Procedural Story Generation with Transformers

UNIVERSITY *of* LIMERICK

OLLSCOIL LUIMNIGH

Niall Dillane

Computer Science and Information Systems

University of Limerick

Submitted to the University of Limerick for the degree of

BSc Computer Systems 2020

1. Supervisor: James Patten
Computer Science and Information Systems
University *of* Limerick
Ireland

2. Second Reader: Dr. Name Surname
Computer Science and Information Systems
University *of* Limerick
Ireland

Showcase day: 18th April 2020

Signature:

Abstract

Procedural content generation (PCG) – the process of generating data algorithmically – is a technique that has applications across a variety of domains. In the research outlined in this report, focus is directed to the use of PCG as a means to generate novel-like stories. The challenge is twofold: research into procedural generation methods, as well as the structure and language of stories, addressing questions such as: "What are the elements of a good story?". Finally, methods to codify these elements must be investigated, in such a way that they can be used by a procedural generation technique, effectively combining the two disciplines.

Declaration

I herewith declare that I have produced this paper without the prohibited assistance of third parties and without making use of aids other than those specified; notions taken over directly or indirectly from other sources have been identified as such. This paper has not previously been presented in identical or similar form to any other Irish or foreign examination board.

The Final Year Project was conducted from 2019 to 2020 under the supervision of James Patten at University of Limerick.

Limerick, 2020

Acknowledgements

First and foremost I would like to thank James Patten; both for his inspiration of the topic researched, and the constant helpful advice and motivation throughout. This extends to all of the lecturers and teaching assistants I have interacted with over the years, each of whom contributed to where I am today.

The wide breadth of open source code, lectures and tutorials available online were also essential in allowing me to carry out this project, as it was all new ground compared to schoolwork that came before it.

Shoutout Christina Applegate

Contents

CONTENTS

List of Tables

LIST OF TABLES

List of Figures

LIST OF FIGURES

1

Introduction

In order to be able to automatically generate stories we first need to understand the structure of a story. Indeed, not only the structure that would make one grammatically correct, but also those devices, structures and styles that help make a good story. This is probably the largest challenge in the project, a formula which is still unclear after millennia of storytelling.

Furthermore, there have been significant advancements over recent years in the Natural Language Processing (NLP) field, with new architectures and technologies appearing and outpacing each other. Choosing a direction which is sufficiently developed but still relatively state of the art will be difficult, especially since few of these have specifically tackled the topic of novel-like stories, instead focusing more on news article or script style content.

However, it is this rapid progress and uncertain nature that make the topic so fascinating. There has been great progress in images, video, even games, but text generation has lagged behind somewhat. This signals that language is more nuanced, difficult to replicate, and still in its infancy.

1.1 Aims and Objectives

The main question that this dissertation addresses is .... In order to address this question, this work focuses on the following issues in the context of .... :

- Firstly, . . .
- Secondly, . . .

1.2 Methodology

In order to address this question the following approach was taken.

1.3 Research Contribution

The primary research contribution is:

1.4 Thesis Outline

The remaining chapters of this dissertation are as follows:

Chapter Two is an introductory discussion to . . .

Chapter Three is a historical review of . . .

Chapter Four presents the . . .

Chapter Five describes the . . .

Chapter Six draws conclusions and evaluates the . . . Lastly, it suggests possible future works.

A series of documents have been included in the Appendix section of this dissertation. These are:

- *Appendix A* outlines . . .
- *Appendix B* presents . . .
- *Appendix C* includes . . .

Attached to this dissertation is a CD containing the following items:

- *folder 1*: . . .
- *folder 2*: . . .

2

Related Research

Much research has been undertaken on this topic, with a variety of approaches.

2.1 Good Stories

First we must discuss the basic issue of what a good story even is, apart from the technology to produce one.

2.1.1 Structure

Structure was the first element to be researched, as the highest level of a story planning process. Popular methods include the Hero's Journey (Campbell 2008), which describes a cycle of sorts: a call to adventure, crossing from the known to the unknown, transformation etc. This is a more granular structure, most known for its application in Star Wars.

