The Poetry Pioneer

Nitin Nihalani

Department of Computing

Imperial College London

A report for

MEng Computing (AI)

Individual Project

Contents

Contents i

1 Introduction 1

2 Background 6

2.1 Poetry Theory . . . 6
2.1.1 The Purpose of Poetry . . . 6
2.1.2 Features of Poetry . . . 7
2.1.2.1 Rhyme . . . 7
2.1.2.2 Rhythm . . . 8
2.1.2.3 Sound Devices . . . 10
2.1.2.4 Structure . . . 11
2.1.2.5 Symbolism and Imagery . . . 12
2.1.2.6 Context and Personification . . . 13
2.1.2.7 Theme . . . 14
2.1.3 Types of Poetry . . . 14
2.1.3.1 Categories . . . 15
2.1.3.2 Popular Types . . . 15
2.2 Brief Overview of Computational Creativity . . . 17
2.3 Brief Overview of Computational Linguistics . . . 17
2.3.1 Syntax . . . 17
2.3.2 Semantics . . . 17
2.3.3 Summarisation . . . 17
2.3.4 Natural Language Generation . . . 17
2.3.5 Grammaticality and Interpretation . . . 17
2.3.6 Discourse Representation Theory . . . 17
2.4 State of the Art . . . 17

3 Project Plan 18

3.1 Analysis . . . 18
3.2 Abstraction . . . 19
3.3 Generation . . . 21

4 Evaluation Plan **24**
4.1 Comparison of Analysis Results to Theory 24
4.2 Turing-style Tests 25
4.3 FACE Descriptive Model 26
4.4 IDEA Descriptive Model 26

Appendix A **28**

References **29**

Chapter 1

Introduction

Poetry is a linguistic art form designed to help convey a particular, well-defined message in a memorable and powerful way. Poets write poems that are succinct in length but dense in meaning by employing a number of techniques, such as rhyme and rhythm (Chapter num). However, they must take the occasional liberty with grammar rules in order to meet the constraints imposed by these techniques as they are more important in ultimately conveying the message. Therefore, poems can be understood as natural language messages with added prioritised constraints and features. Different types of poetry each have their own sets of constraints, features and priorities that define their best usage (Chapter num). For example, limericks are best suited to humour while elegies are best suited to remembrance.

The origins of these constraints is not sufficiently documented given that poetry predates literacy. This means that our understanding of the techniques used in poetry comes from human analysis. There are few better tools for pattern and feature detection than computers, so the aim of this paper is to investigate the effectiveness of computer systems to learn constraints and features in natural language. Analysing poetry is particularly effective because of the wider array of constraints and patterns to be found, allowing us to test the hypothesis that computers are adept to pattern and feature recognition in natural language.

A natural extension to this process is to use the learned techniques to produce novel, creative poetry. Previous attempts at poetry generation involved manually hard-coding rules in various ways to enforce these techniques. While this may be effective at producing poetry that has stricter structure, such as haikus and limericks, they are lacking in the more modern free verse because the mapping of techniques is only to types of poems, not the effects they have on the reader. By learning the techniques from first principles, the system can find the best opportunities for the use of each technique and decide what liberties can be taken in favour of other constraints. This approach should lead to poetry that is adept not only at producing poems of a certain type, but

choosing the right tools for the right message as a human poet would.

The primary objective of this project is to build a poetry generation system that learns constraints and features by analysing a large number of poems and uses what it learned to generate new poetry.

We wish to create a system that:

- identifies a broad list of features in a single poem (Appendix num)
- generalises features of a given class of poems or texts
- learns the features of a wide variety of different classes of poems (Appendix num)
- produces poems, given natural language 'seeds' of inspiration, that are:
  - novel
  - syntactically valid
  - semantically interpretable
  - evident of a set of defined constraints and desired features
- detects relevant opportunities for usage of known poetic techniques

This project makes contributions towards both computational creativity and computational linguistics, in particular:

- We demonstrate the ability for computer systems to assess written natural language works in terms of its structure, common words and phrases, rhetoric and poetic features such as rhyme, rhythm and alliteration. Notably, we will investigate the appropriateness of Discourse Representation Theory as a semantic representation of characters, objects and locations, descriptions of them, relationships between them and the actions that they executed.
- We demonstrate the ability to make generalisations of written features when given a large number of similar texts to compare.
- We investigate the appropriateness of turning those generalisations into predicates and rules in a constraint programming environment.

