

Teaching natural language to computers

Joseph Corneli

Department of Computing
Goldsmiths College, University of London
j.corneli@gold.ac.uk

Miriam Corneli

Former Senior English Language Fellow
(Nepal / Sri Lanka), US Department of State
languagecoach.miriam@gmail.com

Abstract

“Natural Language,” whether spoken and attended to by humans, or processed and generated by computers, requires networked structures that reflect creative processes in semantic, syntactic, phonetic, linguistic, social, emotional, and cultural modules. Being able to produce novel and useful behavior following repeated practice gets to the root of both artificial intelligence and human language. This paper investigates the modalities involved in language-like applications that computers – and programmers – engage with, and aims to fine tune the questions we ask to better account for context, self-awareness, and embodiment.

1 Introduction

In the genesis of intelligent computational systems, one often observes programs that write before they can read, compose before they can listen, and paint before they can see. However, the most successful systems in poetry, music, and visual art are indeed perceptually aware – and derive significant benefits from that ability [Kurzweil, 1990; Screene and Wiggins, 2015; Colton *et al.*, 2015b].

We will argue that the bar by which we judge computational creativity in text generation – and appreciation – can and must be raised, in order to build systems that we can meaningfully communicate with. Our paper takes the form of a necessarily high-level sketch, supported by hand-crafted examples that draw on both standard and custom software. Most of these examples concern computer poetry, but our aim is not to present a technical or aesthetic achievement. Rather, we use the examples to survey the limitations of current systems and to indicate some potentially novel approaches.

Specifically, we highlight three interrelated thematic areas that we think will repay effort.

(Limited) Contextual Understanding Here, we are concerned with what makes a response to some circumstance meaningful. This certainly requires *context* [Ogden and Richards, 1923]. For example, a response in a conversational dialog generally considers the previous elements of the exchange, and perhaps also previous exchanges, and elements

of a shared culture. This understanding does not need to be, nor in general can it be, “complete.” Furthermore, in many cases, the reader or listener will hear meanings that were not there originally [Veale, 2015] – however, the interpolation of meaning by the reader cannot always be relied upon. Later in this paper we include an example of a computer-generated poem that is essentially just verbose babbling fitted to a predefined template. This poem was previously reviewed by a published poet, and it does not stand up well to critical scrutiny [Corneli *et al.*, 2015]. The poem misses any sense of “why” – and the program that generated this poem would not be able to offer what the Provençal poets called a *razo* [Agamben, 1995, p. 51], that is, an exposition of the poem.

(Limited) Self-awareness Here, we focus exclusively on the ability to reason about creativity as a process. Computer programs often have limited metadata about their software processes, for example the “signatures” of functions, specifying the types of input data that the functions will accept, or “contracts” specifying preconditions, postconditions, and invariants. Along with these representations often comes some limited ability to reason formally about code. But few if any contemporary software systems would be able to convincingly make sense of a simple writing prompt, or adapt a dialogical process of response in order to reach an agreement, or respond to feedback from a critic in order to produce a better poem. In the future, such computational abilities with language may be commonplace. These abilities may depend on fairly profound epistemic features, for example, the computer might need to recognize when its “knowledge” is uncertain, and proceed accordingly – perhaps asking for help, or making multiple generative attempts in parallel and assessing which one works better relative to its contextual understanding.

(Limited) Embodiment We will consider program flowcharts [Charnley *et al.*, 2014] as the primary framework with which to describe the computer’s “process” layer. Whether embodied as a flowchart or a Von Neumann machine or something else, computational processes are also physical processes. The manipulations of the nodes and arrows of a flowchart or of some other collection of physical objects, like the flowchart’s corresponding script, or the words of a poem, can (potentially) be thought about with

respect to its gestural content. This definition of *gesture* due to the 12th Century theologian Hugues de Saint-Victor, quoted in [Mazzola, 2016], shows the connection with our other themes: *Gestus est motus et figuratio membrorum corporis, ad omnem agendi et habendi modum*. Mazzola glosses this as follows:

Gesture is the movement and figuration of the body's limbs with an aim, but also according to the measure and modality proper to the achievement of all action and attitude.

