Department of Computer Science
Series of Publications A
Report A-2016-4

Methods and Models in Linguistic and Musical Computational Creativity

Jukka M. Toivanen

To be presented, with the permission of the Faculty of Science of the University of Helsinki, for public examination in Auditorium XIV, University Main Building, Helsinki on June 3rd, 2016 at 12 o'clock noon.

University of Helsinki
Finland

Supervisor
Hannu Toivonen, University of Helsinki, Finland

Pre-examiners
Pablo Gervás, Universidad Complutense de Madrid, Spain
Tapio Takala, Aalto University, Finland

Opponent
Hisar Maruli Manurung, Google Japan Inc., Japan

Custos
Hannu Toivonen, University of Helsinki, Finland

Contact information

Department of Computer Science
P.O. Box 68 (Gustaf Hällströmin katu 2b)
FI-00014 University of Helsinki
Finland

Email address: info@cs.helsinki.fi
URL: http://www.cs.helsinki.fi/
Telephone: +358 2941 911, telefax: +358 9 876 4314

ISSN 1238-8645
ISBN 978-951-51-2203-2 (paperback)
ISBN 978-951-51-2204-9 (PDF)
Computing Reviews (1998) Classification: I.2, I.2.7
Helsinki 2016
Unigrafia

Methods and Models in Linguistic and Musical Computational Creativity

Jukka M. Toivanen

Department of Computer Science
P.O. Box 68, FI-00014 University of Helsinki, Finland
jukka.toivanen@cs.helsinki.fi
http://www.cs.helsinki.fi/u/jmtoivan/

PhD Thesis, Series of Publications A, Report A-2016-4
Helsinki, May 2016, 56+8+79 pages
ISSN 1238-8645
ISBN 978-951-51-2203-2 (paperback)
ISBN 978-951-51-2204-9 (PDF)

Abstract

Computational creativity is an area of artificial intelligence that develops algorithms and simulations of creative phenomena, as well as tools for performing creative tasks. In this thesis, we present various computational methods and models of linguistic and musical creativity. The emphasis is on developing methods that are maximally unsupervised, i.e. methods that require a minimal amount of hand-crafted linguistic, world, or domain knowledge.

This thesis consists of an introductory part and five original research articles. The introductory part outlines computational creativity as a research field and discusses some of the philosophical foundations underlying the current work. The research articles present specific methods and algorithms for automatic composition of poetry and songs. The first article proposes a corpus-based poetry generation method that relies on statistical language modelling and morphological analysis and synthesis. In the second article, we expand that basic model with constraint programming techniques to handle more aspects of the poetic structure and style. The third article presents a method for mining document-specific word associations and proposes using them in poetry generation to produce poems based, for instance, on a specific news story. The fourth article presents a song composition system that utilises constraint programming to produce songs with matching lyrics and music in a transformational way, i.e. is able to modify its own search space and preferences. Transformationality of the system is achieved with a metalevel component that can modify the system's internal constraints leading into new conceptual spaces. Finally, the fifth article discusses possibilities of combining personal biosignal measurements, especially electroencephalography, with techniques of computational creativity and presents an art installation called Brain Poetry based on these ideas.

The current work relies heavily on the use of unsupervised data mining techniques to automatically build models of specific creative domains such as poetry. The proposed methods and models are flexible and they are to a large extent independent of language and style. Thus, they provide a general framework for computational or synthetic creativity in linguistic and musical domains that can be easily expanded in many ways. Applications of this work include pedagogical tools, computer games, and artistic results.

Computing Reviews (1998) Categories and Subject Descriptors:
I.2 Artificial Intelligence
I.2.7 Natural Language Processing

General Terms:
Algorithms, Experimentation, Languages

Additional Key Words and Phrases:
Computational creativity, Computational methods, Language, Poetry, Music, Songs

Acknowledgements

First and foremost, I wish to thank Professor Hannu Toivonen for his excellent supervision and continuous guidance through all phases of academic research. I greatly appreciate your enthusiastic encouragement, constructive critique, and steady support that made this thesis possible.

I would like to offer my special thanks to all current and former members of our Discovery research group at the Department of Computer Science, University of Helsinki, especially to Oskar Gross, Dr Alessandro Valitutti, Anna Kantosalo, Olli Alm, Atte Hinkka, Dr Ping Xiao, Professor Antoine Doucet, Petri Vaittinen, Petri Myllys, Simo Linkola, Mika Hämäläinen, and Khalid Alnajjar. I would also like to express my very great appreciation to all my other co-authors and colleagues. Special thanks go to Dr Matti Järvisalo, Professor Dan Ventura, Professor Frank van der Velde, Professor Martti Vainio, Kasperi Mäki-Reinikka, Aleksander Alafuzoff, Jari Torniainen, Henri Kotkanen, Heikki Paakkanen, and Dr Lassi Liikkanen.

Many thanks to the pre-examiners of this thesis, Associate Professor Pablo Gérvas and Professor Tapio Takala, as well as Marina Kurtén and Paavo Toivanen for their comments that helped to improve the thesis.

I appreciate the financial support from the Doctoral Program in Computer Science (DoCS), the Helsinki Graduate School in Computer Science and Engineering (HECSE), the Finnish Doctoral Program in Computational Sciences (FICS), the Future and Emerging Technologies (FET) programmes within the Seventh Framework Programme for Research of the European Commission, under FET grant numbers 611733 (ConCreTe) and 600653 (PROSECCO), and the Algorithmic Data Analysis (Algodan) Centre of Excellence of the Academy of Finland. I am deeply grateful that I got a chance to do research on interesting topics and travel widely to academic meetings.

Finally, special thanks to my friends and family. My mother Pirkko and father Reijo have always supported me in my studies and have provided endless encouragement and advice. My brothers Arto and Paavo, and sisters Aura and Annasara have shaped me and my interests remarkably. For instance, many thought-provoking concepts, such as artificial intelligence, were introduced to me by my elder siblings when I was a child. In particular, I wish to express my thanks to my wife Aurora for invaluable support, patience, and friendship. It is really wondrous that we can share so many things in life that I consider important and interesting.

In Helsinki, March 24th, 2016
Jukka M. Toivanen

Contents

List of Reprinted Publications

This thesis consists of an introductory part and the following publications that are referred to as Papers I-V in the text. The publications are reprinted at the end of the thesis.

Linguistic Creativity

Paper I: Jukka M. Toivanen, Hannu Toivonen, Alessandro Valitutti, Oskar Gross, "Corpus-Based Generation of Content and Form in Poetry," In *Proceedings of the Third International Conference on Computational Creativity*, pp. 175-179, Dublin, Ireland, 2012.

> *Author's Contribution:* The author implemented the poetry generation system described in the paper. Finnish word association networks were produced in collaboration with Oskar Gross. The author performed the experiments and analyzed their results. He also wrote a major part of the paper.

Paper II: Jukka M. Toivanen, Matti Järvisalo, Hannu Toivonen, "Harnessing Constraint-Programming for Poetry Composition," In *Proceedings of the Fourth International Conference on Computational Creativity*, pp. 160-167, Sydney, Australia, 2013.

> *Author's Contribution:* The author implemented the described poetry generation system in close collaboration with Matti Järvisalo. The author also wrote a major part of the paper.

Paper III: Jukka M. Toivanen, Oskar Gross, Hannu Toivonen, "“The Officer Is Taller Than You, Who Race Yourself!” Using Document Specific Word Associations in Poetry Generation," In *Proceedings of the Fifth International Conference on Computational Creativity*, pp. 355-362, Ljubljana, Slovenia, 2014.

Author's Contribution: The author developed poetry generation methods and Oskar Gross implemented the methods for extracting document-specific word associations. The author wrote a major part of the paper. Paper III is also included in the PhD thesis of Oskar Gross.

Musico-Linguistic Creativity

Paper IV: Jukka M. Toivanen, Matti Järvisalo, Olli Alm, Dan Ventura, Martti Vainio, Hannu Toivonen, "Transformational Creation of Novel Songs," Submitted for publication.

Author's Contribution: The author implemented a song generation system in collaboration with Matti Järvisalo and Olli Alm, and wrote a major part of the paper.

Computational Creativity and Biosignals

Paper V: Jukka M. Toivanen, Frank van der Velde, Aleksander Alafuzoff, Jari Torniainen, Kasperi Mäki-Reinikka, Henri Kotkanen, "Using Biosignals in Computational Creativity – Brain Poetry Installation," Submitted for publication.

Author's Contribution: The Brain Poetry installation was created by the Brains on Art science-art collective consisting of the present author, Kasperi Mäki-Reinikka, Aleksander Alafuzoff, Jari Torniainen, and Henri Kotkanen. The author designed and implemented poetry generation methods used in this work and wrote the first draft of the paper.

Chapter 1

Introduction

1.1 Topic of the Thesis

This thesis deals with computational creativity in the domains of language and music. We adopt a pragmatic viewpoint that sees creativity as the capability to produce work which is both novel and appropriate, i.e. useful and adapts to appropriate domain constraints [66, 83, 84].

The motivation for this work is two-fold: firstly, creativity and generation of poetic texts and songs is an interesting test-bed for methods developed in artificial intelligence, data mining, and natural language processing. Secondly, increased flexibility in language and music generation methods inspires applications in fields that require interaction and adaptation, like dialogue systems and games. In this chapter we introduce the general context of this work.

1.2 Scientific Study of Creativity

Coming up with original insights in the areas of language, science, art, and mathematics, for instance, is not uncommon with human individuals. These various activities that involve the emergence of novel ideas, concepts, and artefacts may not have many shared characteristics but they all can be seen as realisations of creativity. The phenomenon of creativity permeates the whole life as individuals, organisations, and societies need to adapt to changing tasks and environments [49].

However, giving a clear definition for creativity is difficult and that has led some people to argue that creativity is an essentially contested concept [13]. According to these views, creativity is a concept which involves endless disputes about its proper uses on the part of its users. This vagueness has led creativity to be a relatively neglected topic in science despite its importance [84].

Traditionally, scientific study of creativity takes place within psychology, and the number of psychological studies of creativity began to grow in the 1950s. However, this research soon faced many roadblocks [84]: the origins of the research on creativity were in a tradition of mysticism and spirituality, there are impressions of study of creativity lacking theoretical basis and verification through empirical research, theory and methodology of the early scientific work on creativity was apart from the maistream empirical psychology, there are problems with defining and measuring creativity (showing the phenomenon to be either elusive or trivial), many approaches see creativity as an extraordinary and improbable phenomenon (outlier) sometimes present in ordinary structures or processes, and approaches to the study of creativity are often unidisciplinary: viewing a part of creativity as the whole creativity results in a narrow vision [84].

