Toward Digital *Progymnasmata*

Kyle Booten
Neukom Institute for Computational Science
Dartmouth College
Hanover, NH 03755 USA
kyle.p.booten@dartmouth.edu

Abstract

The classical arts of rhetoric described intricate training methodologies for making the writer linguistically flexible and able to avoid stylistic vices. Inspired by the ancient *progymnasmata*, this paper presents Progym, an interactive writing system designed to notice when writers resort to expected language and encourage them to avoid these linguistic elements. Two versions of the system are presented. The first discourages writers from using words that, within a large corpus, are often used to describe a target word. The second discourages writers from using syntactic patterns found in a small corpus. In user studies, Progym did indeed push writers away from these features, though the different versions led to different styles of revision.

Introduction

From its roots in antiquity through its second zenith during the Early Modern period, the arts of rhetoric provided learners with exercises designed to hone their use of language. While much of rhetorical practice was grounded in the imitation of received forms and authors, this does not mean that it did not also foster creativity. Among the ancient Greek *progymnasmata* (a set of preliminary rhetorical exercises) were *ekphrasis*—the description of an object or artwork with a vivid attention to detail; another such exercise was *paraphrase*, the reiteration of a statement with different syntax (Kennedy 2003). The point of these and other *progymnasmata* was not that they themselves produced full or complete texts; rather they were a kind of "gymnastic training for the mind...shaping it for certain activities just as athletics shaped the body" (Webb 2001). A similar spirit of athleticism can be seen much later in Erasmus' treatise on rhetorical education *De Copia* (1512 1978), which recommends various techniques for "diversifying" one's speech or writing and avoiding monotony. Demonstrating a rhetorical exercise meant to promote linguistic flexibility, Erasmus's text offers over a hundred and fifty distinct variations on a simple phrase, the Latin equivalent of "Your letter has delighted me very much."

This paper documents the design of a system that provides computational feedback as a form of rhetorical training in the context of creative writing tasks. Inspired by the gymnastic notion of language found in the rhetorical tradition, and especially by Erasmus' example of forcing oneself into linguistic "copiousness" or flexibility, this system is designed to encourage creativity by steering writers away from particularly common and expected words and syntactic patterns. Like the classical *progymnasmata*, the system is not primarily designed to produce complete or sufficient texts. Rather it is conceived of as a training tool designed to encourage linguistic flexibility. On a technical level, this paper describes techniques for gathering overly-frequent linguistic phenomena using text mining. This paper documents the design of two different versions of this progymnastic system. Results from user studies document the impacts the system's different types of feedback had on the ways that writers used language.

Related Work

Computational Writing Assistants

Within the field of computational creativity, researchers have developed systems that assist humans in the production of creative writing. Some of these computational systems function as collaborators. Say Anything (Swanson and Gordon 2008) functions as a kind of creative Information Retrieval system for narrative composition, returning a sentence from a large collection of texts that is most similar to the human writer's. Inspired by this system, Creative Help (Roemmele and Gordon 2015) uses similar techniques to match human input with a sentence from a large corpus, although it allows writers to more flexibly control how they deploy these sentences. The system approaches interactive storytelling as an Information Retrieval task, with the algorithmic writer returning a sentence from a large collection of sentences that is the most similar to the user's. More recent research from Roemmele (2016) has explored the use of the predictive models of neural networks as an improvement upon traditional techniques of Information Retrieval for offering suggestions to writers as they write. Manjavacas et al. (2017) also used a language model to provide continuations of a human writer's text.

Creative computation research on writing assistants has also drawn on research within the field on the generation of literary texts. For instance, "Co-PoeTryMe" (Oliveira, Mendes, and Boavida 2017) is an interactive version of PoetTryMe (Oliveira 2012), a system for generating poetry

in multiple languages using a combination of networks of semantically-related words and a variety of syntactic and formal constraints, including rhyme and number of syllables. Co-PoeTryMe makes this poetry generation tool interactive by providing an interface for specifying the parameters of the generator and for iterative generating and editing words and lines. Inkwell (Gabriel, Chen, and Nichols 2015) is another system that is both a poetry generator and a poetry-writing assistant. As an assistant, it combines a wide variety of individual functions, such as mimicking a writer's personality and style.

Creative assistants for writing may also provide something like "inspiration" rather than engage in full-fledged collaboration. Gonçalves et al. (2017) demonstrated a system that uses what they call "subliminal priming" to provide writers with feedback to help them get over writer's block. The Poetry Machine (Kantosalo et al. 2014; Kantosalo, Toivanen, and Toivonen 2015), another repurposing of a poetry generation system (Toivanen et al. 2012), offers the writer intitial "fragments" of poetry as a way to help them overcome the difficulty of starting the writing process. Indurkhya (2016) used a similar approach, providing writers (in this case, children) with a combination of related and unrelated words in order to both scaffold the production of a narrative and spur creativity. Researchers have also used crowdsourced images to stimulate creativity and mental well-being during a creative writing task (Gonçalves and Campos 2018).