Introducing this quote requires us to make one significant caveat. Whereas humans tend to perceive ourselves as relatively free beings, able to act according to a purpose and even to choose which purpose to serve, we regard computers in a very different light. At best, a computer can be programmed to optimize its behavior relative to some constraint. This perspective does not sit well with the typical understanding of the English word “aim” – although it would appear that Mazzola freely introduced this concept, instead of sticking with the more literal “agenda.” In any event, the point to make here is that a *computational system* is understood relative to an operating environment, and its behavior is worked out relative to that environment. Under some circumstances we would call this process “programming,” and under somewhat different circumstances we would describe it as “self-programming” or “automatic programming.” In short, it is not necessary to attribute intention to the computer, but – once again – it is necessary to think about its behavior in context.

The remainder of the paper is structured as follows: First, we will explore these themes from a computing standpoint, developing a technical sketch rather than a formal system description. Then we turn to a discussion that evaluates this technical sketching from the point of view of the second author, an English as a Foreign Language teacher with a prior background in consciousness studies. Finally, we draw on this discussion to outline a plan for future computational experiments centered on making sense of language.

2 Motivation

Oscar Schwartz offers the following framing:

Can a computer write poetry? This is a provocative question. You think about it for a minute, and you suddenly have a bunch of other questions like: What is a computer? What is poetry? What is creativity? [Schwartz, 2015]

Computer poetry may sound like a bit of a lark – after all, it's not clear that anyone really needs it. Nevertheless, asking these questions about poetry begins to suggest a way of working with language that has wider implications.

Consider Turing's idea that machines “would be able to converse with each other to sharpen their wits” [Turing, 1951]. This could be realized as a *Q&A site specifically for computers*. The discussions could address all manner of practical concerns, for example, those arising for bots that are engaged in code or editorial review tasks. A reputation system and web of trust could be used to maintain quality control. If

the participating computational systems had sufficient abilities with natural language, this system could be bridged into a Q&A site that is in everyday use by human beings. However, before a computational system would be useful in any Q&A context, it would presumably need to be able to be able to model the meanings of the questions that are being asked reasonably well, and also to be able to compose meaningful responses. For now, we will side-step the Chinese Room-style question of whether the system “really” understands what it is processing [Bishop, 2004] and focus on the more applied question: how would meaningful responses occur? A high-level outline could be something like this [Corneli *et al.*, 2015]: (1) Read and understand the “prompt” to a sufficient degree; (2) Compose a response that “makes sense”; (3) Criticize the response along various dimensions, for instance, does it read well, does it tell a story or develop a character?; (4) Consider how it might be improved. This outline is based on an established process that groups of creative writers use to critique and revise poetry or literary works for publication.

When we turn to computer generated text – say, poetry, to be concrete – in addition to examining the generated poem, a sophisticated audience can also examine the *process* whereby the poem was generated, and read the product against this process (or vice versa). Indeed, the computer can create both poem and process, the later via automatic programming – and offer its own assessments of both as well, as long as it can make sense of the success criteria.

2.1 Related work

Natural language processing often begins with a grammar. If none is available, it may be induced, for instance by using compression techniques [Wolff, 1988]. Both in older [Redington and Chater, 1997] and quite recent work [Hermann *et al.*, 2015], statistical and neural network approaches to corpus-based language understanding have shown strong potential for developing “reasoned” ways of thinking about linguistic structure without the usual grammatical assumptions. Corpus methods help to understand the patterns in the way people use language, and the creative potential of unexpected word combinations [Hoey, 2005].

One example cited by Hoey is Tennyson's famous line *Theirs is not to reason why*. Here, the word *reason* is used with its verb sense, rather than with the noun sense that most readers would expect based on their prior experience with the two-word phrase *reason why*. This unique feature makes the line memorable and interesting. The psycholinguistic properties of the broader phenomenon of “lexical priming” have been extensively studied [Pace-Sigge, 2013]. One empirical result is that priming works differently for native speakers and for non-native ESL speakers, insofar as native speakers are more affected by binding of words within formulas, whereas someone learning a new language tends to only recognize the strings that they have encountered before.

The ascendant status of data-driven methods in natural language processing does not obviate symbolic AI, which continues to be useful for work with specialist languages. For computer programming languages in particular, techniques for *reflection* and, in the case of LISP, *homoiconicity* (i.e., treating code as data) make it possible to write computer pro-

grams that reason about, write, or adapt computer programs.

Artificial languages have been used in video games in creative ways, but not, as yet, for functional communication with or between non-player characters. Multi-agent systems have, however, been used in poetry generation. One example system creates poems based on repeated words and sounds driven by a model of agents' emotional states [Kirke and Miranda, 2013], inspired by earlier work in music [Kirke, 2011]. As for computer programs that read poetry, this is typically limited to reading (and mimicking) surface style, without extending to *meaning* [Carlsisle, 2000] – even if some readers were fooled. Without knowing what's in a poem, it seems difficult to be other than superficial.

