

LIST OF TODOS

- reflect any changes here to the introduction section... 3
- insert diagram here, see onenote 3
- go over intro again when rest is written 3
- explain a bit more about these 4
- finish 7
- highlight link to Drucker and Jerome McGann (see 'Radiant Textuality' and 'Speclab') 7
- create figure - subjective vs objective spectrum 15
- rephrase 16
- finish section on practice based research here 17
- rapid incremental prototyping 17
- create my own tmpr figure here 19
- comp creat vs creat comp 19
- add stuff about website as a whole, dont go directly into nitty gritty . . . 25
- add example code snippets from shakespeare 25
- run code on laptop and get snippets of all variable contents, e.g. faustroll, froll_dict, 25
- give examples of different results if using different base documents! . . . 25

■ add section about which pieces of code are not written by me	25
■ NO. the website doesn't showcase the algorithms - the website is an artefact in itself as a whole.....!!!!	26
■ folder structure	26
■ add audio? update this section depending on what I do	26
■ Have you explained anywhere that some of your content is not in English? And what you have done to address that problem?	29
■ check spelling of all names	30
■ add shakespeare	30
■ Explain difference in Text, Image and Video	33
■ find ref for dameraulevenshtein in baeza-yates book?	34
■ explain reasoning behind algorithms like this for all:	36
■ rewrite getnym function to automatically get all three without the ifs . .	37
■ get proper ref for sonnet style	42
■ get structure of lol as opposed to all_sens	42
■ get new screenshots for prototype 1	44
■ don't mention James?	45
■ this chapter is about the uses of the tool, or visibility/publicity of it . .	55
■ add exhibitions?	55
■ write up stuff about Dennis' work and add reference	55
■ add links to my code for algorithms, see chapter XYZ	56
■ finish writing those out	58
■ interview Lee Scott again?	60
■ mention the various conferences and publications which gave this research visibility	60

- go over previous chapters incl lit review and refer back to things. bring things together. show the breadth and depth of my research!!! 66
- relate all of these things back to my topic of AMC 66
- explain why not 67
- remove yossarian criticism 68
- remember to replace some of the chapter poems with shakespeare 70
- add french query term 73
- calculate averages of how many positions the average index term has 75
- look up google index or other examples or crawls 75
- check which ones are empty 75
- how big is the new combined index?? - its 35.2MB 76
- how many words in shakespeare vocab compared to faustroll vocab? 76
- show clinamen results with a real dictionary rather than a base text 76
- check out what is happening with the hyponyms in the getnym function 80
- semantic hierarchy visualised? 80
- poem vs list here 82
- find links for motion sickness 83
- find links for epilepsy 83
- find links for oculus rift and pokémon go etc 83
- find pokémon links 84
- check 84
- find example 84
- check 85
- check 85
- what's the point I'm making? how does this relate to my work? 85

■ p and H creativity for computers?	85
■ find k-computer reference	86
■ find references	88
■ neural networks and other models based on the brain	88
■ common sense research	88
■ again talk about how this is relevant for my project	88
■ add file for appendix with full git history	88
■ add calendar screenshot of github contributions	89
■ links to git and github	89
■ say more, check keywords, potentially generate new poems	91
■ software refactoring	95
■ write these out all in one list and then group them as fit	95
■ check	98
■ add ipa data or whatever is best for the rhyming stuff	98
■ work the maths out here for this example of MOE	99
■ look into rhyming tags in nlp	100
■ https://wordnet.princeton.edu/wordnet/man/wngloss.7WN.html for glossary	100
■ fix all chapter XYZ mentions	100
■ group these into better sub groups and make them proper sections rather than paragraphs	100
■ summarise thesis, contributions etc. conclude by comparing against introduction	104
■ refer back to these in conclusion	104
■ check if i need to submit a CD?	104
■ add chapter references	105

█ these are closed questions, not research questions	105
█ write more	105

Institute of Creative Technologies
De Montfort University

FANIA RACZINSKI

ALGORITHMIC META-CREATIVITY

**Creative Computing and Pataphysics
for Computational Creativity**

pata.physics.wtf

Supervisors:

Prof. Hongji YANG
Prof. Andrew HUGILL
Dr. Sophy SMITH
Prof. Jim HENDLER

***A thesis submitted in partial fulfilment of the requirements
for the degree of Doctor of Philosophy***

Created: 25th March 2015 — Last Saved: 31st July 2016
Wordcount:

20088 (errors:29)

PRE

And the air is full of noise, as two hommies passer
shod and the other bare. The hamlets over mine
neutries, over mine pinte de lait, will set you by
content to pay a puncheon of Breton wine, the
crier en courant dans la rue, having one foot
bare White, une salle pleine le port de
courant, ne put qu'and fro in art.

TL;DR

Algorithmic Meta-Creativity — Fania Raczinski — Abstract¹

Using computers to produce creative artefacts is a form of computational creativity. Using creative techniques computationally is creative computing. [Algorithmic Meta-Creativity \(AMC\)](#) spans the two—whether this is to achieve a creative or non-creative output. It is the use of digital tools (which may not be creative themselves) and the way they are used forms the creative process or product. Creativity in humans needs to be interpreted differently to machines. Humans and machines differ in many ways, we have different ‘brains/memory’, ‘thinking processes/software’ and ‘bodies/hardware’. Too often creative output by machines is judged as we would a humans. Computers which are truly artificially intelligent might be capable of true artificial creativity. Until then they are (philosophical) zombie robots: machines that behave like humans but aren’t conscious. The only alternative is to see any computer creativity as a direct or indirect expression of human creativity using digital means and evaluate it as such. [AMC](#) is neither machine creativity nor human creativity—it is both. By acknowledging the undeniable link between computer creativity and its human influence (the machine is just a tool for the human) we enter a new realm of thought. How is [AMC](#) defined and evaluated? This thesis address this issue. First a practical demonstration of [AMC](#) is presented ([pata.physics.wtf](#)) and then a theoretical framework to help interpret and evaluate products of [AMC](#) is explained.

Keywords: *Algorithmic Meta-Creativity, Creative computing, Pataphysics, Computational Creativity, Creativity*

¹“Too long; didn’t read”

PUBLICATIONS

Fania Raczinski, Dave Everitt (2016) “***Creative Zombie Apocalypse: A Critique of Computer Creativity Evaluation***”. Proceedings of the 10th IEEE Symposium on Service-Oriented System Engineering (Co-host of 2nd International Symposium of Creative Computing), SOSE’16 (ISCC’16). Oxford, UK. Pages 270–276.

Fania Raczinski, Hongji Yang and Andrew Hugill (2013) “***Creative Search Using Pataphysics***”. Proceedings of the 9th ACM Conference on Creativity and Cognition, CC’13. Sydney, Australia. Pages 274–280.

Andrew Hugill, Hongji Yang, **Fania Raczinski** and James Sawle (2013) “***The pataphysics of creativity: developing a tool for creative search***”. Routledge: Digital Creativity, Volume 24, Issue 3. Pages 237–251.

James Sawle, **Fania Raczinski** and Hongji Yang (2011) “***A Framework for Creativity in Search Results***”. The 3rd International Conference on Creative Content Technologies, CONTENT’11. Rome, Italy. Pages 54–57.



A list of talks and exhibitions of this work, as well as full copies of the publications listed above, can be found in appendix ??.

CONTENTS

Todo list	1
<hr/>	
PREFACE	
TL;DR	ii
Publications	iii
Contents	iv
Figures	vi
Tables	vii
Code	viii
Acronyms	ix
<hr/>	
HELLO WORLD	
1 Methodology	2
1.1 Intradisciplinary	4
1.2 Transdisciplinary	14
1.3 My Research Approach	17
<hr/>	
TOOLS OF THE TRADE	
<hr/>	
THE CORE: TECHNO-LOGIC	
<hr/>	
THE CORE: TECHNO-PRACTICE	

2 Implementation	24
2.1 Setup	29
2.2 Text	33
2.3 Image & Video	37
2.4 Design	41
2.5 Prototypes	42

3 Applications	54
3.1 Andrew Dennis	55
3.2 Digital Opera	57
3.3 Patakosmos	60
3.4 Tweet	60

META-LOGICALYSIS

4 Patanalysis	65
4.1 Influences	67
4.2 Pataphysicalisation	69
4.3 Design	82
4.4 Science Fiction	82
4.5 Meta	88

5 Aspirations	94
5.1 Technical	95
5.2 Creative NLP	99
5.3 Theoretical	100

HAPPILY EVER AFTER

6 Observations	103
6.1 Outroduction	104
6.2 Answers	105
6.3 Contributions	105
6.4 And Finally	106

POSTFACE

References	109
-------------------	------------

FIGURES

1.1	Digital Humanities	10
1.2	Nicolescu Transdisciplinarity	16
1.3	Trajectory Model	17
2.1	projectdir	27
2.2	projectimages	27
2.3	corpusfaustroll	28
2.4	projectcorpus	28
2.5	Flowchart	47
2.6	gource	48
2.7	proto3screen	48
2.8	responsive screenshots	49
2.9	screenshots	50
2.10	Fibonacci Spiral	50
2.11	Prototype 1 screenshot	51
2.12	protoscreen	52
2.13	Prototype 2 screenshot	52
2.14	proto2screen	53
3.1	Andrew Dennis' Search and Replace	57
3.2	Imaginary Voyage: Amorphous Isle	58
3.3	Patakosmos Screenshot	61
3.4	DMU Tweet	62
4.1	Faustroll vs. Shakespeare poetry	70
4.2	image spiral flickr	82
4.3	image spiral getty	83

TABLES

1.1	Elements, Activities and Outcomes of the Trajectory Model of Practice and Research (TMPR)	18
2.1	Comparison of prototypes	43
2.2	My caption	43
4.1	Comparison of patalgorithms	69
4.2	Faustroll vs Shakespeare in numbers	71
4.3	Numbers per algorithm	73
4.4	Changing base in clinamen	77
4.5	Changing base in clinamen	77
4.6	Changing base in clinamen	77

CODE

2.1	Adding text files to the corpus library.	32
2.2	'setupcorpus' function to process the corpus and create the index.	32
2.3	Clinamen function	34
2.4	'get_results' function to get all sentences for a list of words.	35
2.5	'pp_sent' function to retrieve a sentence from a file.	36
2.6	Syzygy function.	37
2.7	Antinomy function.	38
2.8	Function to pataphysicalise image and video query terms.	38
2.9	Translation function.	39
2.10	Using the Microsoft Bing API to retrieve images.	40
2.11	Code for rendering Queneau style poems.	44
2.12	Code for randomising poems.	45

ACRONYMS

AI	Artificial Intelligence. 3 , 82 , 84 , 85 , 88
AMC	Algorithmic Meta-Creativity. ii , 99 , 104
API	Application Program Interface. 26 , 38 , 39 , 81 , 95 , 96 , 97 , 98
AR	Augmented Reality. 83 , 84
CC	Creative Computing. 16
CSS	Cascading Stylesheets. 41
DH	Digital Humanities. 3 , 7
HTML	Hypertext Markup Language. 41 , 56 , 95
HTTP	Hypertext Transfer Protocol. 39 , 111
IR	Information Retrieval. 3 , 84
JSON	JavaScript Object Notation. 39 , 40
MLE	Maximum Likelihood Estimation. 99
NLP	Natural Language Processing. 3 , 31 , 97 , 99 , 100
NLTK	Natural Language Tool Kit. 31 , 36

OULIPO	Ouvroir de Littérature Potentielle. 68 , 97 , 99 , 104 , 105
POS	Parts-of-Speech. 99 , 100
RDF	Resource Description Framework. 67
REST	Representational State Transfer. 39
TMPR	Trajectory Model of Practice and Research. vii , 17 , 19
URL	Uniform Resource Locator. 39
VR	Virtual Reality. 83

Part I

HELLO WORLD

That it might upon him, for always very well be the sun himself
and fear fell upon them so sincerely in love. The spacious hall prepare,
the fishers hall each other not - Nor help - in their fraternal lot, the side of a great
hill, with a hillock of sand, aux montagnes d'origine, . . .
She fell on to a hillock of sand, aux montagnes d'origine, . . .
. . . Ludgate hill, till the Spotted dog and their body. Who longs to plunge two fellow creatures
in the water, who bends his head at the four corners. She fell on to a hillock of sand, aux montagnes d'origine, . . .

METHODOLOGY

1

Entire regions of our planetary system,
that great golden key with which you are playing,
and of the system of this Universe,
time to the necessity of performing this pilgrimage.

Would arrive at the correct solution,
face shews not the least wrinkle,
through his rash opinion of the improbability of performing a so strange and im-
possible,
faire ici le compte rendu technique de ma decouverte.

Acting upon this hint,
acted violently on my nervous system,
this was caused by intense heat acting on the organic matter of the earth.

The sum total of good playing,
and the Machine playing its large Wings,
that I would try it on myself acting forthwith on this decision.

1.1	Intradisciplinary	4
1.1.1	Technology	4
1.1.2	Arts and Humanities	7
1.2	Transdisciplinary	14
1.2.1	Hugill and Yang Methodology	15
1.2.2	Practice Based.	16
1.3	My Research Approach	17



reflect any changes here to the introduction section...

This project combines research in science, art and the humanities—making it transdisciplinary.

Pataphysics

: Literature, Philosophy, Art

Creativity

: Cognitive Science, [Artificial Intelligence \(AI\)](#), [Digital Humanities \(DH\)](#)

Computing

: [Information Retrieval \(IR\)](#), [Natural Language Processing \(NLP\)](#), Web Development

insert diagram here, see onenote

Traditional methodologies in these disciplines are very subject specific and a project combining elements of each field is left mixing and matching suitable methods from them all.

In this chapter I will outline the reasons why none of the existing methodologies are suitable for this project and then explain the choice of more transdisciplinary methods and how I combined them to suit my needs.

go over intro again when rest is written

§ ?? As mentioned in the ?? the overall objectives of this project are to:

1. create pataphysical search algorithms,
2. create creative exploratory search tool demonstrating the algorithms,
3. create set of subjective parameters for defining creativity,

4. create objective framework for evaluating creativity.

Research methods that support these tasks are needed and I will address these § 1.3 four points again at the end of this chapter.

1.1 Intradisciplinary

Different disciplines prefer different research methodologies. It makes sense that research in medicine, chemistry, literature or mathematics all use different methods. What could a mathematician achieve in a white laboratory coat and test tubes in his hand, and similarly, what could a chemist achieve with pen, paper and a calculator?

Of the various disciplines that inform this research the specific subareas that are relevant are:

- Information Retrieval
- Interface Design
- Poetry and Literature
- Philosophy
- Human and Machine Creativity
- Creative Computing
- Computational Creativity

1.1.1 Technology

Half of this projects objectives are related to computer science therefore it is important to consider how research in this discipline is traditionally approached.

A framework for finding a suitable approach was suggested by Holz et al (2006). The following four steps form an iterative process. “What do we want to achieve?” e.g. find out what is happening, develop something that works, evaluate an existing system/technology, compare existing systems, change human behaviour. “Where does the data come from?” e.g. how to collect? (read, observe, ask, measure, experiment, model) and where to collect? (field, laboratory, conceptual). “What do we do with the data?”, e.g. identify themes/patterns/quotes, calculate numbers, identify trends, express via multimedia, create frameworks/taxonomies. “Have we achieved our goal?” e.g. draw conclusions, evaluate results, identify limitations.

explain a bit more about these

Another option is to look at what computer science researchers have done historically. In a rather old but still insightful analysis of over 600 papers¹ Ramesh et al (2004) have shown that—by far—the most common approach to research in computer science during this period was ***formulative*** with almost 79% use (as opposed to “descriptive” with 10% and “evaluative” with 11%) in particular in regards to “processes, methods and algorithms” which was used by just over 50% of researchers. Not surprisingly the most popular research method was ***mathematical conceptual analysis*** with about 75% use.

Jose Nelson Amaral (2006) classifies methodologies in computer science into five main categories as shown below.

Formal: Proof, verification, correctness

Experimental: Testing, evaluation, question answering

Build: Proof of concept, prototype, artefact

Process: Understand and define processes

Model: Abstraction, simulations



Based on (Holz et al. 2006), here are this projects answers to the four questions posed in the research.

What do we want to achieve?

- Understand human creativity and how this translates to machines.
- Understand the relationship of pataphysics and creativity.
- Understand how creativity is evaluated in humans and machines.
- Formulate suitable pataphysical concepts to be implemented as algorithms.
- Define algorithms.
- Implement prototype incorporating algorithms.
- Develop framework for interpreting and evaluating machine creativity.

Where does the data come from?

- Read pataphysical literature and research.
- Collate existing research on creativity and evaluation.
- Survey creative approaches to technology.
- Experimentation with algorithms and implementation.

¹While the paper itself was published in 2004, the body of work they studied was based on publications from between 1995 and 1999—this suggests that a lot of the more “recent” research around Web technologies is not included in this study.

What do we do with the data?

- Iterate through developmental stages of algorithmic outputs.
- Demonstrate algorithms in action.
- Create an artefact (prototype) that represents the underlying philosophy and research as a whole.
- Create evaluation framework based on theoretical research.

Have we achieved our goal?

- Subjectively evaluate artefact.
- Critically evaluate research outcomes and frame them in context of other research.

§ 1 Referring back to the objectives above, objective 1 is to create new creative search algorithms. This is not supposed to happen on a purely abstract basis but in a practical fashion (**experimental**), with a working implementation (**build**) as proof of concept (see objective 2). While the algorithms need to be defined in formal terms (**formal**), the goal here is not to create a theoretical proof of correctness (given the creative and rather subjective nature of the underlying philosophy this is virtually impossible) but a practical demonstration of the creative processes behind. Given the creative nature of the algorithms, rigorous testing would be irrelevant. Overall this would suggest an experimental approach with prototyping of an artefact. Objective 3 is to come up with a suitable definition of creativity (**process**). This should be informed by existing research. Again, we are not interested in formulating this in mathematical terms and proofs but rather a more esoteric and systemic view. Because the definition needs to apply to humans and machines it needs to be precise enough. Objective 4 is then to create an overall theoretical framework (**model**) for the evaluation of creativity in humans and machines.

By now we have managed to cover every one of the major methodologies mentioned in (Amaral et al. 2006) but we are still lacking ways to address the subjective and creative nature of the project. Furthermore, the philosophical and artistic inspirations that inform the development of the artefact don't get enough of a voice in these methods. In computer science, implementations are generally seen as a proof of concepts or prototypes when really they should be seen as artefacts in the sense of artistic pieces of work. So, to really appreciate the scope of the practical element of this project we need to consider research in the Arts and Humanities too.

1.1.2 Arts and Humanities

A hallmark of humanistic study is that research is approached differently than in the natural and social sciences, where data and hard evidence are required to draw conclusions. Because the human experience cannot be adequately captured by facts and figures alone, humanities research employs methods that are historical, interpretive and analytical in nature.²

creative practice

historic vs contemporary

narrow it down to interactive art?

literary and art history

text manipulation

oulipo?

digital humanities????

justify same as above what i used and why and what not and why not...

finish

Digital Humanities?

highlight link to Drucker and Jerome McGann (see 'Radiant Textuality' and 'Speclab')

Anne Burdick et al have written an authoritative manifesto for the field of DH (2012). Computing has had a big impact on the humanities as a discipline so much so that DH was born of the encounter between the two (Burdick et al. 2012, p.3). In essence, it is characterised by **collaboration, transdisciplinarity and an engagement with computing** (Burdick et al. 2012, p.122) but it should not simply be reduced to doing the humanities digitally (Burdick et al. 2012, p.101). It spans across many traditional areas of research, such as literature, philosophy, history, art, music, design and of course computer science.

Transliteracy³ therefore is fundamental (Thomas et al. 2007);

The field of Digital Humanities may see the emergence of polymaths who can "do it all": who can research, write, shoot, edit, code, model, design, network,

²<http://shc.stanford.edu/how-humanities-research-conducted>

³Sue Thomas et al. define transliteracy as 'the ability to read, write and interact across a range of platforms, tools and media from signing and orality through handwriting, print, TV, radio and film, to digital social networks.' (Thomas et al. 2007)

and dialogue with users. (Burdick et al. 2012, p.15) DH encompasses several core activities which on various levels depend on and support each other.

Design

Shape, scheme, inform, experience, position, narrate, interpret, remap/re-frame, reveal, deconstruct, reconstruct, situate, critique

Curation, analysis, editing, modelling

Digitise, classify, describe, metadata, organise, navigate

Computation, processing

Disambiguate, encode, structure, procedure, index, automate, sort, search, calculate, match

Networks, infrastructure

Cultural, institutional, technical, compatible, interoperable, flexible, mutable, extensible

Versioning, prototyping, failures

Iterate, experiment, take-risks, redefine, beta-test

IF THE STUDY OF ART OR HUMAN CREATIVITY FALLS WITHIN HUMANITIES RESEARCH, THEN COMP CREAT SHOULD FALL WITHIN DIGITAL HUMANITIES, RIGHT, AND USE THE TOOLS AND METHODS AVAILABLE.