Progym does not position itself as a "collaborator." Neither does it supply the writer with fragmentary suggestions with the goal that, by integrating them, the writer may make a text more compelling (or merely overcome some of the psychological barriers of writing, such as writers block). Neither does it aim to make the writer feel better while writing. Rather it offers explicitly negative feedback to direct writers to be more creative. In this sense, it is a kind of "coach" (Lubart 2005) as well as a kind of "audience" (Riedl and O'Neill 2009), albeit an opinionated and in fact critical one. The main contribution of this paper is to explore how a system can ask a writer to avoid certain kinds of uncreativity.

Mining Semantic Relations

One version of the Progym system is based on semantic relations between words mined from a large corpus of texts. The notion of mining texts for semantic relations was described by Hearst (1992). Related techniques have been used to mine semantic relationships between words as a way to generate poetry (Toivanen et al. 2012; Veale 2013) and metaphor (Veale and Hao 2007). Veale and Hao's "Jigsaw Bard" (2011) turns semantic relations mined from the web into "a creative thesaurus" of metaphors—in a way, another kind of creative writing assistant.

A main goal of this paper is to take this familiar approach to extracting semantic connections from large corpora and use it in the context of a writing assistant that explicitly wants the user to avoid these statistically-predictable semantic connections between words.

moon (adj)	full new bright young pale old white great high waning
moon (verb)	shine shin[2]hang set sink arise shed light climb cast
moon (noun)	light ray face surface beam disc orbit disk revolution distance
tree (adj)	old great tall large big young green small hollow beautiful
tree (verb)	spread bear wave blossom surround bend bud live hang overhang
tree (noun)	shade branch life root shadow side heart leaf head crown
queen (adj)	young little great beautiful fair good new old dead poor
queen (verb)	send sit die reign wear speak live think smile hear
queen (noun)	room chamber apartment death hand presence command eye taste heart
wolf (adj)	hungry gray old big grey great young large fierce dead
wolf (verb)	howl eat prowl devour creep leap kill roam attack catch
wolf (noun)	head mouth den skin tooth howl fang tongue eye tail

Table 1: Most Frequently Related Words (Lemmatized) Extracted from Project Gutenberg Text

Progym V.1: Avoiding Expected Words

The sun is bright. The sun shines. The sun has beams. Compare these plausible assertions to the following: *The sun is dim. The sun blinks. The sun has banners.*

The first version of the Progym systems aims to steer writers away from the former—that is, from plausible but common descriptions of a topic noun and toward less common ones.[1]

Finding Common Words Common relationships between words were mined from the a selection of the Project Gutenberg corpus using the SpaCy dependency parser (Honnibal and Johnson 2015) , which represents any input sentence as a directional graph of syntactic as well as semantic relationships between words. Using this parser, the following relationships were extracted:

-Adjectival Relations For any noun, the system extracted adjectives that were the child of that noun via an `adjmod` (adjectival modifier) dependency relation. For instance, from the sentence "The old man is weary" it would extract

[1]This can be thought of as encouraging "creativity" in the broad sense that deviation from a statistically-common pattern amounts to a subversion of a "priming" (Hoey 2007).

[2]Ostensibly an artifact of inconsistencies in the lemmatization of verb forms of "shine."

(man,old) and (man,weary), using the lemmatized version of the noun.

-Possessive Relations For any noun, the system found all nouns that were the child of this noun via a poss (possession modifier) dependency relation. For instance, from the sentence "The dog's fur is golden" it would extract (dog,fur), using the lemmatized version of the noun.

-Verb Relations For each noun, the system found the verb that was the parent of this noun via a nsubj (nominal subject) relation. In addition, for each noun, the system found the present participle (tagged VBG) that was the child of the noun via an adjmod relation. For instance, from the sentence "The howling wolf chased me" it would extract (wolf,howl) and (wolf,chase), using the lemmatized version of nouns and verbs.

Using these techniques, fragments were mined from each of 14,928 English-language texts from Project Gutenberg; this is a collection of open source texts of a mostly literary nature, and so it was both convenient and, since I wanted to mine relations that appear in literary language, befitting of the task. Mining fragments was limited to the first 100,000 characters of each text, a limit imposed to ensure a reasonable compute time. Each word and each pair was further verified to be a valid word with the correct part of speech using WordNet (Miller 1995). To deal with the fact that certain uses of words may be idiosyncratic to a particular author, each text within the selection of the Gutenberg corpus was only able to contribute a specific relation between a noun and another word at most once. Using these criteria, an average of 322 Adjectival Relations were discovered for 27,444 nouns, an average of 176 Verb Relations were found for 26,443 nouns, and an average of 45 possessive relations were found for 6,729 nouns. Table 1 shows some of the top nouns, adjectives, and verbs found through these relations for several target nouns.