Just how far the “surface” goes is a question much discussed by poets and translators of poetry. Red Pine, translator of the 13th-14th Century Chinese poet and Buddhist monk Stonehouse (石屋) wrote as follows [Pine, 2014, p. xxiv]:

I don't know how others do it, but when I've tried to think of a metaphor for what I go through, I keep coming up with the image of a dance. [...] I try to get close enough to feel the poet's rhythm, not only the rhythm of the words but also the rhythm of the poet's heart.

A typical approach to poetry generation might take Stonehouse's corpus and notice that he often writes about clouds and mountains and plants, and attempt to generate a poem “in the style of Stonehouse,” referencing some of these typical concepts and aiming to get the number of syllables and the grammatical structure right. However, there is little doubt that a reader with an ear for Chinese poetry or some familiarity with the ideas of Chan Buddhism would recognize these *ersatz* attempts for what they are: “dead words” [Pine, 2014, p. xxiii]. More interesting computer poems in a somewhat related genre (Japanese *haiku*) have been created by programmers working from the premise [Rzepka and Araki, 2015, p. 2497] that a reader's interest in a haiku stems from:

feeling that the poet understands a situation and that we can mentally agree with what she/he (or maybe *it*) shares with us.

The challenges posed by computer poetry serve as a point of departure. “Poetry exercises are used to allow learners to explore the complexities of English” [Parker, 2010] – or another language – and the contexts in which it is meaningful.

3 Exploration

Here are 10 short examples of writing prompts excerpted from the book “642 Things to Write About” [San Francisco Writers' Grotto, 2012]:

1. What can happen in a second.
2. The worst Thanksgiving dish you ever had.
3. A houseplant is dying. Tell it why it needs to live.
4. Tell a story that begins with a ransom note.
5. Write a recipe for disaster.
6. If you had one week to live. . . .
7. What your desk thinks about at night.

8. The one thing you are most ashamed of. . .
9. Describe your best friend.
10. Describe Heaven.

In order to create a computer-generated response to any of these prompts, in addition to understanding what the prompt is saying, some further understanding of the topic is required. The response itself will have various standard features. Many of the following features of *stories* are found (with minor adaptations) in poetry, and other kinds of writing:

A story is not a modular presentation of ideas but a multi-layered work consisting of interdependent characters, plot elements, and settings. [Kim *et al.*, 2014]

Let us consider, then, a simple theory of stories and storytelling, using the prompts above as our domain. One suitable theory would involve a micro-world containing: *A Scenario*, *A Narrator*, *An Audience*, *A Beginning*, *A Middle*, and *An End*. Consider that – with respect to the “Thanksgiving dish” prompt – the computer has presumably never tasted food of any kind. It could, however, “imagine” a scenario in which there is a character who eats a Thanksgiving dish.

3.1 An example Scenario

The scenario could be represented with various relations:

```
Squanto memberOf Patuxet tribe, Patuxet tribe
hasCardinality 1, Thanksgiving isa event,
Thanksgiving hasHost Pilgrims, Thanksgiving
hasGuest Squanto, Thanksgiving hasFood eel,
eel hasCondition burnt
```

Naturally, this might be extended with further information as that comes to light; and in practice we might use a more robust formalism.

3.2 The other components

The *Narrator* would walk through the scenario and say what's there. On a metaphorical level, the narrator's role is somewhat similar to the way a virtual camera moves through a 3D simulation in order to create a film. However, the *Narrator* needs to consider the *Audience* in order to be effective. As indicated in the quote from Kim *et al.*, above, the data in the *Scenario* needs to be structured when telling the story.

Here is one possible presentation of the scenario, embellished with some facts, fictions, and local color, and combined into several sentences that flow reasonably well.

Squanto was the last surviving member of the Patuxet tribe. He attended the first Thanksgiving with the Pilgrims wearing a new buckskin jacket. One of the foods that was served was eel, but it had been rather badly burnt and Squanto didn't find it to his liking.

This text manages to include a range of emotionally evocative, thought provoking, character establishing, and sensory language (“last surviving member,” “first Thanksgiving,” “new buckskin jacket,” “rather badly burnt,” “didn't find it to his liking”) – and at least one unexpected word combination (“Thanksgiving. . . eel”). It also has a discernible *Beginning*, *Middle*, and *End*. It seems appropriate to call it a *story*.

