragicevic, P. (2016). Fair Statistical Communication in HCI. In *Modern Statistical Methods for HCI* (pp. 291 – 330). Springer.

Ferretti, R. P., Lewis, W. E., & Andrews-Weckerly, S. (2009). Do Goals Affect the Structure of Students' Argumentative Writing Strategies? *Journal of Educational Psychology*, 101(3), 577–589.

Fiannaca, A., Paradiso, A., Shah, M., & Morris, M. (2017). AACrobat: Using mobile devices to lower communication barriers and provide autonomy with Gaze-based AAC. *Proceedings of the ACM Conference on Computer Supported Cooperative Work, CSCW*.

Findlater, L. & Wobbrock, J. (2012). Personalized Input: Improving Ten-Finger Touchscreen Typing through Automatic Adaptation. *Proceedings of the 2012*

ACM annual conference on Human Factors in Computing Systems - CHI '12, (pp. 815–824).

Fourney, A., Morris, M. R., & White, R. W. (2017). Web search as a linguistic tool. In *Proceedings of the 26th International Conference on World Wide Web, WWW '17* (pp. 549–557). Republic and Canton of Geneva, CHE: International World Wide Web Conferences Steering Committee.

Fowler, A., Partridge, K., Chelba, C., Bi, X., Ouyang, T., & Zhai, S. (2015). Effects of Language Modeling and its Personalization on Touchscreen Typing Performance. *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems - CHI '15*, (pp. 649–658).

Garver, S., Harriott, C., Chauncey, K., & Cunha, M. (2017). Co-adaptive Relationships with Creative Tasks. *Proceedings of the Companion of the 2017 ACM/IEEE International Conference on Human-Robot Interaction - HRI '17*, (pp. 123–124).

Gehrmann, S., Strobel, H., & Rush, A. M. (2019). GLTR: Statistical detection and visualization of generated text. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics: System Demonstrations* (pp. 111–116).

Geras, K. J., Mohamed, A., Caruana, R., Urban, G., Wang, S., Aslan, O., Philipose, M., Richardson, M., & Sutton, C. (2016). Blending LSTMs into CNNs. In *4th International Conference on Learning Representations (ICLR), workshop track*.

Ghosh, S., Hiware, K., Ganguly, N., Mitra, B., & De, P. (2019). Does emotion influence the use of auto-suggest during smartphone typing? In *Proceedings of the 24th International Conference on Intelligent User Interfaces - IUI '19* (pp. 144–149). New York, New York, USA: ACM Press.

Ghosh, S., Vinyals, O., Strope, B., Roy, S., Dean, T., & Heck, L. (2016). Contextual LSTM (CLSTM) models for large scale NLP tasks. In *KDD Workshop on Large-scale Deep Learning for Data Mining (DL-KDD)*.

Glorot, X. & Bengio, Y. (2010). Understanding the difficulty of training deep feedforward neural networks. *Pmlr*, 9, 249–256.

Goodman, N. D. & Frank, M. C. (2016). Pragmatic Language Interpretation as Probabilistic Inference. *Trends in Cognitive Sciences*, 20(11), 818–829.

Graves, A., Semmel, M., & Gerber, M. (1994). The Effects of Story Prompts on the Narrative Production of Students with and without Learning Disabilities. *Learning Disability Quarterly*, 17(2), 154.

- Green, S., Chuang, J., Heer, J., & Manning, C. D. (2014). Predictive translation memory. *Proceedings of the 27th annual ACM symposium on User interface software and technology - UIST '14*, (pp. 177–187).
- Hard, A., Rao, K., Mathews, R., Ramaswamy, S., Beaufays, F., Augenstein, S., Eichner, H., Kiddon, C., & Ramage, D. (2018). Federated learning for mobile keyboard prediction. *arXiv preprint arXiv:1811.03604*.
- Hart, S. G. (2006). Nasa-Task Load Index (NASA-TLX); 20 Years Later. *Proceedings of the Human Factors and Ergonomics Society Annual Meeting*, 50(9), 904–908.
- Hayes, J. R. & Chenoweth, N. A. (2006). Is Working Memory Involved in the Transcribing and Editing of Texts? *Written Communication*, 23(2), 135–149.
- Heafield, K., Pouzyrevsky, I., Clark, J. H., & Koehn, P. (2013). Scalable Modified Kneser-Ney Language Model Estimation. *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, (pp. 690–696).
- Horvitz, E. (1999). Principles of mixed-initiative user interfaces. *Proceedings of the SIGCHI conference on Human factors in computing systems the CHI is the limit - CHI '99*, (pp. 159–166).