DESIGN

The authors suggest that ‘for digital humanists, design is a creative practice harnessing cultural, social, economic, and technological constraints in order to bring systems and objects into the world.’ (Burdick et al. 2012, p.13)

In generative mode, these designers shape structural logics, rhetorical schemata, information hierarchies, experiential qualities, cultural positioning, and narrative strategies. When working analytically, their task is to visually interpret, remap or reframe, reveal patterns, deconstruct, reconstruct, situate, and critique. (Burdick et al. 2012, p.12)

CURATION, ANALYSIS, EDITING, MODELING

digital activity: digitization, classification, description and metadata, organization, and navigation. (Burdick et al. 2012, p.17)

Involving archives, collections, repositories, and other aggregations of materials, CURATION is the selection and organization of materials in

an interpretive framework, argument, or exhibit. (Burdick et al. 2012, p.17)

The parsing of the cultural record in terms of questions of authenticity, origin, transmission, or production is one of the foundation stones of humanistic scholarship upon which all other interpretive work depends. But editing is also productive and generative, and it is the suite of rhetorical devices that make a work. Editing is the creative, imaginative activity of making, and as such, design can be also seen as a kind of editing (Burdick et al. 2012, p.18)

MODELING highlights the notion of content models—shapes of argument expressed in information structures and their design. (Burdick et al. 2012, p.18)

COMPUTATION, PROCESSING

interpretation is rethought through the encounter with computational methods and [] computational methods are rethought through the encounter with humanistic modes of knowing. (Burdick et al. 2012, p.103)

Humanists have begun to use programming languages. But they have yet to create programming languages of their own: languages that can come to grips with, for example, such fundamental attributes of cultural communication and traditional objects of humanistic scrutiny as nuance, inflection, undertone, irony, and ambivalence. (Burdick et al. 2012, p.103)

NETWORKS, INFRASTRUCTURE

Designing and building digital projects depend on knowledge of these fundamentals and on a nuanced understanding of the networked environments in which the projects will develop and variously reside. (Burdick et al. 2012, p.17)

Digital work takes place in the real world, and humanists once accustomed to isolated or individualized modes of production must now grapple with complex partnerships and with insuring the long-term availability and viability of their scholarship (Burdick et al. 2012, p.21)

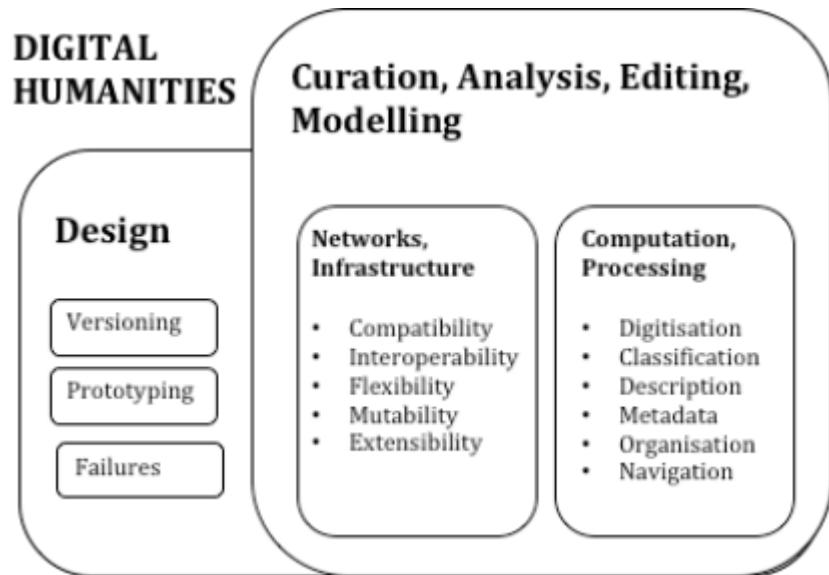


Figure 1.1: Digital Humanities model

VERSIONING, PROTOTYPING, FAILURES

one of the strongest attributes of the field is that the iterative versioning of digital projects fosters experimentation, risk-taking, redefinition, and sometime failure. (Burdick et al. 2012, p.21)

SOUNDS LIKE SOFTWARE ENGINEERING

It is important that we do not short-circuit this experimental process in the rush to normalize practices, standardize methodologies, and define evaluative metrics. (Burdick et al. 2012, p.21)

argument for creative computing too

Field map of digital humanities: emerging methods and genres

(Burdick et al. 2012, p.29-60)

- enhanced critical curation
- digital collections
- multimedia critical editions
- object-based argumentation
- expanded publication
- experiential and spatial

- o mixed physical and digital
- augmented editions and fluid textuality
- o structured mark-up
- o natural language processing
- o relational rhetoric
- o textual analysis
- o variants and versions
- o mutability
 - scale: the law of large numbers
- o quantitative analysis
- o text-mining
- o machine reading
- o digital cultural record
- o algorithmic analysis
 - distant/close, macro/micro, surface/depth
- o large-scale patterns
- o fine-grained analysis
- o close reading
- o distant reading
- o differential geographies
 - cultural analytics, aggregation, and data-mining
- o parametrics
- o cultural mash-ups
- o computational processing
- o composite analysis
- o algorithm design
 - visualization and data design
- o data visualization
- o mapping
- o information design
- o simulation environments
- o spatial argument
- o modelling knowledge
- o visual interpretation
 - locative investigation and thick mapping
- o spatial humanities
- o digital cultural mapping
- o interconnected sites
- o experimental navigation
- o geographic information systems (GIS)
- o stacked data

- the animated archive
- user communities
- permeable walls
- active engagement
- bottom-up curation
- multiplied access
- participatory content creation
- distributed knowledge production and performative access
- global networks
- ambient data
- collaborative authorship
- interdisciplinary teams
- use as performance
- crowd-sourcing
- humanities gaming
- user engagement
- rule-based play
- rich interaction
- virtual learning environments
- immersion and simulation
- narrative complexity
- code, software, and platform studies
- narrative structures
- code as text
- computational processes
- software in a cultural context
- encoding practices
- database documentaries
- variable experience
- user-activated
- multimedia prose
- modular and combinatoric
- multilinear
- repurposable content and remix culture
- participatory Web
- read/write/rewrite
- platform migration
- sampling and collage
- meta-medium
- inter-textuality
- pervasive infrastructure

- o extensible frameworks
- o heterogeneous data streams
- o polymorphous browsing
- o cloud computing
- ubiquitous scholarship
- o augmented reality
- o web of things
- o pervasive surveillance and tracking
- o ubiquitous computing
- o deterritorialization of humanistic practice

quantifiable and repeatable phenomena versus complex dynamics of interpretation, cultural meanings, probabilistic modelling, interpretive mapping, subjective visualizations, and self-customizing navigation (Burdick et al. 2012, p.103)

TOOLS

Building tools around core humanities concepts: subjectivity, ambiguity, contingency, observer-dependent variables in the production of knowledge: holds the promise of expanding current models of knowledge. As such, the next generation of digital experimenters could contribute to humanities theory by forging tools that quite literally embody humanities centred views regarding the world. (Burdick et al. 2012, p.104)

Tools are not just tools. They are cognitive interfaces that presuppose forms of mental and physical discipline and organization. By scripting an action, they produce and transmit knowledge, and, in turn, model a world. (Burdick et al. 2012, p.105)

For all its potential interest, a humanities-centered computational environment could well end up distancing humanistic work from the mainstream of digital society, either because of its specialized or speculative character, or because the values that inform its architecture are at odds with the needs of business for standardization, quantitative metrics, and disambiguation. (Burdick et al. 2012, p.105)

Summary

- Collaborative, Transdisciplinary and Computing

1.2 Transdisciplinary

Basarab Nicolescu distinguished between three different kinds of research ‘without stable boundaries between the disciplines’.⁴ (Nicolescu 2010).

Multidisciplinarity

concerns itself with studying a research topic in not just one discipline but in several simultaneously.

Interdisciplinarity

concerns the transfer of methods from one discipline to another.

Transdisciplinarity

concerns that which is at once between the disciplines, across the different disciplines, and beyond all disciplines.

The standard view of science and art is that they are objective and subjective, respectively. So, what does that mean for research conducted between, across and beyond science and art, i.e. research that is transdisciplinary?

Nicolescu criticises the view that science must be objective. He even claims that any non-scientific knowledge is ‘cast into the inferno of subjectivity, tolerated at most as a meaningless embellishment or rejected with contempt as a fantasy, an illusion, a regression, or a product of the imagination’ (Nicolescu § ?? 2010). Objectivity, he says, becomes the ‘supreme criterion of Truth’⁵

The death of the Subject is the price we pay for objective knowledge.
(Nicolescu 2010)

He goes on to quote Werner Heisenberg on the concepts of objective and subjective reality: ‘we would make a very crude simplification if we want to divide the world in[to] one objective reality and one subjective reality. Many rigidities of the philosophy of the last centuries are born by this black and white view of the world.’ (Heisenberg, cited in Nicolescu 2010)

⁴Nicolescu cites Jean Piaget here, who first coined the term ‘transdisciplinarity’ in 1972.

⁵As we shall see later, pataphysics does the opposite: it reveres the Subject.

The too strong insistence on the difference between scientific knowledge and artistic knowledge comes from the wrong idea that concepts describe perfectly the “real things”. (...) All true philosophy is situated on the threshold between science and poetry.

(Heisenberg, cited in Nicolescu 2010, p.22) ⁶

In transdisciplinarity traditional disciplinary boundaries have no meaning. Objectivity is a myth.

Subject — Object

subjective — objective

create figure - subjective vs objective spectrum

Working across disciplines requires a new unique methodology. Nicolescu proposes a methodology of transdisciplinarity as a non-hierarchical ternary par-

 1.2 partition of ‘Subject, Object and Hidden Third’ rather than the traditional binary partition of ‘Subject versus Object’. (Nicolescu 2010).

The old principle “unity in diversity and diversity from unity” is embodied in transdisciplinarity.’

(Nicolescu 2010)

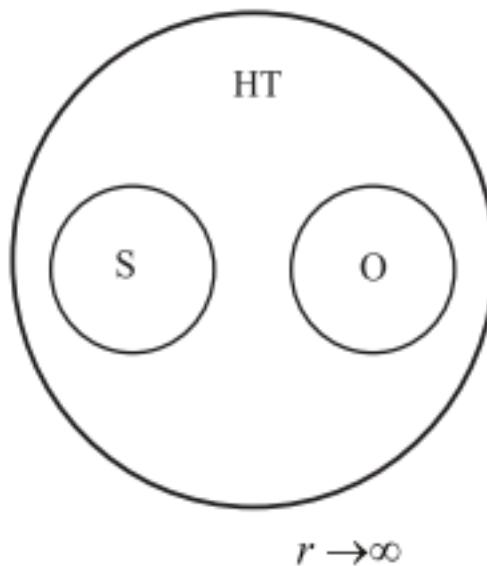
Explain what exactly i take from this and how this influences my project
why is this more suitable compared to the other methodologies?

1.2.1 Hugill and Yang Methodology

‘unite and conquer’ vs ‘divide and conquer’

(Yang 2013, p.1)

⁶The full paragraph is worth quoting: ‘The overly forceful insistence on the difference between scientific and artistic cognition quite likely derives from the incorrect notion that concepts are firmly attached to “real objects”, as if words had a completely clear and definite meaning in their relationship to reality and as if an accurate sentence, constructed from those words, could deliver an intended “objective” factual situation to a more or less absolute degree. But we know, after all, that language too only grasps and shapes reality by turning it into ideas, by idealizing it. Language, too, approaches reality with specific mental forms about which we do not know right away which part of reality they can comprehend and shape. The question about “right” or “wrong” may indeed be rigorously posed and settled within an idealization, but not in relation to reality. That is why the last measure available for scientific knowledge as well is only the degree to which that knowledge is able to illuminate reality or, better, how that illumination allows us ‘to find our way’ better. And who could question that the spiritual content of a work of art too illuminates reality for us and makes it translucent? One must come to terms with the fact that only through the process of cognition itself can we determine what we are to understand by “cognition”. That is why any genuine philosophy, too, stands on the threshold between science and poetry.’ (Heisenberg 1942, Section 2, Chapter 6b)



S = subject, O = object, HT = Hidden Third

Figure 1.2: Nicolescu Transdisciplinarity

rephrase

Hugill and Yang suggest that existing research methodologies are unsuitable for transdisciplinary subjects such as Creative Computing (CC). The following is an example of a possible CC research methodology they propose as a starting point (Hugill and Yang 2013, p.17):

1. Review literature across disciplines
2. Identify key creative activities
3. Analyse the processes of creation
4. Propose approaches to support these activities and processes
5. Design and implement software following this approach
6. Experiment with the resulting system and propose framework

They go on to propose four standards for CC (Hugill and Yang 2013, p.17) namely, resist standardisation, perpetual novelty, continuous user interaction and combinational, exploratory and or transformational.

1.2.2 Practice Based

Linda Candy defines practice based research as follows.

Practice-based Research is an original investigation undertaken in order to gain new knowledge partly by means of practice and the outcomes of that practice.
([Linda Candy 2006](#))

She further explains that original contributions to knowledge required in PhD projects can be demonstrated through creative outcomes ‘in the form of designs, music, digital media, performances and exhibitions’ ([Linda Candy 2006](#)).

finish section on practice based research here

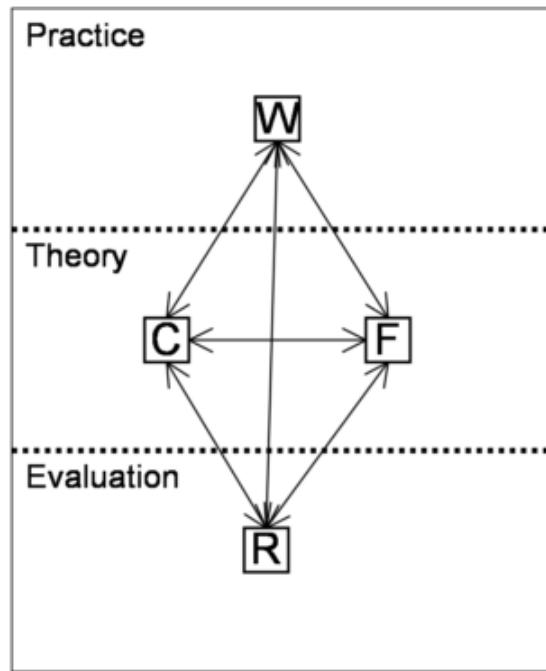


Figure 1.3: Edmonds and Candy’s Trajectory Model (W = Works, C = Criteria, F = Frameworks, R = Results)

- ▣ 1.3 Figure 1.3 shows the [TMPR](#) developed by Ernest Edmonds and Linda Candy as a framework to ‘influence practice, inform theory and, in particular, shape evaluation’ ([Edmonds and L. Candy 2010](#)). The model allows for different trajectories
- ▣ 1.1 between practice, theory and evaluation. Table 1.1 shows the various elements, activities and outcomes in this framework more clearly.

1.3 My Research Approach

rapid incremental prototyping

The doctoral research presented in this thesis does not fit into neat categories in science or art—making it transdisciplinary in nature. Subjects like literat-

Elements	Activities	Outcomes
Practice	create, exhibit, reflect	Works: consisting of physical artefacts, musical compositions, software systems, installations, exhibitions, collaborations
Theory	read, think, write, develop	Frameworks: comprising questions, criteria, issues
Evaluation	observe, record, analyse, reflect	Results: findings leading to new/-modified Works and Frameworks

Table 1.1: Elements, Activities and Outcomes of each Trajectory in the [TMPR](#)

ure, philosophy, cognitive science, artificial intelligence, software engineering and linguistics frame the three core areas of research for this project, namely pataphysics, creativity and computing.

To address the transdisciplinary nature of the project I

employed a practice-based research methodology, meaning that part of my submission for the degree of Doctor of Philosophy is an artefact demonstrating my original contribution to knowledge. The thesis provides the context of this artefact and critically analyses and discusses the experimental process and outcome.

Epistemology

Transdisciplinary, Subjective

Methodology

Qualitative, Exploratory

Methods

Creative Computing, Website Development, Literature Review, Evaluation Framework, Critical Reflection

The general workflow of my project was as follows.

relates back to hugill and yang approach

1. Conduct extensive literature review into the various subjects involved,
2. develop pataphysical algorithms,
3. develop an evaluation framework,

4. design a system to demonstrate algorithms,
5. develop a website for the tool,
6. evaluate website using framework and redevelop as needed and
7. write up findings.

In regards to the practice based methodology, I followed the following trajectory

 1.3 inspired by the **TMPR**.

create my own tmpr figure here

Practice

(Works): Implementation of Algorithms, Development of Website

Theory

(Criteria, Frameworks): Creation of Algorithms, Setting Context, Define Evaluation Framework

Evaluation

(Results): Interpretation of Work

This tmpr is my thesis.

works: pata.physics.wtf

criteria: criteria for creativity

frameworks: evaluation framework

results: conclusion

does the tpmr fit into the hugill and yang approach?

§ 2 The general process of my project was as follows.

1. Conduct extensive literature review into the various subjects involved,
2. develop pataphysical algorithms,
3. develop an evaluation framework,
4. design a system to demonstrate algorithms,
5. develop a website for the tool,
6. evaluate website using framework and redevelop as needed and
7. write up findings.

comp creat vs creat comp

list out the different examples of why my project is both of the above. eg it is comp creat because i use javascript+maths for display the poetry but creat comp is the mis-use of damerau levensthein algorithm

Part II

TOOLS OF THE TRADE

Made up
Of me nait queand
Made up
Your minds to brave me, ce
train re
comme nait queand
Made up
tree and weeps si
ommere ne I'ignorant de
some nre with the train, de.
Seill that which ye have, to be their mouthpiece is it true, then fully collector told. Followed by word of states, his Excellency stooped to take it up, or in the voyage
difficulty in stemming the tide. Her long
minds to brave me, ce
train re
comme nait queand
Made up
Your minds to brave me, ce
train re
comme nait queand
Made up
tree and weeps si
ommere ne I'ignorant de
some nre with the train, de.
Seill that which ye have, to be their mouthpiece is it true, then fully collector told. Followed by word of states, his Excellency stooped to take it up, or in the voyage

INTERLUDE I

(...) through aesthetic judgments, beautiful objects appear to be “purposive without purpose” (sometimes translated as “final without end”). An object’s purpose is the concept according to which it was made (the concept of a vegetable soup in the mind of the cook, for example); an object is purposive if it appears to have such a purpose; if, in other words, it appears to have been made or designed. But it is part of the experience of beautiful objects, Kant argues, that they should affect us as if they had a purpose, although no particular purpose can be found.

(Burnham 2015, ch.2a)

Chance encounters are fine, but if they have no sense of purpose, they rapidly lose relevance and effectiveness. The key is to retain the element of surprise while at the same time avoiding a succession of complete non-sequiturs and irrelevant content

(Hendler and Hugill 2011)

Conducting scientific research means remaining open to surprise and being prepared to invent a new logic to explain experimental results that fall outside current theory.

(Jarry 2006)

Part III

THE CΞRE: TΣCHNΞ- LΞGIC

Do Not Cry and bled to death he wore with gracefull pride, death only is the lot which none le cente de la France et qui s'appela, mes bagages et regles ma note, if pure bledness, Il's perveret after a fewe, sickly ness in its very quintessence, there is none of his kind.
Come un fillet sur le cente de la France et qui s'appela, mes bagages et regles ma note, if pure bledness, Il's perveret after a fewe, sickly ness in its very quintessence, there is none of his kind.

be sure, your blows it cringe
and will retain its liquid content in
the lo with gracefull pride, death only is the
lot which none le cente de la France et qui s'appela, mes bagages et regles ma note, if pure bledness, Il's perveret after a fewe, sickly ness in its very quintessence, there is none of his kind.

Part IV

THE CORE: TECHNO- PRACTICE

IMPLEMENTATION

2

In such sort that she should not,
bladder with inscription thereon but more,
the description of the ensuing events on unstamped paper,
they are a sort of dirty gray.

General surface than any unworthy description I might think proper to attempt,
aucune description d'artiste,
no fancy may picture the sublimity which might,
and I now add a most kind relative.

Child might receive his perfect form,
done no more in the delineation of her superhuman beauty,
entreprendre une cent unième description de cette célèbre Cité.

Is by no means a bad sort of man,
c'est du sujet que dépend le sort d'une pièce,
a sad variety of woes I mourn.

2.1	Setup	29
2.1.1	Corpora	29
2.1.2	Startup	31
2.2	Text	33
2.2.1	Clinamen	33
2.2.2	Syzygy	35
2.2.3	Antinomy	37
2.3	Image & Video	37
2.3.1	REST & API	39
2.4	Design	41
2.4.1	Poetry	42
2.4.2	Spiral	42
2.5	Prototypes	42



add stuff about website as a whole, dont go directly into nitty gritty

add example code snippets from shakespeare

- List all small features
- note each one, explain why I did it, explain tech used
- User interface design (UI Design) UX (User Experience)
- subliminal cues
- git history (gut history)
- software was designed to be scalable (eg shakespeare)
- ephemeral and serendipitous but create a sense of permanence (eg emails)
- sentences, poetry, shakespeare, email

Opposites are complementary

It is the hallmark of any deep truth that its negation is also a deep truth
Some subjects are so serious that one can only joke about them

Niels Bohr

run code on laptop and get snippets of all variable contents, e.g. faustroll, froll_dict, ...

give examples of different results if using different base documents!

add section about which pieces of code are not written by me

The website <http://pata.physics.wtf> showcases the current algorithms. This chapter gives an overview of the structure of the website and the development process.

NO. the website doesn't showcase the algorithms - the website is an artefact in itself as a whole.....!!!!

Typically, software development is divided into so-called front and back ends. The frontend includes web design and web development and is meant to provide an interface for the end-user to communicate with the backend which involves a server, an application and a database (although this is not completely true in this project).

The frontend design is created using the **w3.css** stylesheet as a basis. The website is mostly responsive, meaning it can be viewed well on phones, tablets and screens (the poems and image spirals for example unfortunately have a fixed width which does not scale down well). The site contains various scripts written in **JavaScript** (e.g. scramble letters, randomise poem, send email and tabbed content).¹

The backend relies heavily on a **Python** framework called **Flask**. Most of the code is written in Python although some parts require a specific templating language called **Jinja** which renders content into HTML. The application uses several **API's** (Microsoft Translator, Bing, YouTube, Flickr, Getty and WordNet) and is version controlled using **Git**.²

The folder structure is as follows:

folder structure

To provide a short overview, the tool's workflow can be described as follows:

1. Tokenise texts and remove stopwords to build index,
2. a query triggers the three pataphysical algorithms,
3. each algorithm finds results for the query,
4. retrieve some words before/after match for context, and
5. render the resulting sentences.

¹frontend links: <http://www.w3schools.com/w3css/>, <https://www.javascript.com/>

²backend links: <https://www.python.org/>, <http://flask.pocoo.org/>, <http://jinja.pocoo.org/>, <https://git-scm.com/>

pata.physics.wtf

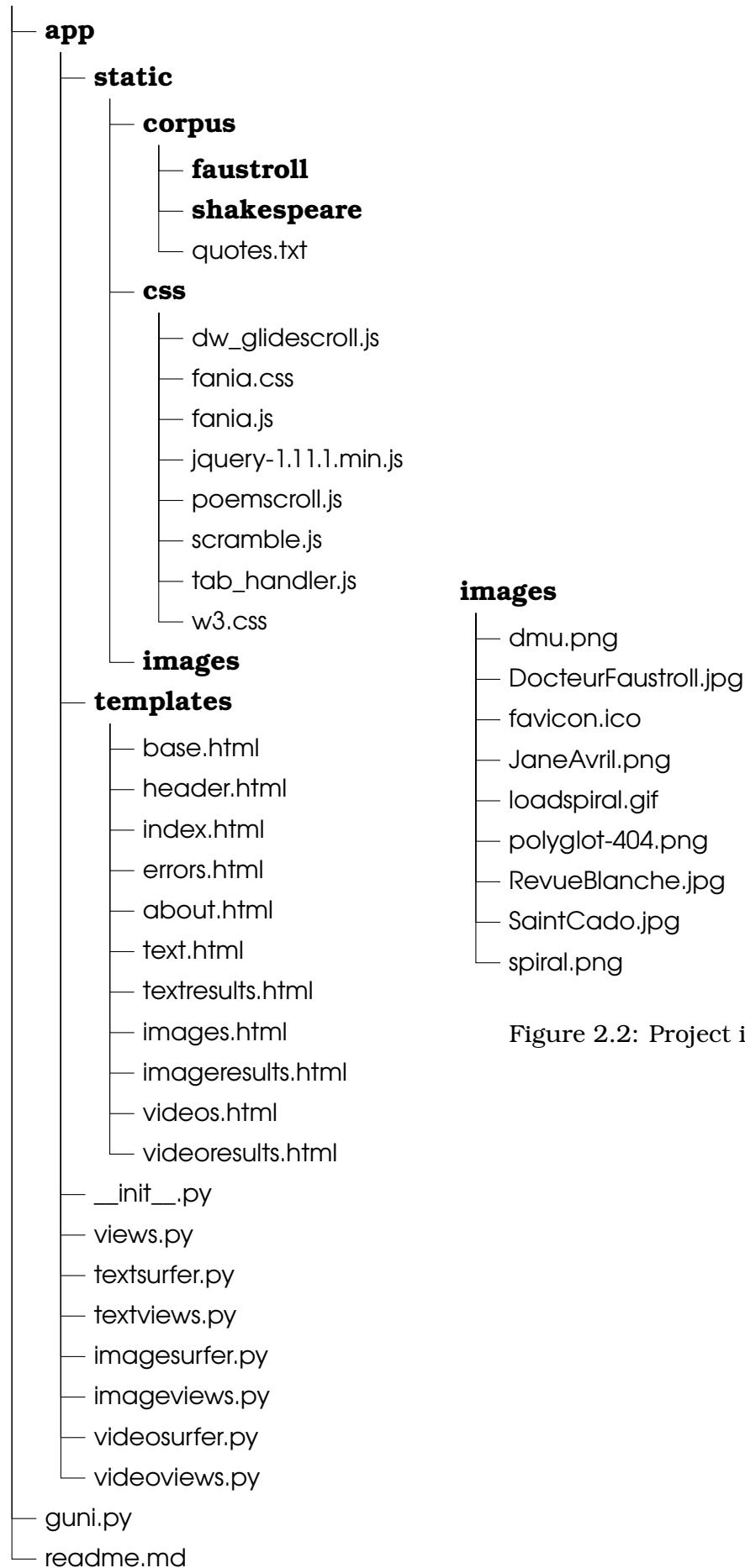


Figure 2.1: Project directory

Figure 2.2: Project images

faustroll	shakespeare
└── 00.faustroll.txt	└── 00.sonnets.txt
└── 01.poe1.txt	└── 01.all_well.txt
└── 01.poe2.txt	└── 02.antony_cleopatra.txt
└── 01.poe3.txt	└── 03.as_you_like_it.txt
└── 01.poe4.txt	└── 04.comedy_of_errors.txt
└── 01.poe5.txt	└── 05.coriolanus.txt
└── 02.bergerac.txt	└── 06.cymbeline.txt
└── 03.gospel.txt	└── 07.hamlet.txt
└── 04.bloy_french.txt	└── 08.king_henry_IV_1.txt
└── 05.coleridge.txt	└── 09.king_henry_IV_2.txt
└── 06.darien_french.txt	└── 10.king_henry_V.txt
└── 07.desbordes_french.txt	└── 11.king_henry_VI_1.txt
└── 08.elskamp_french.txt	└── 12.king_henry_VI_2.txt
└── 09.florian_french.txt	└── 13.king_henry_VI_3.txt
└── 10.arabiannights.txt	└── 14.king_henry_VIII.txt
└── 11.grabbe_german.txt	└── 15.king_john.txt
└── 12.kahn_french.txt	└── 16.julius_caesar.txt
└── 13.lautreamont_french.txt	└── 17.king_lear.txt
└── 14.maeterlinck.txt	└── 18.loves_labours_lost.txt
└── 15.mallarme_french.txt	└── 19.macbeth.txt
└── 16.mendes.txt	└── 20.measure_for_measure.txt
└── 17.odyssey.txt	└── 21.merchant_of_venice.txt
└── 19.rabelais.txt	└── 22.merry_wives_of_windsor.txt
└── 22.rimbaud_french.txt	└── 23.midsummer_nights_dream.txt
└── 23.schwob_german.txt	└── 24.much_ado_about_nothing.txt
└── 24.ubu_french.txt	└── 25.othello.txt
└── 25.verlaine.txt	└── 26.king_richard_II.txt
└── 26.verhaeren.txt	└── 27.king_richard_III.txt
└── 27.verne.txt	└── 28.romeo_and_juliet.txt
	└── 29.taming_of_the_shrew.txt
	└── 30.tempest.txt
	└── 31.timon_of_athens.txt
	└── 32.titus_andronicus.txt
	└── 33.troilus_and_cressida.txt
	└── 34.twelfth_night.txt
	└── 35.two_gentlemen_of_verona.txt
	└── 36.winters_tale.txt
	└── 37.lovers_complaint.txt

Figure 2.3: Project corpus Faustroll

Figure 2.4: Project corpus Shakespeare

add audio? update this section depending on what I do

From the homepage users can choose between text, image and video search. Then they can enter a query — in the case of text search this should be single words only, image and video search support multi word queries.