- We demonstrate constraint programming as an effective tool for guiding the macro- and micro- planning stages of natural language generation
- We demonstrate the use of third party libraries as sufficient for the surface realisation stage of natural language processing
- We provide a tool for poetry creation from natural language seeds of inspiration

Poetry is a mysterious art. Poems are often written to make a point in a powerful and dramatic way that helps the people connect with the author and the emotions that are succinctly put into rhythmic combinations of words and to make it more memorable. Even those who unaware of the subtle techniques employed by poets in order to resonate better with their reader or listener are still able to enjoy high quality poetry because of the way it sounds. Yet, even most humans struggle to understand poems, let alone have the creative ability to generate them.

Furthermore, people reading the same poem can have different interpretations of its underlying goal and meaning. The best of poets take advantage of this by weaving different layers of meaning into very few words, but humans are also apt at finding layers of meaning that were not intended by the poet.

Many previous attempts at poetry generation have abused this admirable human trait by slewing grammatically correct strings of words (a task that has become relatively easy in the field of Natural Language Processing over the past decade) and rely on the reader to find its meaning. However, Chomsky retorted that a valid grammatical syntax can be completely non-sensical, giving the famous example "Colorless green ideas sleep furiously". If this were a line in a poem, a determined reader might read into this as old, locked-away, thoughts of monetary greed threatening to return or something to that effect. However, Veale argues that this 'poetic licence' is not a licence but a contract that allows a speaker to take liberties in language in exchange for real insight.

The approach taken in this project aims to generate meaningful insight in poetic form in a fully autonomous way. This follows from the unfortunate realisation brought forward by Colton et al. that a fully autonomous computer poet cannot induce the emotional connection with the reader that a human can. This is a constraint that prevents the system from producing syntactically correct yet non-sensical strings of

words. This constraint is not a bad thing. In fact, composer Igor Stravinsky argues that constraints assist the creative process rather than hinders it:
"The more constraints one imposes, the more one frees one's self of the chains that shackle the spirit"
- Igor Stravinsky 1947

Poetry comes with more constraints than just meaningful insight. Toivanen et al. 2013 put forward a promising proposal for generating poetry flexibly and effectively using constraint programming.

- Rhyme
- Rhythm, including meter and syllable count
- Alliteration, assonance, consonance, onomatopoeia and other sound devices
- Rhetoric, such as metaphors and similes
- Theme
- Repetition
- Structure
- Unusual words
- Exaggeration and superlatives
- Symbolism
- Tense

These constraints are not a bad thing. In fact, composer Igor Stravinsky argues that constraints assist the creative process rather than hinders it:
"The more constraints one imposes, the more one frees one's self of the chains that shackle the spirit"
- Igor Stravinsky 1947
Similarly, more constraints make poetry generation easier for a logic-based entity as a

computer. The more constraints, the smaller the search space and possible permutations. These constraints can be hard-coded, but the number of sets of constraints can be very large, constantly changing and sometimes too subtle to notice or define logically.

It could be argued that a computer writing poetry is an oxymoron, and a pointless endeavour. However, several attempts have been made in the past at generating poems automatically with several different methods being employed (Chapter num). For motivation, we need look no further than the desire to pass the Turing Test. For a computer to be truly exhibit intelligent behaviour indistinguishable to that of a human, it needs to be able to make an emotional connection with another human. Other reasons for attempting to generate poetry automatically include other transferable skills that a computer could develop, including:

- Semantic understanding of text
- Fluent natural language generation
- Problem abstraction and specification
- Pattern recognition and projection
- Drafting and iterating

Poems are summaries of an author's emotional thoughts and opinions coupled with both common sense and stereotypical knowledge. Therein lie the challenges of poetry generation. It is difficult enough for humans to express their own emotions in words, let alone succinctly. Furthermore, common sense and stereotypical knowledge is derived from the human instinct to abstract conclusions from experiences.

Chapter 2

Background

This chapter gives a brief overview of the features and types of poetry that exist, as well as why people are interested in writing them. This gives context for the brief overview into the fields of computational linguistics and computational creativity, both of which are involved in the task of automatic poetry generation. Finally, we will explain and critique the state of the art of poetry generation that most relate to the approach taken in this paper.

2.1 Poetry Theory

To fully appreciate the task that we are about to undertake, we need to have an understanding of poetry as an art form. Poetry is believed to predate even literacy and as such it has been documented ever since. Over the ages, different styles of poetry have evolved. Here we discuss those styles and the underlying reason for why poetry is written.