Hu, Z., Yang, Z., Liang, X., Salakhutdinov, R., & Xing, E. P. (2017). Toward Controlled Generation of Text. In *Proceedings of the International Conference on Machine Learning (ICML)*.

Hui, J. S., Gergle, D., & Gerber, E. M. (2018). IntroAssist. *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems - CHI '18*, (pp. 1–13).

Ionides, E. L. (2008). Truncated importance sampling. *Journal of Computational and Graphical Statistics*, 17(2), 295–311.

Kannan, A., Kurach, K., Ravi, S., Kaufmann, T., Tomkins, A., Miklos, B., Corrado, G., Lukacs, L., Ganea, M., Young, P., & Ramavajjala, V. (2016). Smart Reply: Automated Response Suggestion for Email. In *KDD*.

Karapapa, S. & Borghi, M. (2015). Search engine liability for autocomplete suggestions: Personality, privacy and the power of the algorithm. *International Journal of Law and Information Technology*, 23(3), 261–289.

Klein, G., Kim, Y., Deng, Y., Senellart, J., & Rush, A. M. (2017). Opennmt: Open-source toolkit for neural machine translation. In *Proceedings of ACL 2017, System Demonstrations* (pp. 67–72).

Kneser, R. & Ney, H. (1995). Improved backing-off for m-gram language modeling. In *Acoustics, Speech, and Signal Processing, 1995. ICASSP-95., 1995 International Conference on*, volume 1 (pp. 181–184).: IEEE.

Koester, H. H. & Levine, S. P. (1994). Modeling the Speed of Text Entry with a Word Prediction Interface. *IEEE Transactions on Rehabilitation Engineering*, 2(3), 177–187.

Krause, M., Garncarz, T., Song, J., Gerber, E. M., Bailey, B. P., & Dow, S. P. (2017). Critique Style Guide. *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems - CHI '17*, (pp. 4627–4639).

Kristensson, P. O. & Vertanen, K. (2014). The Inviscid Text Entry Rate and its Application as a Grand Goal for Mobile Text Entry. *Proceedings of the 16th international conference on Human-computer interaction with mobile devices & services - MobileHCI '14*, (pp. 335–338).

Lau, J. H., Clark, A., & Lappin, S. (2016). Grammaticality, Acceptability, and Probability: A Probabilistic View of Linguistic Knowledge. *Cognitive Science*, (pp. 1–40).

Li, F. C. Y., Guy, R. T., Yatani, K., & Truong, K. N. (2011). The 1Line keyboard: A QWERTY layout in a single line. *UIST'11 - Proceedings of the 24th*

Annual ACM Symposium on User Interface Software and Technology, (pp. 461–470).

Limpo, T. & Alves, R. A. (2018). Effects of planning strategies on writing dynamics and final texts. *Acta Psychologica*, 188(January), 97–109.

Lin, T. Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., & Zitnick, C. L. (2014). Microsoft COCO: Common objects in context. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 8693 LNCS(PART 5), 740–755.

Lipton, Z. C., Vikram, S., & McAuley, J. (2015). Generative Concatenative Nets Jointly Learn to Write and Classify Reviews. *arXiv preprint arXiv:1511.03683*.

Mahyar, N., James, M. R., Ng, M. M., Wu, R. A., & Dow, S. P. (2018). CommunityCrit : Inviting the Public to Improve and Evaluate Urban Design Ideas through Micro-Activities. *Proc. of CHI*, (pp. 1–14).

Martin, R. C., Crowther, J. E., Knight, M., Tamborello II, F. P., & Yang, C.-L. (2010). Planning in sentence production: Evidence for the phrase as a default planning scope. *Cognition*, 116(2), 177–192.