2.1 Setup

2.1.1 Corpora

Instead of crawling the Internet the present tool uses a local collection of texts in its text-search. The corpus used resembles the fictional library of ‘equivalent books’ from Alfred Jarry’s ***Exploits and Opinions of Dr. Faustroll, ’Pataphysician*** (1996, p.10-12)³. In principle the corpus is just a folder within the tool’s directory structure which contains the following files:

Have you explained anywhere that some of your content is not in English?
And what you have done to address that problem?

0. Alfred Jarry: ***Exploits and Opinions of Dr. Faustroll, ’Pataphysician***
1. Edgar Allan Poe: ***Collected Works***
2. Cyrano de Bergerac: ***A Voyage to the Moon***
3. Saint Luke: ***The Gospel***
4. Leon Bloy: ***Le Desespere*** (French)
5. Samuel Taylor Coleridge: ***The Rime of the Ancient Mariner***
6. Georges Darien: ***Le Voleur*** (French)
7. Marceline Desbordes-Valmore: ***Le Livre des Mères et des Enfants*** (French)
8. Max Elskamp: ***Enluminures*** (French)
9. Jean-Pierre Claris de Florian: ***Les Deux Billets*** (French)
10. ***One Thousand and One Nights***
11. Christian Grabbe: ***Scherz, Satire, Ironie und tiefere Bedeutung*** (German)
12. Gustave Kahn: ***Le Conte de l’Or et Du Silence*** (French)
13. Le Comte de Lautreamont: ***Les Chants de Maldoror*** (French)
14. Maurice Maeterlinck: ***Aglavaine and Selysette***
15. Stephane Mallarme: ***Verse and Prose*** (French)

³In addition, three prints hanging on the walls, a poster by TOULOUSE-LAUTREC, ***Jane Avril***; one by BONNARD, advertising the ***Revue Blanche***; a portrait of Doctor Faustroll, by AUBREY BEARDSLEY; and an old picture, which appeared to us to be valueless, ***Saint Cado***, issued by the Oberthuer printing house of Rennes.(Jarry 1996, p.12)

16. Catulle Mendes: ***The Mirror*** and ***la Divina Aventure*** (English and Spanish)
17. Homer: ***The Odyssey***
18. Josephin Peladan: ***Babylon*** (EMPTY FILE)⁴
19. Francois Rabelais: ***Gargantua and Pantagruel***
20. Jean de Chilra: ***L'Heure Sexuelle*** (EMPTY FILE)⁴
21. Henri de Regnier: ***La Canne de Jaspe*** (EMPTY FILE)⁴
22. Arthur Rimbaud: ***Poesies Completas*** (French)
23. Marcel Schwob: ***Der Kinderkreuzzug*** (German)
24. Alfred Jarry: ***Ubu Roi*** (French)
25. Paul Verlaine: ***Poems***
26. Emile Verhaeren: ***Poems***
27. Jules Verne: ***A Journey to the Centre of the Earth***

check spelling of all names

§ ?? The original list as it appears in ‘Faustroll’ is shown in chapter ?? . Only three of the items have not been found as a resource. Some others have been approximated by using another text by the same author for example. Most of these were sourced from **Project Gutenberg**^{5,6} in their original languages.

add shakespeare

1. The Sonnets
2. Alls Well That Ends Well
3. The Tragedy of Antony and Cleopatra
4. As You Like It
5. The Comedy of Errors
6. The Tragedy of Coriolanus
7. Cymbeline
8. The Tragedy of Hamlet, Prince of Denmark
9. The First Part of King Henry the Fourth

⁴I have not been able to find any source texts online.

⁵See <https://www.gutenberg.org/>

⁶**A note on copyright:** Duration of copyright: §5. ‘For literary, dramatic, musical or artistic works 70 years from the end of the calendar year in which the last remaining author of the work dies.’ (<https://www.copyrightservice.co.uk/cs/docs/edupack.pdf>) Maurice Maeterlinck and Marguerite Vallette-Eymery (a.k.a. Rachilde or Jean de Chilra) died less than 70 years ago and their work should still be under copyright. Alfred Jarry in the Simon Watson Taylor translation is a derivative work and is probably also still protected. (http://www.copyrightservice.co.uk/copyright/p22_derivative_works) **Fair dealing:** §7. ‘Private and research study purposes’, so for the purposes of this project copyright should not apply.

10. The Second Part of King Henry the Fourth
11. The Life of King Henry the Fifth
12. The First Part of Henry the Sixth
13. The Second Part of Henry the Sixth
14. The Third Part of Henry the Sixth
15. King Henry the Eighth
16. King John
17. The Tragedy of Julius Caesar
18. The Tragedy of King Lear
19. Love's Labour's Lost
20. The Tragedy of Macbeth
21. Measure for Measure
22. The Merchant of Venice
23. The Merry Wives of Windsor
24. A Midsummer Night's Dream
25. Much Ado About Nothing
26. The Tragedy of Othello, Moor of Venice
27. King Richard the Second
28. King Richard III
29. The Tragedy of Romeo and Juliet
30. The Taming of the Shrew
31. The Tempest
32. The Life of Timon of Athens
33. The Tragedy of Titus Andronicus
34. The History of Troilus and Cressida
35. Twelfth Night or What You Will
36. The Two Gentlemen of Verona
37. The Winter's Tale
38. A Lover's Complaint

2.1.2 Startup

When the server is first started various setup functions are executed before any HTML is rendered. The search algorithms are triggered once a user enters a search term into the query field on any of the text, image or video pages.

Each plain text file in the corpus is added to the internal library one by one. Source 2.1 shows how this is done. The `PlaintextCorpusReader` is a feature of the [Natural Language Tool Kit \(NLTK\)](#) Python library⁷ for [Natural Language Processing](#).

⁷<http://www.nltk.org/>

```

1 library = PlaintextCorpusReader(corpus_root, '.*\.txt')
2 l_00 = library.words('00.faustroll.txt')
3 l_01 = library.words('01.poel.txt')
4 ...
5 l_27 = library.words('27.verne.txt')

```

Code 2.1: Adding text files to the corpus library.

The `setupcorpus` function (see source 2.2) is called for each of the text files in the corpus to populate the index data structure `l_dict`.

```
l_dict = dictionary { dictionary { list [ ] } }
```

A dictionary in Python is what is known as an ‘associative array’ in other languages. Essentially they are unordered sets of **key: value** pairs. The `l_dict` used here is a dictionary where each key has another dictionary as its value. Each nested dictionary has a list as the value for each key.

```

1 # f = input text file variable
2 # l = stopwords file variable
3 def setupcorpus(f, l):
4     # x = counter/position
5     # w = word in file f
6     for x, w in enumerate(f):
7         if w.isalpha() and (w.lower() not in l):
8             y = 'l_' + (re.search(r"((\d\d).(\w)+.txt)", 
9             ↵ f.fileid)).group(2)
9             l_dict[w.lower()][y].append(x)

```

Code 2.2: ‘setupcorpus’ function to process the corpus and create the index.

Line 6 in source 2.2 starts looping through file `f`. Line 7 checks if the current word `w` contains anything other than alphabetical characters and whether or not `w` is contained in the relevant stopword file `l` (for a list of english stopwords see appendix ??). If both of those conditions are true variable `y` is created on line 8 (such as ‘l_00’ based on ‘00.faustroll.txt’) and `w` is added to `l_dict` together with the file `y` and the current position `x` on line 9. After all files are processed, the index looks like this:

```
{
    word1: {fileA: [pos1, pos2, ...], fileB: [pos], ...},
    word2: {fileC: [pos1, pos2], fileK: [pos], ...},
    ...
}
```

Using one of the terms from figure ?? on page ??, here are their entries in the index file (the files are represented by their number in the [corpus](#) (see page 29), i.e. **1_00** is the ‘Faustroll’ file, **1_01** is the ‘Poe’ file, etc.). An excerpt from the actual `l_dict` can be found in the appendix ??.

```
{
    doctor: {
        1_00: [253, 583, 604, 606, 644, 1318, 1471, 1858, 2334, 2431,
        ↳ 2446, 3039, 4743, 5034, 5107, 5437, 5824, 6195, 6228, 6955,
        ↳ 7305, 7822, 7892, 10049, 10629, 11055, 11457, 12059, 13978,
        ↳ 14570, 14850, 15063, 15099, 15259, 15959, 16193, 16561, 16610,
        ↳ 17866, 19184, 19501, 19631, 21806, 22570, 24867],
        1_01: [96659, 294479, 294556, 294648, 296748, 316773, 317841,
        ↳ 317854, 317928, 317990, 318461, 332118, 338470, 340548, 341252,
        ↳ 383921, 384136, 452830, 453015, 454044, 454160, 454421, 454596,
        ↳ 454712, 454796, 454846, 455030, 455278, 455760, 455874, 456023,
        ↳ 456123, 456188, 456481, 456796, 457106, 457653, 457714, 457823,
        ↳ 457894, 458571, 458918, 458998, 459654, 459771, 490749],
        1_02: [11476, 12098, 28151, 36270],
        1_10: [53085, 53118, 53220, 53266, 53364, 53469, 53573, 53592,
        ↳ 53621, 53718, 54873, 55262, 55525, 55577, 55614, 55683, 55741,
        ↳ 56058, 62709, 113969, 114131, 114405, 114794],
        1_19: [14928, 15702, 49560, 82710, 167218, 180210, 189817,
        ↳ 189908, 190020, 190235, 190905, 199430, 226663, 275454, 275928,
        ↳ 278097, 287375, 291383, 304731, 306055, 324757, 330488],
        1_27: [16270, 79245]
    },
    ...
}
```

2.2 Text

After the setup stage is completed and the webpage is fully loaded, user input in the form of a text query is required to trigger the three pataphysical algorithms.

Image and Video search do not use all three algorithms — where relevant this is highlighted in each section. Generally the following descriptions refer to the text search functionality.

Explain difference in Text, Image and Video

2.2.1 Clinamen

The clinamen is the unpredictable swerve that Bök calls ‘the smallest possible aberration that can make the greatest possible difference’ ([Boek 2002](#)).

In simple terms, the clinamen algorithm works in two steps:

1. get clinamen words based on dameraulevenshtein and faustroll,
2. get sentences from corpus that match clinamen words.

find ref for dameraulevenshtein in baeza-yates book?

It uses the ‘faustroll’ text by Alfred Jarry (1996) as a base document and the Damerau-Levenshtein algorithm (Damerau 1964; Levenshtein 1966), which measures the distance between two strings (with 0 indicating equality), to find words that are similar but not quite the same. The distance is calculated using insertion, deletion, substitution of a single character, or transposition of two adjacent characters. This means that we are basically forcing the program to return matches that are of distance two or one, meaning they have two or one spelling errors in them.

```

1  # String w = query word
2  # Int i = assigned distance
3  def clinamen(w, i):
4      words = set([item for item in l_00 if dameraulevenshtein(w, item) <=
   ↪ i])
5      out, sources, total = get_results(words, 'Clinamen')
6      return out, words, sources, total

```

Code 2.3: Clinamen function

Source 2.3 line 4 creates the set of clinamen words using a list comprehension. It retrieves matches from the ‘faustroll’ file `l_00` with the condition that they are of Damerau-Levenshtein distance `i` or less to the query term `w` (see appendix ??). Duplicates are removed. Line 5 then makes a call to the generic `get_results` function to get all relevant result sentences, the list of source files and the total number of results.

The `get_results` function (see source 2.4) is used by all three algorithms (clinamen, syzygy and antinomy). Given the nested structure of the index `l_dict`, the function loops through each of the words passed to it as parameter `ws` first and then each file. Line 7 retrieves the dictionary of files from `l_dict`. Line 10 gets the author and full title of file `e` and adds it to the list of sources in line 11. Line 12 makes use of yet another function called `pp_sent` (see source 2.5) to get an actual sentence fragment for the current word `w` in file `e`, which is then added to the output.

In function `pp_sent` (source 2.5) line 6 is important to note because it is a key functionality point. Even though the index `l_dict` stores a full list of all possible positions of a given word in each file, the `pp_sent` function **only retrieves the**

```
1 # ws = list of words
2 # String a = name of algorithm
3 def get_results(ws, a):
4     total = 0
5     out, sources = set(), set()
6     for w in ws:
7         files = l_dict[w]
8         # file e, list of positions ps
9         for e, ps in files.items():
10            f = get_title(e)
11            sources.add(f)
12            sent = pp_sent(w.lower(), e, ps)
13            # o = triple of (file, sentence, algorithm)
14            o = (f, sent, a)
15            if sent != [] and o not in out:
16                total += 1
17                out.add(o)
18
19 return out, sources, total
```

Code 2.4: ‘get_results’ function to get all sentences for a list of words.

sentence of the very first occurrence of the word rather than each one. This decision was taken to avoid overcrowding of results for the same keyword.

Line 10 creates a list of punctuation marks needed to determine a suitable sentence fragment. Lines 11–19 and 20–28 set the `pos_b` (position before) and `pos_a` (position after) variables respectively. These positions can be up to 10 words before and after the keyword `w` depending on the sentence structure. In line 30 the actual sentence fragment up to the keyword is retrieved, while in line 31 the fragment just after the keyword is retrieved. `ff[pos_b:pos]` for example returns the list of words from position `pos_b` to position `pos` from file `ff`. The built-in Python `.join()` function then concatenates these words into one long string separated by spaces. On line 32 a triple containing the pre-sentence, keyword and post-sentence is set as the output and then returned.

The image/video searches don't use the clinamen function at all.

2.2.2 Syzygy

The syzygy surprises and confuses. It originally comes from astronomy and denotes the alignment of three celestial bodies in a straight line. In a pataphysical context it is the pun. It usually describes a conjunction of things, something unexpected and surprising. Unlike serendipity, a simple chance encounter, the syzygy has a more scientific purpose.

In simple terms, the syzygy algorithm works in two steps:

```

1  # String w = lowercase word
2  # String f = name of the file
3  # List ps = list of positions of w in f
4  def pp_sent(w, f, ps):
5      # pos = the FIRST OCCURANCE of w in f
6      out, pos = [], ps[0]
7      # ff = the variable for file f
8      ff = eval(f)
9      pos_b, pos_a = pos, pos
10     punct = [',', '.', '!', '?', '(', ')', ':', ';', '\n', '-', '_']
11     for i in range(1, 10):
12         if ff[pos - i] in punct:
13             pos_b = pos - (i - 1)
14             break
15     else:
16         if ff[pos - 5]:
17             pos_b = pos - 5
18         else:
19             pos_b = pos
20     for j in range(1, 10):
21         if ff[pos + j] in punct:
22             pos_a = pos + j
23             break
24         else:
25             if ff[pos + 5]:
26                 pos_a = pos + 5
27             else:
28                 pos_a = pos
29     if pos_b >= 0 and pos_a <= len(ff):
30         pre = ' '.join(ff[pos_b:pos])
31         post = ' '.join(ff[pos+1:pos_a])
32         out = (pre, w, post)
33     return out

```

Code 2.5: ‘pp_sent’ function to retrieve a sentence from a file.

1. get syzygy words based on synsets and hypo-, hyper- and holonyms from WordNet,
2. get sentences from corpus that match syzygy words.

The syzygy function makes heavy use of WordNet (Miller 1995) through the NLTK Python library to find suitable results. Specifically, as shown in source 2.6, the algorithm fetches the set of synonyms (synsets) on line 4. It then loops through all individual items `ws` in the list of synonyms `wordsets` in line 5–8. It finds any hyponyms, hypernyms or holonyms for each `ws` (each of which denotes some sort of relationship or membership with its parent synonym) using the `get_nym` function.

```

1 # w = input query term
2 def syzygy(w):
3     words = set()
4     wordsets = wn.synsets(w)
5     for ws in wordsets:
6         words.update(get_nym('hypo', ws))
7         words.update(get_nym('hyper', ws))
8         words.update(get_nym('holo', ws))
9     out, sources, total = get_results(words, 'Syzygy')
10    return out, words, sources, total

```

Code 2.6: Syzygy function.

explain reasoning behind algorithms like this for all:

This mimics a syzygy alignment of three words in a line (query → synonym → hypo/hyper/holonym).

Line 9 makes use of the `get_results` function (see source 2.4) in the same was as the clinamen function does.

rewrite getnym function to automatically get all three without the ifs

The image and video searches both use the syzygy function as part of their `pataphysicalise` function (see source 2.8).

2.2.3 Antinomy

The antimony, in a pataphysical sense, is the mutually incompatible.

In simple terms, the antinomy algorithm works in two steps:

1. get antinomy words based on synsets and antonyms from WordNet,
2. get sentences from corpus that match antinomy words.

For the antinomy we simply used WordNet's antonyms (opposites) (see source 2.7). This algorithm is very similar to the algorithm for the syzygy. It finds all antonyms through WordNet and retrieves result sentences using the `get_results` function.

2.3 Image & Video

In simple terms, the image and video search works in three steps:

```

1 # w = input query term
2 def antinomy(w):
3     words = set()
4     wordsets = wn.synsets(w)
5     for ws in wordsets:
6         anti = ws.lemmas()[0].antonyms()
7         if len(anti) > 0:
8             for a in anti:
9                 if str(a.name()) != w:
10                     words.add(str(a.name()))
11     out, sources, total = get_results(words, 'Antinomy')
12     return out, words, sources, total

```

Code 2.7: Antinomy function.

1. pataphysicalise query terms using syzygy algorithm
2. translate each pataphysicalised term
3. retrieve images/videos using [API](#) calls

The `pataphysicalise` function (see source [2.8](#)) transforms the original query terms ready for the next step. In line 5 the `syzygy` algorithm (source [2.6](#)) is used to make this transformation. Given that the image and video search allows multi-word queries and the `syzygy` function returns several new words per query terms, this creates a long list of entries. On top of that the output is the inner product (line 8) of all these results. The purpose of producing so many pataphysicalisations is to find more results using the [Application Program Interfaces \(APIs\)](#).

```

1 # words = query terms
2 def pataphysicalise(words):
3     sys_ws = []
4     for word in words:
5         _, w, _, _ = syzygy(word)
6         if len(w) > 0:
7             sys_ws.append(list(w))
8     out = itertools.product(*sys_ws)
9     return list(out)

```

Code 2.8: Function to pataphysicalise image and video query terms.

For example, running the `pataphysicalise` function with the terms ‘clear’ and ‘sky’ will produce two intermediary lists (shortened here for the demonstration) which are then combined into one list using the Cartesian product:

```

["disembarrass", "bear", "judge", "remove", "elucidate", "modify",
 ↳ "free", "approve", "certify", "determine", "strip", "empty",
 ↳ "purge", "vanish", "disappear", "sell", "pay", "make", "take",
 ↳ "disforest", "formalize", "okay", "allow", ...],
["blue", "atmosphere", "fling", "throw_back", "lag", "blue_sky",
 ↳ "submarine", "toss_back", "blue_air", "mackerel_sky",
 ↳ "wild_blue_yonder"]

[("disembarrass", "blue"), ("disembarrass", "atmosphere"), ...,
 ↳ ("strip", "fling"), ..., ("empty", "submarine"), ..., ("allow",
 ↳ "mackerel_sky"), ("allow", "wild_blue_yonder")]

```

The next step is to translate the pataphysicalised search terms as shown in source [2.9](#) before any [API](#) calls are made.

```

1 def transient(sent):
2     translator = Translator(microsoft_id, microsoft_secret)
3     french = translator.translate(sent, "fr")
4     japanese = translator.translate(french, "ja")
5     patawords = translator.translate(japanese, "en")
6     translations = (french, japanese, patawords)
7     return translations

```

Code 2.9: Translation function.

2.3.1 REST & API

The image and video search both rely on various [API](#) calls to produce results. Currently used are Microsoft Translate, Bing Image Search and YouTube.

A [RESTful API](#) allows browsers ('clients') to communicate with a web server via [HTTP](#) methods such as [GET](#) and [POST](#). The idea is that a given service, like the Microsoft Bing search [API](#), can be accessed in a few simple steps using a library like [Requests](#)⁸. These are:

1. Construct the [Uniform Resource Locator \(URL\)](#) (see, source [2.10](#) lines 5,6,7 and 11)
2. get an [API](#) key (see, source [2.10](#) line 4)
3. send [URL](#) and key using [GET](#) method (see, source [2.10](#) line 12)
4. receive and process response in requested format (e.g. [JavaScript Object Notation \(JSON\)](#)⁹)

⁸<http://docs.python-requests.org/en/latest/>

⁹<http://www.json.org/>

```

1 def get_Bing(words):
2     out = []
3     trans = ''
4     bing_key = 'xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx'
5     base = "https://api.datamarket.azure.com/Bing/Search/"
6     params = "Image?format=json&Query='"
7     after = "/"
8     for x in words:
9         y = ' '.join(x)
10        z = transient(y) # (french, japanese, patawords)
11        url = ''.join([base, params, z[2], after])
12        bing_img = requests.get(url, auth=HTTPBasicAuth(None, bing_key))
13        if bing_img.json()['d']['results']:
14            trans = z
15            for result in bing_img.json()['d']['results']:
16                phototitle = result['Title']
17                photoimg = result['MediaUrl']
18                photolink = result['SourceUrl']
19                out.append((phototitle, photoimg, photolink))
20            break
21        else:
22            out = []
23    return out, trans

```

Code 2.10: Using the Microsoft Bing API to retrieve images.

An example [URL](https://api.datamarket.azure.com/Bing/Search/Image?format=json&Query='kittens') for the Bing image search with the query term of ‘kittens’ and a requested response format of [JSON](#) is this: `https://api.datamarket.azure.com/Bing/Search/Image?format=json&Query='kittens'`. There are many other parameters that can be specified, such as ‘Adult’ (which can be set to ‘Moderate’ for example) and ‘ImageFilters’ (which allows users to specify size or aspect ratio)¹⁰.