For each noun, a threshold was set either at 3 or at the number of occurrences of the pair at the 90th percentile of all observed relations of that specific type, whichever was higher. This was done to deal with rare nouns or nouns with few relations of a specific type, especially since even relatively few Possessive Relations were extracted overall. For Verb Relations and Adjectival Relations, certain very common words (such as "is" and "such") were treated as stop words and excluded. This process produced, for each noun, a list of Boring Words—Boring Verbs, Boring Adjectives, and Boring Nouns.

Interface

The Progym system is deployed as a web-based interface designed specifically for the user study (see Figure 1). The interface itself is straightforward and minimalistic, presenting the user with a series of ten text input areas. It is intended to be used in the context of an ekphrastic task in which a user must write ten sentences about a specific noun.

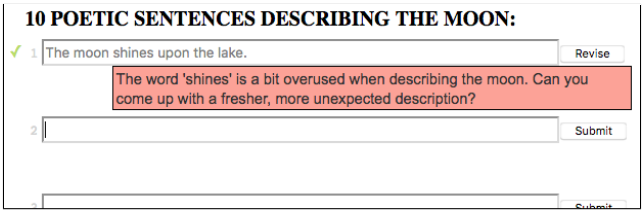

Figure 1: Progym's Interface

Each time the user "submits" a sentence, the system part-of-speech tags the sentence and checks its adjectives, lemmatized verbs, and lemmatized nouns against that noun's Boring Words. If a Boring Word is detected, Progym presents the user with a message asking them to revise it—e.g. "The words 'fluffy' and 'white' are a bit overused when describing a cloud. Can you come up with a fresher, more unexpected description?" If the input sentence contains, for instance, both one of the target noun's Boring Adjective and one of its Boring Verbs, it randomly focuses on one part-of-speech, and at most two different words of this part-of-speech. Users can then revise and resubmit their sentences, once again triggering the system's evaluation so that for the critical comment to disappear all Boring Words must be purged from the sentence.

User Study 1

For the purposes of the user study, Amazon Mturk crowdworkers were asked to write ten "poetic" sentences, each describing a different aspect of *the moon* or *a tree*.[3] These words were chosen as they are both relatively high-frequency nouns with correspondingly ample numbers of Boring Words (for "moon," 29 Boring Nouns, 26 Boring Verbs, and 53 Boring Adjectives; for "tree," 15 Boring Nouns, 56 Boring Verbs, and 111 Boring Adjectives). In addition, from Percy Bysshe Shelley's "To the Moon" to Coleridge's "This Lime-Tree Bower My Prison," both topics have a long history as objects of ekphrastic description. Workers were either given feedback by Progym (n = 33 for moon, n = 42 for tree) or not (n = 44 for moon, n = 47 for tree).

Use of Expected Words

Progym's functions for identifying the use of Boring Words were repurposed for the analysis of the sentences written by the Mturk participants under the four conditions, Tree-Assisted (by Progym's suggestions), Moon-Assisted, Tree-Unassisted, Moon-Unassisted.

Participants could revise a sentence multiple times, and the system recorded each revision to each of the participant's ten sentences. As these writers revised according to Progym's feedback in the assisted conditions, they lessened the number of Boring Words in their texts. Looking at the earliest version of sentences, Moon-Assisted poems had an

[3]The Github library quickstart-mturk was adapted with the permission of its author, user akuznets0v.

average of 7.31 Boring Words (*SD* = 3.17); looking at the most recent (i.e. "final") version of sentences, they had an average of 3.90 (*SD* = 3.60), a statistically significant difference according to a two-tailed t-test, $t(82) = 7.10$, $p < .001$. Looking at the earliest version of sentences, Tree-Assisted poems had an average of 6.03 Boring Words (*SD* = 3.49), while the most recently-revised versions had an average of 3.21 (*SD* = 4.26), also a statistically significant difference according to a two-tailed t-test, $t(64) = 6.81$, $p < .001$.[4]

Likewise, writers who had the assistance of the system ended up with sentences with fewer Boring Words overall than the control (unassisted) condition. The ten-sentence exercises of Tree-Unassisted and Moon-Unassisted conditions had an average of 10.91 (*SD* = 4.76) and 6.93 (*SD* = 3.16) Boring Words, respectively. By contrast, the ten-sentence exercises of Tree-Assisted and Moon-Assisted conditions had an average of 3.90 (*SD* = 3.60) and 3.21 (*SD* = 4.26), respectively. This differences between assisted and unassisted conditions were statistically significant for tree conditions according to a two-tailed t-test, $t(87) = 7.68$, $p < .001$, and for moon conditions, $t(75) = 4.35$, $p < .001$).

Analyzing the unassisted conditions provide a way to check that Progym's sense of what counts as a Boring Noun for a particular word is sensible. Compared to the above-stated average of 10.91 Boring Words for the noun "tree" in the Tree-Unassisted condition, an average of 3.19 (*SD* = 2.32) Boring Words for the noun "moon" (i.e the "incorrect" words) were found, a statistically significant difference, $t(92) = 9.90$, $p < .001$). Likewise, compared to the above-stated average of 6.93 Boring Words for the noun "moon" in the Moon-Unassisted condition, an average of 4.09 (*SD* = 2.50) Boring Words for the noun "tree" were found, a statistically significant difference, $t(86) = 4.62$, $p < .001$. In other words, Progym's noun-specific lists of Boring Words mined from Project Gutenberg texts were predictive of the ways that participants in the user study wrote about these two particular nouns.