2.1.1 The Purpose of Poetry

Merriam-Webster dictionary defines poetry as:
writing that formulates a concentrated imaginative awareness of experience in language chosen and arranged to create a specific emotional response through meaning, sound, and rhythm.

Let us break this down:

- *Formulates* implies a method and system in the process of writing a poem.
- *Concentrated* accentuates the fact that poems are generally short, as they are indeed counted in stanzas in lines rather than paragraphs and pages.

- *Imaginative* confirms the fact that this is a creative act, that has a level of non-determinism and is not necessarily realistic.
- *Awareness of Experience* embodies the need for general background knowledge based on a particular set of experiences.
- *Language Chosen and Arranged* reiterates that this is a methodical and systematic art - decisions are made carefully with precise intention.
- *Create a specific emotional response* finally gives the purpose of the poem - too express a feeling or an idea and elicit an emotional response.
- *Through meaning, sound, and rhythm* describes that this purpose is not reached purely by the words, but other language features.

To summarise the purpose of poetry from this, we would say that it is using one's imagination and experience to trigger empathy about a particular topic from the reader. This means that poetry is a vehicle through which poets can share a very personal message that they want the reader to experience and remember.

We must keep this in mind throughout the project, as it is important to realise that the features of poetry discussed in the next section might be seen as arbitrary rules on form rather than purposeful techniques used to make the language more concise and effective.

2.1.2 Features of Poetry

The definition above mentions the use of *meaning, sound and rhythm* in poetry. These add an extra layer of subtext to poems to help the author remain concise while still eliciting a full emotional response from the reader. We call these techniques *features* of poetry throughout this paper. There are many features of poetry to address, but we have scoped this project down to concentrate on the following popular ones.

2.1.2.1 Rhyme

Two words rhyme when they sound similar when spoken out loud. *Cat* and *fat* in the poem above rhyme, as do *mice* and *nice*. Rhyming words need not spell the same way,

*There once was a big brown **cat***
*That liked to eat a lot of **mice***
*He got all round and **fat***
*Because they tasted so **nice***

Figure 2.1: A rhyming quatrain often used in teaching poetry

for example, *kite* and *height*.

Strict rhyme enforces the exact same sound while weak rhyme only requires that the vowel sounds are the same. Examples of weak rhyme are *turtle* with *purple* and *tragedy* with *strategy*.
A piece of text has a rhyme scheme if there is a pattern of rhyme between its lines. For example, the poem in Figure blah has an $ABAB$ rhyme scheme.

Rhyme can also occur within a line (internal rhyme) or between words in the middle of different lines.

Major Purposes

- Pleasant to hear, making the listener feel more comfortable and listen carefully
- As a mnemonic device
- Used at the end of lines of poetry and songs making the rhythmic structure more distinct

In this project, we will detect and reproduce end-line rhyme and internal rhyme where applicable. It will also be prioritised when producing poetry for which a rhyme scheme is not mandatory but fits the purposes.

2.1.2.2 Rhythm

Rhythm is the pattern of emphasis of syllables that occurs in a line of poetry. There are three major ways of measuring rhythm, often used in tandem - syllabic, quantitative and accentual.

The bartender said
to the neutron, 'For you, sir,
there will be no charge.'

Figure 2.2: A funny Haiku

Syllabic rhythm enforces a certain number of syllables to be used in a particular line of poetry. Haikus are the most famous type of poem with this feature - they are three lines long with the first and last lines restricted to 5 syllables and the second to 7. An example is given in Figure blah.

Quantitative measures uses the fact that some syllables sound longer than others when spoken out loud. Long sounding syllables are *stressed* while short ones are *unstressed*. Accentual measures are similar to Quantitative, but they work on the tendency to stress a particular syllable when spoken out loud, rather than its length. It is important to note that a word's meaning can change depending on stress. For example, '**ob**ject' is a noun whereas 'ob**ject**' is a verb.

Lines of pre-defined patterns of stressed and unstressed syllables are called *meters*. Lines with meter are made up of individual units called *feet*. The five major foot types in poetry are given in Table blah.