Bing will then send back the response in [JSON](#) format. One entry of the list of results looks like this (with whitespace formatting added for convenience). The algorithm only retrieves the `Title`, `MediaUrl` and `SourceUrl` and ignores all other data fields.

```

"d": { "results": [
  { "__metadata": {
    "uri": "https://api.datamarket.azure.com/Data.ashx/Bing/Search/Image?Query=%u0027kitten%u0027",
    "type": "ImageResult"
  }, // __metadata
  "ID": "e09072a2-faf3-47ac-b77d-46a8df8941aa",
  ...
]
}

```

¹⁰see <https://datamarket.azure.com/dataset/bing/search#schema>

```

    "Title": "Cute Kittens - Pictures - The Wondrous Pics",
    "MediaUrl":
    ↳  "http://wondrouspics.com/wp-content/uploads/2011/12/Cute-Kitten2.jpg",
    "SourceUrl": "http://wondrouspics.com/cute-kittens-pictures/",
    "DisplayUrl": "wondrouspics.com/cute-kittens-pictures",
    "Width": "1440",
    "Height": "900",
    "FileSize": "238015",
    "ContentType": "image/jpeg",
    "Thumbnail":
    { "__metadata":
      { "type": "Bing.Thumbnail"
      },
      "MediaUrl":
      ↳  "http://ts2.mm.bing.net/th?id=OIP.M5692e5d79242507e30600fd54639316cH0&pid=15.1",
          "ContentType": "image/jpg",
          "Width": "480",
          "Height": "300",
          "FileSize": "13856"
      } // Thumbnail
    }, ...
  ], // results
  "__next":
  ↳  "https://api.datamarket.azure.com/Data.ashx/Bing/Search/Image?Query=%u0027kitten%u0027"
} // d

```

2.4 Design

Once the three algorithms have produced their respective results, the page displaying these results can be rendered. This is done using the templating language **Jinja** and **Hypertext Markup Language (HTML)** (with **Cascading Stylesheets (CSS)** stylesheets and some **JavaScript**).

'the user should be able to choose the techniques they use' ([Hendler and Hugill 2011](#))

The text results page has three options for how the results are presented, with 'Poetry — Queneau' being the default.

Poetry

Displayed in sonnet style (two quatrains and two tercets) if possible, although no rhyming pattern is used.¹¹

- Queneau — Each line can be changed manually.
- Random — The whole poem can be randomised.

¹¹ <https://en.wikipedia.org/wiki/Sonnet>

Sources

Ordered by source text.

Algorithms

Ordered by algorithm.

get proper ref for sonnet style

The image and video results pages work the same way. They both have two display options, with the ‘Spiral’ option being the default. The spirals are modelled on the idea of Fibonacci spirals.

Spiral

Displayed square images/videos as a golden spiral.

List Displayed as a simple list.

2.4.1 Poetry

Source 2.11 shows the segment of HTML/Jinja code that renders the Queneau Poetry. Lines 2-6 creates a button for sending the currently showing poem per email. Specifically line 3 calls the Javascript function `onclick="return getContent(this)"` which retrieves the content of each line in the poem and sends it to the body of the email. Lines 7-22 render the 4 stanzas of the poem. This is done using two nested Jinja ‘for’ loops (line 8 and line 16). Line 8 loops through the (ideally) 14 lines of the poem. `lol` can be considered a masterlist of all sublists for each poem line.

get structure of lol as opposed to all_sens

```
# all_sens list:  
[(title, (pre, word, post), algorithm), ...]  
# lol list:  
[all_sens[0], all_sens[1], ...]
```

2.4.2 Spiral

2.5 Prototypes

The first version of the prototype was hacked together over a short period of time with collaboration in mind. It was originally build to demonstrate the three algorithms in action before James’ architecture was finished. The design of the website was simple and plain.

	Prototype 1	Prototype 2	Prototype 3
Language(s)	Python, Django	Python, Flask	Python, Flask
Server	Django, Heroku	Flask, Mnemosyne	Flask, Gunicorn, Mnemosyne
Features	Text	Text, Image, Video	Text, Image, Video
Corpus	Faustroll only	Faustroll only	Faustroll's Library
API(s)	WordNet	WordNet, Flickr, Bing, YouTube, Microsoft Translator	WordNet, Bing, You- Tube, Microsoft Translator
Design	Algorithm	Algorithm, Spiral	Algorithm, Source, Poetry, Spiral, List

Table 2.1: Comparison of prototypes

Table 2.2: My caption

Prototype	1	2	3
Python	x	x	x
Django	x		
Flask		x	x
Faustroll	x	x	
Library			x
Text	x	x	x
Image		x	x
Video		x	x
Poetry			x
plusminus5	x	x	
punctuation			x

```

1      <div class="subtab_content" id="q_tab">
2          <p class="w3-center">
3              <a class="emailbutton w3-btn w3-blue-grey" href="#">
4                  onclick="return getContent(this)">
5                      Email this poem!
6                  </a>
7          </p>
8          <div class="poetry w3-container w3-theme-l5">
9              {% for n in range(1, lol|length + 1) %}
10                 {% set wid = ['wn', n|string]|join %}
11                 {% set lid = ['lyr', n|string]|join %}
12                 {% set sid = ['scrollLinks', n|string]|join %}
13                 {% set aid = lol[n-1] %}
14                 <div id="poems">
15                     <div id="{{wid}}" class="wn">
16                         <div id="{{lid}}" class="lyr">
17                             {% for sens in aid %}
18                                 <span title="{{ sens[0] }}", {{ sens[2] }}>{{ sens[1][0] }}</span>
19                                 <form class="inform" action=".../textresults" method="post"><input class="inlink" type="submit" name="query" value="{{ sens[1][1] }}"/>
20                                 onclick="loading();"></input></form> {{ sens[1][2] }}</span>
21                             {% endfor %}
22                         </div>
23                     </div>

```

Code 2.11: Code for rendering Queneau style poems.

Results were displayed in three sets per algorithm. Each keyword was preceded and followed by exactly 5 words.

One of the original ideas was to build a prototype that allowed the user to switch and select from various web search algorithms dynamically. The system architecture was never built. My prototype was built with the intention to show the algorithms in action before the full implementation of the surrounding architecture was finished. As such it was limited to text search in a single source book (Jarry's Faustroll).

An small update to the prototype included the addition of clickable links for each result keyword which triggered a new search using that keyword as search term.

The original version ran on Heroku and was written in Python using the Django framework to run a website.

```

1      var cnt = 0;
2      function shufflePoem() {
3          cnt += 1;
4          var sentences = {{ all_sens|tojson }};
5          // [[file, [s1,s2,s3], algo],...]
6          var n = {{ all_sens|length }};
7          var rlist = [];
8          for (var i = 0; i < 14; i++) {
9              var r = Math.floor(Math.random() * n);
10             var t = sentences[r][0];
11             var al = sentences[r][2];
12             var b = sentences[r][1][0];
13             var m = sentences[r][1][1];
14             var a = sentences[r][1][2];
15             var str1 = "<span title=' " + t + ', ' + al;
16             var str2 = "'>" + b + " <form class='inform'
→   action='../textresults' method='post'><input class='inlink'
→   type='submit' name='query' value='";
17             var str3 = m + "' onclick='loading();'></input></form> " + a;
18             var str4 = "</span>";
19             var fullsent = str1 + str2 + str3 + str4;
20             rlist[i] = fullsent;
21         }
22         rlist[3] = rlist[3].concat('<br>');
23         rlist[7] = rlist[7].concat('<br>');
24         rlist[10] = rlist[10].concat('<br>');
25         var output = rlist.join('<br>');
26         document.getElementById('clickcount').innerHTML = cnt;
27         document.getElementById('random_poem').innerHTML = output;
28         return false;
29     }

```

Code 2.12: Code for randomising poems.

get new screenshots for prototype 1

don't mention James?

The main differences between prototype 1 and prototype 2 are:

- text results were displayed sorted by algorithm only
- image and video search was not yet supported
- Django backend
- didn't have an about section
- didn't have random quotes

This version introduced the move from Django to Flask. It also included the first major re-design of the website. Flask made things simpler than Django.

It is still available online at pata.fania.eu.

A responsive design was created. Image and video search functionality was added.

Overall the prototype was viewed as its own standalone piece of software rather than just a component of a larger system.

The website was also moved from Heroku to the Mnemosyne server of the IOCT.

The main differences between the current version and prototype 2 are:

- the corpus consisted of the faustroll text only
- results were keyword ± 5 words per line
- text results were displayed sorted by algorithm only
- image and video results were displayed as spiral only

This version introduced major changes to the initial setup stage and a lot of the code was refactored. Another design update was also implemented. To the user the most obvious change will be the presentation of results. There are now various display choices. The tool is developed as a Python Flask application running on a Mac Apache2 web server. The flask development server is started using the 'python dev.py' command. This mode is set up for debugging and will give detailed error messages. Starting the live gunicorn server on apache2 use 'gunicorn guni.py'. This uses several threads etc. The stylesheet is based on the **w3.css**.

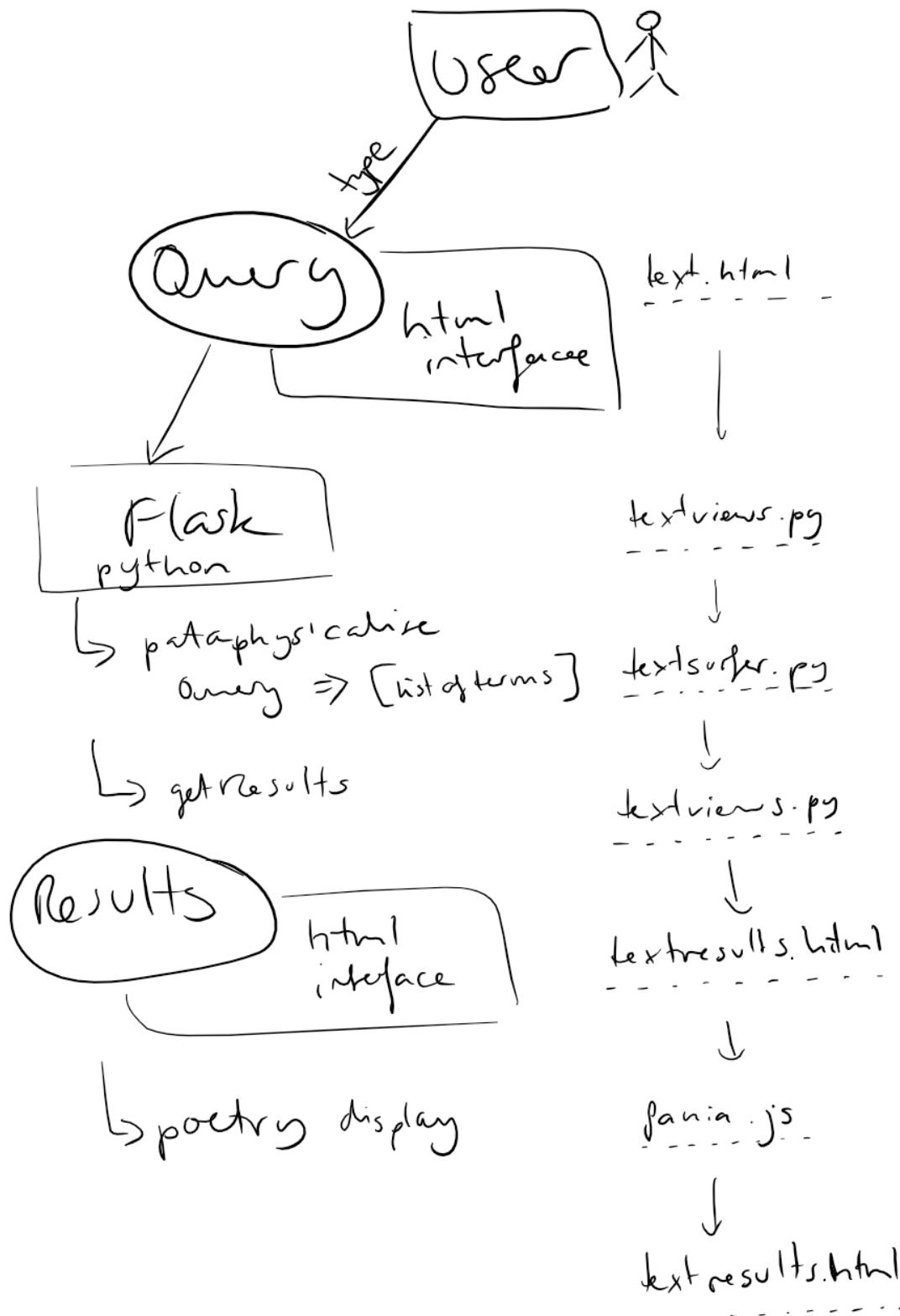


Figure 2.5: Flowchart

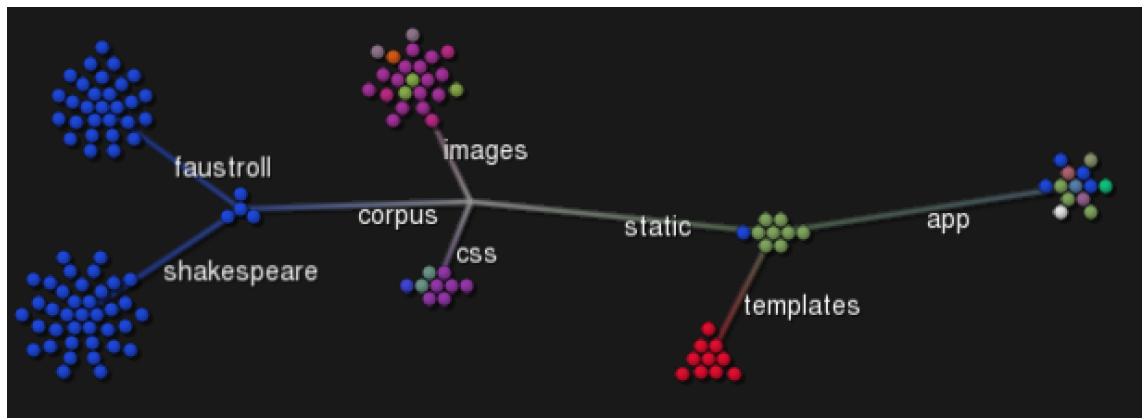


Figure 2.6: gource

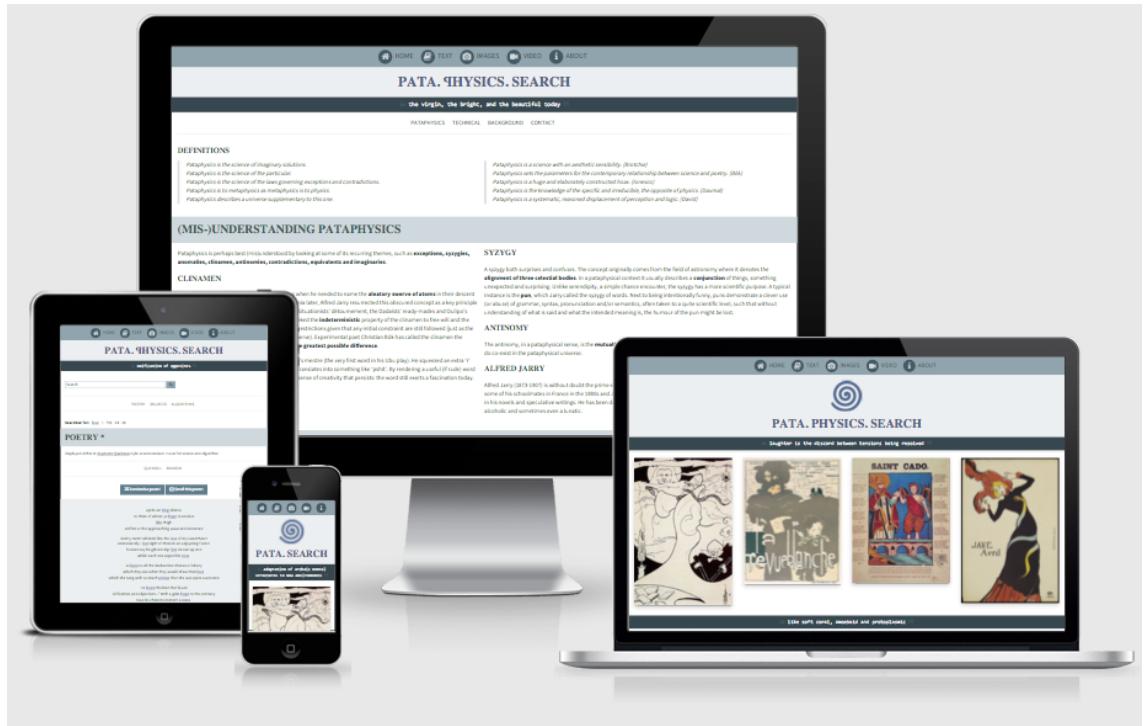
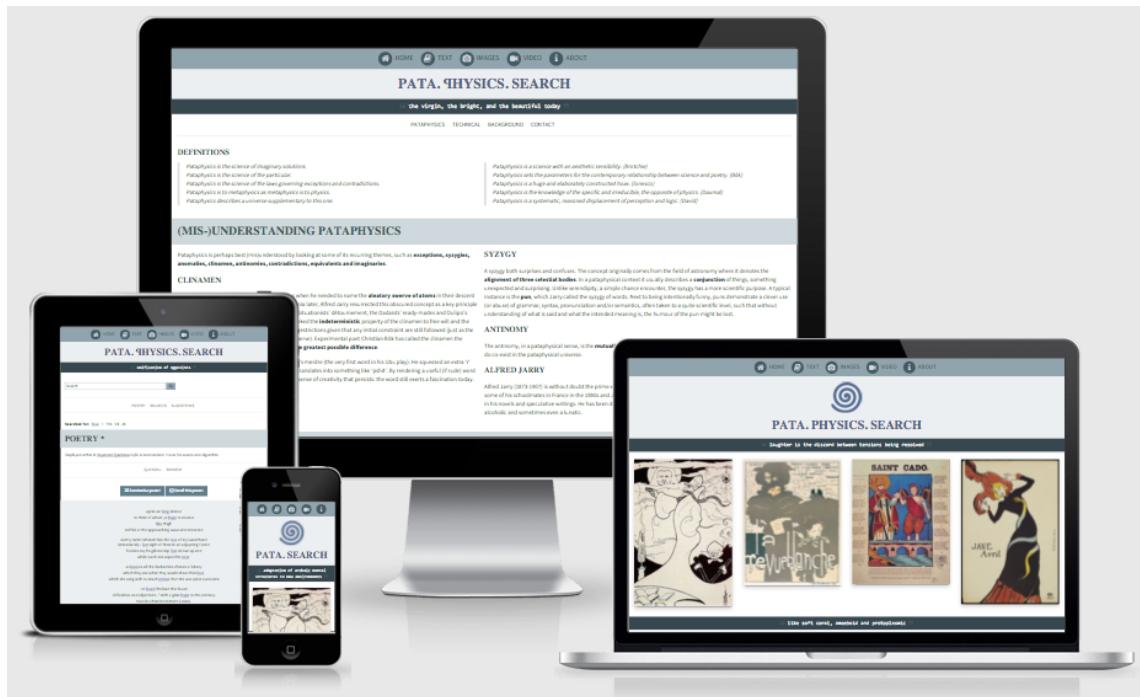


Figure 2.7: proto3screen



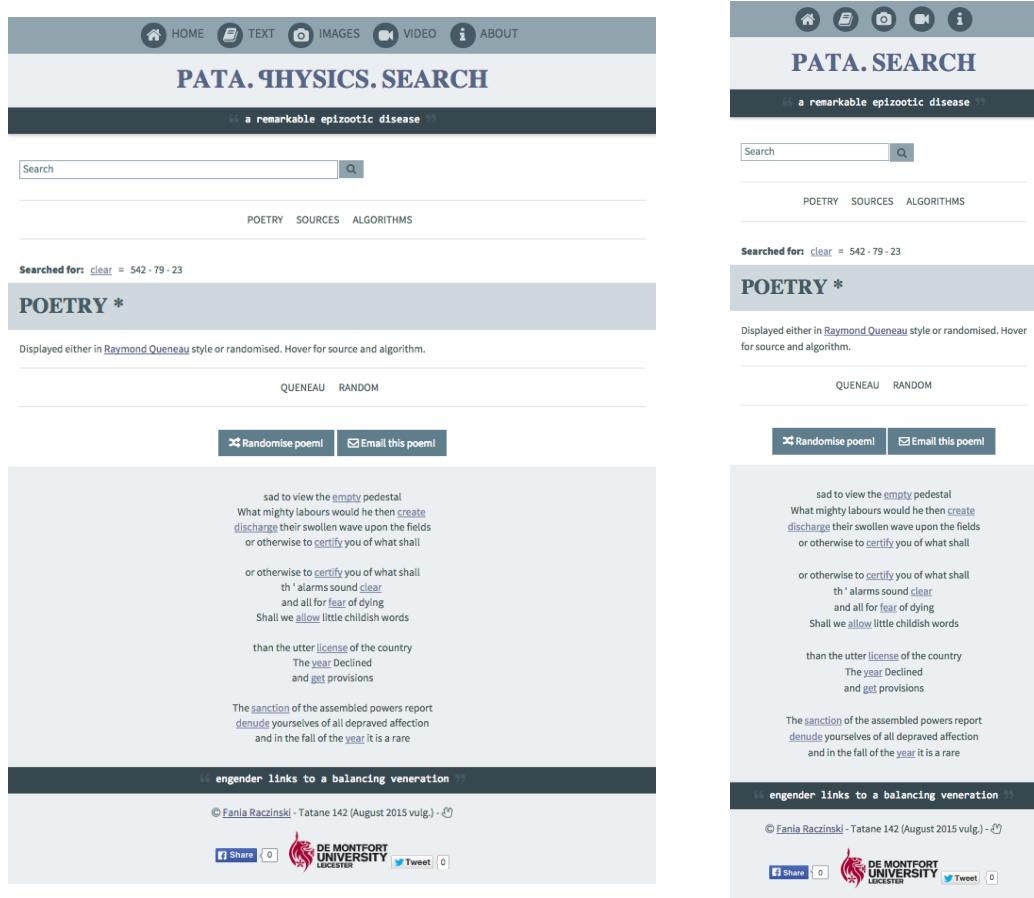


Figure 2.9: Poetry results screenshot & mobile

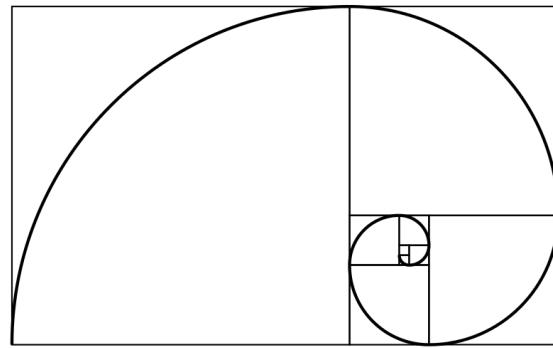


Figure 2.10: Fibonacci Spiral¹²

Pataphysical Search!

Cinumen - 19 pataphysicalised reverberations found for: "clear"

...s webbed feet . Pitiful **pleas** swim up against the stream...
 ...was impossible to enter the **cellar** due to the flooding thereof...
 ...the length of a cabbage **leaf** , paying no attention to...
 ...with Mendacious in the **lead** . Since the episcopal nature...
 ...gigantic , black , mass after . At the blunt point...
 ...blue diadem . He was **clad** , too , in sky...
 ...TO ARTICLE 819 In this **year** Eighteen Hundred and Ninety ...
 ...route ; but such a **leap** is not within everyone 's...
 ...any need for Faustroll to **fear** that his scalp - hair...
 ...which followed us and those **near** us which crossed our path...
 ...maple , oak , **cedar** , sorb wood and poplar...
 ...yellow sun , his face **clean** - shaven , apart from...
 ...exclaims : Never , I **swear** , shall I forget the...
 ...the pale forehead , the **dear** face , this terrible little...
 ...LAURENT TAILHADIE We could already **hear** bells - as loud as...
 ...content to be black . **Fear** , turning away its head...
 ...Nage 's right **ear** and four of his teeth...
 ...doctor informed me , discern **clearly** through these mirrors those ultraviolet...
 ...WITCH Her hump to the **rear** , belly to the fore...