Qualities of Revision

To analyze the ways that participants wrote when confronted with Progym's criticism, for all sequential pairs of revisions $((s_0, s_1), (s_1, s_2)...)$ the Levenshtein distance in terms of tokens was calculated, with one outlier removed.[5] Figure 2 shows the distribution of the frequency of lengths of revisions produced by users in the Inspiration-Assisted condition. There were 323 revisions total, with an average of 4.31 revisions per participant (*SD* = 3.58). The majority of revisions were of an edit distance of 1.

blue	→	azure
face	→	visage
changing	→	periodic
limbs	→	appendices
surface	→	topography
bends	→	swoops
beautiful	→	sightly
beautiful	→	spellbinding

Table 2: Example Revisions Made Toward Rarer Word

What were the nature of these one-token changes? By encouraging writers to avoid common words, the system also pushed writers toward greater linguistic diversity. Those revisions were gathered in which the user's original sentence and first revision of this sentence were equal in number of tokens but differed by exactly one token—i.e. in which one token (w_0) was "replaced" by another (w_1). Out of the 108 w_0 tokens, there were only 64 unique ones. By contrast, there were 102 unique w_1 tokens, a statistically significant difference according to a chi-squared test, $\chi^2(1) = 35.62$, $p < .001$. In essence, the collection of "revised" words was more varied than the collection of "unrevised" words.

It was hypothesized that pressure from Progym may encourage writers to eschew common words, replacing them with rare ones. Google Ngram Viewer[6] provides a way of roughly testing whether one word is more common than the other. For each pair of sequential revisions that were equal in number of tokens but differed by one word, Ngram Viewer was used to check whether the word in the first sentence, w_n, or the word that replaced it, w_{n+1}, was the more frequent.[7] Out of 167 of such comparisons, w_{n+1} was the rarer word in 116 (69%), a statistically significant difference according to a chi-squared test, $\chi^2(1) = 25.30$, $p < .001$. The difference was a bit more extreme looking only at those revisions in which the first version of a sentence was equal in number of tokens to its "final" version but differed by one word; of these (w_{first},w_{last}) pairs, w_{last} was the rarer word 76% of the time (65 out of 87), a statistically significant difference, $\chi^2(1) = 21.25$, $p < .001$. This suggests that Progym inspired participants to use less-frequent words. Table 2 shows a sample of the single word revisions in which a word was substituted by a rarer one.

Progym V.2: Beyond the Word

The second version of Progym differs from the first in two respects. First, rather than focus on individual words, it encourages the users to turn away from too-common syntax. Second, rather than compare the writer to specific relations mined and distilled from a very large number of texts, it compares the writer to a relatively small number of exam-

[4] Analysis of the data revealed that participants did not always heed the study-task's exhortation that they write ten sentences in ten different text boxes; sometimes they wrote more than one sentence in a text box. To control for the length of users' writing, calculations in section result from analysis of the first actual sentence of each user's ten input texts, as determined by the SpaCy parser's sentence tokenization.

[5] Several of these pairs contained edit distances much greater than the average. Upon closer inspection, these can be explained by the fact that entire poems by poets such as Robert Frost were submitted, with the previous or sequential "revision" of that line being much shorter and in fact unrelated. Revisions of an edit distance greater than or equal to 150 were excluded.

[6] https://books.google.com/ngrams

[7] Datapoints for the year 2000, the default most recent year, were compared. Automatic spelling correction was applied using the PyEnchant library.

you can do	←	You can do anything you want to do, you just need to push yourself sometimes to get them done.
VB your NN	←	Focus your energy and you can make leaps and bounds
RB VB up	←	NEVER GIVE UP
you are JJ	←	You are smart and intelligent.
do n't VB	←	Don't give up. You'll be glad you didn't.
if you VBP	←	If you stop now, all the work you've put in thus far will have been for nothing.

Table 3: Rhetorical Stubs Used by Progym V.2 (Most Frequent in Corpus), with Examples

ples. Using the same interface as before, "inspiring" sentences were gathered from Amazon Mturk crowdworkers. These workers were told: "Imagine that you are writing for somebody who needs your words to help them accomplish a difficult task or overcome some adversity." In all, ten sentences each from 49 crowdworkers were collected.

These sentences became a small corpus of examples to which Progym would compare any new inspiring sentence, testing its novelty against them. The goal of this version of Progym is to push users away from the one-word edits typical of interactions with V.1 by focusing on longer syntactic units rather than individual words. It does so by comparing the syntax writers use to begin their inspiring sentences.