- McCallum, A., Freitag, D., & Pereira, F. (2000). Maximum entropy Markov models for information extraction and segmentation. In *Proceedings of the International Conference on Machine Learning (ICML)*.
- McCormick, P. A. (1997). Orienting attention without awareness. *Journal of experimental psychology. Human perception and performance*, 23(1), 168–180.
- Meyer, J. (2004). Conceptual issues in the study of dynamic hazard warnings. *Human Factors*, 46(2), 196–204.
- Morash, V. S., Siu, Y.-T., Miele, J. A., Hasty, L., & Landau, S. (2015). Guiding Novice Web Workers in Making Image Descriptions Using Templates. *ACM Transactions on Accessible Computing*, 7(4), 1–21.
- Mott, M. E., Williams, S., Wobbrock, J. O., & Morris, M. R. (2017). Improving Dwell-Based Gaze Typing with Dynamic, Cascading Dwell Times. *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems - CHI '17*, (pp. 2558–2570).
- Ngoon, T. J., Fraser, C. A., Weingarten, A. S., Dontcheva, M., & Klemmer, S. (2018). Interactive Guidance Techniques for Improving Creative Feedback. *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems - CHI '18*, (pp. 1–11).

Orita, N., Vornov, E., Feldman, N., & Daumé III, H. (2015). Why discourse affects speakers' choice of referring expressions. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)* (pp. 1639–1649). Stroudsburg, PA, USA: Association for Computational Linguistics.

Pang, B. & Lee, L. (2005). Seeing stars: Exploiting class relationships for sentiment categorization with respect to rating scales. In *Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics, ACL '05* (pp. 115–124). USA: Association for Computational Linguistics.

Polacek, O., Sporka, A. J., & Butler, B. (2013). Improving the methodology of text entry experiments. *4th IEEE International Conference on Cognitive Infocommunications, CogInfoCom 2013 - Proceedings*, (pp. 155–160).

Prabhavalkar, R., Alsharif, O., Bruguier, A., & McGraw, L. (2016). On the compression of recurrent neural networks with an application to LVCSR acoustic modeling for embedded speech recognition. In *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)* (pp. 5970–5974).

- Quinn, P. & Zhai, S. (2016). A Cost-Benefit Study of Text Entry Suggestion Interaction. *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*, (pp. 83–88).
- Raaijmakers, J. G. W. & Jakab, E. (2013). Rethinking inhibition theory: On the problematic status of the inhibition theory for forgetting. *Journal of Memory and Language*, 68(2), 98–122.
- Radford, A., Jozefowicz, R., & Sutskever, I. (2017). Learning to generate reviews and discovering sentiment. *arXiv preprint arXiv:1704.01444*.
- Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., & Sutskever, I. (2018). Language Models are Unsupervised Multitask Learners. unpublished manuscript from <https://openai.com/blog/better-language-models/>.
- Rauschenberger, R. (2003). Attentional capture by auto- and allo-cues. *Psychonomic Bulletin & Review*, 10(4), 814–842.
- Reddy, S. & Knight, K. (2016). Obfuscating Gender in Social Media Writing. *Proceedings of ACL Workshop on Computational Social Science*, (pp. 17–26).
- Reyal, S., Zhai, S., & Kristensson, P. O. (2015). Performance and User Experience of Touchscreen and Gesture Keyboards in a Lab Setting and in the

Wild. *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems - CHI '15*, (pp. 679–688).

Rough, D., Vertanen, K., & Kristensson, P. O. (2014). An evaluation of Dasher with a high-performance language model as a gaze communication method. *Proceedings of the 2014 International Working Conference on Advanced Visual Interfaces - AVI '14*, (pp. 169–176).

Rush, A. M., Chang, Y.-W., & Collins, M. (2013). Optimal beam search for machine translation. In *EMNLP* (pp. 210–221).

Sordani, A., Galley, M., Auli, M., Brockett, C., Ji, Y., Mitchell, M., Nie, J.-Y., Gao, J., & Dolan, B. (2015). A neural network approach to context-sensitive generation of conversational responses. In *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics (NAACL)*.

Stoop, W. & van den Bosch, A. (2014). Using idiolects and sociolects to improve word prediction. In *Proceedings of the 14th Conference of the European Chapter of the Association for Computational Linguistics* (pp. 318–327).

Swaminathan, A. & Joachims, T. (2015). Counterfactual risk minimization. In *Proceedings of the 24th International Conference on World Wide Web Com-*

panion (pp. 939–941).: International World Wide Web Conferences Steering Committee.