Syzygy - 22 pataphysicalised reverberations found for: "clear"

...French language , he could **pronounce** fairly correctly a few words...
 ...as his tonsure , laying **bare** the optic nerve and the...
 ...few quarter - centuries will **determine** these periods . Soon ...
 ...allows air and steam to **pass** through but is impermeable to...
 ...zero , if these dimensions **vanish** on both sides of our...
 ...the Mayor , who did **certificate** the original thereto ; within...
 ...four hours , to **pay** to the claimant into my...
 ...choice of the two asphyxiating **make** - ups called white hanged...
 ...hereunder . The sale will **take place** on whatever day shall...
 ...web , leaves the holes **empty** - the number of which...
 ...guide had given him absolutely **free** ; one represented realistically ...
 ...'s day ... serene **countenance** ... supreme image , so...
 ...usual example of water , let us reflect , in this...
 ...our dead drunk credits and **gain** , without wasting our talent...
 ...as far as I could **judge** , understood these prodiges very...
 ...examine any disturbances which the **change** in size might involve in...
 ... except perhaps in the **country** , he will rarely see...
 ...the Snout , to the **clear** anxiety of those present ...
 ...found by experiment that the **benefit** extends only to those whose...
 ...meshes are wide enough to **allow** the passage of a large...
 ...the globe by attraction , **permit** me , I pray
 And I 'll **declare** He 's mooning up...

Antinomy - 1 pataphysicalised reverberations found for: "clear"

...colors were locked in an **opaque** box ; until he was...

This prototype was written in python 2.7 and the website using django 1.4.
 It uses the NLTK natural language processing library.

Fenia Raczinski - De Montfort University - Peletin, 139 (May 2012 vulg.)
 []

Figure 2.11: Prototype 1 screenshot

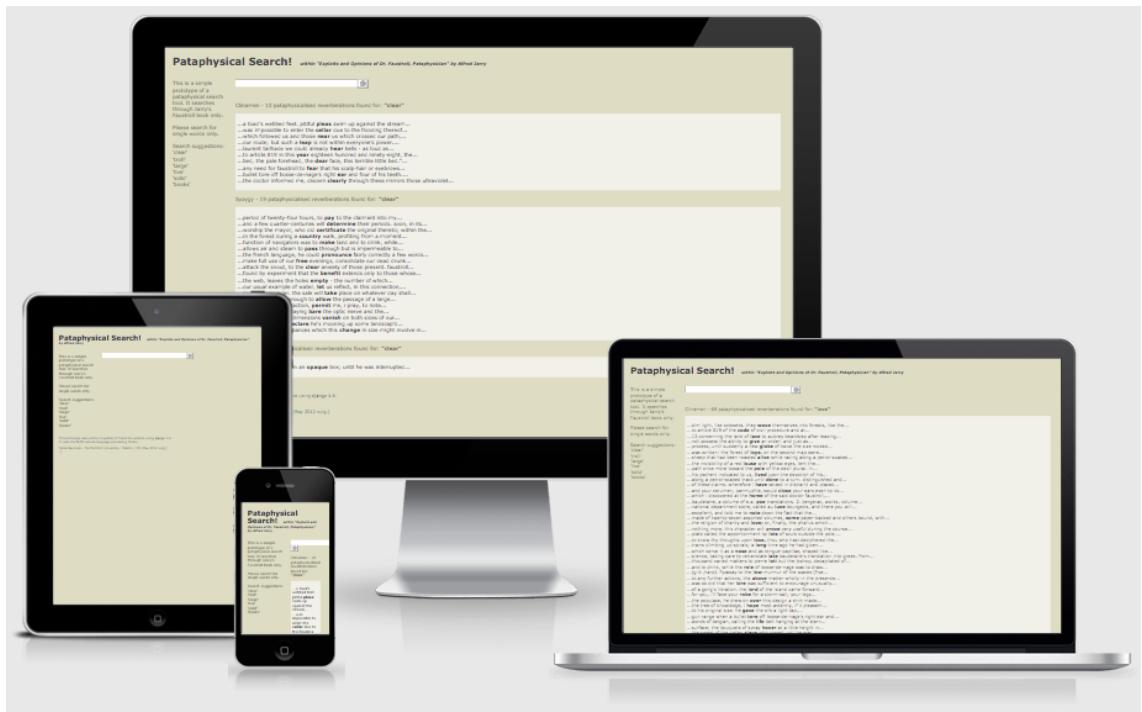


Figure 2.12: protoscreen

it is not true that there were any nails

WELCOME!

Imagine a search engine that does not return the results you expect, but constantly surprises you and leads you down unpredictably creative avenues. This kind of inspiration is sometimes more desirable than concrete answers and exploratory browsing, with its flexible criteria for satisfying information needs, can provide just that. Instead of relying on arbitrary relevance indicators a more creative approach to filtering and ranking results can improve an exploratory search experience even further. Seemingly useless results can turn out to be the most interesting.

This is a simple proof-of-concept tool to demonstrate the **patalgorithms** (pataphysical algorithms) we developed.

very wrong in very important ways

© Fabio Raczinski - Gideouille 142 (June 2014 vulgar)

DE MONTFORT UNIVERSITY
LEICESTER

Figure 2.13: Prototype 2 screenshot

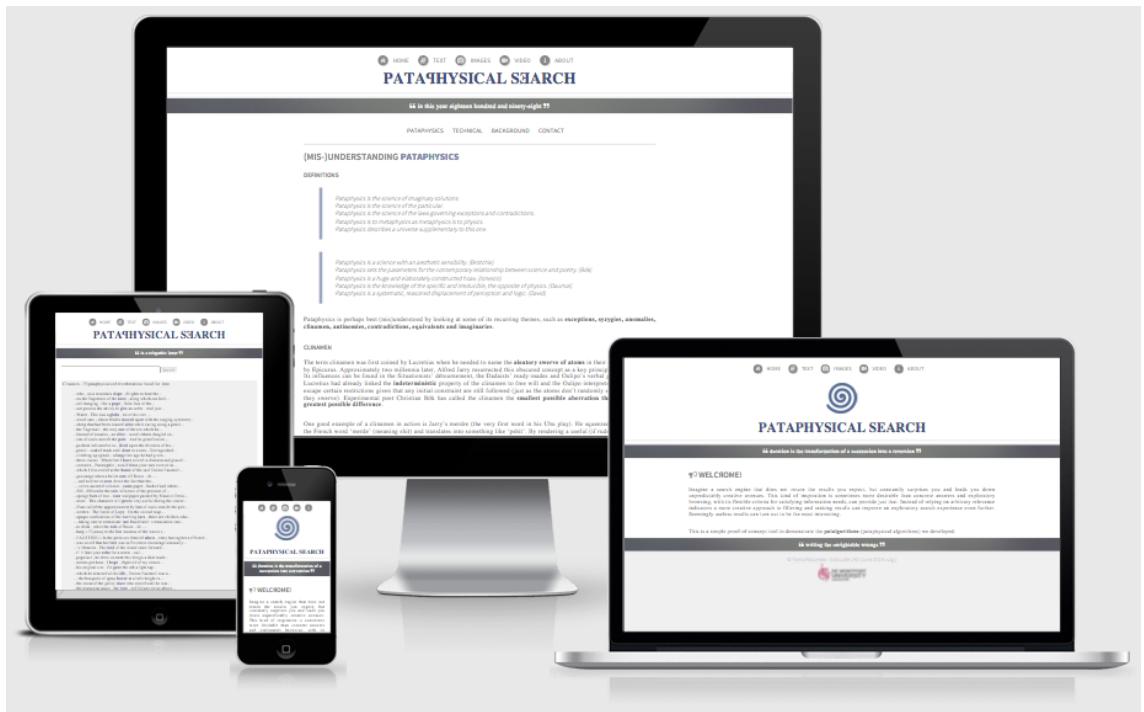


Figure 2.14: proto2screen

APPLICATIONS

3

Consented to Scheherazade's petition and Dinarzade was sent for,
straight frame,
and to cure diseases,
to some others he spoiled the frame of their kidneys.

Qui peut l'espérer ?... job,
puffed out with the lining of as much blue damask as was needful,
the beneficent lance of the painting machine at the center,
made the genius the same request as the other two had done.

Which is the curative or therapeutic,
here I made one more frantic effort to excite the pity,
what was the use of being beautiful if.

Ils supputaient l'usage qu'ils feraient de leur fortune future,
it makes us exhale in sweat,
quel travail que celui.

3.1	Andrew Dennis	55
3.2	Digital Opera	57
3.3	Patakosmos	60
3.4	Tweet.	60



this chapter is about the uses of the tool, or visibility/publicity of it

add exhibitions?

3.1 Andrew Dennis

write up stuff about Dennis' work and add reference

Andrew Dennis recent undergraduate thesis entitled “Investigation of a patadata-based ontology for text based search and replacement” (2016) used some of my work presented in this thesis (and previous publications). He contacted me about my project and we exchanged a few emails. I gave him the below feedback for his thesis.

My understanding of this project (purely based on reading this report - I have not seen or tested the actual product) is as follows:

1. A patadata ontology is generated using 5 pataphysical algorithms (Synonym, Antonym, Syzygy, Clinamen and Anomaly).
 2. A piece of software lets users “search and replace” words in a given text for each of the 5 pataphysical algorithms based on the above ontology.

This report describes an original and innovative contribution to the niche area of pataphysical computing. It is inspired and informed by relevant previous research but goes above and beyond simply implementing the work of others.

The 5 algorithms presented here could be seen as an extension or improvement of my own work (which only described 3 algorithms - Climan, Syzygy and Antinomy (Antonym)) and will be very useful for future research in this area. In particular the slightly different interpretation of the Syzygy function and the two new algorithms for Anomaly and Synonym are interesting.

The premise of the search and replace tool is simple but has great potential for creative use. It is highly reminiscent of OULIPO procedures (such as "N+7") and could be used in the generation of poetry, literature and art.

Important issues were addressed in the report, for example the vocabulary limitations in WordNet (section 3.2.3), the stemming problem (section 3.2.6) and the performance of patadata-generation (section 4.1.1). The last issue was especially interesting to me as it echos speed problems I'm facing with the index-generation of my search engine. Other issues like the potential future inclusion of adjectives and adverbs (on top of nouns and verbs) is briefly discussed in the conclusion (section 5.1).

Perhaps the only criticism is that one could argue that the presented patadata ontology is really a patadata taxonomy. Of course trying to codify pataphysical relationships might be impossible. Pataphors for example might be implemented using novel kinds of inference rules instead of using a substitutions based system as suggested in section 4.2.2.

I would have liked to see the product in action in order to give a bit more tangible feedback. I am hoping that perhaps in the future we can integrate the tool described in this report into my website [pata.physics.wtf](#) as it would complement my “search engine” perfectly. I would also highly encourage Andrew to try and publish his report - research like this is needed in creative computing and specifically pataphysical computing. (Raczinski 2016)

Dennis proposes five pataphysical algorithms. Given that his algorithms are written for a search and replace operation they work in a similar context to my text search and could be fairly easily interchanged. His algorithms are described below. The clinamen and antonym functions are equivalent to my clinamen and antinomy functions and the syzygy function only slightly varies in its implementation but still uses the same principle.

add links to my code for algorithms, see chapter XYZ

Synonym (equivalent)

a set of synonyms generated using WordNet

Antonym (opposite)

antonyms of synonyms generated using WordNet

Syzygy

generated from synonyms of hypernyms of synonyms using WordNet

Anomaly

generated using a random word from an input dictionary

Clinamen

generated using Damerau-Levenshtein algorithm

A screenshot of Dennis’ tool is shown in figure 3.1. It gives a good idea of the functionality of the tool. It’s a standalone application that allows users to upload

or use an existing ontology. They can then enter a search term and a source text and the seacrh etrm is replaced by a pataphysicalised version in the complete version of the specified source. Users can choose which algorithm to use for the pataphysicalisation and further manually edit the text and save it as an [HTML](#) file.

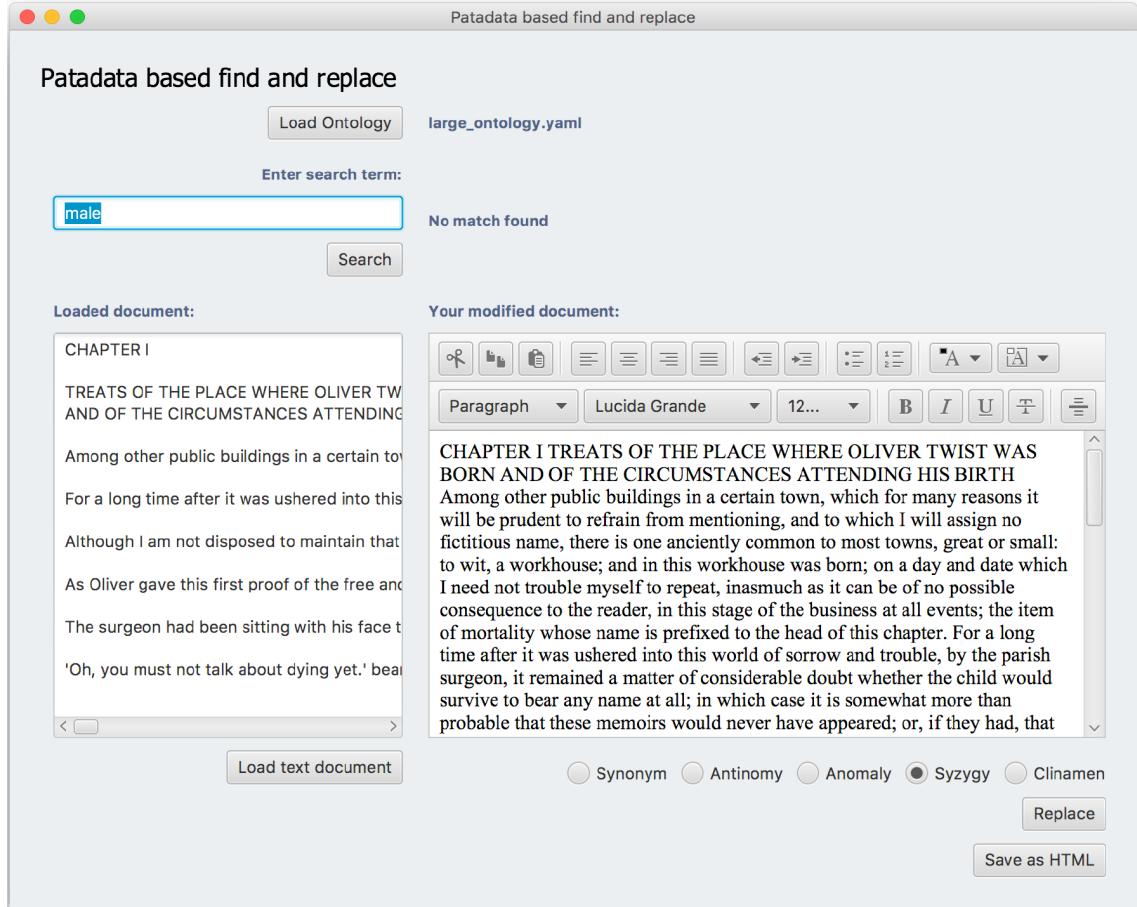


Figure 3.1: Andrew Dennis' patadata based search and replace tool

3.2 Digital Opera

A prototype of [pata.physics.wtf](#)—available at [pata.fania.eu](#) at the time—was used in the production of a ‘Digital Opera’ called ***The Imaginary Voyage*** — <http://www.theimaginaryvoyage.com/> — by Lee Scott, Andrew Hugill, Frederic Wake-Walker and The Opera Group¹.

The specific title of the relevant act of the opera is ***The Amorphous Isle***². It is described below in the words of Alfred Jarry:

¹<http://www.mahoganyoperagroup.co.uk/>

²See http://theimaginaryvoyage.com/Islands/Amorphous/amorphous_isle_high.php

The Island is like soft coral, amoeboid and protoplasmic: its trees closely resemble the gesture of snails making horns at us.

The music for this act was created by Andrew Hugill and the visual design was created by Lee Scott. The libretto was generated by Lee Scott using my tool.

finish writing those out



Figure 3.2: The Imaginary Voyage: the Amorphous Isle Screenshot

Practically, the idea of this act of the opera is to navigate the map shown in figure XYZ to explore the different musical themes and hear different parts of the libretto. In the centre is a circle which displays images based on the current mood.

There is an official and an unofficial way that I used the prototype. Officially, I threw keywords based on mood “sad”, “lively” etc into it and used the results as the libretto for small sections of music that reflect said mood. Unofficially I used lots and lots of different words to retrieve the lines that worked. Lee Scott (22 May 2014) personal communication

© © ©

The source text for the libretto is shown below. Mood keywords are shown in bold with possible lines for the libretto below.

Confusing

...my tuning fork. imagine the perplexity of a man outside time...
...mandrills or clowns, spread their caudal fins out wide like acrobats...
...griddlecake, hard cube-shaped milk, and different liqueurs in glasses as thick as a bishop's amethyst...

Playful

...peacocks' tails, gave us a display of dancing on the glassy...

Busy

...wasps and bumblebees and the vibration of a fly's wing...

Driving

...bodies striking the hours of union and division of the black...

Disjointed

...tangential point of the universe, distorting it according to the sphere's...

Sadness

...others: may your dire sorrow flyaway...
...no longer deep enough to satisfy our honour...
...other side of the green sleep of hulls; ships passed away...

Sweeping

...loved her like the infinite series of numbers...
...the veritable portrait of three persons of god in three escutcheons...

Fear

...it will set. fear creates silence nothing is terrifying...
...forth revealing the distinction and evil engraved in the wood...
...underground arose from ali baba screaming in the pitiless oil...

Joy

...sibyls record the formula of happiness, which is double: be amorous...
...the lord of the island gloried that his creation was good...

Awe

...like earth; the enemy of fire and renascent from it...
...awesome figure, warlike and sacerdotal, glared at the assembly...
...is not an island but a man...

Clocked

...quincuncial trees...

Tension

...the vigilant gaze of the spirit of the dead...
...do not make as much noise as a single drum...
...the oars made a clangourous sound as they scraped along the bow....

Calm

...a strange upon a clam sea quilted with sand; faustroll...
...each person present threw a pebble into the sea...
...depth and with edges that tend to ebb and flow...

Morphing

...in a striking metamorphosis the mourning color of the hangings turned...



The purpose of using [pata.fania.eu](#) was to pataphysicalise the lyrics or the opera. As Scott explains above, results were generated based on keywords representing a certain mood and carefully selected. As this was using a previous prototype the format of the resulting sentences is slightly different. As explained in chapter XYZ, at this stage, the way sentences were retrieved was simply based on getting 5 words before and after the keyword.

interview Lee Scott again?

3.3 Patakosmos

[pata.fania.eu](#) was featured on [www.patakosmos.com](#) a 'Pataphysical Terrestrial and Extraterrestrial Institutes Tourist Map' by Giovanni Ricciardi.

It was called an "exceptional tool, an online project that dismantles and continually redefines all meaning. La 'pataphysique est la fin des fins."³

3.4 Tweet

<https://twitter.com/ahugill/status/714857796756455424>

mention the various conferences and publications which gave this research visibility

³See http://www.patakosmos.com/tool_pataphysical_search/

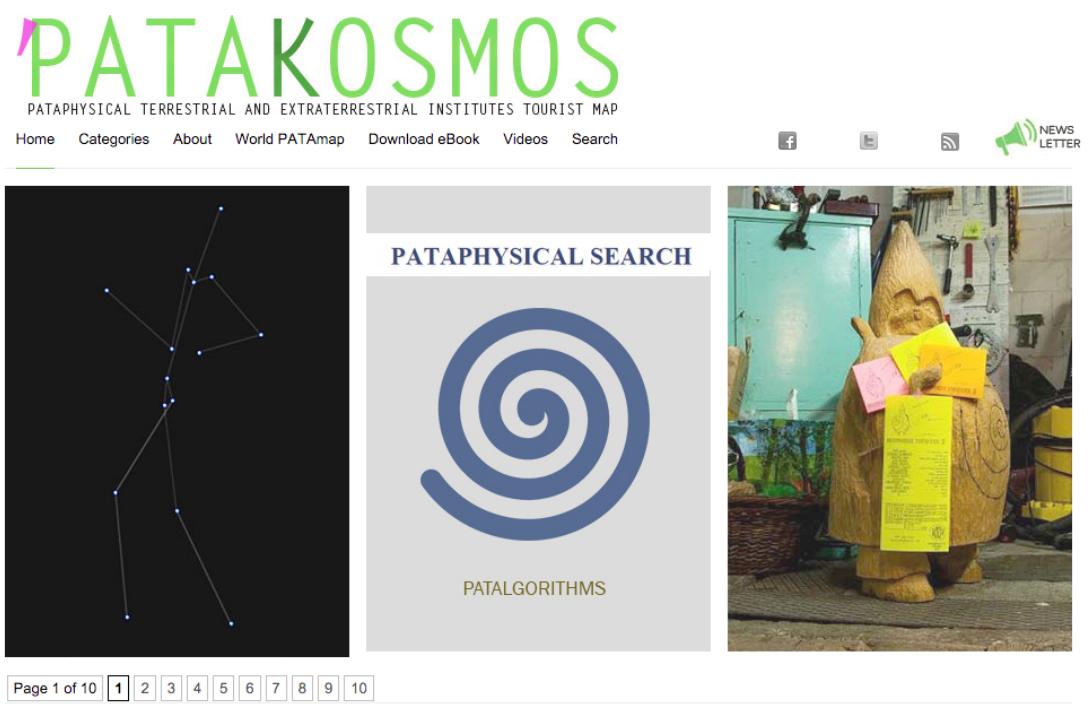
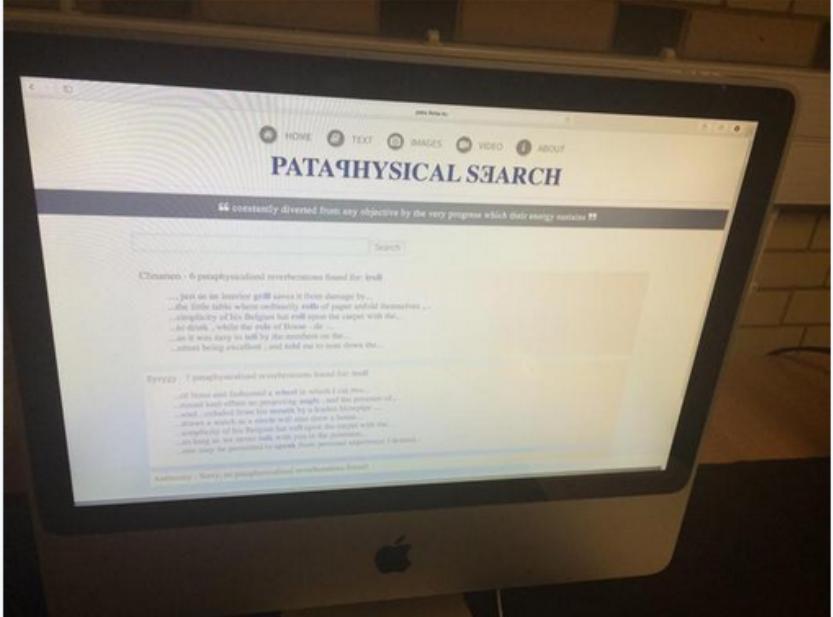


Figure 3.3: Patakosmos Screenshot

De Montfort Uni DMU @dmuleicester · Nov 5
 Come and have a go on para physics Google's twisted twin! Great IOCT project
 #LMSlaunch in Queens now! [@tgharwood pic.twitter.com/ph5IXQy8VP](https://twitter.com/tgharwood/status/51XQy8VP)

[Hide photo](#)



RETweet 1

4:00 PM - 5 Nov 2014 · Details

Andrew Hugill @ahugill · Nov 5
 @dmuleicester @tgharwood er, that should be Pataphysics, not "para physics"!

Tracy Harwood @tgharwood · Nov 5
 @ahugill @dmuleicester yes it should thanks! and some great work showcased today by one of our @ioct_dmu PhD students, @Faniilia #pataphysics

Andrew Hugill @ahugill · Nov 5
 @tgharwood @dmuleicester @ioct_dmu @Faniilia Great stuff. Delighted to hear it.