For each sentence in the example sentences, at most the first three tokens were either represented as this token's part-of-speech tag or, if this token was in a list of stop words[8], the token itself. For instance, the sentence "Focus your energy and you can make leaps and bounds" is represented as *(VB, your, NN)*. Figure 3 shows the most frequent stubs in this small corpus with examples. This technique of building abstract—but not totally unlexicalized—representations of text is inspired by the "stretchy patterns" described by Gianfortoni, Adamson, and Rosé (2011). Since the goal of this exercise was to target patterns that may be overused in specifically inspiring sentences (rather than sentences in general), the top 20 most frequently used of such patterns in an excerpt of the Gutenberg Corpus were excluded, leaving 308 in all (see Table 2) .

Progym V.2 asks users to generate inspiring sentences, testing how they begin against these banned "Rhetorical Stubs" found in the previously-gathered example sentences. When there is a match between the writer's sentence and one of the examples, Progym once again provides feedback like this: "The phrase 'You are ready' reminds me of other inspiring messages, like 'You are amazing and nothing can stop you.' Could you try making yours a little more creative?" Rhetorical Stubs are meant to strike a balance between the semantic openness of merely a part-of-speech take sequence and the specificity of the sequence of tokens themselves, drawing attention away from the choices of words toward the underlying structure of the sentence. In other words, while one may substitute the participle "running" with any number of words (e.g."sprinting," "hustling," and "galloping'), one may not so easily replace a closed-class word such as "you." The design choice of the "Rhetorical Stub" was made to stimulate revisions unlike those one-word revisions users made when interacting with V.1.

[8]Here the standard list in the Natural Language Toolkit (Bird, Klein, and Loper 2009) was used and supplemented with tokens to accommodate how the SpaCy parser tokenizes contractions (e.g. "'ll").

User Study 2

Amazon Mturk crowdworkers were tasked with writing ten inspiring sentences, either assisted by Progym (n = 35) or unassisted (n = 38).

Use of Rhetorical Stubs

Progym's function for identifying the use of banned Rhetorical Stubs was re-purposed for the analysis of the sentences written by the Mturk participants under two conditions, Inspiration-Assisted and Inspiration-Unassisted.

As writers revised according to Progym's V.2's feedback in the assisted conditions, they lessened the number of banned Rhetorical Stubs in their texts. Looking at the earliest version of sentences (i.e. before any revision based on Progym's suggestions), the poems of the assisted condition had an average of 3.69 banned Rhetorical Stubs (*SD* = 1.74); looking at the most recent (i.e. "final") version of sentences, they had an average of 1.26 (*SD* = 1.87), a statistically significant difference according to a two-tailed t-test, $t(68) = 5.55$, $p < .001$. Participants writing with assistance of Progym V.2 ended up with sentences with fewer banned Rhetorical Stubs compared to the control (unassisted) condition. The ten-sentence exercises of Inspiration-Unassisted and Inspiration-Assisted had an average of 4.47 (*SD* = 2.02) and 1.26 (*SD* = 1.87), respectively. This difference was statistically significant according to a two-tailed t-test, $t(71) = 6.94$, $p < .001$.

To test whether the Progym V.2's small number of Rhetorical Stubs were, as one would expect, a reasonable "training set," a comparison was made between the number of banned Rhetorical Stubs in the Inspiration-Unassisted condition and (as an example of non-inspirational sentences written under similar experimental conditions) the Unassisted Tree and Moon conditions from the previous user test. One would expect the banned Rhetorical Stubs generated from the example sentences have better "coverage" of additional inspiring sentences than uninspiring ones. (Otherwise, those Rhetorical Stubs may simply be characteristic of sentences generally produced by Mturk workers, no matter what the rhetorical or expressive purpose.) Indeed, compared to an average of 4.47 of those Rhetorical Stubs found in the Inspiration-Unassisted condition, there were an average of 1.41 (*SD* = 1.52) found in the collection of Moon and Tree-Unassisted conditions, a statistically significant difference according to a two-tailed t-test, $t(127) = 9.33$, $p < .001$. In this case, even a small number of example sentences were predictive of the

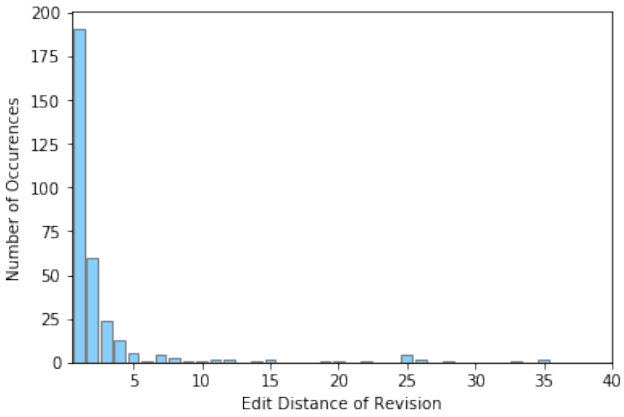

Figure 2: Revisions with Progym V.1 (Tree and Moon)