These issues have led scientific study of creativity to the margins of psychological research. Nevertheless, some work has been carried out in many different schools of psychology, including psychodynamic, Gestaltist, and cognitive approaches [84]. Often, different schools of psychology have concentrated on different aspects of creativity. The approaches of cognitive psychology have tended to ignore personality and social structure whereas the approaches of social and personal psychology have tended to neglect mental representations and underlying cognitive processes [84]. Besides psychology, creativity has also been studied in philosophy, history, social sciences, and computer science with their own focuses, and often it is thought that understanding creativity requires a multidisciplinary approach. Research approaches within computer science can be roughly divided into two general categories: creativity support systems and computational creativity.

1.3 Computational Creativity

Computational creativity is generally considered as a subdiscipline of artificial intelligence (AI) having strong connections to many neighbouring fields including cognitive science, linguistics, and arts. Colton and Wiggins [9] have defined the field in the following way: *"Computational creativity is the philosophy, science and engineering of computational systems which by taking on particular responsibilities, exhibit behaviours that unbiased observers would deem to be creative."*

Thus, computational creativity has many roots as a scientific discipline. On one hand, the field has tried to shed light on creative phenomena by formalising different kinds of creative activities in one way or another. On the other hand, there has been a strong computer-scientific emphasis on developing more or less practical generative algorithms and systems in different domains [9]. The interest here lies both in generative methods and their results for their own value, and in using these methods to support human creativity. In contrast to creativity support tools widely studied in the field of human computer interaction (HCI) [see, e.g., 77], computationally creative systems should perform creative behaviours themselves and take responsibility for the produced artefacts, not only serve as tools that facilitate creative behaviours [9].

Because of the inherent challenges of the field, many researchers have adopted a practical viewpoint that considers production of something novel and useful or high in quality to be creative [84]. In many cases, computational creativity researchers have been more interested in the engineering aspects of the work than studying invariant characteristics of different creative phenomena. Thus, some of the research can be seen as a sort of artisan work. Generative systems have been developed in many different domains, including mathematics [6, 7, 10], science [39, 44], visual arts [54], music [16, 32], poetry [12, 51], narrative [24], and jokes [93].

On the other hand, some researchers have put more emphasis on studying general characteristics of creativity with computational models and simulations. This branch of the work has especially close ties with cognitive science, and often the developed systems have the goal of producing creative thought computationally in a manner that simulates what people do [2]. For instance, Langley et al. [43] have developed computational models that automatically rediscover scientific laws by making use of certain heuristics, i.e. problem solving guidelines. The basic idea of these models is to search for patterns in a dataset in order to find hidden relationships between variables in the data, and for instance, the BACON system could rediscover Kepler's third law of planeraty motion based on simple inference rules and some data that was available to Kepler [43]. Later these approaches have been extended with the ability to transform data sets and reason with qualitative data and scientific concepts [49]. Also in artistic domains many generative systems implement cognitive science theories including, for instance, the Divago concept generator [52, 71] and the IDyOM model of music listening [69].

Margaret Boden [3, 4] is generally credited for starting the discussion on the nature of creativity in computational settings. Boden has proposed three general classes of creativity: combinatorial, exploratory, and transformational creativity, which have been useful in the formalisation attempts.

Combinatorial creativity results from unusual combinations of ordinary elements and ideas, and for instance, many types of analogies, metaphors, and artistic forms can be seen as belonging to this class. Exploratory creativity is exploration in some well-defined conceptual space of ideas or artefacts whereas transformational creativity also requires modifications of the conceptual space where the search is performed [4]. Wiggins has later formalised these types of creativity as search [94]. In this formalism transformational creativity is seen as search on the metalevel of the conceptual space. An interesting question here is how the conceptual spaces are or need to be defined and how representations of artefacts and conceptual spaces affect what kind of manipulation and search can be performed.

Also, distinctions between psychological or personal creativity (P-Creativity) and historical creativity (H-Creativity) and between improbabilistic and impossibilist creativity have been used. P-creativity involves production of artefacts or ideas that are novel for the agent itself but not novel for the whole society at large whereas H-creativity involves production of ideas or artefacts that are novel for the society as a whole and also historically without precedents [2, 4]. Improbabilistic creativity is tightly linked to combinatorial creativity, meaning creation of novel combinations of familiar ideas. Higher levels of creativity involve more sophisticated approaches to mapping, exploring, and transforming conceptual spaces. This type of creativity is impossibilist in the sense that ideas may be generated which could not have been generated in preceding conceptual spaces. In essence, generation of such ideas is only made possible by some transformation of the conceptual space [2]. The concepts developed by Boden have been useful in developing the computational creativity theory. However, Boden's theory is actually rather vague what it comes to actual implementations.

1.4 Can a Computer Be Creative?

At the current stage of research, computational creativity raises more questions than it answers. The underlying question in the field is of course whether a computer can, even in principle, be creative. Besides that, what properties should a system have in order to be perceived as creative? Should the creativity of a given system be assessed based on its output or based on the process that produced that output? Should a computational system include motives behind the generative processes and how should these motives be implemented? Can a computer exhibit creativity independently of its creator or programmer? How is an evaluation function for different genres of art formed? Can creative behaviour be reproduced using high-level symbols or does it require sub-symbolic processing?

Some psychological studies have addressed the question of what human creativity actually is. In many cases, creativity is not something magical or even seemingly breaking its internal rules. For instance, Goldenberg et al. [25] have identified some structured procedures that can produce results judged to be creative. However, here the evaluation was performed for the created results, and the creativity of the procedure itself was not evaluated. Many of the processes identified here could be seen as instances of combinatorial creativity.

According to the study by Goldenberg et al. [25], systematic use of procedures and templates identified by the researchers leads to ideas and results which are comparable to the best creative ideas in advertising. In their study, they conclude that by utilising these systematic methods even a computer can produce seemingly creative results which they call "creative sparks". However from the perspective of computational creativity, it is not very interesting to take some creative results made by humans and to develop a mechanized procedure that produces seemingly similar results. Some researchers in computational creativity think that also the process, not only the artefacts, should be judged as creative, because in many creative domains the process of creating an artefact is often an important factor in the evaluation of that artefact [8]. Also, it is claimed that computational creativity should be understood as different from human creativity in many ways, although not necessarily fundamentally different. Analogously, aeroplanes replicate the process of flying in a different way than many animals such as birds and bats do, although fundamental principles of physics are the same [22].

Many arguments have been made in favor of the claims that computers can not exhibit any true intelligence or creativity [59]. These arguments include, for instance, that (1) computers can do only what they are programmed to do, (2) computers cannot learn, and (3) computers cannot exceed the internal rules defined in their software. However, recent achievements in machine learning, for instance, have given contrary evidence [33]. Also, it has been argued that the computer should have real motivations behind the generative processes, and in many cases computationally creative systems clearly fall short in this respect. However, Boden argues that while motivation and emotion may have a central role in many creative processes, especially in humans, they are not the prime element of (computational) creativity [2]. The problem is not that motivation and emotion would be outside the reach of computational systems – on the contrary, there is a lot of research in this area – but in computational creativity the central theme should be how (instead of why) novel ideas and artefacts are produced in the human mind and in artificial computational systems [2].

1.5 Computational Creativity and Occam's Razor

Occam's razor, also known as the principle of parsimony, is often considered to be one of the fundamental doctrines of science. It states in its original form that "Nunquam ponenda est pluralitas sine necessitate" (i.e., plurality must never be posited without necessity) [90]. Occam's razor was formulated as criticism of scholastic philosophy, as its theories grew ever more detailed without making more accurate predictions. Likewise, in arts, one can claim that the desired artistic effect be achieved in an economical way. An overly complicated artistic process should not be seen as better than a simpler one if it does not produce better results. By process we mean here a procedural implementation of a more abstract model of a given artistic domain, such as an algorithm that produces poetry

However, it seems that people tend to see more complex systems as more creative even if the results of the system are only as good as results of some simpler system. Briefly, the less one understands an artist's mind the more creative the work by that artist seems [59]. Thus, when building artificially creative systems it seems that more complicated systems are deemed to be creative more easily. This perception is very central to computational creativity in general. If creativity cannot be explained it cannot be turned into a mechanical process. Once one understands the process it does not seem so creative anymore [59].

However, we argue that in this case perception of creativity is not the same thing as creativity. Based on Occam's razor, generative systems that are based on flexible and general principles of data mining and machine learning should be valued more highly than complex expert systems usually tuned to work in some narrow domain, assuming that the artefacts produced by them are equivalent in quality.

1.6 Research Questions

In this thesis, we seek to answer the question of how maximally unsupervised methods that require a minimal amount of hand-crafted resources can be used in computational creativity, for instance, to produce poems and songs. This research question divides into the following more detailed research questions:

I *How can unsupervised corpus-based methods be used in flexible language generation, especially for poetry? By unsupervised methods we mean here methods that require a minimal amount of hand-crafted linguistic, world, and domain knowledge.*

II *What kind of declarative methods can be utilised to control properties of both content and form in automated poetry and song composition and how can language and music be unified into a coherent whole in generative processes?*

III *How can transformational creativity of a generative system be modelled in an explicit way?*

IV *How can the methods developed contribute to arts and how can the results be given personal meaningfulness?*

1.7 Contributions of this Thesis

The scientific contributions of this thesis are in the five original publications I–V.

Research Paper I: Corpus-Based Generation of Content and Form in Poetry

In this paper, we propose a corpus-based poetry generation method that uses statistical language analysis and morphological analysis and synthesis to produce poetry in a maximally unsupervised way (Research question I).

Research Paper II: Harnessing Constraint-Programming for Poetry Composition

This paper presents a constraint-based approach to control finer characteristics of poetic content and form with respect to the Paper I (Research question II).

Research Paper III: "The Officer Is Taller Than You, Who Race Yourself!" Using Document Specific Word Associations in Poetry Generation

This paper presents a method for finding document-specific content in a given document and a method to produce poetry loosely based on this content (Research question I).

Research Paper IV: Transformational Composition of Novel Songs

> In this paper, we present a method for combining language and music into a unified whole with a constraint-based approach that enables the system to transform its own workings (Research questions II and III).

Art Paper V: Brain Poetry Installation – Using Personal Biosignals in Computational Creativity

> In this paper, we propose a way for providing more context around the generative methods by linking them to measurements of personal biosignals (Research question IV). The Brain Poetry installation which implements these ideas, also serves as a large-scale example of embedding computational creativity in the societal settings and showcases the cultural impact of the methods developed.