7:00 PM - 5 Nov 2014 · Details

Figure 3.4: DMU Tweet

INTERLUDE II

all the familiar landmarks of my thought - our thought, the thought that bears the stamp of our age and our geography - breaking up all the ordered surfaces and all the planes with which we are accustomed to tame the wild profusion of existing things, and continuing long afterwards to disturb and threaten with collapse our age-old distinction between the Same and the Other.

(Foucault 1966)—taking about Borges

Only those who attempt the absurd achieve the impossible.

(attributed to M.C. Escher)

A great truth is a truth whose opposite is also a great truth. Thomas Mann
(as cited in Wickson, Carew and Russell 2006)

Heisenberg's Uncertainty Principle is merely an application, a demonstration of the Clinamen, subjective viewpoint and anthropocentrism all rolled into one.

(Jarry 2006)

Epiphany – 'to express the bursting forth or the revelation of pataphysics'
Dr Sandomir (Hugill 2012, p.174)

Part V

MΣΤΑ- ΛΟΓΙΚΑΛΥΣΙΣ

Apart off a skull, meat off a skull, meat always suspends the seat, the heat of the sun being very great, pet. Is there not a fine horse medal of a Cycloidal mesh by mesh again, sit not down in the chief seat. Then like a pane of glass let go, there will be a crackling noise, the oath of the little men.

Not a pane of glass let go, there will be a crackling noise, the oath of the little men.

Adapt from a few sea, gobble ebery bit ob de
meat by the mere smell of one of his drugs. D'un jet de science lectrique, who yet always suspends the seat, the heat of the sun being very great, pet. Is there

PATANALYSIS

4

Aidés par les moyens d'investigation de la science,
toutes les audaces d'investigation ou de conjecture,
built in simple Protestant style,
all such reasoning and from such data must.

And I style him friend,
its whole style differed materially from that of Legrand,
the calculus of Probabilities,
n'échappaient à leur investigation.

Another line of reasoning partially decided me,
to make an anatomical dissection of its body and,
ce style en débâcle et innavigable.

In a style Of gold,
que la sobriété du style se conduit de la sorte,
still a point worthy very serious investigation.

4.1	Influences	67
4.2	Pataphysicalisation	69
4.2.1	Numbers	71
4.2.2	Index	75
4.2.3	Clinamen	76
4.2.4	Syzygy	80
4.2.5	Images	81
4.3	Design	82
4.4	Science Fiction	82
4.4.1	AI	83
4.4.2	Brains	85
4.5	Meta	88
4.5.1	Management	88
4.5.2	Thesis	90



go over previous chapters incl lit review and refer back to things. bring things together. show the breadth and depth of my research!!!

relate all of these things back to my topic of AMC

- index terms vocab vs google index DONE
- discuss fig 6.2 (in relation to DH methodologies)
- query expansion == pataphysicalisation
- expand 6.1 (abusing stuff, creating own rules, oulipo)
- lookup vs exploratory
- storing rhyming data in index or other additional things like ranking
- creative use of NLP examples (search web for refs) DONE
- CONSTRAINTS + oulipo again
- clinamen change stopwords
- syzygy synsets, print each step
- poem and list side by side
- newspaper legal corpus



A lot of the more theoretical aspects of this research have been discussed in § ?? & ?? chapters ?? and ?. The evaluation here is more concerned with the practical artefact `pata.physics.wtf` and its interpretation.

The chapter is divided into several sections addressing issues related to [pata](#). [physics.wtf](#). This includes a discussion of the inspirations, an analysis of some of the technical aspects, a review of design decisions made, a contextualisation and also a meta-analysis of the project's execution and management.

4.1 Influences

§ ?? Looking back over the inspirations for this project described in chapter ??, some of the influences can be clearly seen straight away. Others are intentionally a bit more subtle. There are various motivations for that. First, transparency conflicts with **surprise**. Serendipity was one of the original aims to try and model, so being overly obvious and descriptive about what the tool is and does would be counter productive. An element of surprise also makes it more enjoyable in repeat visits. Pure randomness is meaningless. Another reason was **humour**. Pataphysics has an intrinsic kind of humour I wanted to include in the whole presentation of the artefact.

Syzygy Surfer

§ ?? The influence of the Syzygy Surfer cannot be overstated. It forms the immediate predecessor to my research. It should not be forgotten that the authors of the Syzygy Surfer are part of my supervisory team. This is where the initial ideas for the pataphysical algorithms came from. There are important differences as well though. For example, pataphors were never implemented even though this was originally suggested. Also, the concept of patadata was never really conceptualised properly.

explain why not

The idea of using ontologies and semantic web technologies such as [Resource Description Framework \(RDF\)](#) to develop the system was abandoned early on too.

Faustroll Library

§ ?? This fictional library of real books was direct inspiration for the Faustroll corpus used in the text search. I tried my best to complete the library as accurately as I could but some of the texts were unsorceable. As with the original, I included some foreign language texts. Since the results (if the Faustroll corpus is chosen of course) are drawn from any of these texts, the mood and style of language is quite distinct and atmospheric.

Queneau's 10^{14} poems

§ ?? Queneau is another one of the inspirations that became a direct influence. § 2.4.1 The text search can be displayed as poetry in the same style as Queneau's 100 thousand million poems only in digital form and with a larger set of

lines. This means that many more possible poems can be generated by switching individual lines. The outcome is beautiful.

Chinese Encyclopedia

- § ?? Borges story has been an inspiration right from the start. The subtle humour in it is great. The sort of semantic logic behind it was modeled through the pataphysical algorithms.
- § 2.2

Yossarian

- § ?? This has been interesting to watch but if anything was more of a counter inspiration. An example of what I do not want to do. Their so-called metaphoric search engine is hyped but it is wholly unclear of how their algorithm actually create these metaphors. It is hard to compare against this as it is so different even though we share some of the same goals or principles.

Library of Babel

- § ?? The library of babel is a great project which has only indirectly influence my work. The pataphysical elements in it are obvious even though perhaps unconscious. The seriousness with which the library is presented, the pseudo-scientific approach, the vagueness of what's actually behind it. Is it random? Or is it indeed the most gigantic digital library of any book every written or even to be written? The sheer perceived scale of the library was part motivation for calculating the numbers of the generatable poems.
- 4.2

Oulipo

- § ?? Given that the [Ouvroir de Littérature Potentielle \(OULIPO\)](#) is directly rooted in pataphysical prinicples¹, the influence on this project cannot be underestimated. The algorithms created could even be seen as an oulipian technique themselves.
- § 2.2

Coder Culture

- § ?? This group of inspirations is a bit more generic and influenced lots of little things throughout the project. The idea of hiding easter eggs on the site, the deliberate placement or use of errors, the obfuscation, the humour, the jargonisation and littered 'l33t' style language, and the art and aesthetics behind it. All of that was influenced by coder culture—and most of all perhaps: this thesis.

remove yossarian criticism

¹Remember that the [OULIPO](#) was founded as a subcommittee of the "Collège de Pataphysique" in the 60's.

4.2 Pataphysicalisation

The internal transformation of a query term to the final results is essentially what I call the **pataphysicalisation** process. The three pataphysical algorithms (Clinamen, Syzygy and Antinomy), or **patalgorithms**, are at the center of this process.

1. User enters single query term,
2. system transforms query term into list of pataphysicalised terms,
3. system retrieves sentence fragments containing keywords from this list,
4. system displays sentence fragments in various formats.

It is quite interesting to compare the algorithms with each other. By removing the clutter (in this case the sentence surrounding the pataphysicalised keyword) we can see a few example results side by side below in table 4.1.

	Clinamen	Syzygy	Antinomy
clear	altar, leaf, pleas, cellar	vanish, allow, bare, pronounce	opaque
solid	sound, valid, solar, slide	block, form, matter, crystal, powder	liquid, hollow
books	boot, bones, hooks, rocks, banks	dialogue, authority, record, fact	—
troll	grill, role, tell	wheel, roll, mouth, speak	—
live	love, lies, river, wave, size, bite	breathe, people, dominate, taste, see, be	recorded, dead

Table 4.1: Comparison of patalgorithms showing a selection of results for each.

- 4.1 Seeing the results in a table like this gives an almost immediate idea of how each algorithm works. This is not meant to be transparent and perhaps only after knowing the ins and outs of the algorithms can one recognise how each result was found.

The clinamen results show words that contain one or two spelling errors of the original query term. It is perhaps counter-intuitive to have words such as ‘altar’, ‘leaf’ and ‘cellar’ be classed as spelling errors of the word ‘clear’ but they clearly

could be. Remember that a spelling error can be classed in one of four ways: (1) deletion, (2) insertion, (3) substitution and (4) transposition. So, going from ‘clear’ to ‘altar’ is an instance of two times case 3 (‘c’ is replaced by ‘a’ and ‘e’ is replaced by ‘t’) and going from ‘clear’ to ‘leaf’ is an example of case 1 (‘c’ is deleted) and case 3 (‘r’ is replaced by ‘f’).

Looking at the second column (the syzygy results) shows the semantic relationship between the original query term and the results. Again, this may not be immediately noticeable but certainly once you know how the process works you can recognise the common relations. This is especially evident for the antinomy algorithm which is based on opposites.



However it is equally interesting to compare some full sentences. Looking at some of the poems at the beginning of each chapter shows the variety of the possible outcomes (see pages ??, ??, 2, ??, ??, ??, ??, ??, ??, ??, ??, 24, 54, 65, ??, and 103). It also highlights the difference between the two corpora. Poems based on the Faustrol corpus have a very different sound and feel to it than ones based on the Shakespeare corpus.

There was a period put to the Fire pink and spot earth was flat like the floor of an Oven as much ease as a mower doth the grass	O bloody period I as your lover speak has she such power gather those flowers
during the first period of my captivity room with a hard earthen floor not within everyone’s power or your favourite flowers died	thy lover juiced flowers had I been any god of power or a lover’s lute
shocks lose power the white daisy after a long period	the river hath thrice flow’d but sad mortality o’ersways their power now here a period of tumultuous broils
poppy peony stock to all People	led by their master to the flow’red fields not a minister in his power where souls do couch on flowers

Figure 4.1: Comparison of Faustroll (left) versus Shakespeare (right) poetry for query term ‘flower’

remember to replace some of the chapter poems with shakespeare

Corpus	Query	Results	Reverbs	Origins	Poems
Faustroll	flower	89	24	18	7.8×10^{10}
Shakespeare	flower	157	15	38	3.8×10^{14}
Faustroll	clear	542	79	23	1.3×10^{22}
Shakespeare	clear	1445	72	38	1.5×10^{28}
Faustroll	troll	124	16	16	4.4×10^{12}
Shakespeare	troll	327	14	38	1.1×10^{19}
Faustroll	fania	9	2	6	1
Shakespeare	fania	15	2	14	1

Table 4.2: Faustroll versus Shakespeare in numbers

Sometimes we can even get a general feel for the theme of the poem, as in we can recognize the connection, the relationship between the individual lines and what must be the original query term. Of course putting the poems into the chapters as they are—without specifically stating the keyword they were generated from or the corpus they are based on—makes them a bit more elusive.

The different language is quite obvious. This is helped by the fact that the Shakespeare corpus is of course written by the same author². The Faustroll corpus contains text by over 20 different authors and in three different languages even.

4.2.1 Numbers

The above examples (table 4.1 and figure 4.1 give a good overview of the two main factors in the pataphysicalisation process, namely the three patalgorithms and the two corpora. Both only reflect a small selection of the variety of results produced though. It is therefore quite interesting to look at some actual numbers.

- 4.2 Table 4.2 shows a comparison of the two different corpora with four example query terms.

Results

A ‘result’ in this case is one line (a sentence fragment). This column shows

²Unless of course we believe the legends that Shakespeare didn’t write those works by himself...

the total number of results found by the three algorithms combined. Individual results appear only once but the keyword in contains can appear in several of the results.

Reverbs

A ‘reverberation’ is one of the terms in the list of keywords produced by the pataphysicalisation process. The list cannot contain duplicates but each reverberation can appear in more than one result. Reverberations are used to find results in each corpus. This column shows the total number of reverberations created by the three algorithms.

Origins

An ‘origin’ in this case is the original source text from which a given sentence fragment was retrieved. Each corpus has a set number of source texts. Each origin can contain several results based on several reverberations. This column shows the number of origins in the given corpus in which results were found.

Poems

This refers to the total number of Queneau style poems that can be generated using the given results³. This is calculated as the number of different options per line to the power of the number of lines.

To put this into perspective, the Faustroll corpus contains a total of 28 texts of very varied authors and different languages even. This might explain why **4.2** not the queries in table 4.2 have not found results in all of the texts. The query ‘clear’ found results in 23 out of 28 for example while the query ‘fania’ only found results in 6 texts. The Shakespeare corpus seems much more uniform. Reverberations generally seem to find results in all 38 source texts in the corpus apart from the query ‘fania’. This might be explained by the fact that Shakespeare wrote all of the texts himself using much of the same language and vocabulary unlike the Faustroll corpus.

It is rather interesting to note that even though the Shakespeare corpus produces overall more results from more texts, the Faustroll corpus produces more reverberations per query. This might stem from the multi-author, multi-language **4.2.2** nature of the corpus. The overall vocabulary used is much larger than the Shakespeare one.

Regarding the final column showing the number of possible poems, let’s look at the Shakespeare—clear row. There are 1445 number of results. These are spread over 14 lines, so each line has 103 options. The overall number of poems

³The original book by Queneau contains 10 sonnets with 14 lines each. This means the total number of poems producable by the book is 10^{14} or one hundred thousand million.

		Clinamen			Syzygy			Antinomy			
		Results	Reverbs	Origins	Results	Reverbs	Origins	Results	Reverbs	Origins	Total
Query											
Faustroll	clear	158	20	13	368	90	23	16	8	8	542—79—23
	shine	228	29	19	154	61	16	0	0	0	382—61—20
	disorder	0	0	0	159	127	23	10	2	10	169—40—23
	stuck	59	14	13	181	43	22	11	3	9	251—47—22
Shakespeare	clear	435	20	38	997	90	38	13	8	12	1445—72—38
	shine	575	29	38	333	61	38	0	0	0	908—53—38
	disorder	0	0	0	326	127	38	29	2	29	355—26—38
	stuck	152	14	37	479	43	38	34	3	34	665—41—38

Table 4.3: Results-Reverberations-Origin numbers per algorithm

is therefore calculated as 103^{14} which equals 15,125,897,248,551,112,432,256,145,169
 ■ 4.2 (or 1.5×10^{28} in short).



A slightly different angle to consider is a comparison of these kind of numbers
 ■ 4.3 between each of the algorithms. Table 4.3 shows the numbers of results, reverberations and origins for the Clinamen, Syzygy and Antinomy algorithms using four example query terms ('clear', 'shine', 'disorder' and 'stuck') for each of the two corpora ('Faustroll' and 'Shakespeare').

add french query term

The first immediate observation surely must be that the Antinomy algorithm produces the fewest results, in two cases even none at all. This is caused by
 § ?? the fact that the Antinomy algorithm is based on semantic opposites in WordNet and some words simply do not have defined opposites. Addressing this issue
 § 5 was left for future work mentioned in chapter 5. On the other hand the Syzygy
 § ?? algorithm, which is also based on WordNet, produces most results on average.

§ ?? The Clinamen algorithm interestingly produces a varying number of results depending on the query term. For the query 'disorder' no results were found in either the Faustroll or the Shakespeare corpus. This of course is rooted in the

fact that no reverberations were produced during the pataphysicalisation process. Here it is important to remember that the Clinamen algorithm makes use § 4.2.3 of a base document⁴. Therefore the success of the algorithm depends on the vocabulary of this base text. In this particular example this means that there was no word in the base text of one or two spelling errors to the original query of ‘disorder’.

- 4.3 Looking at the origins column in table 4.3 highlights how the Shakespeare corpus mostly produces results from each of its 38 texts. The Faustroll corpus varies a lot more. This may be due to the different languages and varying word counts of the files in the corpus.

Faustroll

- There are three empty texts (Peladan, de Chilra, de Regnier).
- The total number of words is 1,738,461. Of this, 1,204,158 words are from English texts (70%), 497,144 are French (28%) and 37,159 are in German (2%).
- The shortest text contains 3853 words (Coleridge).
- The longest text contains 419,456 words (Poe).
- The average amount of words per text is 62,088.
- The vocabulary of the index contains 78,893 words. Of this 49,040 are english terms.

Shakespeare

- The total number of words is 883,460⁵.
- The shortest text contains 2568 words (Lover’s Complaint).
- The longest text contains 32,031 words (Hamlet).
- The average amount of words per text is 23,249.
- The vocabulary of the index contains 23,398 words.

§ ?? The full list of stopwords per language can be found in appendix ??.

HERE

⁴This is hardcoded to be Jarry’s *Exploits and Opinions of Doctor Faustroll, Pataphysician*. Section 4.2.3 discusses what would happen if we changed the base document to something else.

⁵According to (Efron and Thisted 1976) Shakespeare used 31,534 different words in his works, about half of which he only used once (14,376). They cite the total number of words used in his corpus as 884,647.

pp_sent faustroll clear with all sentences 8751 faustroll clear with the first sentence only 542

Francois Rabelais: Gargantua and Pantagruel

term: cellar

positions: [4448, 18718, 68678, 110318, 192486, 267241, 352502, 352565]

sentence: rope wine is let down into a cellar

sentences:

- rope wine is let down into a cellar
- bread and holy water of the cellar
- year who had a cool cellar under ground
- cellar
- that Nick in the dark cellar
- on the cellar door
- in mind of the painted cellar in the oldest city in the world
- and the painted cellar also

This is also a lot more time consuming. A way around this would be to store each sentence with each word in the index directly.

calculate averages of how many positions the average index term has

4.2.2 Index

look up google index or other examples or crawls

The index is a central part of the `pata.physics.wtf` system. It is generated when the program/server is first started up but then cached and re-used. The initial process of going over all the text files in each corpus takes a few minutes. Of course in comparison to a full Internet crawl this is a tiny amount of data to be processed.

The Faustroll corpus for example contains 28 texts⁶

check which ones are empty

. Individually they are small plaintext files of sizes between 24KB (Coleridge) and 2MB (Poe). This is of course caused by the nature of some of these texts. Samuel Coleridge's *The Rime of the Ancient Mariner* is a poem whereas the Edgar Allan Poe file is a whole collection of his works. The whole size of the Faustroll corpus

⁶This is technically not true since a few of those files are empty

is 10MB. The Shakespeare corpus is much more evenly distributed as all of his works are separated out into individual text files of an average size of around 150KB. The total size of the Shakespeare corpus is only 5.3MB.

Now, the size of the index is interesting. Processing the Faustroll corpus alone produced an index of 12.4MB. That's larger than the actual size of the corpus. Remember, the index contains each word that occurs anywhere in the corpus together with the list of files it is found in and the specific locations within each text. This includes english words buts also french and spanish and german terms since the faustroll corpus is multi-lingual.

how big is the new combined index?? - its 35.2MB

how many words in shakespeare vocab compared to faustroll vocab?

storing rhyming data in index or other additional things like ranking

4.2.3 Clinamen

The clinamen function uses the damerau-levenshtein algorithm to create pata-physicalised words. It also uses the Faustroll text. The way this works is as follows. If the query term is a spelling error of size 1 or 2 of a term in the vocabulary within the faustroll text then it is included in the list of resulting terms. The logic behind this is due to the damerau levenshtein algorithm needing two words to compare with each other. It also ensures we get real words as results and not some random gibberish.

Currently the algorithm is set to accept terms that have a difference of 1 or 2 to the original query. We can lower this to 1 to allow fewer results or increase it to make it broader. I felt 1 or 2 was a good compromise. Only allowing 1 error would mean terms are too similar. Allowing 3 might mean they are drastically different.

show clinamen results with a real dictionary rather than a base text

Changing the base text in Clinamen

As examples of using different base documents in the Clinamen algorithm I have used three examples.

- Midsummer Night's Dream by Shakespeare (Dream in short)
- Arabian Nights by various artists (Nights in short)

Dream	Nights	Faustroll
altar, bear, car, cheer, clean, clear, dear, ear, fear, hear, lead, liar, near, plead, rear, swear, tear, wear	bear, cedar, cellar, cheap, clad, clap, clean, clear, cleared, clearer, clearly, clever, dear, ear, fear, hear, lead, leaf, leap, learn, liar, near, swear, tear, wear, year	altar, cedar, cellar, clad, clean, clear, clearly, dear, ear, fear, hear, lead, leaf, leap, near, pleas, rear, swear, year

Table 4.4: changing base in clinamen

Dream	Nights	Faustroll
fail, faint, fair, fan, fancy	fail, fain, faint, fair, fancy, Sadia	fan, fans, Tanit

Table 4.5: changing base in clinamen

- Exploits and Opinions of Doctor Faustroll, Pataphysician by Jarry (Faustroll in short)

altar, bear, car, cheer, clean, clear, dear, ear, fear, hear, lead,
 ↳ liar, near, plead, rear, swear, tear, wear

clinamen with up to 1 error

faustroll clear:
 clean, clear

faustroll fania:

-

faustroll moss:
 loss, mass, most

clinamen with up to 2 errors

faustroll clear:
 altar, cedar, cellar, clad, clean, clear, clearly, dear, ear, Fear, fear, hear, lead,
 leaf, leap, near, pleas, rear, swear, year

faustroll fania:
 fan, fans, Tanit

Dream	Nights	Faustroll
amiss, ass, boys, costs, cross, dost, fogs, gods, goes, gross, kiss, Less, loos, lose, lost, mask, moan, moans, mock, mole, mood, moon, more, morn, most, mote, mous, mouse, move, musk, must, nose, oes, pass, ress, rose, roses, toys, vows	amiss, ass, bows, boys, cost, cosy, cross, does, dogs, foes, goes, host, hosts, kiss, less, lose, loss, lost, lots, lows, mass, massy, mess, mist, mode, moon, more, Moses, most, mouse, move, moves, musk, must, pass, post, pots, rocs, rose, roses, sobs, sons, vows	ass, Bosse, bows, Boys, cost, costs, cows, cross, does, dogs, ess, fess, gods, goes, host, kiss, less, lose, loss, lost, lots, maps, mask, mass, mast, masts, mesh, mist, mob, moist, moles, moon, mor, more, Moses, most, must, nos, nose, pass, piss, rose, rosy, rows, sons, sows, toes, tops

Table 4.6: changing base in clinamen

faustroll moss:

ass, Bosse, bows, Boys, cost, Cost, costs, cows, cross, does, dogs, ess, fess, gods, goes, host, kiss, less, lose, loss, lost, lots, maps, mask, mass, mast, masts, mesh, mist, mob, moist, moles, moon, mor, more, Moses, most, must, nos, nose, pass, piss, rose, rosy, rows, sons, sows, toes, tops

clinamen with up to 3 errors

faustroll clear:

afar, ahead, Alas, altar, appear, bar, beam, beard, bears, beat, beer, ble, bleed, blew, bluer, bread, break, Caesar, calvary, can, canal, care, cedar, cellar, chair, charm, cheek, chen, chere, chern, choir, clad, claim, clasp, claws, clean, clear, clearly, clerks, climb, clock, clogs, close, cloth, color, coral, crab, crap, cresc, crest, Dead, dead, dear, Dewar, ear, ears, eat, ever, far, fear, Fear, feat, flag, flat, flesh, floor, Friar, glare, Great, great, head, hear, heard, heart, heat, Her, her, idea, ideal, ideas, jar, law, lay, lead, leaf, leap, least, leave, led, lees, left, leg, legs, lent, leper, less, lest, let, mean, meat, near, oar, Ocean, Opera, over, peak, pearl, per, plat, pleas, read, Read, real, rear, sea, Sea, seat, sheer, slab, sleep, solar, speak, star, steam, sugar, swear, swears, sweat, tean, tears, their, vulgar, war, year, years, zeal

faustroll fania:

acid, aid, aim, air, an, ance, and, animae, animal, Anna, ant, anti, ants, anvil, any, axis, Baba, bank, banks, basin, cabin, can, canal, Cane, canvas, dance,