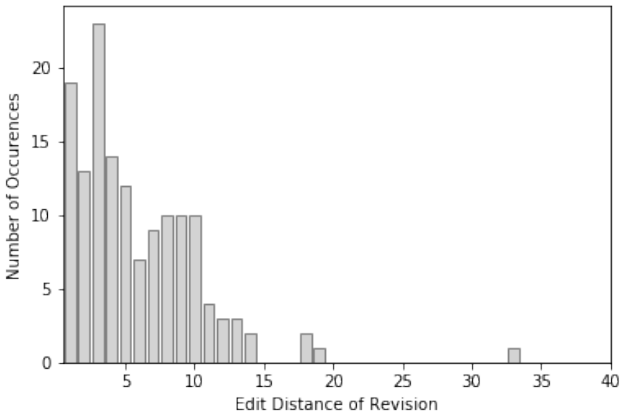

Figure 3: Revisions with Progym V.2 (Inspiring Sentences)

kinds of Rhetorical Stubs that would be written in other examples of inspiring sentences.

Coverage Due to Delexicalization Making Rhetorical Stubs is more computationally complex than simply using, for instance, the first three tokens from the example inspiring sentences in the training data (what might be called "Simple Stubs" [n = 402]). However, because they are more "general" (i.e. mostly delexicalized), the Rhetorical Stubs made out of these had better coverage over the data. Compared to a per-poem average of 4.47 of those Rhetorical Stubs found in the Inspiration-Unassisted condition, there were only 2.47 Simple Stubs (*SD* = 2.02), a statistically significant difference according to a two-tailed t-test, $t(74) = 4.25$, $p < .001$. Delexicalizing was thus an effective way to "stretch" data.

Qualities of Revision

Once again, all sequential pairs of revisions $((s_0, s_1), (s_1, s_2)...)$ were analyzed for the edit-distance (in terms of tokens) between the two. Figure 3 shows the distribution of the frequency of lengths of revisions produced by users in the Inspiration-Assisted condition (as in the calculations for V.1, with several outliers removed). Comparing this to the frequency of lengths of revisions produced by users interaction with V.1, which were mostly a single token in length, these revisions show a tendency toward revisions of multiple tokens. There were 143 revisions total, with an average of 4.09 per participant (*SD* = 2.89). The average edit distance of the revisions created by participants using V.1 was 3.00 (*SD* = 5.35, median = 1), while the average edit distance of the revisions created by participants using V.2 was 5.87 (*SD* = 4.52, median = 5), a statistically significant difference according to a two-tailed t-test, $t(464) = 5.57$, $p < .001$. Moreover, as can be seen by comparing and Figure 2 and Figure 3, the lengths of revision completed with V.2 are more diverse. While for V.2 the top revision length was indeed 3 (reflecting the fact that the prompt drew attention to a Rhetorical Stub made from three tokens), revisions were more likely to be other lengths than revisions made with V.1 were likely to be lengths other than 1. This diversity can be described statistically: the entropy of the revisions performed with V.2 (n = 143) was 2.55 bits. By contrast, the entropy of a random sample of the same number of revisions performed with V.1 was 1.34 bits, this lower entropy signalling less diversity in the revision lengths.

Like Progym V.1, V.2 seemed to encourage linguistic diversity. For each sequential pair of revisions, the first or "unrevised" Rhetorical Stub (rs_0) and the subsequent revision (rs_1) were gathered. Out of the 93 rs_0 patterns, there were only 56 unique ones. By contrast, there were 85 unique rs_1 patterns, a statistically significant difference according to a chi-squared test, $\chi^2(1) = 22.98$, $p < .001$. The collection of "revised" Rhetorical Stubs was more diverse than the collection of "unrevised" ones. By putting pressure on writers to avoid certain common Rhetorical Stubs, Progym nudged them toward linguistic variation.

There was no evidence that revisions using V.2 led to an increase in the rarity of words within a text, though the consideration of this was limited by the small number of (w_n, w_{n+1}) word pairs (n = 15). Of these, w_{n+1} was the rarer word in 9 of them—not a statistically significant difference, $\chi^2(1) = 0.60$, $p > .05$.

Another Pattern of Revision For all sequences of revision of at least length 2 (i.e. in which the writer revised a sentence once and then revised again, n = 37), were gathered, and the first, second, and last (final) versions of these sentences were compared. In 6 of these, the writer first changed the sentence such that one of the first three tokens was different but it still matched the same "forbidden" Rhetorical Stub as the original sentence before ultimately revising the sentence more dramatically in a way that manifested a different Rhetorical Stub. For instance:

- You are *enough* just as you are.
- You are *perfect* just as you are.
- your attitude determines your direction [sic]

In such cases, it seems that the flexibility of the Rhetorical Stub has pushed the writer beyond simply swapping out a word with another related word of the same part of speech.