Another method is the three (or five) act structure (Trottier 1998), wherein the story is broken down into distinct sections, typically: setup, confrontation, resolution. This is a bit more versatile, which was something to consider for the later steps of "filling in the gaps" with PCG. There was a balance to be struck: providing some structure so that the generation has some level of cogency, but also allowing flexibility so that not all stories are the same. Following this, a tree-based approach became appealing, at the higher levels resembling more of a novel plan with plot points, and is eventually fleshed out below into sentences

and phrases. This is a familiar concept in Computer Science, and it seems like the best way to maintain a coherent story. Some optional user interaction would be ideal – e.g. "Tell me a story about a dog with superpowers" – a kind of "seed". These could also be generated autonomously.

2.1.2 Wiki page

You can learn more about Latex by clicking here.

2.2 Another Section

2.2.1 Another Subsection

2.2.1.1 Another Subsubsection Title

3

Chapter 3

3.1 Insert a Figure

We refer to a figure in the text like this: (Figure 3.1).

Figure 3.1: Title of Figure - Description. [Source: (**?**)]

3.2 Creating a list

- **(2000)** item 1.
- **(2004)** item 2.
- **(2010)** item 3.
- **(2013)** item 4.

3.2.1 Creating a Table

Table 3.1: Table Title

	Resolution	Min	Max
Gyroscope	4.5mV / °/s	0.27 °/s	406 °/s
Accelerometer	600mV /g	0.002g	2g
Magnetometer	385mV/ gauss	0.317 gauss	6 gauss

3.3 Quote

An in-text quote is declared in the following way:

> Recycling is the way forward! Recycling is the way forward! Recycling is the way forward! Recycling is the way forward! Recycling is the way forward! Recycling is the way forward! Recycling is the way forward! Recycling is the way forward! Recycling is the way forward! Recycling is the way forward! Recycling is the way forward! Recycling is the way forward! Recycling is the way forward! Recycling is the way forward! Recycling is the way forward! Recycling is the way forward! Recycling is the way forward! Recycling is the way forward! Recycling is the way forward! (CommonSense, p. 0)

Another way of doing it is:

There are in our existence spots of time,
That with distinct pre-eminence retain
A renovating virtue, whence-depressed
By false opinion and contentious thought,
Or aught of heavier or more deadly weight,
In trivial occupations, and the round
Of ordinary intercourse-our minds
Are nourished and invisibly repaired;
A virtue, by which pleasure is enhanced,
That penetrates, enables us to mount,
When high, more high, and lifts us up when fallen.

(**?**, verses 208-218)

3. CHAPTER 3

4

Math formulae in Latex

4.1 Math

In-text Math

- A vector vector $\hat{A}_{ba}\{x_{ba}, y_{ba}, z_{ba}\}$
- Another vector $R^b\{\hat{t}_{1b}, \hat{t}_{2b}, \hat{t}_{3b}\}$ as:

A series of Equations:

$$\hat{t}_{1b} = \hat{A}_{ba} \tag{4.1}$$

$$\hat{t}_{2b} = \frac{\hat{A}_{ba} \times \hat{M}_{bm}}{\mid \hat{A}_{ba} \times \hat{M}_{bm} \mid} \tag{4.2}$$

$$\hat{t}_{3b} = \hat{t}_{1b} \times \hat{t}_{2b} \tag{4.3}$$

$$R^a = R^b(R^i)^T = [\hat{t}_{1b}, \hat{t}_{2b}, \hat{t}_{3b}][\hat{t}_{1i}, \hat{t}_{2i}, \hat{t}_{3i}]^T \tag{4.4}$$

where the symbol ^{T} denotes transposition.