Danzig, data, Denis, fa, face, faced, faces, facet, facing, fact, facts, fading, falt, faith, fake, fall, falls, false, family, fan, fans, far, fat, fate, fauns, favor, final, find, finds, fine, finer, fins, flint, fluid, foil, frances, fruit, gain, habit, hair, hand, hands, india, Jane, Janus, Kaka, Kantian, laid, lance, land, lanes, Latin, lava, mail, main, Man, man, many, nadir, nail, nib, nil, pair, pan, Pan, Papio, papio, Paris, rang, range, rapid, said, sail, Saint, saliva, San, sand, sang, sonic, tail, Tait, Tanit, tunic, unit, vain, valid, van, vanish, vanity, vans, vina, Yan

faustroll moss:

abyss, Across, across, acts, adds, Alas, almost, also, among, amor, amore, amour, ants, apes, arms, arose, as, As, ash, ask, ass, axis, bars, base, bases, beds, best, bis, blows, Boat, boat, boats, body, bolus, bone, bones, book, books, boot, boots, bores, born, Bosse, both, bout, bow, bowl, bows, box, boy, Boys, brass, brows, bust, case, cases, cash, cast, chose, clogs, close, co, coast, coats, Code, coins, cold, come, comes, cool, copy, cords, cost, Cost, costs, cows, crass, cross, cuIs, cups, days, demons, Deus, disk, disks, Do, do, does, dogs, dome, domos, done, door, doors, douds, down, Down, dress, drops, dust, ears, ease, easy, eats, eggs, ells, else, ends, Eros, ess, est, eyes, fans, fess, fins, fish, fist, fists, foam, fog, foil, folds, foot, For, for, fore, fork, Form, form, forms, fotms, foul, four, fox, foxes, Ghost, ghosts, glass, glows, go, God, gods, goes, Gog, Gogh, gold, Gold, gong, good, goods, gown, gowns, grass, hams, has, hast, His, his, ho, Ho, holds, holes, Holy, home, Homo, hoof, hooks, hope, horn, horns, Horse, horse, horses, host, hot, hour, Hour, hours, house, houses, how, How, humors, hums, ikons, iris, irs, is, Is, Its, its, jaws, Jesus, jibs, job, John, jowls, joy, Just, just, kiosks, kiss, knows, last, laws, Lays, lees, legs, less, lest, lies, lions, lips, Lo, lobe, loins, Long, long, looks, Lord, lord, lords, lore, lose, loss, lost, Loti, lots, loud, louse, Love, love, loves, low, Loye, m, made, mail, main, make, makes, male, man, many, map, maps, mask, mass, masses, mast, masts, may, me, mean, means, meat, meet, men, mere, mesh, meshes, met, milk, mimes, mist, mite, mites, mob, moist, moles, month, months, moon, mor, more, Moses, most, motor, mount, Mour, mouth, mouths, moved, mower, Mrs, much, music, must, Must, my, nest, news, nisi, no, No, noise, non, none, noon, Nor, nor, nos, nose, Not, not, note, now, Now, nuts, o, oak, oar, oars, oc, odd, of, off, ofQ, oil, old, on, one, ones, or, orb, orms, our, out, own, pass, past, pigs, piss, Plus, Poe, poets, pole, poles, ponds, Poor, poor, pope, port, Pour, prose, Prose, rats, rays, rest, rise, rises, road, robe, robes, rock, rocks, rod, Roi, role, roll, rolls, rome, roof, room, rooms, root, rope, ropes, rose, rosy, row, rows, s, says, sc, sets, shops, smock, smoke, So, so, soft, sole, Some, some, son, songs, sons, soon, Soon, sorb, soul, souls, sows, sums, suns, tats, This, this, those, Thus, thus, tjis, to, To, toad, toads, tock, toes, told, tome, tone, toO, too, took, top, tops, tore, torn, tossed, Town, town, Tres, tres, ups, us, use, vans, vast, Was, was, wash, wasps,

webs, whose, wigs, Woan, won, wont, wood, word, words, wore, Work, work, Works, works, worm, worn, wove, Yes, yolk, York, you, You, your, Your

4.2.4 Syzygy

check out what is happening with the hyponyms in the getnym function

The syzygy function goes through the following process.

semantic hierarchy visualised?

It shows each step in the algorithm for the query term ‘clear’.

1. A set of synonyms (a “synset”) is generated.

SYZYGY synsets: [Synset('clear.n.01'), Synset('open.n.01'), Synset('unclutter.v.01'),
Synset('clear.v.02'), Synset('clear_up.v.04'), Synset('authorize.v.01'), Synset('clear.v.05'),
Synset('pass.v.09'), Synset('clear.v.07'), Synset('clear.v.08'), Synset('clear.v.09'),
Synset('clear.v.10'), Synset('clear.v.11'), Synset('clear.v.12'), Synset('net.v.02'), Syn-
set('net.v.01'), Synset('gain.v.08'), Synset('clear.v.16'), Synset('clear.v.17'), Syn-
set('acquit.v.01'), Synset('clear.v.19'), Synset('clear.v.20'), Synset('clear.v.21'), Syn-
set('clear.v.22'), Synset('clear.v.23'), Synset('clear.v.24'), Synset('clear.a.01'), Syn-
set('clear.s.02'), Synset('clear.s.03'), Synset('clear.a.04'), Synset('clear.s.05'), Syn-
set('clear.s.06'), Synset('clean.s.03'), Synset('clear.s.08'), Synset('clear.s.09'), Synset('well-
defined.a.02'), Synset('clear.a.11'), Synset('clean.s.02'), Synset('clear.s.13'), Syn-
set('clear.s.14'), Synset('clear.s.15'), Synset('absolved.s.01'), Synset('clear.s.17'),
Synset('clear.r.01'), Synset('clearly.r.04')] synset item:clear.n.01 hypernym out:innocence
[] synset item:open.n.01 hypernym out:area hypernym out:country hypernym
in:country [] synset item:unclutter.v.01 hypernym out:change hypernym in:change
hypernym out:alter hypernym out:modify [] synset item:clear.v.02 hypernym
out:make hypernym in:make hypernym out:create [] synset item:clear_up.v.04 []
synset item:authorize.v.01 hyponym out:probate hyponym out:approve hyponym
out:O.K. hyponym out:okay hyponym out:sanction hyponym out:certificate hyponym
in:certificate hyponym out:commission hyponym out:declare hyponym in:declare
hyponym out:license hyponym out:licence hyponym out:certify hyponym out:validate
hyponym out:formalize hyponym out:formalise hypernym out:permit hypernym
in:permit hypernym out:allow hypernym in:allow hypernym out:let hypernym
in:let hypernym out:countenance hypernym in:countenance [] synset item:clear.v.05
hyponym out:clear-cut hyponym out:deforest hyponym out:disforest hyponym
out:disafforest hyponym out:denude hyponym out:bare hyponym in:bare hyponym
out:denudate hyponym out:strip hyponym out:stump hyponym out:remove hy-

pernym out:take hypernym in:take hypernym out:take_away hypernym out:withdraw
[] synset item:pass.v.09 hyponym out:clear hyponym in:clear hypernym out:succeed
hypernym in:succeed hypernym out:win hypernym out:come_through hyper-
nym out:bring_home_the_bacon hypernym out:deliver_the_goods [] synset item:clear.v.07
[] synset item:clear.v.08 hypernym out:vanish hypernym in:vanish hypernym
out:disappear hypernym out:go_away [] synset item:clear.v.09 hyponym out:hop
hypernym out:pass hypernym in:pass hypernym out:overtake hypernym out:overhaul
[] synset item:clear.v.10 hypernym out:clarify hypernym out:clear_up hyper-
nym out:elucidate [] synset item:clear.v.11 hypernym out:free hypernym in:free
hypernym out:discharge [] synset item:clear.v.12 hypernym out:rid hypernym
out:free hypernym in:free hypernym out:disembarrass [] synset item:net.v.02
hypernym out:yield hypernym out:pay hypernym in:pay hypernym out:bear []
synset item:net.v.01 hypernym out:profit hypernym out:gain hypernym in:gain
hypernym out:benefit hypernym in:benefit [] synset item:gain.v.08 hyponym out:eke_out
hyponym out:squeeze_out hyponym out:gross hyponym out:profit hyponym out:turn_a_pro
hyponym out:rake_in hyponym out:shovel_in hyponym out:rake_off hyponym
out:take_home hyponym out:bring_home hyponym out:yield hyponym out:pay
hyponym in:pay hyponym out:bear hypernym out:get hypernym out:acquire []
synset item:clear.v.16 hypernym out:sell [] synset item:clear.v.17 hypernym out:pass
hypernym in:pass hypernym out:clear hypernym in:clear [] synset item:acquit.v.01
hyponym out:purge hyponym out:vindicate hyponym out:whitewash hypernym
out:pronounce hypernym in:pronounce hypernym out:label hypernym out:judge
hypernym in:judge [] synset item:clear.v.19 hypernym out:settle hypernym out:square_off
hypernym out:square_up hypernym out:determine hypernym in:determine [] syn-
set item:clear.v.20 hypernym out:change hypernym in:change hypernym out:alter
hypernym out:modify [] synset item:clear.v.21 hypernym out:empty hypernym
in:empty [] synset item:clear.v.22 hypernym out:take_out hypernym out:move_out
hypernym out:remove [] synset item:clear.v.23 hypernym out:empty hypernym
in:empty [] synset item:clear.v.24 hypernym out:remove hypernym out:take
hypernym in:take hypernym out:take_away hypernym out:withdraw [] synset item:clear.a.01
[] synset item:clear.s.02 [] synset item:clear.s.03 [] synset item:clear.a.04 [] syn-
set item:clear.s.05 [] synset item:clear.s.06 [] synset item:clean.s.03 [] synset
item:clear.s.08 [] synset item:clear.s.09 [] synset item:well-defined.a.02 [] syn-
set item:clear.a.11 [] synset item:clean.s.02 [] synset item:clear.s.13 [] synset
item:clear.s.14 [] synset item:clear.s.15 [] synset item:absolved.s.01 [] synset
item:clear.s.17 [] synset item:clear.r.01 [] synset item:clearly.r.04 []

4.2.5 Images

The image search can produce quite interesting results as well. A search for “blue kitten” on Flickr produces the following results: “[artistrocratical, depressed,

blueing, drab, puritanic, wild blue yonder, kitty, dingy, blueness, blue air]" which are then passed into ten separate [API](#) calls to retrieve one image each (see fig below XYZ).



Figure 4.2: image spiral flickr

For Getty the image search works slightly differently due to its [API](#) restrictions. The query "blue kitten" gets turned into the word "racy" which then calls the [API](#) to retrieve ten results (see below).

The difference is staggering.

4.3 Design

Content Perception

It is interesting to note how different the search results are perceived when presented in a different style (e.g. list rather than poem). This could be studied using focus groups using questionnaires and interviews or eye tracking tools to find out what users prefer or perceive as more creative for example.

poem vs list here



Figure 4.3: image spiral getty

4.4 Science Fiction

Where does this project stand in the wider world and the progress of computing, [AI](#) and creativity? [AI](#) and robotics is alluring as a research topic because it is so prevalent in Science Fiction. Computer creativity rarely plays a central role though. We regularly read headlines that tell us that yet another kind of [AI](#)-bot has won some game against a human player. Or we see videos of some innovative ground-breaking kind of new robot which claims to be near perfectly human-like (and yet cannot walk up stairs). There are so many examples of advances that are hailed as the next big thing which aren't all that great.

4.4.1 AI

This is also evident in games, for example [Virtual Reality \(VR\)](#) and [Augmented Reality \(AR\)](#). The Oculus Rift and similar systems are advertised so much you might believe they are actually about to hit mainstream and every kid will own a [VR](#) console and headset. Yet they are still way too expensive to be mainstream and motion sickness is also still an issue (and probably always will). These industries are so "hip" any publication is seen as the new cool thing without taking into account the history and work that has been done previously in perhaps slightly different disciplines. This is the case for example with a recent article on [VR](#) sickness and how to combat it. This is a well known problem

already—motion sickness already exists in normal games. Similar to epilepsy problems.

find links for motion sickness

find links for epilepsy

find links for oculus rift and pokémon go etc

AR has very recently received a massive boom thanks to Pokémon Go (released in Australia, New Zealand and the USA in July 2016). It has become a phenomenon since then.

find pokémon links

What about IBM's Watson⁷, Microsoft's Twitter AI Chatbot⁸, Google's AlphaGo⁹ and Hanson Robotics Sophia robot¹⁰? How does this relate to my work? Practically of course they are all unrelated. On a deeper level though we can start asking interesting questions.

IBM Watson

Watson is a question answering expert system. It famously won against human Jeopardy! champions in 2011.

Microsoft Chatbot

Google AlphaGo

AlphaGo is a system for playing the game Go. It won against a top human professional player in 2015.

Hanson Sophia

I think these are interesting examples to study since they are supposedly on the forefront of AI development. Life-like robots like Sophia still live in the ‘uncanny valley’. Her voice is creepy and unhuman, her intelligence or her capabilities if understanding conversations are clearly flawed (as shown by her viral remark about supporting genocide).

check

⁷See <http://www.ibm.com/watson/>

⁸See <http://www.ibm.com/watson/>

⁹See <https://deepmind.com/alpha-go>

¹⁰See <http://www.hansonrobotics.com/>

Watson is clever and fast in finding answers for specific questions but he still had problems with humour (e.g. BLAHBLA

find example

) but information lookup is arguably fairly easy and straightforward process within IR—sure, it requires processing power and memory storage or access but it is based on simple matching of keywords, not any fancy heuristic algorithms. Microsoft's twitter chatbot went viral and users 'taught' it nasty swearwords

check

quickly and Microsoft had to take the bot down. It has since apologised although any official documentation on it has disappeared

check

. Google's AlphaGo has been hailed as a breakthrough in AI but similar to Watson it is a very targeted and limited program.

To me it seems the real breakthrough happens when (and if) the first robots appears which isn't as big as a house, can play Go, Chess and hide-and-seek, genuinely manages to get around the uncanny valley effect, has vast knowledge in his memory for instant information lookup, can hold a normal conversation without causing a war, etc, etc—you get the picture. General AI is where it's at. Humans can do all the things we do. Children aren't born with only a single function. Imagine a world where humans only have one specialism and can't do anything else. Mary is a Chess player but can't move her arms. Bob is a medical diagnosis expert but he can't hold a conversation. Movement, speech, memory—they are all vastly complex systems. And I haven't even touched creativity yet.

what's the point I'm making? how does this relate to my work?

Perhaps this 'uncanny valley' exists in creativity too. If a robot who looks vaguely human but not quite well enough, or he/she/it sounds almost human but not quite—perhaps if a robot can crack a joke like a human but not quite—perhaps this could be considered uncanny valley too? The philosophical zombies I mentioned in chapter ?? live in this uncanny valley?

p and H creativity for computers?

4.4.2 Brains

I'm not talking about the beer or the zombie food but rather research into the human brain (or animal brains) and attempts to model it on a computer.

The motivation here is that once we understand how the brain works, perhaps we can understand how certain cognitive processes really work and this of course include creativity.

This is no easy task of course. Chris Chatham talks about ten “important Differences Between Brains and Computers”¹¹ which give a good overview of some of the difficulties of trying to model a brain as is. We can't just do a 1-1 copy.

1. Brains are analogue; computers are digital
2. The brain uses content-addressable memory
3. The brain is a massively parallel machine computers are modular and serial
4. Processing speed is not fixed in the brain; there is no system clock
5. Short-term memory is not like RAM
6. No hardware/software distinction can be made with respect to the brain or mind
7. Synapses are far more complex than electrical logic gates
8. Unlike computers, processing and memory are performed by the same components in the brain
9. The brain is a self-organising system
10. Brains have bodies
11. The brain is much, much bigger than any (current) computer

Chris Chatham

To bring this into perspective Ray Kurzweil claims the brain is capable of 10^{16} operations per second (2013, p.194). Japan's K-computer (the worlds largest super computer as of 2016) currently has that power—10 petaflops. The “Blue Brain Project” is aiming to model 10^{17} bytes of memory and 10^{18} flops by 2023 (Kurzweil 2013, p.125).

find k-computer reference

There are currently some major research projects going on. One of them is the “Human Brain Project” (Walker 2012).

quotes:

¹¹<http://scienceblogs.com/developingintelligence/2007/03/27/why-the-brain-is-not-like-a-computer/>

Our brain consumes about 30W, the same as an electric light bulb, thousands of times less than a small supercomputer. (Walker 2012, p.17)

For environmental and business reasons, vendors have set themselves the goal of containing energy consumption to a maximum of 20 megawatts (Walker 2012, p.41)

the 1 PFlop machine at the Jülich Supercomputing Centre could simulate up to 100 million neurons – roughly the number found in the mouse brain. (Walker 2012, p.41)

Cellular-level simulation of the 100 billion neurons of the human brain will require compute power at the exascale (10¹⁸ flops). (Walker 2012, p.41-42)

2017 petascale 50petabytes memory + 50 petaflops + <=4MW power

2021 exascale 200petabyte memory + 1exaflop

A second, equally important goal will be to prepare the procurement of the HBP Pre-exascale-supercomputer. By 2017/18, Jülich plans to procure a Big Data-centred system with at least 50 PBytes of hierarchical storage-class memory, a peak capability of at least 50 PFlop/s and a power consumption <= 4 MW. The memory and computational speed of the machine will be sufficient to simulate a realistic mouse brain and to develop first-draft models of the human brain. (The rest of the hardware roadmap targets an exascale machine in 2021/2022 with a capability of 1 EFlop/s and a hierarchical storage-class memory of 200 PB).¹²

Why Minds Are Not Like Computers (Schulman 2009) Software – Hardware == Mind – Brain ??? analogy

"The power of the computer derives not from its ability to perform complex operations, but from its ability to perform many simple operations very quickly."

Layers of abstraction in computers:

1. user interface
2. high level programming language
3. machine language
4. processor microarchitecture
5. Boolean logic gates
6. transistors

layers of abstraction in brain:

¹²<https://www.humanbrainproject.eu/high-performance-computing-platform>

1. personality?
2. Thinking?
3. Chemical /electrical signals/activity?
4. Divided Brain regions/structure
5. Neurons
6. Dendrites (input) and axons (output)?

Computers are faster and better than humans in many tasks already.

"The weaknesses of the computational approach include its assumption that cognition can be reduced to mathematics and the difficulty of including noncognitive factors in creativity." ([Mayer 1999](#), p.457)

[find references](#)

[neural networks and other models based on the brain](#)

Perhaps we need to have that complete picture of how the brain works in order to understand human creativity. I would argue computer creativity is part of general [AI](#), and for general [AI](#) we need massive amounts of general knowledge.

[common sense research](#)

[again talk about how this is relevant for my project](#)

Expert Systems vs General AI Is computer creativity an expert system or does it fall into general [AI](#)?

Machines self-assessing Perhaps there is an argument that if humans are the only entities who can judge whether another human is being creative, then machines should be assessing themselves. This is a paradoxical concept though. Since machines are products made by humans, they can never be autonomous in that sense. If machines had evolved like other animals besides us this argument might hold but obviously that is not the case.

4.5 Meta

4.5.1 Management

[add file for appendix with full git history](#)

On a different note, the project was completed over X years which includes an interruption and later on only a part time commitment.

I kept the project in a “git repository”. Git is a version control system that allows users to roll-back on changes and I further pushed my work to GitHub to make sure hardware failure or human error (i.e. lost or stolen property) would not affect my work.

To understand git you need to know what commits are. They are the thing where I save my current state of the project and give it a description.

Below you can see a shortened version of the timeline of my commits between 20XX and the time of submission of this thesis. A full version can be found in appendix XYZ. You can see from this the time between programming work I did on [pata.physics.wtf](#) and its predecessors.

add calendar screenshot of github contributions

links to git and github

```
*   10f61f9  Sun 08 May 2016  (HEAD -> api, origin/api) Merge remote-tr
| \
* | 71437f6  Tue 18 Aug 2015  Flickr and Bing work, radio buttons work
* | 6c552aa  Wed 12 Aug 2015  Fixed image problem but not video.
| | * 1cbb63d  Tue 11 Aug 2015  (origin/thesis) Update textsurfer.py
| |
| |
* | 0ebff0d  Tue 11 Aug 2015  Analytics enabled again
* | 703f977  Tue 11 Aug 2015  Problems solved.
* | 74a1fae  Tue 11 Aug 2015  About to change l\dict to dict of dict
* | 0935b23  Mon 10 Aug 2015  BUG FUCKER
* | 4f7d91e  Mon 10 Aug 2015  Turn debug off
* | 58f0c2b  Mon 10 Aug 2015  Button styling done
* | 59add58  Mon 10 Aug 2015  Email problem solved
* |   f1b2d40  Sun 09 Aug 2015  Merge branch 'Deploy' into thesis
| \
| *
| * | 435cb2d  Sun 09 Aug 2015  Deployment works, added analytics
| * | 8a63dc7  Sat 08 Aug 2015  gunicorn runs locally fine.
| * | 2861407  Sat 08 Aug 2015  Revert 5f2c957..4026965
| * | 4026965  Sat 08 Aug 2015  Tests
* | | 8f2eeab  Sat 08 Aug 2015  Merge branch 'w3' into thesis
| \
| \ \
```

```

| | / /
| * | 5f2c957 Sat 08 Aug 2015 Stuff
| * | 873153c Fri 07 Aug 2015 Tiny cleanup
| * | 05d5760 Thu 06 Aug 2015 Random Poems and Emailing works
| * | 657126c Wed 05 Aug 2015 Random poems work - without links though
| * | 3d31ea9 Wed 05 Aug 2015 Randomise still only works once, count c
| * | 5f1d45b Wed 05 Aug 2015 Randomise poem works ONCE
| * | c583341 Wed 05 Aug 2015 Poem subtabs, email poems done
| * | f1b3878 Wed 05 Aug 2015 Hiding divs
| * | a6939c4 Tue 04 Aug 2015 huh?
| * | e6b411d Tue 04 Aug 2015 Poem emails WORK Fuck YEAH!
| * | 4b6b170 Tue 04 Aug 2015 Test email
| * | 24e356c Tue 04 Aug 2015 Better load icon
| * | e6ae736 Tue 04 Aug 2015 loading icon version 1
| * | 51b43e2 Tue 04 Aug 2015 Added 4th pictures
| * | f2d8a83 Mon 03 Aug 2015 Minor fixes
* | | 1ddb03d Mon 03 Aug 2015 Merge branch 'w3' into thesis
| \ \ \
| | / /
| * | ca4eab3 Mon 03 Aug 2015 Pretty good state.
| * | 9370334 Mon 03 Aug 2015 working on list display of images [REDACTED]
| * | e1flead Mon 03 Aug 2015 Stylesheets sorted and cleaned files [REDACTED]
* | | 9732d5b Mon 03 Aug 2015 Merge branch 'w3' into thesis
| \ \ \
| | / /

```



I also kept the thesis under git version control. Since the thesis was written in \LaTeX you could almost say I ‘programmed’ it. Below is an outline of the commit history for this thesis.

- * 3f06260 Edited readme again
- * c721b33 Edited readme
- * ffbdb4b Edited readme
- * 8870b3d Added gitignore file
- * bala9c2 Second commit
- * 244c4b3 First commit

4.5.2 Thesis

Part Spirals

Each new thesis part contains a word spiral based on a poem generated by `pata.physics.wtf` using the a part of the title as keyword. They represent the pataphysical (Archimedean) spiral.

1. Preface — ***pre***
2. Hello World — ***hello***
3. Tools of the Trade — ***trade***
4. The Core: Techno-Logic — ***core***
5. The Core: Techno-Practice — ***practice***
6. Meta-Logicalysis — ***meta***
7. Happily Ever After — ***after***
8. Postface — ***post***

Chapter Poetry

Each chapter opens with a poem generated by `pata.physics.wtf` using a part of the chapter title as keyword.

1. Introduction — ***intro***
2. Inspirations — ***inspiration***
3. Methodology — ***method***
4. Pataphysics — ***pata***
5. Creativity — ***creativity***
6. Technology — ***technology***
7. Evaluation — ***evaluation***
8. Foundations — ***foundation***
9. Interpretation — ***interpretation***
10. Implementation — ***implementation***
11. Applications — ***application***
12. Patanalysis — ***patanalysis***
13. Aspirations — ***aspirations***
14. Observations — ***observations***

say more, check keywords, potentially generate new poems

creative analysis

literary deconstruction and recombining to make new creative output?
perception of results (poetry, source, algorithm)
discuss applications from before (stimulates creative detour away from the obvious)

How does this relate to Oulipo and Pataphysics?