Foot Type	Pattern	Example
iamb	unstressed - stressed	des**cribe**
trochee	stressed - unstressed	**po**em
spondee	stressed - stressed	**popcorn**
anapest	unstressed - unstressed - stressed	meta**phor**
dactyl	stressed - unstressed - unstressed	**po**etry

Table 2.1: The major poetic foot types with their corresponding pattern and an example to illustrate

The metre is formed by repeating feet, typically with up to six feet:

- Monometer: 1 foot
- Dimeter: 2 feet
- Trimeter: 3 feet
- Tetrameter: 4 feet

- Pentameter: 5 feet
- Hexameter: 6 feet

All Shakepeare's sonnets are written in iambic pentameter, i.e. five repetitions of unstressed-stressed syllables. His Sonnet II is given as an example:
*When **for**ty **win**ters **shall** be**siege** thy **brow***

Major Purposes

- Introduces a melody based on the natural intonations of speech
- Add a level of predictability and subconscious structure
- Emphasizes the message by putting stress on the words that matter

Rhythm is a fundamentally important feature of poetry, so this project aims to be able to detect and reproduce it consistently and to a high level of accuracy.

2.1.2.3 Sound Devices

This project considers four types of sound devices.

The first is **onomatopoeia** - words that imitate or suggest sounds of particular sources. For example, the word 'Pow' is used to describe the sound of a punch or 'tick-tock' for clocks. This technique has mostly been used in comic books to help the reader experience the sound of the scene to go with the image.

The next three devices are repetitions of a pattern of similar sounds, like rhyme. **Consonance** is the repetition of similar consonant sounds (e.g. 'pitter patter' repeats the 'p', 't', and 'r' sounds), while **assonance** is that of vowels (e.g. 'doom and gloom' repeats the 'oo' sound). **Alliteration** is a special case where the repeated sound occurs at the beginning of consecutive words. 'Zany zebras zigzagged through the zoo' has alliteration on the letter 'z'.
Major Purposes

- Poets use onomatopoeia to help describe actions or atmosphere richly. A famous example is the nursery rhyme 'Old MacDonald', which uses onomatopoeia of the

sounds that animals make to describe the farm, figuratively placing the reader in the farm itself.

- Alliteration, consonance and assonance are pleasant to listen to when spoken out loud.
- Can be used to add drama to an action.
- Sometimes used to suggest danger.

We aim to detect and reproduce all forms of sound devices explained here. However, there will be a limit to the onomatopoeic vocabulary and it will be unable to create brand new onomatopoeia.

2.1.2.4 Structure

The structure of the poem is the organisation of lines in a poem. The main unit is the stanza, which is a fixed amount of lines grouped by rhythmical pattern.

There are four major types of stanza:

- Couplet: 2 lines
- Tercet: 3 lines
- Quatrain: 4 lines
- Cinquain: 5 lines

Stanzas can also be called verses, which have the added property that they tend to rhyme. A chorus is a special type of verse that is repeated every other verse. Similarly, envoys are the short final verse of a poem.

Features of the structure of a poem include:

- The number of stanzas
- The number of lines per stanza

- The number and positions of repeated lines
- The number and positions of repeated stanzas

The Haiku in Figure blah has a single tercet structure with no repetitions. Songs are generally several stanzas long, with a chorus interleaving longer non-repeating verses.

Major Purposes

- Helps to guide the reader through the story.
- Forces the poet to be more succinct and purposeful.
- Manages the storyline - changes in stanza often suggest a change in perspective or message.
- Repetition helps drive home the main message.
- Ties several thoughts together into one continuous flow.

We concentrate on detecting and reproducing accurate structures of common poetry in this project.

2.1.2.5 Symbolism and Imagery

Symbolism and imagery are general terms for creating an overall image in the reader's mind by describing a subject or object as something else.

Techniques include:

- *Metaphor*: an object is described as another object with a set of desirable characteristics. For example, saying someone is 'drowning in debt' describes the debt as an ocean in which the subject is drowning, suggesting that there is too much to handle, and may even insinuate the subject's downfall.
- *Simile*: an object or action is specifically described using an adjective or adverb, but compared to another object that is a stereotypical example of that description. The phrases 'like a' and 'as a' are often used, e.g. 'Runs like a cheetah', 'Slippery as an eel'

- *'Hyperbole'*: unrealistic exaggeration, often used in tandem with metaphor e.g. 'Cried a river of tears'.
- *'Powerful Verb'*: a more exciting way to describe an action using unusual verbs, e.g. 'Wormed through the crowd'.