1.8 Structure of the Thesis

This thesis consists of five original research papers and an introductory part. The introductory part provides background and context for the research work presented in the papers and it concludes by summing up the scientific contributions of this work and discussing them briefly.

Chapter 2 gives an overview of linguistic and musical computational creativity, discusses previous work in these areas, and describes the context of the present research. In this chapter, we also present the underlying principles and design motives of the methods developed.

Chapter 3 presents the computational methods in automated poetry and song composition.

Chapter 4 describes results produced by the methods and their societal impact.

Chapter 5 summarizes contributions of the thesis and answers the research questions.

Finally, Chapter 6 discusses the work and provides an overall conlusion.

Chapter 2

Linguistic and Musical Computational Creativity

Use of natural language is a creative task on its own, and different methods to produce text in various forms have received a fair amount of attention in the past [9]. Computational linguistic creativity encompasses diverse areas of text generation, such as generation of stories, poetry, aphorisms, and news stories. Likewise, in musical computational creativity, a large number of algorithms and methods have been presented that generate different aspects of music, including melody, harmony, and rhythm [75].

In this thesis, the main emphasis is on computational poetry that is seen as a special case of creative text generation, and on songs that combine both musical and linguistic creativity. This chapter describes the background of computational poetry and song composition and outlines some general design principles for flexible computational creativity systems. These design principles reflect the research questions presented in the previous chapter. In all our methods, we aim for flexibility, unsupervision, and a diverse range of results. The technical content of the methods proposed in this thesis is presented in Chapter 3.

2.1 Computational Poetry and Songs

Poetry is a form of literature that has tight unity between the aesthetic and rhythmic qualities of language and its semantic content [38]. Several different poetic devices, including rhythm, metre, rhyming, sound symbolism, ambiguity, and metaphor are often used to evoke emotions and mental images, and to achieve musical or incantatory effects. These poetic devices are very important in constructing the overall form of a poem and in highlighting meaning in content.

Composing poetry in an automated way is an intimidatingly complex task. Several levels of language, such as syntax, semantics, lexical choice, and phonetics, need to be taken into account. In most forms of poetry, the system should also aim to convey some message or coherent meaning via the produced text. In addition, the production of poetic texts is not a well-defined or precise task. Various forms of text can be seen as poetry, and various interpretations of the same text arise through an individual reader's subjective knowledge, experiences, and impressions [50].

However, despite the many difficulties, computational poetry generation has attracted considerable attention as a research topic [see e.g. 23]. It provides challenges for developing flexible language generation methods and serves as an ultimate AI problem. In Papers I-IV, we present novel methods and discuss various aspects of automated poetry writing.

Like poetry, automated composition of songs, i.e. of combinations of music and lyrics is a challenging task. Automatic composition of song lyrics can be seen as essentially identical to the problem of automated poetry writing but, in addition, a system should be able to compose music and to make it match with lyrics [88].

Given that components for music and poetry generation exist, it is tempting to consider a sequential approach to song writing. Starting with lyrics the system can generate matching music, or vice versa. Such a system is, however, not optimal as it needs first to analyse, for instance, the lyrics in order to produce matching music, and this analysis can be error-prone [88]. In Paper IV, we present an approach that combines production of both the music melody and lyrics in a single generative phase.

2.2 Previous Work on Automated Poetry Composition

Various approaches have been developed to produce poetry automatically. According to Gervás [21] these approaches can be roughly divided into four different groups: template-based poetry generation systems, generate and test systems, evolutionary systems, and case-based reasoning systems.

In template-based poetry generation, templates specifying the grammar and form of a poem are filled in with suitable words in such a way that the defined constraints for content and form are satisfied. Notable examples of template-based poetry generation systems include the Full-FACE poetry generator [12], and the Pemuisi system [74]. The Full-FACE poetry generation system [12] uses a template and corpus-based approach to generate poetry according to given constraints on rhyme, meter, stress, sentiment, word frequency, and word similarity. The system is also argued to invent its own aesthetics and framings of its work. The Pemuisi system, on the other hand, utilises templates, constraint satisfaction, and various linguistic resources to produce topical poetry in Indonesian. In many respects, this system is similar to the system presented in Paper II although published slightly later. The chart generation system by Tobing and Manurung [85] utilises a dependency parser to extract predicate argument structures from input articles and generates poetry based on this structure and additional requirements for poetic form. This system can be seen as a more sophisticated version of the template-based generation method. The poetry generation methods presented in Papers I, II, and III can be seen as belonging to this category but in our case the templates are constructed automatically and they provide instance-based grammar. Our methods are further described in Chapter 3.

In generate and test approaches, one component of the system produces random word sequences satisfying certain constraints. Then, another component of the system evaluates the results and selects the best ones to be output. For instance, the WASP system [19] uses several construction heuristics to generate metrical Spanish poetry out of input words and verse patterns. Other examples of generate and test-based poetry generators include, for instance, the work by Netzer et al. for generating haiku poetry [63] and Markov chain-based methods to produce creative text [42].

Evolutionary poetry generation methods utilise methods of evolutionary computation, such as genetic algorithms, to produce poetry. Examples of evolutionary poetry generation include the work by Levy [46] and the work by Manurung [50, 51]. The Poevolve system by Levy [46] uses a neural network trained with human users as a fitness function. The system randomly generates a population of poems and applies mutation, crossover, and direct-copy operators to evolve better poems. The system developed by Manurung et al. [50] uses a grammar-driven formulation to generate metrically constrained poetry out of a given topic. This approach performs stochastic hillclimbing search within an explicit state-space, moving from one solution to another. The explicit representation is based on a hand-crafted transition system.

In case-based reasoning approaches, existing fragments of text with already known semantics are retrieved based on user queries and then combined and modified in order to satisfy given constraints for poetic content and form. Examples of systems that use case-based reasoning to produce poetry include the COLIBRI [15] and ASPERA [20] systems. For instance, ASPERA composes poetry by retrieving text fragments from a database and applying various operations to these fragments in order to compose a poem that satisfies given constraints for the poetic form while keeping semantic content as intact as possible. Another example in this category is the system by Wong et al. [95] that generates haiku poems based on texts harvested from the internet. Also, our instance-based grammar extraction in poetry generation can be seen as belonging to this class.

2.3 Previous Work on Automated Song Composition

Algorithmic music composition is an active research field, and many different methods have been developed in this area. Roads [75] provides a comprehensive overview of the work in this field. Our intent is not to review all that work but, instead, concentrate on automated song composition, i.e. on systems and approaches that share both musical and linguistic aspects.

Related work here can be roughly divided into three different categories. One line of research has concentrated on composing matching music for given texts. For instance, Monteith et al. [61] have developed a system that extracts linguistic stress patterns from given lyrics and composes a melody with matching note durations. In addition, the system tries to fulfill certain aesthetic metrics in order to make the music and text match each other better. Monteith et al. [60] have also generated musical accompaniments for stories. This system tries to make the emotional content of stories and music fit together.

Another line of research has concentrated on composing matching lyrics for music. Oliveira et al. [67] have generated text based on rhythm by matching stressed beats in music with stressed syllables. Likewise, Ramakrishnan et al. [73] have generated Tamil lyrics for given melodies. This method is based on using conditional random fields to analyse an input melody in order to find out what kind of music notes are accompanied by a certain kind of syllables, and a probabilistic sentence generation module to compose lyrics with matching stress patterns.

Finally, few systems exist for composing music and lyrics together. In their earlier work, the present author et al. [88] have generated simple songs with a sequential approach. This system writes lyrics first, and then composes matching music. Since the system performs both tasks, the music composition module can directly utilise information provided by the lyrics-writing module. This additional information on the intended sentiment etc. makes the match between music and lyrics tighter than in a purely sequential system.

Scirea et al. [78] have also automatically generated both music and lyrics. Their system produces songs based on text documents, composing lyrics by extracting important words from a given text and filling these words into simple templates. A melody is then composed by using a Markov chain evolution algorithm.

In Paper IV, we present a system that composes melody and lyrics in a single generative phase by using declarative programming. This system also modifies its own goals and sense of aesthetics. Methods utilised by the system are further described in Chapter 3.

2.4 Design Principles for Computationally Creative Systems

2.4.1 Supervised versus Unsupervised Methods

In many cases a deep but narrow body of knowledge is enough to solve a problem in a specific domain. This insight has led to construction of numerous *expert systems* that show intelligent behaviour in specific domains, including production planning [28], psychiatric treatment [62], resource utilisation [55], probabilistic fault diagnosis [45], and agriculture planning [72] among many others.

On the other hand, in more general AI settings, it has proven very hard to construct complex computational systems by manually specifying their internal workings. This has led to a view that some form of machine learning is necessary in order to achieve behaviours that could be deemed intelligent or creative [76]. One basic question here is whether it is, even in principle, feasible to manually include all the necessary rules, constraints and their interactions in the system, or does one need automated ways to handle all the complexity of the task.

For instance, in poetry generation, defining all necessary knowledge of world and language that is required for writing syntactically and semantically sound poems about arbitrary topics and in rich language does not seem achievable by manual inclusion of knowledge, at least in cases where the topics and forms of poetry are not very restricted. However, different poetry generation systems may have different goals. In some cases, the aim might be to produce high-quality poems about a very specific topic. In this case, fine-tuning a system for this goal may restrict its ability to produce poetry on other topics and in other styles. A poetry generation system containing a deep and narrow body of knowledge on a specific type of poetry can produce surprisingly good results in that area, but different approaches are most probably needed to expand that skill in order to produce more versatile results. Our aim has been to develop as general methods as possible and to avoid fine-tuning and manual inclusion of knowledge.

This has led us to adopt an approach that is heavily based on unsupervised data mining methods. We have emphasised the utility of a *maximally unsupervised* approach: we want to learn as much of the necessary knowledge directly from corpora as possible, instead of using manually annotated data sets and hand-coded rules. This approach does not necessarily produce as good results in the first place as systems that are carefully tuned to produce high-quality instances in a specific narrow style and on a limited number of different topics by manual specification of rules. However, we see the possibilities of data mining- and machine learning-based systems as very promising due to their flexibility. Besides, these approaches can be seen as more independent of the programmer. Increasingly large data sets, such as Google n-grams [57], and more powerful computational resources also provide growing support for this approach. For instance, deriving robust estimates of natural language statistics has only recently become widely available to researchers [47].

2.4.2 Stochastic versus Deterministic Methods

A purely stochastic system is such that its states are randomly determined according to some probability distribution. Thus, the behaviour of a stochastic system can be analysed statistically but not predicted precisely and it can be classified as non-deterministic. It has been argued that creativity itself would essentially be a stochastic process [82]. In science, for instance, new findings and theories are often results of a constrained stochastic behaviour. Because of the prevalence of such serendipitous elements in many creative processes creativity might be rooted in some chance element that is not present in a well-defined computer program.