3.3 How to come up with stories?

Let’s start with a parse:

```
(TOP
 (NP
  (NP (DT The) (JJS worst)
       (NNP Thanksgiving) (NN dish))
  (SBAR (S (NP (PRP you)) (ADVP (RB ever))
          (VP (VBD had)))) (. .)))
```

Here are some associated word meanings from WordNet:¹

| Word | Gloss from WordNet |
|--------------------|--|
| (DT The) | determiner |
| (JJS worst) | (superlative of ‘bad’) most wanting in quality or value or condition |
| (NNP Thanksgiving) | commemorates a feast held in 1621 by the Pilgrims and the Wampanoag |
| (NN dish) | a particular item of prepared food |
| (PRP you) | pronoun |
| (RB ever) | at any time |
| (VBD had) | serve oneself to, or consume regularly |

There are some other interesting word senses available, and knowing which one to pick, or how (and how much) to combine various senses seems like a bit of an art form. Should we consider a *short prayer of thanks before a meal* when thinking of “Thanksgiving”? For now, we will say a short prayer and tentatively assume that a (NN dish) is what you eat, rather than what you eat off of. “Lexical priming” [Hoey, 2005] techniques would help make the relevant distinction here.

But supposing we get this far, now what? We’ve moved from one sentence to several quasi-sentences, without getting that much closer to a “*Scenario*” like the one envisaged above. One possibility is that the WordNet expansion could be sufficient give us relevant keywords and phrases, from which a small corpus could be built (e.g., by doing a web search for the glosses) and then mined to learn relations between the items in that corpus. Alternatively or additionally, these meanings might be connected to a pre-computed model of linguistic meaning drawing from a much larger background corpus [McGregor *et al.*, 2016].

Mining significant associations from large scale text corpora is something people have explored in various ways. Finding subject-verb-object triples, in particular, is a popular method: one well-known algorithm is presented by Rusu *et al.* [2007]. This is sometimes called “building a semantic model.” Other more sophisticated approaches might draw on associations with a pre-existing ontology [Kiryakov *et al.*, 2004] – the particular benefit of the Rusu *et al.* approach is that it can be implemented using a simple parsing-based method. The basic theme of building semantic models of text goes back to Quillian [1969] – about whom there will be more to say later. For now, we just remark that in addition to expanding the writing prompt, we may also want to draw on some “stock” associations stored in a background knowledge base. We illustrate the method from Rusu *et al.* by applying it to the beginning of the novel *Frankenstein* (Table 1).

You will rejoice to hear that no *disaster* has accompanied the commencement of an enterprise which you have *regarded* with such evil *forebodings*. . . [Shelley, 1831] (emphasis added)

| | | | |
|----|------------------|-------------------|--------------------|
| 1. | <i>disaster</i> | <i>regarded</i> | <i>forebodings</i> |
| 2. | <i>yesterday</i> | <i>increasing</i> | <i>confidence</i> |
| 3. | <i>London</i> | <i>fills</i> | <i>delight</i> |
| 4. | <i>feeling</i> | <i>understand</i> | <i>feeling</i> |
| 5. | <i>breeze</i> | <i>gives</i> | <i>foretaste</i> |

Table 1: 5 triples extracted from *Frankenstein*

As far as text understanding goes, the result in Table 1 is not particularly encouraging. However, even a low-fidelity database of background knowledge would allow us to extend the story. Perhaps Squanto would decide that the burnt eel *gives* a *foretaste* of things to come. To make this association, we might use methods similar to the ones used to reason about ConceptNet triples – which have been employed to good effect in text and concept generation within the FloWr framework [Llano *et al.*, 2016], which is described below.

3.4 Can FloWr flowcharts be used to solve the composition problem?

FloWr [Charnley *et al.*, 2014] is a flowcharting system with basic text processing abilities. Metadata describing the “ProcessNodes” from which FloWr’s flowcharts are formed can be used to pose and solve simple automatic programming problems. A writing prompt like “worst Thanksgiving dish” can be interpreted as a constraint – or, more broadly, a fitness function – that steers the generative process, and trickles through to guide the choice of functional components and, eventually, words. By “fitness function,” we understand that text may be composed iteratively, and improved along the way, relative to some context of evaluation. The simple examples in Figure 1 illustrate problems that can be solved quite straightforwardly:

“Give me a list of mixed *adjectives* and *nouns*.”