$$\hat{t}_{2b} = \frac{\hat{A}_{ba} \times \hat{M}_{bm}}{\mid \hat{A}_{ba} \times \hat{M}_{bm} \mid} \tag{4.5}$$

$$= \frac{(y_{ba}z_{bm} - y_{bm}z_{ba}, z_{ba}x_{bm} - z_{bm}x_{ba}, x_{ba}y_{bm} - x_{bm}y_{ba})}{\mid A_{ba} \mid\mid M_{bm} \mid \sqrt{1 - (\hat{A}_{ba} \cdot \hat{M}_{bm})^2}}$$

$$= \frac{-0.2338, -0.0931, -0.0651}{0.2601}$$

$$= (-0.8989, -0.3579, -0.2503)$$

which if normalised gives:

$$= (-0.8995, -0.3581, -0.2504)$$

A matrix

$$R^b = [\hat{t}_{1b}, \hat{t}_{2b}, \hat{t}_{3b}] = \begin{pmatrix} -0.2698 & -0.8995 & 0.3439 \\ 0.0035 & -0.3581 & -0.9337 \\ 0.9629 & -0.2504 & 0.0997 \end{pmatrix}$$

5

Chapter 5

5.1 References

Open the file 'references_myphd' with BibTex. The file is located in '/7_references/'

The file contains a series of templates for the proper formatting of references according to the Harvard referencing style. Click on each reference in the text below to see how Latex has formatted the reference on the References section.

- **Book**: (**?**, p. 10) (page number required only if referring to an in-text quote)
- **Book Chapter**: (**?**)
- **Book Contribution**: (**?**)
- **MUSIC CD/DVD**: (**?**)
- **Journal Article**: (**?**)
- **Patent**: (**?**)
- **PhD/Master Dissertation**: (**?**)
- **Conference Paper**: (**?**)
- **Technical Report**: (**?**)

- **Unpublished Report**: (**?**)
- **Web Videos**(**?**) (. . . not the name of the person who uploaded the video though)
- **Web Paper**: (**?**)
- **Web Journal Article**: (**?**)
- **Webpage**: (**?**)
- **Downloaded Images or Sounds**: (**?**)

You can also you this kind of referencing if needed in the text:
Surname (**?**) said that . . .

6

Conclusions and Future Directions

6.1 Summary

6.2 Conclusions

6.3 Contributions

6.4 Future Work

6. CONCLUSIONS AND FUTURE DIRECTIONS

Appendix A

Insert a figure

Figure 1: Title of Figure - Description. [Source: (**?**)]

Appendix

Appendix B

Title

Type here you text as usual . . .

Appendix

Appendix C

Code for estimating attitude

AHRS_TRIAD.C

```
/*
 *
 *  Copyright (c) 2013, Giuseppe Torre
 *  All rights reserved.
 *
 *  Redistribution and use in source and binary forms, with or without
 *  modification, are permitted provided that the following conditions are met:
 *      * Redistributions of source code must retain the above copyright
 *        notice, this list of conditions and the following disclaimer.
 *      * Redistributions in binary form must reproduce the above copyright
 *        notice, this list of conditions and the following disclaimer in the
 *        documentation and/or other materials provided with the distribution.
 *      * Neither the name of the BREAKFAST LLC nor the
 *        names of its contributors may be used to endorse or promote products
 *        derived from this software without specific prior written permission.
 *
 *  THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS_IS" AND
 *  ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED
 *  WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE
 *  DISCLAIMED. IN NO EVENT SHALL BREAKFAST LLC BE LIABLE FOR ANY
 *  DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES
 *  (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES;
 *  LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND
 *  ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT
 *  (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS
 *  SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
 *
 */



#include "ext.h"
#include "ext_mess.h"
#include <string.h>
#include <stdio.h>
#include <math.h>
#define EPSILON 1e-6

//*****************CALIBRATION STUFF************************//
/* INTRODUCE THE OFFSET VALUES IN THE FOLLOWING VARIABLES IN ADC values */
#define ACCELX_OFFSET 2043 //initial offset in accelerometer X
#define ACCELY_OFFSET 2053 // initial offset in accelerometer Y
#define ACCELZ_OFFSET 2264//initial offset in accelerometer Z

#define MagX_OFFSET 1917 //initial offset in Magnetometer X
```