Teevan, J., Kaur, H., Williams, A. C., Iqbal, S. T., Thompson, A. L., & Lasecki, W. S. (2018). Creating Better Action Plans for Writing Tasks via Vocabulary-Based Planning. *Proceedings of the ACM on Human-Computer Interaction*, 2(CSCW), 1–22.

Torrance, M. (2015). Understanding Planning in Text Production. In *Handbook of Writing Research* chapter 5, (pp. 72–87). New York, NY: Guilford Press, 2 edition.

Torrance, M. & Galbraith, D. (2006). The processing demands of writing. *Handbook of Writing Research*, (pp. 67–82).

Torrance, M., Johansson, R., Johansson, V., & Wengelin, Å. (2016). Reading during the composition of multi-sentence texts: an eye-movement study. *Psychological Research*, 80(5), 729–743.

Trnka, K., McCaw, J., Yarrington, D., McCoy, K. F., & Pennington, C. (2009). User interaction with word prediction: The effects of prediction quality. *ACM Trans. Access. Comput.*, 1(3).

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., & Polosukhin, I. (2017). Attention Is All You Need. In *Advances in neural information processing systems* (pp. 5998–6008).

Vertanen, K., Fletcher, C., Gaines, D., Gould, J., & Kristensson, P. O. (2018). The impact of word, multiple word, and sentence input on virtual keyboard decoding performance. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*, CHI '18 New York, NY, USA: Association for Computing Machinery.

Vertanen, K., Gaines, D., Fletcher, C., Stanage, A. M., Watling, R., & Kristensson, P. O. (2019). VelociWatch : Designing and Evaluating a Virtual Keyboard for the Input of Challenging Text. *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*, (pp. 1–14).

Vertanen, K. & MacKay, D. J. (2010). Speech Dasher: Fast Writing using Speech and Gaze. *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, (pp. 595–598).

Vertanen, K., Memmi, H., Emge, J., Reyal, S., & Kristensson, P. O. (2015). VelociTap: Investigating Fast Mobile Text Entry using Sentence-Based Decod-

ing of Touchscreen Keyboard Input. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems - CHI '15* (pp. 659–668).

Wade-Stein, D. & Kintsch, E. (2004). Summary Street: Interactive Computer Support for Writing. *Cognition and Instruction*, 22(3), 333–362.

Wang, S. & Manning, C. D. (2012). Baselines and bigrams: Simple, good sentiment and topic classification. In *Proceedings of the 50th annual meeting of the association for computational linguistics: Short papers-volume 2* (pp. 90–94).: Association for Computational Linguistics.

Ward, D. J., Blackwell, A. F., & MacKay, D. J. C. (2000). Dasher - A Data Entry Interface Using Continuous Gestures and Language Models. *Proceedings of the 13th annual ACM symposium on User interface software and technology*, 2, 129–137.

Wickens, C. D., Dixon, S. R., & Johnson, N. (2006). Imperfect diagnostic automation: An experimental examination of priorities and threshold setting. *Proceedings of the Human Factors and Ergonomics Society Annual Meeting*, 50(3), 210–214.

Yang, Q., Cranshaw, J., Amershi, S., Iqbal, S. T., & Teevan, J. (2019). Sketching nlp: A case study of exploring the right things to design with language

intelligence. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*, CHI '19 (pp. 1–12). New York, NY, USA: Association for Computing Machinery.

Yantis, S. & Jonides, J. (1984). Abrupt visual onsets and selective attention: Evidence from visual search. *Journal of Experimental Psychology: Human Perception and Performance*, 10(5), 601–621.

Yasuoka, K. & Yasuoka, M. (2011). On the prehistory of qwerty. *ZINBUN*, 42.

Zhai, S. & Kristensson, P. O. (2012). The word-gesture keyboard: reimagining keyboard interaction. *Communications of the ACM*, 55(9), 91–101.

Zhao, S., Chevalier, F., Ooi, W. T., Lee, C. Y., & Agarwal, A. (2012). Auto-ComPaste: Auto-completing text as an alternative to copy-paste. In *Proceedings of the International Working Conference on Advanced Visual Interfaces, AVI '12* (pp. 365–372). New York, NY, USA: ACM.

Zizzo, D. J. (2010). Experimenter demand effects in economic experiments. *Experimental Economics*, 13(1), 75–98.