(the italicized terms are “independent” variables that tell the WordSenseCategoriser node how to behave); or

“Give me a list of 5 rhyming couplets built of text from The Guardian and Twitter mentioning ‘eels’.”

It may make sense to include considerably more abstraction in the description of larger and more complex flowcharts. For example, a flowchart discussed by Corneli *et al.* [2015] includes 28 nodes and generates the following poem (and others that are similar):

Oh dog the mysterious demon
Why do you feel startle of attention?
Oh demon the lonely encounter
ghostly elusive ruler
Oh encounter the horrible glimpse
helpless introspective consciousness

¹NB., WordNet contains no entries for determiners or pronouns.

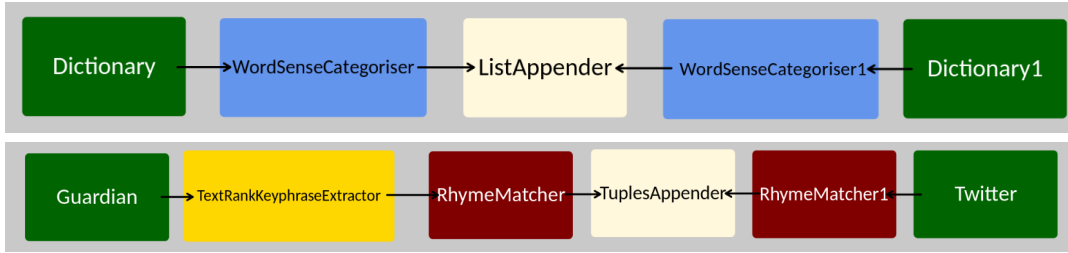


Figure 1: Two simple FloWr flowcharts

| | | |
|-------------------------------|----------------------------|--|
| number N | Dictionary | list of words of length N |
| list of words, word sense | WordSenseCategoriser | list extracting words with the word sense |
| input string(s) | TextRankKeyphraseExtractor | list of key phrases extracted from the strings |
| phrases, number M of phonemes | RhymeMatcher | tuple of couplets with M rhyming phonemes |
| number N, word | Twitter | some N tweets containing the word |
| tuples | TuplesAppender | a list combining the tuples |
| lists | ListAppender | a list combining the lists |

Table 2: Triples describe the functional mapping from input to output for selected FloWr nodes

Would the most succinct description of the flowchart be an approximately 28 clause sentence that is *equivalent* to the flowchart? Perhaps, in the current case, everything can be compressed down to the following template (fixed for this flowchart), and potentially further with optimizations:

Oh THEME the COLLOCATE SIMILAR
Why do you feel INVERSION of DESIRED?
Oh SIMILAR the COLLOCATE-OF-SIMILAR SIMILAR-TO-SIMILAR
SIMILAR-TO-COLLOCATE SIMILAR-TO-COLLOCATE' COLLOCATE-OF-SIMILAR
Oh SIMILAR-TO-SIMILAR' the COLLOCATE-OF-SIMILAR-TO-SIMILAR SIMILAR-TO-SIMILAR-TO-SIMILAR
SIMILAR-TO-COLLOCATE SIMILAR-TO-COLLOCATE COLLOCATE-OF-COLLOCATE

The connection between this template, its instantiation, and the putative prompt, “Write a poem about an old dog who is afraid of attention,” is tenuous at best. Nevertheless, reasonable hope exists for future work that would generate models – and a *Narrator* – tailor-made to a given prompt.

FloWr’s ProcessNodes define a micro-language denoting the available ways in which the system can transform input data to output data (Table 2). In a meaningful expansion of a given prompt, many choices would have to be made. New flowcharts built in response to writing prompts or other contextual data would constitute the system’s core “learnings.” In short, poetry and process need to be thought about together.

4 Discussion

In this section, we will briefly review human language learning from a second language teaching perspective, and then draw comparisons with the foregoing description of a hypothetical computer language learner.

First, why are primary language learning and second/foreign language learning often considered separately? One of the biggest differences between the two cases is that after learning a first language, “neural pathways” have been set

down, so that second language information has to be encoded “along with” or “beside” the first language pathways. New neural connections have to be formed to maintain the memory of the second language, whereas the first language has been implanted quite thoroughly. So when learning verbs, nouns, etc., the person (child, adult) learns or has learned, for instance, “*stand up!*” first and then that “*levantate!*” means a similar thing, in Spanish, or that “*chair*” = “*silla.*” As they progress in the second language, what we find is a process that linguists call “interlanguage” [Selinker, 1972], or a language that is not quite English, and not totally Spanish yet, because of the variations in the two languages in structure, phonemes, morphemes, allophones, semantics, and so on.