Major Purposes

- Explain complex concepts concisely.
- Induce empathy from the reader by relating it to something they understand.

This paper attempts to detect use of metaphor and simile as well as reproduce it. However, the context around the use of the metaphor and simile is not considered, so they may not be generated at the most appropriate time. We do not consider Hyperbole and Powerful Verb as they are similar to Metaphor, but it is worth noting that they exist because including them should be an accessible extension for future work.

2.1.2.6 Context and Personification

Poetry is similar to storytelling in that it has characters around which the poem is written. Understanding who or what they are, their descriptions and their actions are all part of the underlying message that the poet wants to get across. This is the context of the poem.

Personification is a technique used by poets to give inanimate objects life, expressing actions and descriptions as if it were human. This is a powerful technique that relates to imagery, helping poets make more abstract messages clearer. For example, 'the moon smiled' gives the moon life by describing it as having performed a human-like action with full intention of doing so. Noting the use of personification can make the context of the poem clearer, as often inanimate objects are the subject of the poem.

Major Purposes

Context is the underlying message in its bare form. It is the story that the poet wishes to tell and guides the use of all other features.

In this paper, we aim to extract characters and differentiate them by their descriptions and actions. This is vital in understanding the poem and can help us generalise the uses of features when attempting to produce a coherent story as the backbone to the generated poem. Furthermore, it will help determine the type of poem (narrative, lyrical, descriptive etc.) and will help guide generation of poems of a particular type.

2.1.2.7 Theme

Theme is a very abstract term that is difficult to define. It can be thought of as the worldly context of the poem, often requiring better understanding and background knowledge of the poet. For example, Maya Angelou's *I Know Why The Caged Bird Sings* is about oppression of African Americans during the time that she was alive. It was were way of helping readers empathise with her situation, by personifying a caged bird and the similarities she shares with it. The theme is therefore *oppression*. However, without knowing Angelou's life situation, it would have been impossible to determine.

Theme is beyond the scope of this project, both in interpretation of poetry and generation. However, we will attempt themes that can be summarised into a single word, often called topics. For example, we will try to emulate the styles of 'Love' poems or 'War' poems from the past. This will help give more meaning to the features described here and help in using them in the right context during the generation phase.

2.1.3 Types of Poetry

We define a type of poetry as a particular form of poem with a set of unique features, including those described in the previous section. Some types are very popular and have had their styles, features and purposes documented and taught. Out of these grew categories of different types that tend to be used for similar purposes.

This project attempts to derive these categories and some popular types of poetry. They are discussed here.

2.1.3.1 Categories

There are many types of poem all with different form. However, there are only three main categories of purpose for a poem:

1. *Lyrical* poems have an identifiable speaker whose thoughts and emotions are being expressed in the poem. This means that poems of this category have very few characters, a song-like structure and tends to be in a reflective tone, generally using a lot of symbolism. Maya Angelou's *I Know Why The Caged Bird Sings* is an example of this, along with many songs.

2. *Descriptive* poems describe the surroundings of the speaker. This is identifiable by the use of adjectives and complex imagery. Many objects may appear in this type of poem to be able to give an in-depth description of the environment and atmosphere. There will be very few action verbs used.

3. *Narrative* poems concentrate on telling a story. It therefore has a coherent plot line, several characters with explicit relationships between them, action and climax. Ballads and Epics are types of narrative poems.

4. Some popular poem types do not fall under any one bracket as they can be used in any of the above categories. Examples include Haikus and Limericks.

This project aims to be able to place any poem accurately into one of these categories.

2.1.3.2 Popular Types

As well as determining the category of poems, we aim to be able to detect and reproduce some popular types of poetry. For this project, we will concentrate on:

- *Haiku:* single tercet structure with 5-7-5 syllabic rhythm.

- *Limerick:* single cinquain structure with AABBA rhyme scheme. Lines 1, 2 and 5 have 7-10 syllables, while lines 3 and 4 have 5-7 syllables. The first line tends to begin with "There was a..." and ending with a person or location. Limericks are usually used for humour as the last line is generally a punchline.