However, computers can be made to simulate random behaviour, and many commonly used methods in machine learning and artificial intelligence, like simulated annealing, stochastic neural networks, and genetic algorithms, are stochastic in nature [1]. Also, many current methods and systems in computational creativity share this characteristic at least partially.

According to our view, such stochastic elements may have an important role in many types of computationally creative systems, especially in combinatorial ones. Even in cases in which the search space contains only valid artefacts, some heuristics are most likely necessary to restrict the search space because of combinatorial explosion. In the case of poetry, such heuristics may include, for instance, estimates of syntactic well-formedness and statistical information on semantics.

2.4.3 Model of Analysis and Synthesis

We propose our data mining-based approach to computational creativity as a *model of analysis and synthesis*. This approach can be summarized as follows: a model of some creative domain, such as poetry, is learnt by using unsupervised data mining methods, and then this model is used to compose new instances in that domain. The model of analysis and synthesis can be seen as an instance of the idea usually known as Pasteur's dictum: "In the fields of observation chance favors only the prepared mind...". Gathering as much knowledge in an area as possible is the most important component in being creative in that domain. For instance, a good enough language model can be seen as the basic building block of linguistic creativity.

However, this approach to computational creativity is only one of the possible alternatives. We propose calling this approach synthetic creativity analogously to the suggestion by John Haugeland, father of the term GOFAI (Good Old-Fashioned Artificial Intelligence), who also proposed that AI should more properly be referred to as synthetic intelligence [29].

2.4.4 Semantics, Syntax, Morphology, and Poetic Devices

Building models of natural languages is very difficult for many reasons. First of all, languages are very complex systems [79]. All the syntactic and semantic quirks of natural languages can certainly not be described by a couple of hundred rules and tricks, perhaps not even by several hundred thousand such tricks. Merely describing how verbs are used in English has proved difficult [79]. In traditional approaches these regularities are described as rules which are then learned based on the limited information available. In this form, the task of language learning seems so complex that innate grammatical information is needed in order to accomplish it [79]. Besides syntactic and morphological complexities, the prevalence of linguistic ambiguity and extra-linguistic elements in communication complicate things further. Thus, describing a natural language with a rule-based approach has proved laborious if not problematic. In this thesis, we utilise data mining to model natural language semantics, and in order to define specific syntactic and morphological structures, we utilise existing instances of language use.

In approaches known to the author, the content of a poem comes basically from one of two different sources: a logic-based representation [50, 51] or a bag-of-words style approach. In Papers I and III, we outline an approach belonging to the latter class that is based on use of statistical word associations. The semantics of the poetry produced this way largely emerges from a bag-of-words-style approach that puts emphasis on novel associations between words and mental images that they raise in the mind of the reader. The meaning of a word emerges from the meanings of the words surrounding it. The main difficulty with logic-based representations is that they tend to be rigid. Building and using them is laborious and resource intensive. Thus, some AI researchers have argued that logical reasoning is better suited for confirming results of thinking than for thinking itself [59]. However, the question of how things are represented when we think about them is a matter of great importance and logic-based representations certainly work on many occasions. In the context of poetry and, more broadly, any language generation, the statistical methods that utilise networks of meanings seem more interesting to us due to their flexibility.

Thus, the current approach shares much of the philosophy behind statistical natural language processing. Also, according to some linguistic theories, natural language grammar and semantics are are probabilistic and variable rather than fixed and absolute [80]. This conception of grammar follows from the view that one's competence in a given language changes in accordance with one's experience and familiarity with that language [79]. This view of language grammar as a stochastic phenomenon provides the foundation for modern statistical natural language processing.

In poem composition, we solve many issues with the form and especially grammar by using a so-called grammar corpus. The idea is to utilise existing instances of language use from this corpus and to re-use that exact grammatical structure in generated poetry. This is done by substituting words in the existing piece of text by new words that are related to a given topic in suitable morphological forms.

In poetry a large number of different poetic devices can be used to make texts appear more poem-like. These include rhyming, alliteration, meter, and many other qualities of the poetic form. These constraints can be handled, for instance, by using a constraint programming approach outlined in Paper II. According to some studies, for instance, rhyming can highly enhance the appeal of poetry or aphorisms and even affect their perceived truthfulness [56].

2.4.5 Declarativity and Transformationality

In Papers II and V, respectively, declarative constraint programming is used to model tasks of automated poetry and song composition. Constraint programming enables declarative specification of conceptual spaces and provides a computationally efficient way to perform search in those spaces by using highly optimised off-the-shelf constraint solvers [30]. We argue that this declarative approach to modelling of creative processes provides many advantages such as computational efficiency, transparent computational tools for studying creative conceptual spaces and their elements (Paper II for poetry, Paper IV for songs), and a way to implement explicit transformationality in a creative system by self-modification of constraints (Paper IV).

For instance, many forms of poetry are governed by strict rules specifying stress and syllable structures, rhyming patterns and selection of words with certain qualities. By modelling such interacting rules and constraints with constraint programming, complex generative settings can be handled efficiently. In addition, declarativity enables expressing the tasks of poem and song construction at the conceptual level in an intuitively appealing way.

The declarative approach also enables explicit modelling of transformational creativity in the sense of Wiggins [94]. By self-modification of constraints shaping the system's conceptual space, the system can transform its search space and generate artefacts that were impossible to generate in the preceding conceptual spaces. The capability for a system to adjust its own rules can be achieved with a metalevel component that carries out such changes.

The declarative approach to computational creativity satisfies our goal for flexibility, and the constraint-based systems can be easily expanded. Methods of constraint-programming for automated poetry writing and transformational song composition are further discussed in Chapter 3.

Chapter 3

Methods for Automated Poetry and Song Composition

This chapter presents the methods for automated poetry and song composition developed in this work. In poetry generation, our goal is to develop maximally unsupervised methods that require a minimal amount of handcrafted resources to automatically produce poetry. To achieve this goal, we have employed text-mining, constraint programming, and morphological analysis and synthesis methodologies in a corpus-based approach. The basic idea has been to use two different corpora, on one hand, to provide semantic content in the form of word associations, and on the other hand, to provide specific grammatical and poetic structures. In automated song composition, a constraint-based approach is used to produce songs that have relatively tight unity between their textual and musical features and to enable transformational creativity by self-modification of the system's rules.

First, we present our general corpus-based poetry generation method that composes poetry automatically by using word associations and instance-based grammars constructed with morphological analysis and synthesis. Then, we describe in some detail our methods for extracting general and document-specific word associations that provide semantic content for poetry in the proposed poetry generation algorithm. Next, we propose how the basic framework for poetry generation can be extended with constraint-satisfaction techniques to control finer elements of poetic content and form. Finally, we present methods for composing songs with relatively high textual and musical unity in a transformational way. Examples produced by the proposed methods can be found in Appendix B.

3.1 Corpus-Based Poetry Generation

Our corpus-based poetry generation method uses word association networks to control content and semantic coherence of the produced poetry. These word-association networks are constructed automatically and they can contain either common-sense world knowledge (Papers I and II) or document-specific content (Paper III). The word associations provide the necessary world knowledge and a distributional semantics-based model [37] of natural language semantics. The idea is that by using a semantically connected set of words when generating a poem, the internal coherence of the poem can be roughly controlled.

Many issues with the form, including the natural language grammar, we solve by using existing instances of actual language use extracted from a *grammar corpus*. The grammar corpus can contain poetry, for instance, but other genres of text can be used as well. Now, the basic form of the corpus-based poetry generation procedure (see also Figure 3.1) can be outlined as follows, with semantic content coming from background word associations (Paper I) or document-specific word associations (Paper III).

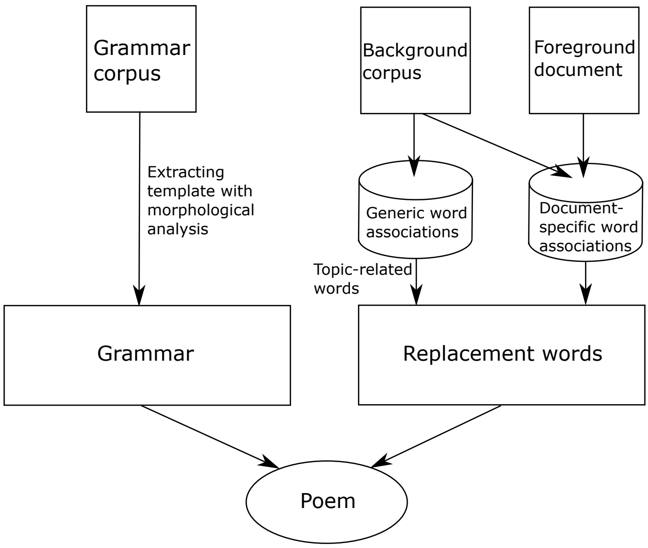
1. Obtain content words to be used in the poem. These words are either generically associated with a given topic (Papers I and II) or contain document-specific information (Paper III).

2. Obtain grammar for the poem. At this step, a piece of text of the desired length is selected from the grammar corpus. This piece can be one stanza of poetry, for instance. Words in this text are analysed morphologically for their part of speech, singular/plural, case, verb tense, clitics etc.

3. Produce a new poem out of the content words and grammar by word substitution. The substitute words are transformed into similar morphological forms with the original words.

Substituting words with other words in the same form requires varying degrees of syntactic and morphological analysis and adaptation depending on the language in question. Especially for morphologically rich languages such as Finnish, the morphological analysis is essential and non-trivial. In these languages much of the language's syntactic and semantic information is carried by morphemes joined to the root words. This word substitution method has been developed to work in Finnish, English, and German. In Finnish and German, the morphological analysis and synthesis are carried

Figure 3.1: An overview of basic corpus-based poetry generation procedure.



out using Helsinki Finite State Tools (HFST) [48]. The English version uses Morpha and Morphg morphological tools [58] and Stanford POS tagger [91].

The presented procedure serves as our basic framework for generating poetry in a maximally unsupervised way. The main advantage of this method is its flexibility: a wide range of poems in different forms and styles can be produced. The main limitation of the method is in handling exact semantics of the resulting poetry. Interconnectedness of content words is not enough to guarantee meaningful internal semantics. Next we discuss the construction of word association networks in some more detail.