Appendix

```
#define MagY_OFFSET 1813 // initial offset in Magnetometer Y
#define MagZ_OFFSET 1667//initial offset in Magnetometer Z
/* WE SHOULD CALIBRATE THE FOLLOWING VALUES FOR EACH PARTICULAR IMU axis */
#define ACCELX_Resolution 536
#define ACCELY_Resolution 661
#define ACCELZ_Resolution 559
#define MagX_Resolution 186
#define MagY_Resolution 189
#define MagZ_Resolution 170
//****************** END CALIBRATION STUFF  ************************//


void *this_class; // Required. Global pointing to this class

typedef struct _triad // Data structure for this object
{
        t_object m_ob;   // Must always be the first field; used by Max

        Atom m_args[9];  // we want our inlet to be receiving a list of 10 elements

        long m_value;    //inlet

        void *m_R1;              //these are all the outlets for the 3 X 3 Matrix
        void *m_R2;              // R1 --> firts top left cell ..R2 middle cell of the first
             row ...and so on
        void *m_R3;                                     //
        void *m_R4;                                     // End Outlets
             ---------------------------------------
} t_triad;

                        void *triad_new(long value);


                        void triad_assist(t_triad *triad, void *b, long msg, long arg, char *
                             s);
                        void triad_free(t_triad *triad);



                        void triad_list(t_triad *x, Symbol *s, short argc, t_atom *argv);



                        void MatrixByMatrix(double *Result,double *MatrixLeft,double *
                             MatrixRight);

                        void Matrix2Quat(double *Quat,double *Matrix);
                        void Quat2Matrix(double *Matrix, double *Quat);
                        void inverseQuat(double *InvQuat, double *RegQuat);
                        void NormQuat(double *YesQuat, double *NotQuat);
                        void Slerp(double *NewQuat, double *OldQuat, double *CurrentQuat);
                        void NormVect(double *YesVect, double *NotVect);
                        double orientationMatrix[9];
                        double Result[9];
                        int     i;
                        double InvorientationMatrix[9];

                        double temp[6];
                        double ref[6];
                        double vectAx[3];
                        double vectAy[3];
                        double vectAz[3];
                        double vectBx[3];
                        double vectBy[3];
                        double vectBz[3];
                        double MagnCrosProd_A;
                        double MagnCrosProd_B;
                        double accnorm, magnorm, earthnorm, VectAynorm, VectAznorm,
                             VectBynorm, VectBznorm;
```

```
                        double m[9], n[9];
                        double quat_e[4];
                        double invquat_e[4];
                        double mult;
                        double quat_new[4];
                        double quat_old[4];
                        double ecs,accex_ADCnumber,y,accey_ADCnumber,z,accez_ADCnumber,mx,
                            magnx_ADCnumber,my,magny_ADCnumber,mz,magnz_ADCnumber;
// SLERP Variables
                        double trace, Suca;
                        double tol[4], omega, sinom, cosom, scale0, scale1, tez,
                            orientationMatrixA[9];

int main(void)
{
        // set up our class: create a class definition
        setup((t_messlist**) &this_class, (method)triad_new, (method)triad_free, (short)
            sizeof(t_triad), 0L,A_GIMME, 0);
    addmess((method)triad_list, "list", A_GIMME, 0);
        addmess((method)triad_assist, "assist", A_CANT, 0);
        finder_addclass("Maths","triad");
        post("....I'm_TRIAD_Object!....from_AHRS_Library...",0);
        return 0;
}

/* ------------triad_new ---------------*/

void *triad_new(long value)
{
        t_triad *triad;
        triad = (t_triad *)newobject(this_class);       // create the new instance and return
            a pointer to it
        triad->m_R4 = floatout(triad);
        triad->m_R3 = floatout(triad);
        triad->m_R2 = floatout(triad);
        triad->m_R1 = floatout(triad);

return(triad);
}

etc.

etc.
```

Appendix

References

Campbell, J. (2008), *The hero with a thousand faces*, Vol. 17, New World Library. 3