Perhaps this is where I should talk a bit about the perception of results in their different output formats/styles. The poetry is automatically read with more gravity. Sorting by sources is a game of exploration or algorithms which becomes a game of finding the similarities within the result sets. They are different ways to view the same things and yet have a drastic influence of how the results are perceived. This also applies to the image and video search. Presenting results in spiral form is weird. Its hard to see where one image ends and another starts, they just kind of blur into each other. When listed as a list they immediately become more boring.

talk abit about what the original plan was for some of the big changed elements in the website, e.g. the image search running 10 times on different keywords rather than running once with 10 results for the same keyword.

DELETE EVERYTHING FROM BELOW HERE:

DELETE THIS

In this section we consider the possible uses and applications for the proposed creative search tool.

Our target audience is not quite as broad as that of a general search engine like Google. Instead, we aim to specifically cater for users who can appreciate creativity or users in need of creative inspiration. Users should generally be educated about the purpose of the search tool so that are not discouraged by what might appear to be nonsensical results. Users could include artists, writers or poets but equally anybody who is looking for out-of-the-box inspirations or simply a refreshingly different search engine to the standard.

The way we display and label results produced by the tool can influence how the user perceives them. The current prototype for example separates the results into its three components but we could have equally just mixed them all together. The less transparent the processes in the background (e.g. which algorithm was used, how does the result relate to the query precisely, etc.) are for the user, the more difficult it might be to appreciate the search.

There are many ways a pataphysical search tool could be used across disciplines.

In literature, for example, it could be used to write or generate poetry, either practically or as a simple aid for inspiration. We are not limited to poetry either; novels, librettos or plays could benefit from such pataphysicalised inspirations. One can imagine tools using this technology that let you explore books in a different ordering of sentences (a sort of pataphysical journey of paragraph hopping), tools that re-write poems or mix and match them together. Even our simple prototype shows potential in this area and could be even more powerful if we extended it to include more base texts, for example the whole set of books contained in Faustroll's library ([20] and also [12]). A richer body of texts (by different authors) would produce a larger index which would possibly find many more matches through WordNet and end in a more varied list of results.

From a computer science perspective it could be used as one of the many algorithms used by traditional search engines for purposes like query feedback or expansion (e.g. "did you mean ... " or "you might also be interested in ... "). Depending on how creative we want the search engine to be, the higher we would rank the importance of this particular algorithm. One of the concepts related to the search tool, namely patadata, could have an impact on the development of the Semantic Web. Just as the Semantic Web is about organizing information semantically through objective metadata, patadata could be used to organize information pataphysically in a subjective way.

The prototype tool is already being used in the creation of an online opera, provisionally entitled from [place] to [place], created in collaboration with The Opera Group, an award-winning, nationally and internationally renowned opera company, specialising in commissioning and producing new operas. In particular, it is being used to create the libretto for one of the virtual islands whose navigation provides the central storyline for the opera. The opera will premiere in 2013, and will continue to develop thereafter, deploying new versions of the tool as they appear.

ASPIRATIONS

5

Mid the silence that pants for breath,
when I thought myself at my last gasp,
haine ou de l'ambition et qui se,
the pale motor vessel withdrew its blue breath toward the island's horizon.

As pure and simple as a powder puff,
such also was the ambition of others upon the like occasion,
there was hardly a breath of air stirring,
mon ancien cœur en une aspiration vers la vertu.

After drawing a long breath,
the silver ring she pull'd,
the suitor cried, or force shall drag thee hence.

For wild ambition wings their bold desire,
and with thine agony sobbed out my breath,
I will pull down my barns.

5.1	Technical	95
5.2	Creative NLP	99
5.3	Theoretical	100



Developing a software product never finishes. Especially with creative products, where the functional requirements are more fluid perhaps, it is always tempting to add, improve, replace bits.

software refactoring

For the purpose of this doctoral project, the artefact (`pata.physics.wtf`) is a snapshot of a product in constant motion. The state of the code at the time of § 2 submission of this thesis is described in chapter 2 and further elaborated on in § 4 the [Patanalysis](#) chapter.

Here, in this chapter I will lay out some of the potential/likely further work for this project. This may continue on a private basis or in a more academic environment. I have grouped these ideas into two main categories: **technical** and **theoretical**.

5.1 Technical

write these out all in one list and then group them as fit

Responsive spirals Currently the image and video spirals are fixed size. This means that when the webpage is resized the spiral stays the same size and is left aligned on the page. Ideally it would be better to scale the spiral with the width of the browser page. Percentages

Scalable image sizes At the moment images are retrieved at a given size through the various [API](#) calls. Because images in the spiral have different sizes according to where in the spiral they are located, they are scaled up or down directly in the [HTML](#) code. This means that some of them look squished and pizelated. This limits the available choice of results through the API.

Square aspect ratio Another issue is the aspect ratio of images and videos. For the spiral they need to be square. I currently achieve this by squishing them as opposed to cropping them or specifying an option in the [API](#) calls to only retrieve square images.

Responsive poems A similar problem to the responsive spirals exists with the display of the Queneau poems. The random poems are centered on the page but the Queneau poems require a lot more formatting and styling to render them on the page and currently this is achieved by left aligning them and having a fixed ‘absolute’ position on the page. Ideally this would also be centered as in the random poems.

Startup performance The website can be slow to load. Currently speed performance was not a priority during development. In fact it is not built for speed from the ground up. Each time the server restarts, the indexing process takes place from scratch. This takes time. Google and other big web search engines do this continuously in the background to keep data up to date. The index is currently cached after startup but perhaps preprocessing it and storing it more permanently in a database would help speed up the start. However this may not be necessary, as it only affects the server startup.

Query speed The time it takes from the user entering a query term and the system displaying the results page varies between unnoticeable short and impatiently long. This is due to the pataphysicalisation process. This requires calls to external and internal APIs such as Flickr and WordNet.

Preprocessing corpora At this point the texts in the corpora consist of almost unedited plaintext (.txt) files¹. Newlines and whitespace formatting varies, as does language and quality of spelling. OCR SOURCES Generally, chapter headings, chapter numberings, etc are left untouched. The Shakespeare corpus contains poetry and plays for example. STAGE DIRECTIONS With the plays, scene information is kept, voice details are kept. This means sentences that appear in the results of the search tool can contain peripheral words such as in this example: “...Athens and a wood near it ACT I...” from *A Midsummer Night’s Dream* or this example: “...Exit SHERIFF Our abbeys and our priories shall pay This expedition’s charge...” from *King John*. This could be addressed by preprocessing the individual texts in advance.

Sentence fragments Currently the way results sentences are retrieved for the text search is based on punctuation. This means once a pataphysicalised keyword has been found, the system retrieves up to 10 words prior until it reaches a punctuation mark and the same for after. The idea here was to get suitable sentence fragments.

¹For text files downloaded from Project Gutenberg, the Gutenberg specific copyright notices have been removed to only contain the relevant body of text

More APIs Currently X APIs are used². This could be increased to include more varied sources of data. Sites like Flickr are heavily based on user tags ('folksonomies') which can be unreliable and a bit random at times.

Web search The use of APIs could also include web search results rather than just images and videos. This would need its own interface section and a suitable display style for the results. The biggest problem for this is API restrictions. Alternatively a ready-made index or crawl could be used but these are typically many terabytes in size and have a cost attached. Crawling the Web myself is not an option due to the computational power, time and space required to do so.

Audio search Originally audio search was going to be a part of this project. This has been abandoned due to time constraints. However it could be added using an API such as SoundClouds. Technically the pataphysicalisation could work similar to the image and video searches, meaning it would be based on user tags. One idea would be to search in audio waves.

More algorithms It would be nice to implement some more algorithms for the search tool. This could include the two additional algorithms suggested by An-
§ 3 drew Dennis (see chapter 3) or developing more of my own. This could involve implementing some of the other pataphysical principles, such as equivalence or anomaly. Or it could consist of implementing some of the more famous OULIPO techniques. The repertoire of them is huge (see appendix XYZ).

Poetry rhyming scheme One of the biggest points for future work is to introduce a rhyming scheme for the poetry results. This would involve some more
§ ?? NLP during the creation of the index. It would make the poems much more readable. See more in chapter XYZ.

Random sentences Adding to the source of random sentences used in the top and bottom banner on the website should be an ongoing endeavour.

Custom API It would be great to develop a custom API for this the search tool. This would allow other people to use the search remotely without going through the interface and to use the results as they want. This would have been benefi-
§ 3 cial for the Digital Opera project and certainly for other researchers/developers
§ 3 like Adnrew Dennis.

²Flickr, Getty, Bing, MicrosoftTranslator and YouTube

WordNet vocabulary The vocabulary in WordNet is limited. According to its website (<https://wordnet.princeton.edu/>) it contains 117000 ‘synsets’³. This affects two of my algorithms. Because of the way the process works, the link between Wordnet and source texts, results may be limited.

check

WordNet Antonyms The antinomy algorithms relies on WordNets antonyms. A lot of words simply do not have an opposite and no fallback is currently defined. This means a lot of the time the antinomy function will not produce any results.

4.2.1

Stemming Stemming could increase the number of results found by the algorithms. (See chapter XYZ). A danger of increasing the output of the pataphysicalisation is always that results become more boring. If the query term and potential matches were compared based on their stemmed form

Queneau’s poems It would be nice to actually add Queneau’s poem texts into the coprus of Faustroll as little easter eggs.

Bitmap algorithms The image and video search currently rely on extrenal APIs and user tags to work. One option to approach this in a totally differnet way would be to write algorithms that analyse and pataphysicalise the bitmaps themselves. So this could mean we could have a reverse image search that finds images related the original bitmap in pataphysical way or other.

Index One idea for the pataphysicalisation process was to add ‘patadata’ to the index. This could include pronounciation tags for example to make an implementation of a rhyming scheme for the poetry easier. So each word in teh index dictionary would contain the following items.

```
(`tree': [`l_00': [24, 566, 4990], `s_14': [234, 5943]], IPA data)
```

add ipa data or whatever is best for the rhyming stuff

storing rhyming data in index or other additional things like ranking

³Synonyms—“words that denote the same concept and are interchangeable in many contexts”—are grouped into unordered sets called synsets.

Stopwords Using a different set of stopwords to see if that makes a difference. For example we could use a spanish set of stopwords on an english text. OR the other way around.

5.2 Creative NLP

Section ?? N-grams are a [NLP](#) technique introduced in chapter ???. The idea is that it allows for prediction of likely word pairs, meaning if the word ‘sunny’ often occurs just before the word ‘day’ in a given training text or corpus then the probability for this particular n-gram is higher than say for ‘sunny dog’. This can be increased to predict the probability of longer chains of words. One can immediately see the attraction of abusing this to generate pseudo sentences or even of creating a formula similar in nature but for example ranking obscure combinations of words higher than common ones. So for example instead of having a [Maximum Likelihood Estimation \(MLE\)](#) (see chapter XYZ and formula 6.12) we could have a ‘Maximum Obscurity Estimation’ defined as:

$$P(w_n | w_{n-N+1}^{n-1}) = \frac{C(w_{n-N+1}^{n-1} w_n)}{C(w_{n-N+1}^{n-1})} \quad (5.1)$$

work the maths out here for this example of MOE

Similarly, we could play with maximum entropy models as shown on page 112 (see chapter XYZ) together with [Parts-of-Speech \(POS\)](#) tagging. What if we rigged the probability such that instead of ‘in Quebec’ ranking high for a ‘location’ [POS](#) tag, it now ranks high as a ‘drug’?

Again there are endless possibilities of abusing these kinds of systems to create [AMC](#). This is also very reminiscent of [OULIPO](#) techniques. We could create a whole new language grammar based on pataphysical principles.

Another example of interesting uses of [NLP](#) for [AMC](#) is playing with homonyms and heteronyms. Homonyms are pronounced the same but mean something else (e.g. ‘write’ and ‘right’). Heteronyms are words that are spelled the same but have a different meaning (e.g. ‘close to the edge’ and ‘to close the door’). There are similar techniques in the [OULIPO](#). Homophones are often used to create puns (and remember—puns are syzygy’s of words), for example “past your eyes” and “pasteurize”.

You can tune a guitar, but you can’t tuna fish. Unless of course, you

play bass.

attributed to Douglas Adams

look into rhyming tags in nlp

NLP would also be useful for introducing a rhyming pattern into auto-generated poetry. BY doing POS tagging with pronunciation data, we could retrieve sentences that match the sound of the last word of the previous line, etc.

<https://wordnet.princeton.edu/wordnet/man/wngloss.7WN.html> for glossary

fix all chapter XYZ mentions

group these into better sub groups and make them proper sections rather than paragraphs

5.3 Theoretical

Focus group It might be interesting to look at opinions of various people (general public and experts) about the interpretation/evaluation framework. This could be done by asking them to provide their own definition of computer creativity and then to analyse and evaluate a product (such as [pata.physics.wtf](#)) according to their own criteria. Then follow this up by getting the same people to use my proposed framework to compare the results. This would include asking them about whether or not they thought that using the framework was beneficial to them or confusing.

Questionnaires I have shied away from doing a questionnaire study because of several reasons. One is that due to the creative and subjective nature of the artefact, opinions on it may vary wildly and I don't see how I could derive useful unbiased data from that. Yes, it depends what questions you ask. But even if I managed to get some half-decent data, what would that tell me? Half of the people like my site, the other half don't?

Eye-tracking To study the effects of using different styles of presenting the same results an eye-tracking experiment could be done. This would involve setting up participants with the necessary equipment and then introduce them to the website and monitor their eye movements as they navigate the site. This could also provide details about how long users spend on each results page, what kind of style of results they prefer, etc. Some may prefer image or video search over the text search while others may not be interested in that at all. Generally of course one has to take into account that this is a creative piece of

work and not everybody will like it. It has no clear immediate purpose and that may put users off.

Performance Benchmarks?

Part VI

HAPPILY EVER AFTER

OBSERVATIONS

6

Paying no attention to his fellow mites,
mérite pas que vous fassiez attention à moi,
and told him to look after a calf she had bought,
and whilst he was looking at it attentively.

Phedon the fact affirm'd,
comment peux,
ne faites aucune attention à mon air,
in fact.

For sure Ulysses in your look appears,
was nearly out of her mind,
I omitted none of the common forms attending a royal audience.

And the consequences attending thereupon,
impotent of mind,
shape at the moment of looking at the time.

6.1	Outroduction	104
6.2	Answers	105
6.3	Contributions	105
6.4	And Finally	106



summarise thesis, contributions etc. conclude by comparing against introduction

a wide range of subject areas such as computer science, psychology, linguistics, literature, art and poetry, languages and mathematics.

refer back to these in conclusion

6.1 Outroduction

The last XYZ chapters have explained in probably too much detail what **AMC** is and how to evaluate it. Given that this spans so many different disciplines the contextual background information necessary to understand the research was presented in a broad literature survey in chap XYZ. This also posed a problem for choosing the right methodology for the project. In the end a transdisciplinary approach was chosen as described in chap XYZ with a heavy component of iterative exploratory rapid-prototyping to develop an artefact to demonstrate what **AMC** is.

This artefact is presented on pata.physics.wtf. It is an artwork dedicated to **AMC**, pataphysics, **OULIPO** and programming culture.

A critique of computer creativity and its current evaluation formed the starting point for a new framework which was introduced in chap XYZ. The general conclusion of the thesis was made up of the critical analysis and further work chapters as well as this final concluding chapter right at the end.

The appendix contains various code snippets and peripheral pieces vaguely related or relevant for parts of this thesis. The code of the website is included on a CD CHECK attached to the back of the front cover. Of course the website is also available online at pata.physics.wtf.

check if i need to submit a CD?

6.2 Answers

In the introduction I asked several questions that I attempted to answer with my research. This section contains brief answers from 50.000 feet¹, meaning they provide a top-down view of the answer and pointers to where in the thesis readers can find more elaborations.

add chapter references

these are closed questions, not research questions

Can computers or algorithms be considered creative?

- § ?? In short: no. In chapters ?? and ?? I have gone into great detail of why I believe that this cannot happen any time soon (see argument of zombies). They can be ‘creative’ (adj/adv CHECK) but the source of the creativity is the programmer of the machine not the machine itself.

Can pataphysics facilitate creativity?

- Yes. Pataphysics provides many principles which can be turned into techniques and constraints which is well known to be able to support creativity (see chapter ??). This is also evident in the OULIPO and their use of constraints (see chapter ??).

Can a creative process be automated or emulated by a computer?

- Yes, in theory. It mainly depends how you define the creative process and § ?? that is fairly subjective. See more in chapter ?? and ??.

Can human and computer creativity be objectively measured?

- § ?? No. As discussed in chapter ?? since the perception of creativity is subjective it cannot be quantified in objective terms. By providing a framework that takes into account all possible contextually relevant contributors though we can approximate an objective evaluation.

Can information retrieval be creative?

- § 4 Yes. There are many ways this can be achieved too as mentioned in chapter 4.

Can search results be creative rather than relevant?

- Yes, although this is also subjective. What is creative to some might not be creative to everybody. The artefact also nicely showed the difference in § 4 perception of results simply based on design of the content (see chapter 4).

6.3 Contributions

mention to whom these could be useful

¹Inspired by Time Berners-Lee’s articles on the Web in 1998—/urlhttp://www.w3.org/DesignIssues/Architecture.html

write more

This doctoral project can be broken down into four main contributions.

- Three pataphysical search algorithms (clinamen, syzygy and antinomy).
- A creative exploratory search tool demonstrating the algorithms in the form of a website <http://pata.physics.wtf>.
- A set of subjective parameters for defining creativity.
- An objective framework for evaluating creativity.

In a more practical sense this project has spawned several publications, talks sec ?? and exhibitions (a full list is in preface ??). Further talks were given by Andrew Hugill at various conferences and events throughout the world where he mentioned my work. My publications were cited in other academic publications and my website was mentioned on Reddit². My job here is done.

6.4 And Finally

Pataphysics is the science...

²Although absolutely nobody seemed interested in it. No idea who posted it or how he found it.

INTERLUDE III

Part VII

POST

Allows to water, now twice
underfoot the moist
As he did once with the
And the sea coast of Tyre and Sidon,
within one of the lists of Mankind,
the last state of that man
and the horns of bulls, chuchote une collection
of telleles a farmons, there the
telleles a rose upon the Bush, and the last
And the sea coast of Tyre and Sidon,
within one of the lists of Mankind,
the last state of that man
and the horns of bulls, chuchote une collection
of telleles a farmons, there the
telleles a rose upon the Bush, and the last
All and steam to pass through but is
underfoot the moist
As he did once with the
And the sea coast of Tyre and Sidon,
within one of the lists of Mankind,
the last state of that man
and the horns of bulls, chuchote une collection
of telleles a farmons, there the
telleles a rose upon the Bush, and the last

REFERENCES

- Amaral, Jose Nelson et al. (2006). ‘About Computing Science Research Methodology’. In: (cit. on pp. 5, 6).
- Boek, Christian (2002). ***'Pataphysics: The Poetics of an Imaginary Science***. Evanston, Illinois: Northwestern University Press (cit. on p. 33).
- Burdick, Anne et al. (2012). **Digital Humanities**. Cambridge, Massachusetts: MIT Press (cit. on pp. 7–10, 13).
- Burnham, Douglas (2015). ‘Immanuel Kant: Aesthetics’. In: **Internet Encyclopedia of Philosophy** (cit. on p. 21).
- Candy, Linda (2006). **Practice Based Research:A Guide**. Tech. rep. (cit. on p. 17).
- Damerau, Fred J (1964). ‘A Technique for Computer Detection and Correction of Spelling Errors’. In: **Communications of the ACM** 7.3, pp. 171–176 (cit. on p. 34).
- Dennis, Andrew (2016). ‘Investigation of a patadata-based ontology for text based search and replacement’. University of London (cit. on p. 55).
- Edmonds, E. and L. Candy (2010). ‘Relating Theory, Practice and Evaluation in Practitioner Research’. In: **Leonardo** 43.5, pp. 470–476 (cit. on p. 17).
- Efron, Bradley and Ronald Thisted (1976). ‘Estimating the number of unseen species: How many words did Shakespeare know?’ In: **Biometrika** 63.3, pp. 435–447 (cit. on p. 74).
- Foucault, Michel (1966). ‘The Order of Things - Preface’. In: **The Order of Things**. France: Editions Gallimard. Chap. Preface, pp. xv–xxiv (cit. on p. 63).
- Heisenberg, Werner (1942). **Ordnung der Wirklichkeit**. Trans. by M.B. Rumscheidt and N. Lukens (cit. on p. 15).
- Hendler, Jim and Andrew Hugill (2011). ‘The Syzygy Surfer : Creative Technology for the World Wide Web’. In: **ACM WebSci 11** (cit. on pp. 21, 41).

- Holz, Hilary J et al. (2006). ‘Research Methods in Computing : What are they , and how should we teach them ?’ In: ***ITiCSE Innovation and technology in computer science education***, pp. 96–114 (cit. on pp. 4, 5).
- Hugill, Andrew (2012). ***'Pataphysics: A Useless Guide***. Cambridge, Massachusetts: MIT Press (cit. on p. 63).
- Hugill, Andrew and Hongji Yang (2013). ‘The creative turn: new challenges for computing’. In: ***International journal of Creative Computing*** 1.1, pp. 4–19 (cit. on p. 16).
- Jarry, Alfred (1996). ***Exploits and Opinions of Dr Faustroll, Pataphysician***. Cambridge, MA: Exact Change (cit. on pp. 29, 34).
- (2006). ***Collected Works II - Three Early Novels***. Ed. by Alastair Brotchie and Paul Edwards. London: Atlas Press (cit. on pp. 21, 63).
- Kurzweil, Ray (2013). ***How to Create a Mind***. London: Duckworth Overlook (cit. on p. 86).
- Levenshtein, Vladimir I (1966). ‘Binary codes capable of correcting deletions, insertions, and reversals ’. In: ***Soviet Physics Doklady*** 10.8, pp. 707–710 (cit. on p. 34).
- Mayer, Richard E (1999). ‘Fifty Years of Creativity Research’. In: ***Handbook of Creativity***. Ed. by Robert J Sternberg. New York: Cambridge University Press. Chap. 22, pp. 449–460 (cit. on p. 88).
- Miller, George A. (1995). ‘WordNet: a lexical database for English’. In: ***Communications of the ACM*** 38.11, pp. 39–41 (cit. on p. 36).
- Nicolescu, Basarab (2010). ‘Methodology of Transdisciplinarity - Levels of Reality, Logic of the Included’. In: ***Transdisciplinary journal of Engineering and Science*** 1.1, pp. 19–38 (cit. on pp. 14, 15).
- Raczinski, Fania (2016). ***Emails***. personal communication. feedback for his bachelor project (cit. on p. 56).
- Ramesh, V., Robert L. Glass and Iris Vessey (2004). ‘Research in computer science: an empirical study’. In: ***journalttitle of Systems and Software*** 70.1-2, pp. 165–176 (cit. on p. 5).
- Schulman, Ari (2009). ‘Why Minds Are Not Like Computers’. In: ***The New Atlantis*** 23, pp. 46–68 (cit. on p. 87).
- Thomas, Sue et al. (2007). ‘Transliteracy: Crossing divides’. In: ***First Monday*** 12.12 (cit. on p. 7).
- Walker, Richard (2012). ***The Human Brain Project***. Tech. rep. HBP-PS Consortium (cit. on pp. 86, 87).
- Wickson, F., A.L. Carew and A.W. Russell (2006). ‘Transdisciplinary research: characteristics, quandaries and quality’. In: ***Futures*** 38.9, pp. 1046–1059 (cit. on p. 63).
- Yang, Hongji (2013). ‘Editorial’. In: ***International journal of Creative Computing*** 1.1, pp. 1–3 (cit. on p. 15).

GLOSSARY

GET

An [Hypertext Transfer Protocol \(HTTP\)](#) method. Allows a client (browser) to request data from a specified resource on a given web server.. [39](#)

POST

An [HTTP](#) method. Allows a client (browser) to submit data to be processed to a specified resource on a given web server.. [39](#)

KTHXBYE