3.2 Extracting Semantic Structure from Masses of Text

In the basic scenario, the semantic content of generated poetry is controlled by a simple network that contains information on common-sense word associations. This network that we call *background graph* is automatically constructed from a *background corpus*, for instance Finnish or English Wikipedia. The background graph is constructed from a given background corpus by using the log-likelihood ratio test (LLR) that is based on a multinomial model of word co-occurrences [14]. However, other techniques, such as latent semantic analysis (LSA) and latent Dirichlet allocation (LDA) could be used for mining the word associations as well.

Background associations are defined in the following way. For a given pair of words $\{x, y\}$, the multinomial model has four parameters $p_{11}, p_{12}, p_{21}, p_{22}$ that correspond to the probabilities of their co-occurrences like in the contingency table below.

	x	$\neg x$	Σ
y	p_{11}	p_{12}	$p(y; C)$
$\neg y$	p_{21}	p_{22}	$1 - p(y; C)$
Σ	$p(x; C)$	$1 - p(x; C)$	1

Probabilities $p(x; C)$ and $p(y; C)$ are the marginal probabilities of words x and y occurring in a sentence in corpus C, respectively.

The LLR test is based on the likelihoods of two such multinomial models, a null model and an alternative model. For both models, the parameters are estimated from relative frequencies in corpus C. For the null model independence of words x and y is assumed (i.e., by assigning $p_{11} = p(x; C)p(y; C)$ etc.), whereas for the alternative model maximum likelihood estimates are assigned from their observed frequencies (i.e., in general $p_{11} \neq p(x; C)p(y; C)$).

Then, the log-likelihood ratio test is defined as

$$LLR(x, y) = -2 \sum_{i=1}^{2} \sum_{j=1}^{2} k_{ij} \log(p_{ij}^{null}/p_{ij}), \tag{3.1}$$

where k_{ij} is the respective number of occurrences [14]. This test serves as a measure of how much the observed joint distribution of words x and y differs from their distribution under the null hypothesis of independence. In other words, the test tells us now how strongly two given words are associated. Edges in the background graph are constructed to connect any two words x and y that are associated with $LLR(x, y) > t$ where t is an empirically chosen threshold.

3.3 Document-Specific Content

The LLR-based method for finding common-sense word associations can be extended to find word associations that are semantically relevant for a specific document, as described in Paper III. The idea is to neglect too generic word associations such as “Los” and “Angeles” that do not carry much document-specific information. To achieve this, a reference of commonness is needed for judging the amount of document-specific novelty. A large and generic background corpus is used as this reference, and then a given *foreground* document is contrasted to this background. Associations that are prevalent in both the foreground document and background corpus, are not novel. Instead we are interested in associations that are uncommonly strong in the foreground document with respect to the background corpus. This method for extracting document-specific word associations has also been used as a central component in a document summarization method [26].

In this case the LLR equation becomes

$$D_{LLR} = 2\sum_{i=1}^{2}\sum_{j=1}^{2} k_{ij}(\log(p_{ij}) - \log(q_{ij})),$$

where, k_{ij} denotes the respective number of word occurrences and p_{ij} and q_{ij} denote the estimated probabilities for background and foreground, respectively. Here, the respective foreground document is included in the background model in order to improve the statistical robustness of the method.

The more the co-occurrence distribution of a given word pair in the foreground deviates from the distribution in the background, the higher values the LLR test now gives. In the case that a word pair only co-exists in the document, its joint co-occurrence probability is estimated under the assumption that the words are mutually independent.

3.4 Finer Control of Poetic Content and Form with Constraint Satisfaction Methods

In Paper II, we present how additional characteristics of poetic form and content can be controlled declaratively by using constraint satisfaction methods. This approach is based on expressing the task of automated poem composition as interacting constraints describing what the desired poems are like. Then, computationally efficient, off-the-shelf constraint solvers can be used to find such poems based on certain input data, such as candidate words for different places in the poem. Constraints can also be used to define what are desirable properties in solutions, and the resulting artefacts can be optimised based on these properties. Thus, constraint programming enables intuitively appealing, transparent, and flexible description of conceptual search spaces, description of aesthetics in those spaces, and efficient exploration with the constraint solver. Many aspects in poetic form and content, such as metrical patterns, rhyming structures, lexical choice, and even natural language syntax and semantics can be modelled that way.

Our general framework for constraint-based poem composition, described in Paper II, consists of two subcomponents: one specifies the conceptual space and an aesthetic with constraints and the other explores the specified space based on the aesthetic using a constraint solver. More specifically, the specifier component can generate a large number of mutually dependent choices of words for different positions in the poem, as well as dependencies between them. These candidates are called *data*. Then, the explorer component that consists of a constraint solver and a static constraint library is used to search for suitable poems based on that dynamically generated data by the specifier component. Data triggers varying constraints in the static constraint library based on its properties.

We implement the task of poetry composition as an answer set programming (ASP) problem instance expressed with *predicates* [18, 64, 81]. These predicates can be generated dynamically and they express, for instance, (1) the number of lines and the number of words (poem skeleton) on each line in the poem, (2) a list of words called candidates that can be used for each word position in the poem skeleton, (3) requirements on the desired poetic structure, including rhymes, meter, etc., and (4) other possible interdependencies between the words which can have both syntactic and semantic effects. In the present implementation the specifier component provides the poem skeleton with a fixed number of word places and suitable candidate words for all these places. For instance, the following poem skeleton marked with required parts-of-speech for every word position is automatically extracted from a grammar corpus by the specifier component:

N_SG VB, N_SG VB, N_SG VB!
PR_PS ADJ N_PL ADJ PRE PR_PS N_SG:
– C ADV, ADV ADV DT N_SG PR VB!
DT N_SG PRE DT N_PL PRE N_SG!

Here PR denotes pronoun, VB denotes verb, PR_PS denotes possessive pronoun, ADJ denotes adjective, N_SG denotes singular noun, N_PL denotes plural noun, C denotes conjunction, ADV denotes adverb, DT denotes determiner, and PRE denotes preposition. Then, the specifier component conveys semantically interconnected word candidates matching the required parts-of-speech as data to the explorer component.

Actual poem generation is expressed via *rule-based constraints* that dictate what a desired poem is like. These constraints constitute the static constraint library, and they can be re-used in any instances of poem generators just by generating data that activates them. Finally, state-of-the-art constraint solvers, such as Clasp [17], can be used in the explorer component to find poems that satisfy these constraints.

A two-line basic model for generating poetry can be seen in Figure 3.2. This model simply states that exactly one of the given candidate words must be selected for each word position in the poem skeleton. The predicates used in this model are described in Table 3.1. They specify how many lines there are in the poem, how many words there are on each line, and what kind of word candidates there are for every position in the poem.

Table 3.1: Predicates used in the basic ASP model for poem composition.

Predicate	**Interpretation**
`rows(X)`	the poem has `X` rows
`positions(X,Y)`	the poem contains `Y` words on row `X`
`candidate(W,I,J,S)`	the word `W`, containing `S` syllables, is a candidate for the `J`th word of the `I`th line
`word(W,I,J,S)`	the word `W`, containing `S` syllables, is at position `J` on row `I` in the generated poem

```
% Generator part
{ word(W,I,J,S) } :- candidate(W,I,J,S).                                   (G1)

% Testing part: the constraints
:- not 1 { word(W,I,J,S) } 1, rows(X), I = 1..X, positions(I,Y), J=1..Y. (T1)
```

Figure 3.2: Answer set program for generating poetry: the basic model.

This model consists of two parts: the *generator part* (Rule G1) and the *testing part* (Rule T1). Rule G1 of the generator part states that each candidate word for a specific position in the poem skeleton may be considered to be chosen for that position in the generated poem. Rule T1 of the testing part states that exactly one word must be selected for a specific position in the generated poem.

This very basic model can be extended in many ways. Further qualities of the poetic form, such as rhyme and syllable structures can be controlled by introducing additional predicates and rule-based constraints. An extended model in which specific rhyme and syllable structures, as well as minimum and maximum occurrences of words are specified, can be seen in Figure 3.3. Predicates of this extended model are listed in Table 3.2.

In this model, the predicate `must_rhyme/4` is used to provide pairwise word positions in the poem that should rhyme. Knowledge of rhyming candidate words is provided by the `rhymes/2` predicate. Similarly, the predicate `nof_syllables/2` defines the required number of syllables in a certain line and the predicate `syllables/2` defines the number of syllables

Table 3.2: Predicates used in extending the basic ASP model for poetry composition.

Predicate	Interpretation
must_rhyme(I,J,K,L)	the word at position J on row I and the word at position L on row K are required to rhyme
rhymes(X,Y)	the words X and Y rhyme
nof_syllables(I,C)	the Ith row of the poem is required to contain C syllables
min_occ(W,L)	L is the lower bound on the number of occurrence of the word W
max_occ(W,U)	U is the upper bound on the number of occurrence of the word W

```
% Generator part

{ word(W,I,J,S) } :- candidate(W,I,J,S).                              (G1)
rhymes(Y,X) :- rhymes(X,Y).                                           (G2)
syllables(W,S) :- candidate(W,_,_,S).                                 (G3)

% Testing part: the constraints

:- not 1 { word(W,I,J,S) } 1, rows(X), I = 1..X, positions(I,Y), J=1..Y. (T1)
:- word(W,I,J,S), word(V,K,L,Q), must_rhyme(I,J,K,L), not rhymes(W,V).   (T2)
:- Sum = #sum [ word(W,I,J,S) = S ], Sum != C, nof_syllables(I,C),       (T3)
   I = 1..X, rows(X).
:- not L { word(W,_,_,_) } U, min_occ(W,L), max_occ(W,U).                (T4)
```

Figure 3.3: Answer set program for generating poetry: extending the basic model.

for a given candidate word. Obviously, these predicates can then be used to specify certain rhyming and metrical structures for the generated poetry.

In the generation phase, rule T3 ensures that the number of syllables on each line of the poem matches the required number of syllables for that line. This extended model also controls repetition of words in the poems by using predicates min_occ(W,L) and max_occ(W,U) and rule T4 that constrains the number of word occurrences to be within certain lower and upper bounds. In Paper II, we outline further possibilities to control the content and form of the produced poetry in a similar way.

A different kind of aesthetics can be specified for the generated poetry by utilising soft constraints. The constraint solver then attempts to search for results that satisfy these soft constraints as well as possible. For instance in ASP, this is achieved by using the optimization statements offered by the language. This optimization also enables defining the system's internal aesthetics in an appealing way.

3.5 Transformational Song Composition with Constraint Programming

In Paper IV, we describe a song generation system that generates songs in a transformational way. This means that the system can reflect its actions and change its own conceptual space. The system uses constraint programming to define a conceptual space of possible songs that then enables using efficient constraint solvers to search for suitable solutions. Declarative constraint programming enables explicit transformationality by a metalevel component that can change the constraints of the system.