Language is also very multi-modal, and if we think again in terms of neurons and brain pathways, the word “dog” not only brings up images of a furry canine animal but all kinds of other associations. Pet dogs. Large and small dogs. Dogs with different color fur (visual cortex). *Dog* as a verb (motor cortex associations). The smell of wet dogs (olfactory bulb). Fear of dogs (limbic system). Favorite pet dogs you have loved in your life (emotions). The spelling or sound of the word D-O-G as opposed to D-A-W-G (auditory and visual cortex). If you are from certain countries perhaps even dog meat. Imagining a “demon dog” will invoke networks running all over the brain [Schlegel *et al.*, 2013]. Now what about computers, do they have a similar symbolic or multi-modal operating ability? People, as they become literate, learn to write and sound out letters at the same time – and the foundations of first language learning draw heavily on a sensory-motor channel [Iverson and Thelen, 1999; Hernandez, 2013].

Computers may be said to have a first language with several dialects, highly constrained by grammar – namely, byte-code and programming languages. The flowcharts described above begin to recover a degree of multi-modality, and a process orientation that is similar in certain partial respects to sensory-motor experience. We might also think of flowcharts

as akin to neurons and cortexes, as above. After all, there is a kind of embodiment even in the brain – it is an active and evolving organ [Doidge, 2007]. Flowcharts are rather different from classical neural network models, but one common feature is that they would need to be “trained” if they are to understand and express language.

Again, poetry could be part of the way forward. Exercises from a book like *Writing Simple Poems: Pattern Poetry for Language Acquisition* [Holmes and Moulton, 2001] might be used to teach computers as well as humans. Note that grammar and poetry are very different, and perhaps complementary. Thus, for example, a “Learn English!” notebook found in Japan with *Subject – Verb – Direct Object – Indirect Object – Prepositional Phrase – Adjectival Phrase* written at the top of each page offers a useful rubric for Japanese students, since the verb comes at the end of the sentence in Japanese (SOV). But grammar will only get you so far. Consider Chomsky’s famous nonsense statement, “Colorless green ideas sleep furiously.” A poetic gesture – like prefixing that phrase with a description of “planted tulip bulbs” – is able to make some sense out of nonsense by adding context!²

Social context is also likely to be relevant: peer learning is very useful for human language learners [Rancière, 1991; Raw, 2014]. One idea would be to adapt the Q&A model mentioned in Section 2 as a “social” site about poetry.

5 Future work

Quillian’s “The Teachable Language Comprehender: A Simulation Program and Theory of Language” [1969] took the novel – and fundamental – approach of understanding things in such a way that new understandings could be added directly to its knowledge base.³ When reading a piece of text, the TLC program would search its memory for related information that it could use to make sense of the input. More specifically, a given text would be expanded using “form tests” which extracted meaningful pieces of the text, and connected these to items stored in memory. Quillian writes that “ultimately, a human-like memory should relate descriptive knowledge of the world to perceptual-motor activity, in a manner like that indicated by Piaget” – but deems this to be “far beyond our present scope” [Quillian, 1969, p. 474].

Future research might use FloWr to develop a TLC-like library of “form tests” and generative tools that would add a multi-modal aspect to knowledge representation. To be sure, a flowchart-based representation of poetic process would be quite different from the embodied sensory-motor experience of humans. Nevertheless, computational processes that allow us to model text generation contextually, procedurally, and gesturally can help to understand the way linguistic meaning comes to be. This is not something we can readily learn from parsing, corpus-based modeling, or grammar-based text

²<http://www.linguistlist.org/issues/2/2-457.html#2>

³In practice, “While the monitor can add TLC’s encoded output to the program’s memory, the program itself makes no attempt to do so, nor to solve the problems inherent in doing so” [Quillian, 1969, p. 473].

generation alone. There is exciting potential for future experiments with natural language that strives to capture and express shades of meaning and the “feel” of the language. Experimentation is necessary: if we have learned anything about language, it is that “learners should be motivated to speak bravely” [Wang, 2014].