Discussion

Two versions of Progym were tested. Each version effectively steered writers away from certain linguistic elements that the system desired them to avoid. The two versions of Progym led to different styles of revision: participants writing with V.1 produced mostly single-word changes, shifting a common word to a rare one. Those writing with V.2 engaged in more extensive revision in terms of the number of tokens changed. Both small and large corpora of examples were useful for creating a background of "expected" language against which writers were asked to depart and encouraging linguistic diversity. This study was limited in the sense that it focused on only on several conditions (the Tree vs. Moon conditions, and the Inspiring conditions). Future research could explore a wider set of each of these.

Conclusions

This paper's title begins with the word "toward" in order to make clear that its goal is to test the validity of a path. The main conclusion to be drawn from it is that even relatively simple techniques for predicting what users will write can be used to steer them away from these predictable moves and encourage linguistic diversity as well as different techniques of revision. One may imagine, further down the path, a wide variety of digital *progymnasmata* that would train writers to spurn mundane formulations or vary their styles.

Future versions of digital progymnastic systems could no doubt make use of more complex computation to determine whether a writer is veering into some too-common pattern or formulation. For instance, one might use a more complex statistical approach to identify clichés (Smith, Zee, and Uitdenbogerd 2012) or make use of statistical models of character types (Bamman, O'Connor, and Smith 2013) to detect when users are falling into common tropes. Likewise, a more complicated interface could allow the writer to have more control over the system—for instance, by specifying that they want to practice avoiding familiar syntactic construction or words, by adjusting the level of difficulty, or by specifying certain discourses that they want to depart from (e.g. the syntactic constructions of Romantic poetry in particular). A larger problem is how to address the fact that writers may use "boring" words or syntactic constructions in nonetheless interesting ways. For instance, while to write that "the moon is white," may be overly expected, to write that "the moon is white like your Toyota Prius" may seem less so. Likewise, a sentence may use expected words organized in rhetorically powerful ways; a more complete system would keep an eye out for figures such as *anaphora* or *chiasmus* (Dubremetz and Nivre 2015). However, just as simple systems of text generations can serve as a baseline for more complex systems (Montfort and Fedorova 2012), it is useful to explore a pair of relatively straightforward techniques for steering writers away from "predictable" language to which more complex ones may be later compared.

This paper has focused on the way that Progym "mediated" (Vygotsky 1980) writers' writing process. However, while crowdsourcing interactions with the system allowed for statistical analysis of these interactions, this research could be complemented by a more naturalistic and qualitative study of student or professional writers using this system. Further research into this and other literary interfaces could and should explore how they could be taken up in particular educational contexts over longer time-scales of literacy (Lemke 2000). One might reasonably wonder whether training with such tools over periods of time has effects on one's mode of composition the same way that attending a spin class every week has effects on one's body. Further research could also focus more closely on the perception of overall "creativity"—whether writers feel as though the system makes them more creative, and whether readers perceive texts written with this system as more creative.

Unexplored too are the political and ideological potentials of this kind of progymnastic exercise. Researchers have begun both to critique and attempt to reverse the biases (especially gender and racial biases) in large data sets and the models trained upon them (Bolukbasi et al. 2016). One could imagine a kind of progymnasmata that targets overly familiar and biased ways of talking about male or female characters, for instance, and encouraging the writer's departure from stereotypical use of language (such as a tedious insistence that a queen be "fair"; see again Table 1).

Work in computational creativity has focused on how to make creative writing more pleasant, less cognitively and psychologically taxing (Kantosalo et al. 2014; Gonçalves et al. 2017; Gonçalves and Campos 2018). Progym clearly aims to make the task of writing harder rather than easier. Future research could also consider the psychological aspects of users' interactions with intentionally-critical progymnasmatic systems and could consider techniques of gamification to motivate writers to engage with them.

Acknowledgements

Thanks to the anonymous reviewers for their thoughtful suggestions.

References

Bamman, D.; O'Connor, B.; and Smith, N. A. 2013. Learning latent personas of film characters. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, volume 1, 352–361.

Bird, S.; Klein, E.; and Loper, E. 2009. *Natural language processing with Python: analyzing text with the natural language toolkit*. " O'Reilly Media, Inc.".

Bolukbasi, T.; Chang, K.-W.; Zou, J. Y.; Saligrama, V.; and Kalai, A. T. 2016. Man is to computer programmer as woman is to homemaker? debiasing word embeddings. In *Advances in neural information processing systems*, 4349–4357.

Dubremetz, M., and Nivre, J. 2015. Rhetorical figure detection: The case of chiasmus. In *Proceedings of the Fourth Workshop on Computational Linguistics for Literature*, 23–31.

Erasmus, D. 1512 1978. *De Utraque Verborum ac Rerum Copia*. Marquette University Press.

Gabriel, R. P.; Chen, J.; and Nichols, J. 2015. Inkwell: A creative writer's creative assistant. In *Proceedings of the 2015 ACM SIGCHI Conference on Creativity and Cognition*, 93–102. ACM.

Gianfortoni, P.; Adamson, D.; and Rosé, C. P. 2011. Modeling of stylistic variation in social media with stretchy patterns. In *Proceedings of the First Workshop on Algorithms and Resources for Modelling of Dialects and Language Varieties*, 49–59. Association for Computational Linguistics.