- *Sonnet:* 14 lines, each in iambic pentameter with an ABABCDCDEFEFGG rhyme scheme, i.e. three quatrains followed by a rhyming couplet.
- *Elegy:* usually used to mourn the dead, its lines alternates between dactylic hexameter and pentameter in rhythm. It has no particular rhyme scheme, although does still use rhyme.
- *Ode:* Description of a particular person or thing, using plenty of similes, metaphors and hyperbole.
- *Ballad:* Tells a story and has a number of quatrains, each with an AABB rhyme scheme. Lines alternate between iambic tetrameter and iambic trimeter.
- *Cinquain:* as the name suggests, this has 5 lines. They are not rhymed, but have a 2-4-6-8-2 syllabic pattern.
- *Riddle:* Riddles describe things without telling what it is, using anaphora to refer to it. Ususally told in a number of rhyming couplets.
- *Free Verse:* No particular features attached to this type.

Some of these poems are harder to read and generate than others, particularly in terms of context. However, this selection covers the main features of poetry that we would like to address so it a good way to evaluate this project.

2.2 Brief Overview of Computational Creativity

2.3 Brief Overview of Computational Linguistics

2.3.1 Syntax

2.3.2 Semantics

2.3.3 Summarisation

2.3.4 Natural Language Generation

2.3.5 Grammaticality and Interpretation

2.3.6 Discourse Representation Theory

2.4 State of the Art

Chapter 3

Project Plan

This project is split up into three main phases. The first two phases, Analysis and Abstraction, is planned to span roughly the first two terms. The final stage, Generation, is to be completed in the third term. Some implementation details are included in this section.

3.1 Analysis

First we analyse a single poem in great depth. This means that we write algorithms to detect the use of:

- Rhyme and internal rhyme
- Rhythm, including meter and syllable count
- Alliteration, assonance, consonance, onomatopoeia and other sound devices
- Structure, tense and repetition

Then we try to understand the context of the poem to extract:

- Characters
- Objects
- Locations
- Descriptions
- Relationships
- Actions
- Point of View
- Symbolism, such as metaphors and similes

The implementations of these detectors use:

- Basic string parsing algorithms
- Python Natural Language Toolkit, a suite of text processing libraries, corpora and lexical resources
- The CMU Pronunciation Dictionary, which gives phonetic spellings of 133,000 English words in ARPAbet
- WordNet, a lexical database that provides conceptual-semantic and lexical relations of 155,287 English words
- VerbNet, a lexical database that groups verbs by semantic and syntactic linking behaviour
- CLiPS Pattern, a web-mining module for Python
- Written Sound, a dictionary of onomatopoeia to their meanings and associated objects
- Discourse Representation Theory
- Metaphor Magnet, a web application that maps commonplace metaphors in everyday texts

This stage is currently on schedule and expected to be completed within two weeks of the submission of this paper. The tasks that remain are an improved grammar and lexicon for parsing to a Discourse Representation Structure (DRS) and metaphor detection using Metaphor Magnet.

Note that the Theme of the poem is not be addressed at this stage. The algorithm for detecting Theme uses a very similar process to Abstraction as explained below, so has been moved to that phase.

3.2 Abstraction

A prerequisite to this stage is that all of the output from the detectors is as general as possible, leaving all options open for consideration during this phase. They should also be in an easily comparable format such as numerical scores and strings. The first task of this stage is to make sure that this is the case and make any adjustments if necessary.

The major algorithm to be written in this stage will analyse a large set of poems, compare each aspect of the results individually and find significant overlaps. The overlapping features will be saved as a template for poems of that type. Any single type of poem may have several templates. For example, overlapping features of all limerick include its rhythm and *AABBA* rhyme scheme, but only some start with *"There once was a"*, so one template will be created with that start string and one without.

We then take a large list of poems of the same Theme, as determined by classifications on poets.org [https://www.poets.org/search.php/prmResetList/1] and run it through this algorithm. We will then be able to analyse and generate poetry with Theme-associated constraints. For example, love poems may talk about roses and flowers, so we associate these objects as having a theme of love. It may also, for example, reveal that limericks are mostly humourous.

Thereafter, we will find abstractions of the following types of poems:

- Haiku
- Limerick
- Sonnet
- Elegy
- Ode
- Ballad
- Cinquain
- Riddle
- Free Verse

The data acquired in this stage will then be stored in a relational database.

This algorithm can do most of its processing unsupervised and in parallel so should not take very much time. However, to be on the safe side we will employ agile methodologies to avoid running out of time for the generation phase. Firstly, we will only run abstraction on a limited number of themes. Then we will work on Haikus initially

because they are short and have relatively simple structure. We will then move on to the Generation phase. Once we are able to generate Haikus, we will return to this stage and abstract the next type of poem down the list and try to generate it. We will oscillate between Abstraction and Generation in this way until the end of the list and all themes are completed, or we run out of time.