6 Acknowledgments

Joseph Corneli’s work on this paper was supported by the Future and Emerging Technologies (FET) programme within the Seventh Framework Programme for Research of the European Commission, under FET-Open Grant number 611553 (COINVENT). Thanks to Teresa Llano for conversations about her work on the “demon dog” poem, quoted above.

References

- [Agamben, 1995] Giorgio Agamben. *Idea of prose*. SUNY Press, 1995. (First published in 1985, as *Idea della prosa*.)
- [Bishop, 2004] J.M. Bishop. A view inside the Chinese room. *Philosopher*, 28(4):47–51, 2004.
- [Carlsisle, 2000] Judith P Carlsisle. Comments on Kurzweil’s Cybernetic Poet. *AMCIS 2000 Proceedings*, page 123, 2000.
- [Charnley *et al.*, 2014] John Charnley, Simon Colton, and Maria Teresa Llano. The FloWr framework: Automated flowchart construction, optimisation and alteration for creative systems. In Simon Colton, Dan Ventura, Nada Lavrač, and Michael Cook, editors, *Proc. 5th International Conference on Computational Creativity*, 2014.
- [Colton *et al.*, 2015a] S. Colton, H. Toivonen, M. Cook, and D. Ventura, editors. *Proc. Sixth International Conference on Computational Creativity, ICCO 2015*. Association for Computational Creativity, 2015.
- [Colton *et al.*, 2015b] Simon Colton, Jakob Halskov, Dan Ventura, Ian Gouldstone, Michael Cook, P Blanca, et al. The Painting Fool Sees! New Projects with the Automated Painter. In Colton et al. [2015a].
- [Corneli *et al.*, 2015] Joseph Corneli, Anna Jordanous, Rosie Shepperd, Maria Teresa Llano, Joanna Misztal, Simon Colton, and Christian Guckelsberger. Computational Poetry Workshop: Making Sense of Work in Progress. In Colton et al. [2015a].
- [Doidge, 2007] Norman Doidge. *The brain that changes itself: Stories of personal triumph from the frontiers of brain science*. Penguin, 2007.
- [Hermann *et al.*, 2015] Karl Moritz Hermann, Tomas Kocisky, Edward Grefenstette, Lasse Espeholt, Will Kay, Mustafa Suleyman, and Phil Blunsom. Teaching machines to read and comprehend. In *Advances in Neural Information Processing Systems*, pages 1684–1692, 2015.
- [Hernandez, 2013] Arturo Hernandez. *The Bilingual Brain*. OUP USA, 2013.
- [Hoey, 2005] Michael Hoey. *Lexical priming: A new theory of words and language*. Psychology Press, 2005.