Gonçalves, F., and Campos, P. 2018. Enhancing your mental well-being and creativity while writing: A crowdsource-based approach. In *IFIP Working Conference on Human Work Interaction Design*, 17–35. Springer.

Gonçalves, F.; Caraban, A.; Karapanos, E.; and Campos, P. 2017. What shall i write next?: Subliminal and supraliminal priming as triggers for creative writing. In *Proceedings of the European Conference on Cognitive Ergonomics 2017*, 77–84. ACM.

Hearst, M. A. 1992. Automatic acquisition of hyponyms from large text corpora. In *Proceedings of the 14th conference on Computational linguistics-Volume 2*, 539–545. Association for Computational Linguistics.

Hoey, M. 2007. Lexical priming and literary creativity. *Text, discourse and corpora: Theory and analysis* 729.

Honnibal, M., and Johnson, M. 2015. An improved non-monotonic transition system for dependency parsing. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, 1373–1378. Lisbon, Portugal: Association for Computational Linguistics.

Indurkhya, B. 2016. On the role of computers in creativity-support systems. In *Knowledge, Information and Creativity Support Systems: Recent Trends, Advances and Solutions*. Springer. 213–227.

Kantosalo, A.; Toivanen, J. M.; Xiao, P.; and Toivonen, H. 2014. From isolation to involvement: Adapting machine creativity software to support human-computer co-creation. In *ICCC*, 1–7.

Kantosalo, A. A.; Toivanen, J. M.; and Toivonen, H. T. T. 2015. Interaction evaluation for human-computer co-creativity. In *Proceedings of the Sixth International Conference on Computational Creativity*. Brigham Young University.

Kennedy, G. A. 2003. *Progymnasmata: Greek textbooks of Prose Composition and Rhetoric*. Brill.

Lemke, J. L. 2000. Across the scales of time: Artifacts, activities, and meanings in ecosocial systems. *Mind, Culture, and Activity* 7(4):273–290.

Lubart, T. 2005. How can computers be partners in the creative process: classification and commentary on the special issue. *International Journal of Human-Computer Studies* 63(4-5):365–369.

Manjavacas, E.; Karsdorp, F.; Burtenshaw, B.; and Kestemont, M. 2017. Synthetic literature: Writing science fiction in a co-creative process. In *Proceedings of the Workshop on Computational Creativity in Natural Language Generation (CC-NLG 2017)*, 29–37.

Miller, G. A. 1995. Wordnet: a lexical database for english. *Communications of the ACM* 38(11):39–41.

Montfort, N., and Fedorova, N. 2012. Small-scale systems and computational creativity. In *International Conference on Computational Creativity*, 82.

Oliveira, H. G.; Mendes, T.; and Boavida, A. 2017. Co-poetryme: a co-creative interface for the composition of poetry. In *Proceedings of the 10th International Conference on Natural Language Generation*, 70–71.

Oliveira, H. G. 2012. Poetryme: a versatile platform for poetry generation. *Computational Creativity, Concept Invention, and General Intelligence* 1:21.

Riedl, M. O., and O'Neill, B. 2009. Computer as audience: A strategy for artificial intelligence support of human creativity. In *Proc. CHI Workshop of Computational Creativity Support*.

Roemmele, M., and Gordon, A. S. 2015. Creative help: a story writing assistant. In *International Conference on Interactive Digital Storytelling*, 81–92. Springer.

Roemmele, M. 2016. Writing stories with help from recurrent neural networks. In *Thirtieth AAAI Conference on Artificial Intelligence*.

Smith, A. G.; Zee, C. X.; and Uitdenbogerd, A. L. 2012. In your eyes: Identifying clichés in song lyrics. In *Proceedings of the Australasian Language Technology Association Workshop 2012*, 88–96.

Swanson, R., and Gordon, A. S. 2008. Say anything: A massively collaborative open domain story writing companion. In *Joint International Conference on Interactive Digital Storytelling*, 32–40. Springer.

Toivanen, J.; Toivonen, H.; Valitutti, A.; Gross, O.; et al. 2012. Corpus-based generation of content and form in poetry. In *Proceedings of the third international conference on computational creativity*. University College Dublin.

Veale, T., and Hao, Y. 2007. Comprehending and generating apt metaphors: a web-driven, case-based approach to figurative language. In *AAAI*, volume 2007, 1471–1476.

Veale, T., and Hao, Y. 2011. Exploiting readymades in linguistic creativity: A system demonstration of the jigsaw bard. In *Proceedings of the 49th annual meeting of the association for computational linguistics: Human language technologies: Systems demonstrations*, 14–19. Association for Computational Linguistics.

Veale, T. 2013. Linguistic readymades and creative reuse. *Journal of Integrated Design and Process Science* 17(4):37–51.

Vygotsky, L. S. 1980. *Mind in society: The development of higher psychological processes*. Harvard university press.

Webb, R. 2001. The progymnasmata as practice. *Education in Greek and Roman Antiquity* 289–316.