3.5.1 Metalevel Control and Transformational Creativity

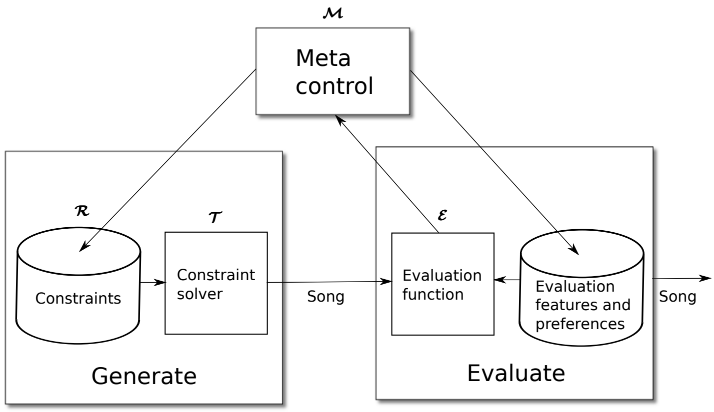
As described in Chapter 1, Boden's transformational creativity can be seen as metalevel search [94]. By operating on a metalevel with respect to pure exploratory creativity, a computationally creative system can transform its search space leading into ground-level artefacts that were impossibilist in the preceding conceptual spaces [2].

In Paper IV, we propose a practical, generic architecture for transformational creativity based on constraint programming, and present a song generation system that implements these ideas. Use of declarative constraints to define a conceptual space of the system enables the system to make changes to this space by modifying the constraints that shape it. The general architecture of the system can be seen in Figure 3.4.

The presented system works in an iterative way. First, it generates a song or several songs with default constraint settings. Then, it assesses the produced songs based on a pool of evaluation features included in the evaluation component of the system and adjusts its internal goals for subsequent songs based on this evaluation. Next, the system transforms the constraints in several different ways and produces new songs with all these different constraint settings. Then, the system evaluates these songs and selects the song that best satisfies the goals set in the previous iteration. After that, the goals for subsequent songs are adjusted again based on this best song, and the iterative process continues.

The system's capability to reflect and change its internal workings increases its autonomy. This self-reflection could be futher enhanced by providing the system an ability to communicate with other agents and get feedback concerning the artefacts it has produced [31].

Figure 3.4: Constraint programming allows the metalevel to modify the search space of the system specified by constraints $\mathcal{R}$ as in Wiggin's framework [94]. The evaluation component $\mathcal{E}$ is based on a set of features and target values for them, modifiable by the metalevel $\mathcal{M}$. A constraint solver provides the traversal function $\mathcal{T}$.



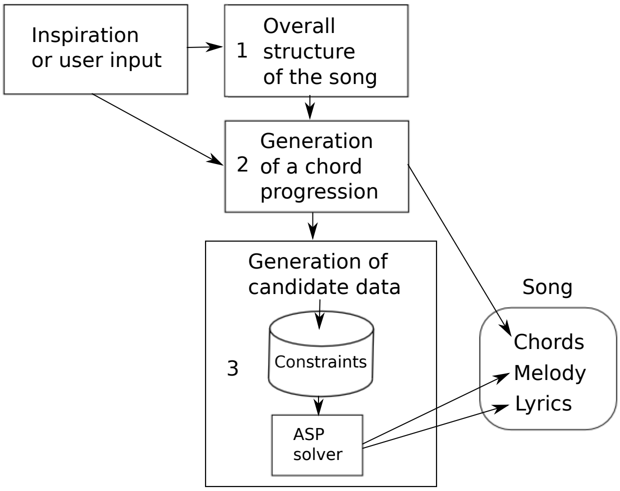
3.5.2 Ground-level Song Generation

On the technical implementation level, generation of songs, i.e. combinations of music and language, provides interesting challenges. In many existing generative systems that combine language and music, the generative process is inherently sequential: either music or language is generated first and then matching music or language is generated. In the song composition system presented in Paper IV, both the lyrics and melody of a song are generated together based on a joint constraint program.

The system aims to generate a variant range of songs that are easy to sing and play. The output consists of music notation with the melody, lyrics, and chords for accompaniment. Chords are presented with chord symbols. For output, we use Lilypond notation [65] that can be easily converted into pdf sheet music or transformed automatically into audio with MIDI synthesis if needed. A song is generated in three phases, gradually adding detail (the architecture of the ground-level song generation can be seen in Figure 3.5).

In the composition process, an overall section structure (e.g. ABAB) is selected first from a collection of different alternatives. Then, chord progressions are generated for each section type and repeated in all sections of the same type. Finally, melody and lyrics are generated using constraint programming, in two steps: generation of so-called candidate data consisting of possible notes and segments of lyrics, and application of ASP to compose songs out of this data using the current constraints.

Figure 3.5: Song composition architecture on the ground-level.



Melody and lyrics are generated with a constraint satisfaction approach in chunks of 4 bars so that they fit the overall structure and the underlying chord progression. Constraints enable modification of certain aspects of the produced songs while keeping other aspects intact. For the melody notes, a couple of different general characteristics are defined. These characteristics include note duration, pitch, consonance value with respect to the underlying chord, boolean value for belonging to the key signature, and boolean value for belonging to the underlying chord. In addition, the interval structure can be derived and controlled in the ASP program based on the note pitches. Lyrics of the song are constructed jointly with the melody by selecting one phrase from a set of phrases for every four bars in such a way that stressed syllables are accompanied with longer note durations and unstressed syllables with shorter note durations. Ground-level generation of songs and transformationality of the system are discussed in more detail in Paper IV.

Chapter 4

Evaluation and Cultural Applications

In this chapter, we take a closer look at computationally created artefacts and at applications of the methods presented in the previous chapter. The aim is to give an overview of mainly non-technical contributions of this work.

We discuss evaluation of computational creativity and present some preliminary evaluation results for computationally produced poems. Then, we move on to describe the Brain Poetry art installation and discuss how computationally created artefacts can be given personal meaning by connecting them to measurement of personal biosignals. After that, we overview poetry publications, exhibitions, and media coverage resulting from the work in this thesis. Then, we describe use of the methods developed in pedagogical tools, and finally at the end of the chapter, we discuss computational poetry in the context of literature.

4.1 Evaluation of Computationally Produced Artefacts

Evaluating progress in computational creativity is conceptually very difficult [34]. Many different approaches have been used to evaluate computationally produced artefacts, most prominently Turing-style tests and measurement of the societal impact of the aforementioned artefacts in publications and exhibitions.

Alan Turing [92] proposed the idea of measuring whether a digital computer can think or exhibit intelligent behaviour with an imitation game, usually known as the Turing test. Turing presented the idea of a machine behaving in an indistinguishable way from that of a human in a natural language conversation with a human judge. If the human judge cannot reliably distinguish between a machine and a human then the machine has passed the Turing test. In its original form, the test was intended to operationalize the vague concept of what it means for a machine to think or to exhibit intelligent behaviour. Turing-style tests have been applied several times also in computational creativity. Usually in these tests, human-created and machine-generated artefacts are evaluated according to certain qualities by a human judge who does not know which artefacts belong to which class.

However, Turing tests have been criticised [see e.g. 13] because they measure human-likeness instead of intelligence. For instance, aeroplanes are tested on how well they fly, not how much they resemble birds (reference to Russell and Norvig). Besides that, some human behaviour is not intelligent and some intelligent behaviour might not be human. For instance, Colton et al. [11] have criticised the Turing test in computational creativity settings for various reasons. Different styles of creativity are not equally recognised, contextual framing information is ignored, and evaluations are performed independently of context.

Other frameworks for evaluation have been proposed, as well. The tripod by Colton [8] is one of the earliest formal frameworks for evaluating computational creativity. Later Colton et al. [11, 70] have proposed the FACE and IDEA models as alternatives. The FACE model (Frame, Aesthetic, Concept, and Expression of a concept) represents ways to assess context, aesthetics, and concepts of interest, and also addresses how they are expressed. The IDEA model represents an Iterative Development Execution Appreciation cycle that assumes an ideal audience and measures the effect of a single creative act on that audience.

These frameworks seem potentially very useful in comparing different computationally creative systems, and they provide a promising approach to evaluating computational creativity systems and methods in the long term. However, we also see Turing-style test settings, and especially measurement of the societal impact of the computationally produced artefacts, as useful springboards towards methods and systems that exhibit real machine creativity. Due to problems prevalent in measuring the value of the produced artefacts, a better approach could be to measure their impact [9]. For instance, artistic publications and exhibitions can provide valuable forms of evaluating the impact of computationally creative systems in natural settings. In the following sections, we discuss the work of this thesis from that perspective.

4.2 Initial User Study

Because the intended audience of poetry – computational or not – consists of people, the most pragmatic way of evaluating computer poetry is by an empirical validation by human subjects. Thus, in the early phase of our research, we decided to carry out a preliminary study, described in Paper I, in which computationally produced poetry was evaluated using a panel of twenty subjects.

In this study, each subject independently evaluated a set of 22 poems. One half of these poems were human-written from the grammar corpus used in the poetry-generation experiments and the other half were computer-generated poems produced by the method described in Paper I with at least half of the words replaced. The subjects were not informed about the origin of the poems and the poems were presented in random order. Each text (poem) was evaluated by each subject. The first question to answer was if the subject considered the piece of text to be a poem or not, with a binary yes/no answer. Then each text was evaluated qualitatively along six dimensions: (1) How typical is the text as a poem? (2) How understandable is it? (3) How good is the language? (4) Does the text evoke mental images? (5) Does the text evoke emotions? (6) How much does the subject like the text? These dimensions were evaluated on a scale from one (very poor) to five (very good).

Figure 4.1: Is this a poem or not? The whiskers indicate an interval of 66.7 % of poems around the median. Points indicate the best and worst poems in the both groups.