3.3 Generation

The process of generating a poem begins with a seed of inspiration. It could come in the form of a command from the user, for example "write me a limerick about a computer that is bored with data but finds poetry fun". Or it could come from another source, perhaps the top trending Tweet for that day. A type, and optionally theme, for the poem is also given, which imposes constraints on the features and form of the poem as well as preferences for the words to be used.

The inspiration is converted into a DRS, as in the Analysis phase. As part of generating the requested type of poem, we wish to aim for the a DRS of that found in the generalisation stage for the requested poem. This enforces descriptive poems to have more descriptions or narrative ones to have more action, for example.

The corresponding template for the requested poem is loaded. If there are multiple candidate templates, one is chosen at random. This randomness is encouraged as it could ensure non-determinism, making the poem seem more creative. An added feature would be to allow the user to alter the constraints manually and add words or phrases to be excluded from the poem. If given time, the generator could autonomously 'mutate' low priority constraints as another way of including non-determinism.

The poem template is then filled as far as possible using the seeds only. The following algorithm is then executed to complete the rest of the poem:

```
For each line in the poem:
  If full sentence and valid structure,
```

```
    continue.
  Elif full sentence and invalid structure,
    rephrase to fit structure (using error code if provided).
  Elif almost full sentence,
    use collocations to fill gaps.
  Elif non-empty sentence,
    determine goal of sentence based on target DRS.
    find phrases/similes/metaphors to fit.
  Elif empty sentence,
    find association to subject/object of prev line.
    add association as subject/object of sentence.
Check against constraints of type of poem.
If not all constraints are met,
  Restart loop with error codes to help rephrasing process.
```

Words selected during the algorithm are weighted to include more alliteration and other features. A dry run example is given in the appendix. Rephrasing entails replacing a word or phrase in a sentence with a substitute that fixes the broken constraint without breaking anything further.

This algorithm will be implemented using all the tools listed in the Analysis section, as well as from the following sources and libraries:

- Oxford Collocations Dictionary, a source of word pairings and phrases that occur with greater than chance probability
- Phrase Finder, a database of 2048 English sayings, phrases, idioms, expressions
- Wordnik, a multi-faceted dictionary and semantic language database collected from a variety of sources
- LinguaTools DISCO, a tool to derive semantic similarity between words based on statistical analysis of large text collections
- ConceptNet 5, a semantic general knowledge network
- Thesaurus Rex, a monster thesaurus generated from web texts

Once the poem is generated, the user can once again ask for certain changes in constraints. The most useful one will perhaps be rephrasal of certain words or phrases by asking for them to be excluded. For example, if the phrase "bored to death" was used in the poem, the user could ask for the word death to be replaced. This would involve running the same algorithm as above, but with the word 'death' removed and the added constraint that the word 'death' cannot be used. This would enter the 'almost full sentence' elif condition and use collocations, possibly using 'tears' instead.

Chapter 4

Evaluation Plan

We will employ four methods of evaluating this project.

4.1 Comparison of Analysis Results to Theory

The Analysis and Abstraction phases attempt to discover the common features of any type of poetry. Current theory of these common features have been derived and documented by poets through their own analysis. We wish to investigate whether this system is able to find everything documented that has been documented by poets. It is likely that the system could only find a subset but it will be interesting to see which points were missed. A full comparison will be made and explanation for any missing features will be attempted, as well as possible algorithms to detect them that could have been implemented.

A particularly interesting result would be if the system manages to find a common feature of a type of poetry that poets have not yet found. If even one case of this occurs then this would make a strong case for the ability of computers to understand, interpret and find patterns in natural language. Further investigation would surely be warranted into how much a computer system would be able to find that human readers have overlooked.

Note that this section does not expect the system to infer the effects of any of the poetry features on the reader, just simply detect the use of the features.

4.2 Turing-style Tests

The simplest way to check the quality of poems produced would be to show people a poem either generated by this system or written by a human without telling them and asking them to determine whether the author was man or machine.

However, this is highly dependent on the reader's fluency of English, understanding of poetry and imagination. A poor English speaker with little to no understanding of poetry and with a far reaching imagination could easily be fooled into thinking that a piece of text was written by a human rather than a computer.