- [Holmes and Moulton, 2001] Vicki L Holmes and Margaret R Moulton. *Writing simple poems: Pattern poetry for language acquisition*. Ernst Klett Sprachen, 2001.
- [Iverson and Thelen, 1999] Jana M Iverson and Esther Thelen. Hand, mouth and brain. The dynamic emergence of speech and gesture. *Journal of Consciousness Studies*, 6(11-12):11–12, 1999.
- [Kim et al., 2014] Joy Kim, Justin Cheng, and Michael S Bernstein. Ensemble: exploring complementary strengths of leaders and crowds in creative collaboration. In *Proc. 17th ACM conference on Computer supported cooperative work & social computing*, pages 745–755, 2014.
- [Kirke and Miranda, 2013] Alexis Kirke and Eduardo Miranda. Emotional and Multi-agent Systems in Computer-aided Writing and Poetry. In *Procs. of the Artificial Intelligence and Poetry Symposium (AISB’13)*, pages 17–22, Exeter University, Exeter, UK, 2013.
- [Kirke, 2011] Alexis Kirke. *Application of intermediate multi-agent systems to integrated algorithmic composition and expressive performance of music*. PhD thesis, University of Plymouth, 2011.
- [Kiryakov et al., 2004] Atanas Kiryakov, Borislav Popov, Ivan Terziev, Dimitar Manov, and Damyan Ognyanoff. Semantic annotation, indexing, and retrieval. *Web Semantics: Science, Services and Agents on the World Wide Web*, 2(1):49–79, 2004.
- [Kurzweil, 1990] Ray Kurzweil. *The age of intelligent machines*. MIT Press, 1990.
- [Llano et al., 2016] Maria Teresa Llano, Simon Colton, Rose Hepworth, and Jeremy Gow. Automated Fictional Ideation via Knowledge Base Manipulation. *Cognitive Computation*, pages 1–22, 2016.
- [Mazzola, 2016] Guerino Mazzola. Melting Glass Beads—The Multiverse Game of Gestures and Strings. *Glass Bead, Site 0: Castalia, the Game of Ends and Means*, 2016.
- [McGregor et al., 2016] Stephen McGregor, Matthew Purver, and Geraint Wiggins. Words, concepts, and the geometry of analogy. In Dimitrios Kartsaklis, Martha Lewis, and Laura Rimell, editors, *Proceedings of Workshop on Semantic Spaces at the Intersection of NLP, Physics and Cognitive Science*. 2016.
- [Ogden and Richards, 1923] Charles Kay Ogden and Ivor Armstrong Richards, editors. *The Meaning of Meaning. A Study of the Influence of Language upon Thought and of the Science of Symbolism*. Harcourt, Brace and Company, Inc., 1923.
- [Pace-Sigge, 2013] Michael Pace-Sigge. The concept of lexical priming in the context of language use. *ICAME Journal*, 37:149–173, 2013.
- [Parker, 2010] Philip M. Parker. An introduction to “Graph Theoretic Poetry”, 2010. Published on totopoetry.com.
- [Pine, 2014] Red Pine. *The Mountain Poems of Stonehouse*. Copper Canyon Press, 2014.
- [Quillian, 1969] M. Ross Quillian. The Teachable Language Comprehender: A Simulation Program and Theory of Language. *Commun. ACM*, 12(8):459–476, August 1969.
- [Rancière, 1991] Jacques Rancière. *The ignorant school-master: Five lessons in intellectual emancipation*. Stanford University Press, 1991.
- [Raw, 2014] Laurence Raw. The Paragogy of Adaptation in an EFL context. In D. Cartmell and I. Whelehan, editors, *Teaching Adaptations*, pages 26–40. Palgrave Macmillan, 2014.
- [Redington and Chater, 1997] M. Redington and N. Chater. Probabilistic and distributional approaches to language acquisition. *Trends in Cognitive Sciences*, 1(7):273–281, 1997.
- [Rusu et al., 2007] Delia Rusu, Lorand Dali, Blaž Fortuna, Marko Grobelnik, and Dunja Mladenić. Triplet extraction from sentences. In *Proc. 10th International Multiconference Information Society-IS*, pages 8–12, 2007.
- [Rzepka and Araki, 2015] Rafal Rzepka and Kenji Araki. Haiku Generator That Reads Blogs and Illustrates Them with Sounds and Images. In *Proceedings of the 24th International Conference on Artificial Intelligence*, pages 2496–2502. AAAI Press, 2015.
- [San Francisco Writers’ Grotto, 2012] San Francisco Writers’ Grotto. *642 Things to Write About*. Chronicle Books, 2012.
- [Schlegel et al., 2013] Alexander Schlegel, Peter J. Kohler, Sergey V. Fogelson, Prescott Alexander, Dedeepya Konuthula, and Peter Ulric Tse. Network structure and dynamics of the mental workspace. *Proc. National Academy of Sciences*, 110(40):16277–16282, 2013.
- [Schwartz, 2015] Oscar Schwartz. Can a computer write poetry?, May 2015. Lecture at TEDxYouth@Sydney.
- [Screene and Wiggins, 2015] Francis Screene and Geraint A Wiggins. Learning large scale musical form to enable creativity. In Colton et al. [2015a].
- [Selinker, 1972] Larry Selinker. Interlanguage. *International Review of Applied Linguistics in Language Teaching (IRAL)*, 10(1-4):209–232, 1972.
- [Shelley, 1831] Mary Wollstonecraft Shelley. *Frankenstein, or, The modern Prometheus*. H. Colburn and R. Bentley, 1831.
- [Turing, 1951] Alan Turing. Intelligent machinery, a heretical theory, 1951. Lecture at ‘51 Society’, Manchester.
- [Veale, 2015] Tony Veale. Game of tropes: Exploring the placebo effect in computational creativity. In *Proceedings of the Sixth International Conference on Computational Creativity June*, page 78, 2015.
- [Wang, 2014] Zhiqin Wang. Developing accuracy and fluency in spoken English of Chinese EFL learners. *English language teaching*, 7(2):110, 2014.
- [Wolff, 1988] J.G. Wolff. Learning syntax and meanings through optimization and distributional analysis. *Categories and processes in language acquisition*, 1(1), 1988.