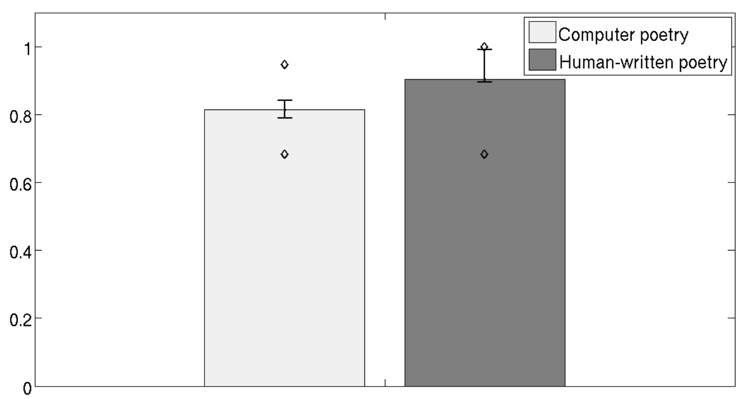
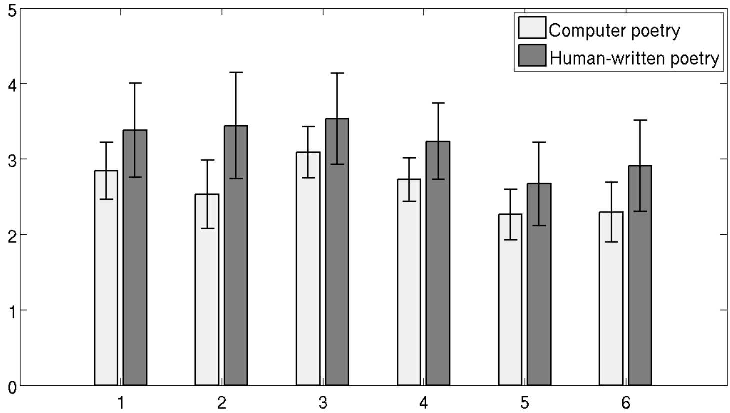


Figure 4.2: Subjective evaluation of computer-generated and human-written poetry along six dimensions: (1) typicality as a poem, (2) understandability, (3) quality of language, (4) mental images, (5) emotions, and (6) liking. Results are averaged over all subjects and poems. The whiskers indicate one standard deviation above and below the mean.



The results of the study are shown in Figures 4.1 (Question 1) and 4.2 (Questions 2-7) as averaged over subjects and poems. In all of the questions, the evaluation results have a similar pattern. Human-written poetry is deemed to be better than computer-generated and the differences are statistically highly significant. However, the average difference between the two categories is not large, and in many cases, there is overlap in the ranges of scores. Thus, some computer-generated poems have been evaluated as good as some human-written ones. The most notable difference is seen in the understandability of the texts (dimension 2). However, the quality of the language in computer-generated poetry (dimension 3) is deemed to be relatively good despite the simplicity of the natural language-generation technique. In this evaluation, the poems were in Finnish and one might expect somewhat different results in other languages.

In comparison to the aforementioned study, a better way to assess the success of the methods developed may be to measure the societal impact of computer-generated artefacts in societal settings, for instance, in publications, exhibitions, and media.

4.3 Poetry Publications, Exhibitions, and Media Coverage

Computational poetry produced with the presented methods – collectively known as the computer poet P. O. Eticus – has been published in several different venues including a book of pictures and haiku poetry by Heikki Paakkanen and P.O. Eticus (an example page can be seen in Figure 4.3) [68], Finnish literature journals Parnasso [87] and Särö [86], Czech literature journal Psí víno [89], an art project with computational poems and paintings based on them [27], and the Brain Poetry installation (Paper V). Original poems have been published in journals and news paper reports in Finnish, English, and German, translations of the poems in addition in Czech and Japanese.

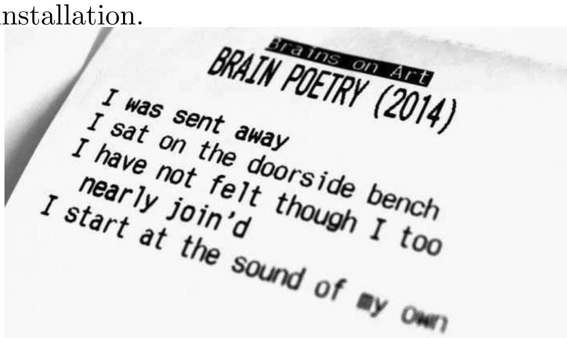
Figure 4.3: A page from the book Tee se kotona, kiitos! by Heikki Paakkanen and P.O. Eticus. P.O. Eticus wrote the original Finnish poem. The text in Japanese is a translation of this poem by Miika Pölkki. Reprinted with permission.

The research on poetry generation has also been covered in many electronic and press articles and radio and TV shows. A full listing of all the media coverage can be found in Appendix A. These publications, exhibitions, and media coverage show some societal impact of the methods developed but this interest may be caused, to a large extent, by the extraordinary nature of the poet. It is difficult to draw comprehensive conclusions about the quality of the methods and artefacts produced by them based on reader and media interest. However, embedding methods of computational creativity in societal settings on a larger scale shows some promise. The Brain Poetry art installation presented in Paper V serves as an example, and next we will discuss it in some more detail.

4.4 Brain Poetry Art Installation

Brain Poetry art installation, described in Paper V, creates poetry from an EEG signal, measured from a viewer's brain. The signal determines the style of a poem from a set of possible styles. In the installation, the measured brain activity is projected in real time on a screen. After a poem based on the signal has been generated, the poem is projected on the screen as well and printed out with a miniature thermal printer (an example of a brain poem can be seen in Figure 4.4).

Figure 4.4: A brain poem that is based on brainwave measurements of a viewer of the installation.



4.4.1 EEG Analysis and Poetry Generation

In the installation, EEG measurements are used to classify people into different groups based on their individual peak alpha frequency (IPAF). IPAF is an individually variable but relatively stable feature of the EEG power spectrum [40], thought to reflect alpha oscillations that are produced by thalamocortical feedback loops [41].

Based on the IPAF classification, viewers are divided into five groups. Each of these groups is assigned a different poetry generation method and style. This mapping between IPAF groups and poetry styles is arbitrary, but it remains constant for the duration of the installation. Thus, due to the stableness of IPAF, returning visitors typically receive similar poems as before. It is important to note that the arbitrary designation of poetry categories does not diminish the observed experiences of personal meaningfulness.

Poetry generation algorithms used in this work are partly based on methods described in Paper I. In addition, some other, simple methods are used to produce poetry in more varying styles. These methods are based on analysing existing text corpora for finding short text fragments with specific qualities like a certain metric structure and combining these fragments stochastically to produce new poems.

4.4.2 Connecting Personal Biosignal Measurements and Computational Creativity

Human connection is very important in much of the arts. We propose that linking methods of computational creativity to measurement of personal biosignals can provide personal meaning for the generated artefacts.

Electrophysiological biosignal measurements provide many interesting possibilities for combining algorithmic generation of art and people. These measurements can be based, for instance, on EEG (like in Brain Poetry), electrocardiography (ECG), electromyography (EMG), or electro-oculography (EOG). EEG provides an especially interesting case because it directly reflects the most fundamental human qualities, thinking and feeling. Connecting computational creativity and measurement of personal biosignals is further discussed in Paper V.

4.4.3 Reception of the Artwork

Brain Poetry has been exhibited in several places, including XL Art Space in Helsinki in autumn 2013, the Brain Waves art exhibition in Helsinki in spring 2014, and most notably, the Frankfurt International Book Fair in October 2014.

At the Frankfurt Book Fair, for instance, approximately 100,000 people visited the Guest of Honor Pavilion of Finland where Brain Poetry was one of the three exhibited art works. During the six days that Brain Poetry was exhibited there, almost 4500 unique poems were generated.

Brain Poetry also received considerable media attention at the Frankfurt Book Fair. The media coverage included German, Brazilian and Finnish newspapers, German and Finnish radio shows, and German television shows. In order to evaluate the reception of the installation, we analysed a sample of the electronic press articles and blog posts (19 in total) qualitatively using the following criteria: (1) how much of the whole article was specifically dedicated to Brain Poetry, (2) what kind of positive and negative expressions were used to describe the installation, and (3) whether the overall tone of the section on Brain Poetry was positive, neutral or negative. All the articles were analysed by five reviewers and the results of this analysis are based on their collective view.

Based on this analysis, we concluded that Brain Poetry attracted a lot of attention as most reports dedicated a section, in some cases two sections, to the installation. Thirteen of the reports had a generally positive tone, two had a generally negative tone, and four a neutral tone. Positive adjectives used to describe the installation included fun, cool, amusing, and beautiful. Some articles reported the grammar of the generated poetry to be surprisingly good. On the other hand, some reports described the poems as weird and creepy. Also, the undestandability of the generated poems was often questioned. In many cases, merely a technical description was provided.

4.5 Pedagogical tools

Pedagogical tools and educational games provide one application area for the presented methods. Kantosalo et al. [35, 36] have developed an interactive poetry-writing tool for learning about language and literature based on some of the presented methods. A prototype of this system has been tested in a Finnish school with children of age ten to twelve years. Their work also studies the possibility to transform computational creativity systems into interactive tools that support mixed processes of human-computer creativity [35].

The basis of the co-creative poetry writing system is in the methods described in Paper I. In addition, the system can provide poetic fragments satisfying certain criteria. For instance, these fragments can be in a certain meter or contain certain words. The fragments are then combined in novel ways and modified with the word substitution method in order to provide a building block of poetry writing for the user [35].

In a typical use case, a user of the system provides keywords, and based on them the computer generates a few lines of poetry that can be modified and extended by the user. The user can also ask for inspiration from the computer in the form of new poetry fragments and words with certain qualities such as specific rhymes for specific places in the poem. Text produced by the system adapts automatically to the modifications done by the user [35]. Such co-creative tasks as poetry writing can also provide motivation for school children and even older students [36].

4.6 Computational Poetry in the Context of Literature

Several literary schools have developed formal and mechanized methods for writing poetry. One of the most well-known of these schools is Oulipo [53] (French Ouvroir de littérature potentielle) that, among other things, devised constrained and methodical forms of poetry writing. Oulipo used already established techniques like palindromes and lipograms but also developed novel methods often having their origins in mathematical problems and ideas. Constraints used by the Oulipians included, for instance, S+7 (sometimes known as N+7) where every noun in a text is replaced with the seventh noun after it in a dictionary [53]. Thus, the current algorithmic methods for poetry generation have predecessors also in the context of literature.

According to our view, the literary interestingness of computationally generated poetry lies in the quantitative analysis of culture, and in the condensation of the knowledge gained by that analysis into new viewpoints in novel poetic forms. For instance, analysis of millions of books is not possible for a human reader but a computer can perform this analysis and thus produce something that was not conceivable by a human. Expressing the results of such computerized analyses automatically can provide valuable insights into the language and culture. Computer-generated poetry can also provide results that deviate from average speech and literature and, thus by breaking the rules of everyday language, provide new insights into the language and the world in general [86].