Therefore, we propose a survey that asks for these three measures to be told truthfully. It then randomly shows a poem and asks whether it was written by a human or computer. This way, we can get a demographic of those who are fooled and those who are not so that we can see at what level of poetry literacy this system is.

Further, we plan to approach real poets from literary institutions and university departments and ask them to do this survey in person. If they incorrectly believe even one poem to be written by a human and not a computer, then this project is a definite success. For the cases where this does not arise, having an in-person survey will enable us to ask for feedback on what gave us away, which can be used to improve the system and future attempts at poetry generation.

However, Pease and Colton 2011b make several arguments that Turing-style tests are not appropriate for poetry generation. We believe that it has its place because ultimately this project is for user consumption as well as research and experimentation. Furthermore, poems attempt to create an emotional connection with the reader, something that cannot be determined other than with a human reader.

Having said that, Colton, Charnley and Pease 2011 described the FACE and IDEA Descriptive Models for evaluating computational creativity projects. We believe these models provide a useful evaluation methodology alongside Turing-style tests and so are just as much part of the evaluation of this system as described in Section 4.3 and 4.4.

4.3 FACE Descriptive Model

A full FACE model has four symptoms: examples, concepts, aesthetics and framing information.

- *Examples* will be showcased by the templates generated by the Analysis and Abstraction phase.
- *Concepts* are of the form of the algorithm described for the Generation phase as it takes input from the user or online, the results from the Abstraction phase and several third party libraries to output a poem.
- *Aesthetics* are assessed by running the Analysis phase over the poem again. In fact, this happens several times during the creation of the poem. Any faults are reported back to the next iteration of that poem.
- *Framing Information* is the poem created.

Therefore, we can see that this system should fully abide by the FACE Descriptive Model. We will evaluate the results mathematically as per Colton, Charnley and Pease 2011.

4.4 IDEA Descriptive Model

A full IDEA model has six stages to which the software can reach. We want our software to be in the, fourth or *Discovery stage*.

- *Developmental stage*: this system has a full Abstraction phase to avoid the case that all creative acts undertaken by this system are purely based on inspiring examples. So this system will have surpassed this stage.
- *Fine-tuning stage*: the Abstraction phase only looks for a limited number of overlapping features to provide the template, leading to higher level abstraction. For example, it does not use part-of-speech tags from previous examples or any low level abstractions. We believe the system should be able to surpass this level.

- *Re-invention stage*: the system is able to work off a template provided by the Abstraction phase, but also able to mutate the templates and add or remove restrictions both automatically and guided by the user. Therefore, the creative acts are not restricted only to those that are known and should be able to surpass this stage as well.
- *Discovery stage*: the ability to work off templates derived from Analysis and Abstraction imply that the system is able to generate works that are sufficiently similar to be assessed with current contexts. However, given the flexibility of the mutation and user-guidance ability, it can also produce works that are significantly dissimilar. We believe the system to be able to reach this stage.
- *Disruption and Disorientation stages*: Since templates and constraints on the creative work that are imposed are the results of analysing and abstracting existing works, it is not the case that this system solely produces poetry that is too dissimilar to those known by theory.

Therefore, we can see that this system should reach the desired *Discovery stage* of the IDEA Descriptive Model. We will evaluate the results mathematically as per Colton, Charnley and Pease 2011.

Appendix A

The features of peoms we wish to be able to detect:

- Structure
  - Number of stanzas
  - Number of lines per stanza
  - Number of repeated lines
  - Repeated phrases
  - Repeated words
  - Unusual words
- Rhyme
  - Rhyme scheme between lines
  - Rhyme scheme within a line
  - Presence of regular rhyme scheme
- Rhythm
  - Number of syllables
  - Stress pattern for each line
- Line patterns
  - Alliteration
  - Assonance
  - Consonance
- Rhetoric
  - Similes
  - Metaphors
  - Onomatopoeia
- Contextual
  - Tense
  - Point of view
  - Characters
  - Objects
  - Locations
  - Descriptions
  - Actions
  - Relationships

References

Chobotov, V. (2002). *Orbital Mechanics, Third Edition*. AIAA Education.

Koci, R. (2010). World, view and projection matrix unveiled. http://robertokoci.com/world-view-projection-matrix-unveiled/.

Meeus, J. (1997). *Mathematical Astronomy Morsels*. Willmann-Bell.

Yu, D.T. (2005). Computer graphics. http://cse.csusb.edu/tong/courses/cs420/notes/.