Chapter 5

Answers to the Research Questions

In this thesis, we have presented unsupervised methods and models for automated composition of poetry and songs and described the underlying ideas behind our approach. Now, we return to the research questions specified at the beginning and briefly summarise the answers we got and the observations we made:

> I *How can unsupervised corpus-based methods be used in flexible language generation, especially in poetry? By unsupervised methods we mean here methods that require a minimal amount of hand-crafted linguistic, world, and domain knowledge.*

In Paper I, we present a corpus-based poetry generation method that uses statistical word association analysis, morphological analysis and synthesis, and existing instances of language use to handle semantic, syntactic, and morphological aspects of poetry composition in an automated way. We argued this approach to be flexible and to a large extent language independent which is emphasised by the fact that we have developed a working system in three different languages, Finnish, English, and German. Due to the flexibility of this approach, it can produce versatile results and it can easily be used as a basic building block in more sophisticated poetry generation systems. The main deficiencies of this approach lie in its handling of semantics. Rigorous treatment of the natural language semantics could be based on logical formalisms such as the formalisms used by Manurung in his work [50], but combining the current flexible and unsupervised approach with logical formalisms seems very challenging. Another way to achieve semantically more sound poetry could be based on refining the unsupervised approach, for instance, by using deep belief nets to represent semantic information (see, e.g. [5] for use of deep belief nets in natural language processing). Also, guaranteeing constantly high-quality output is tricky though this can be seen as a general difficulty in most of the computational creativity systems.

In Paper III, we present a method for mining document-specific word associations and for using these associations in a poetry generation system to produce poetry that is based on a specific news story. This approach shares the pros and cons of the basic corpus-based poetry generation method presented in Paper I. Poems based on certain documents seem to reflect the original documents in some way, but the rigorous handling of semantics and control of what the poems say exactly would require substantial further developments. This is also an aspect that should be rigorously evaluated in future work.

> II *What kind of declarative methods can be utilised to control the properties of both content and form in automated poetry and song composition and how can language and music be unified into a coherent whole in generative processes?*

In Paper II, we extend the corpus-based poetry generation approach with constraint satisfaction methods to handle more aspects in the poetic content and form. By using answer set programming and off-the-shelf constraint solvers, many interacting poetic characteristics, such as rhyme, alliteration, meter, aspects of semantic content, and natural language syntax can be efficiently satisfied or optimised. The declarative approach based on answer set programming also provides a conceptually clear approach to control the complexity of the task. On the other hand, the current method requires more supervision in the construction phase than the approach presented in Paper I. However, this approach is still rather flexible and language independent.

In Paper IV, we present a song generation system that is likewise based on a declarative answer set programming approach and an off-the-shelf constraint solver. This system combines music and lyrics generation into a single generative phase by using constraint programming. The advantage of this approach in comparison to previous work on automated song composition is in its seamless combination of language and music. We have also carried out a small qualitative evaluation and concluded that the system can produce songs with a reasonable good fit between the melody and the lyrics. The described system is also able to produce a range of variant results and can be used as a flexible building block in more skillful song generation systems. Many aspects in the system, such as composition of musical structures and harmonies, are kept intentionally simple. These aspects need to be further extended in future work. In this approach, we have also mainly concentrated on a surface-level fit between the rhythmic qualities of language and melody note durations. This fit needs to be made deeper by matching, for instance, the semantic qualities of lyrics to specific characteristics in music. Also, a full-scale evaluation of the songs produced this way would be required. This system serves as a starting point when answering research question IV.

> III *How can transformational creativity of a generative system be modelled in an explicit way?*

In Paper IV, we describe how transformational creativity can be explicitly modelled with a constraint programming-based approach and present a song generation system that modifies its own goals and conceptual space in a transformational way. This is achieved by modelling the automated song composition process with declarative constraint programming. Then, a metalevel component of the system can control the system's conceptual space by modifying the constraints leading into new conceptual spaces. The capability to transform its own rules and goals enables the system to exercise creative autonomy.

> IV *How can the methods developed contribute to arts and how can the results be given personal meaningfulness?*

In Paper V, we present an art installation called Brain Poetry that combines some of the poetry generation methods developed with electroencephalography measurements. The main idea of this article is that computationally created artefacts can be made more interesting and relevant by connecting them to objectively measurable human qualities, such as different characteristics of the EEG signal. This paper also shows how some of the methods have been embedded in societal settings at large and discusses reception of the artwork.

In conclusion, this thesis contributes to the field of Computational Creativity in several ways. We have developed novel unsupervised methods for automatically producing poetry and songs, we have shown how these methods can produce artistic results, we have demonstrated the societal impact of the methods developed and artefacts produced by them, and we have presented the current methods in a wider context.

Chapter 6

Discussion and Conclusions

This thesis has proposed novel methods in automated poetry and song composition. The emphasis has been on developing maximally unsupervised methods with a minimal amount of hand-crafted linguistic, world, or specific domain knowledge.

We have seen how conceptually relatively simple methods can be used to produce artefacts in specific domains, such as poetry and songs. The amount of creativity exhibited by these methods is still open for debate. However, we claim that generation of novel and interesting content in various domains and media is an interesting and growing field of both scientific research and commercial applications. We are only taking the first steps in this area but possibilities, for instance, in the computer gaming industry and personalised web content creation seem huge.

We have proposed to use the term synthetic creativity to describe our approach in computational creativity. Here the idea is that a model is learned from data and then used to produce new artefacts in that area. This model can also be modified internally to produce artefacts which have their basic rooting in the already observed data but also more novel qualities, i.e. these artefacts are not just combinationally new instances of knowledge already incorporated into the system. In this way, even some transformational qualities of creativity could be achieved.

Evaluation of the results is difficult. However, several different frameworks for evaluating the results have been proposed. In our case, assessing the societal impact of the computationally created artefacts seems to be the best way to measure their value. In the present work, the publications of poetry and artistic work leveraging the methods developed here show such impact. However, more work on cultural applications and a longer time span is needed for making reliable judgements. Assessing whether the interest in computationally produced artefacts is long-lasting remains to be seen in the future. In an endeavour to develop better and more creative methods one needs to compare the societal impact of different methods and approaches. These comparisons are problematic because there are many variables besides the methods themselves that affect the observations.

Computational creativity is a nascent field, and thus there is little unifying theory or methodology. As creativity is a very complex phenomenon and the language used to describe creativity is plastic, it might even be unrealistic to speak of any unified theories for computational creativity or creativity in general. However, we argue that some generative principles are more general than others. We think that the unsupervised and data-driven nature of the methods gives more generality to the present approaches as it makes them more flexible and independent of the programmer.

In future work, more fine-grained semantic processing should be incorporated into the existing models and methods. For instance, deep belief net-based methods for modelling language semantics and syntax seem to provide very interesting possibilities. Use of hybrid models that include statistical and other techniques seem to provide one way to tackle many problems inherent in modelling of natural language and music, and to construct more interesting generative systems. We consider the methods presented here to be useful building blocks when constructing more skillful generative systems in language and music.

References

[1] C. M. Bishop. *Pattern recognition and machine learning.* Springer-Verlag New York, 1st edition, 2006.

[2] M. A. Boden. What is creativity. In M. A. Boden, editor, *Dimensions of creativity*, pages 75–117. MIT Press, 1994.

[3] M. A. Boden. Creativity and artificial intelligence. *Artificial Intelligence*, 103(1-2):347–356, 1998.

[4] M. A. Boden. *The Creative Mind: Myths and Mechanisms*. Routledge, London, 2nd edition, 2004.

[5] R. Collobert, J. Weston, L. Bottou, M. Karlen, K. Kavukcuoglu, and P. Kuksa. Natural language processing (almost) from scratch. *The Journal of Machine Learning Research*, 12:2493–2537, 2011.

[6] S. Colton. Refactorable numbers – a machine invention. *Journal of Integer Sequences*, 2:Art. 99.1.2, electronic only, 1999.

[7] S. Colton. Automated conjecture making in number theory using HR, Otter and Maple. *Journal of Symbolic Computation*, 39(5):593–615, 2005.

[8] S. Colton. Creativity versus the perception of creativity in computational systems. In *AAAI Spring Symposium: Creative Intelligent Systems*, pages 14–20, Palo Alto, California, 2008.

[9] S. Colton and G. A. Wiggins. Computational creativity: The final frontier? In *Proceedings of the 18th European Conference on Artificial Intelligence*, pages 21–26, Montpellier, France, 2012.

[10] S. Colton, A. Bundy, and T. Walsh. Automatic concept formation in pure mathematics. In *Proceedings of the 16th International Conference on Artificial Intelligence*, volume 2, pages 786–791, San Francisco, California, 1999.

[11] S. Colton, J. Charnley, and A. Pease. Computational creativity theory: The FACE and IDEA descriptive models. In *Proceedings of the Second International Conference on Computational Creativity*, pages 90–95, Mexico City, Mexico, 2011.

[12] S. Colton, J. Goodwin, and T. Veale. Full-FACE poetry generation. In *Proceedings of the Third International Conference on Computational Creativity*, pages 95–102, Dublin, Ireland, 2012.

[13] S. Colton, M. Cook, R. Hepworth, and A. Pease. On acid drops and teardrops: Observer issues in computational creativity. In *Proceedings of the 50th Annual Convention of the Society for the Study of Artificial Intelligence and the Simulation of Behaviour, AISB*, Goldsmiths, London, UK, 2014.

[14] T. Dunning. Accurate methods for the statistics of surprise and coincidence. *Computational linguistics*, 19(1):61–74, 1993.

[15] B. Díaz-Agudo, P. Gervás, and P. A. González-Calero. Poetry generation in COLIBRI. In Susan Craw and Alun Preece, editors, *Advances in Case-Based Reasoning*, volume 2416 of *Lecture Notes in Computer Science*, pages 73–87. Springer Berlin Heidelberg, 2002.

[16] K. Ebcioğlu. An expert system for harmonizing chorales in the style of JS Bach. *The Journal of Logic Programming*, 8(1):145–185, 1990.

[17] M. Gebser, R. Kaminski, B. Kaufmann, and T. Schaub. Clingo = ASP + control: Preliminary report. *arXiv preprint arXiv:1405.3694*, 2014.

[18] M. Gelfond and V. Lifschitz. The stable model semantics for logic programming. In *Logic Programming, Proceedings of the Fifth International Conference and Symposium*, pages 1070–1080, Seattle, Washington, USA, 1988.

[19] P. Gervás. WASP: Evaluation of different strategies for the automatic generation of Spanish verse. In *Proceedings of the AISB-00 Symposium on Creative & Cultural Aspects of AI*, pages 93–100, Birmingham, UK, 2000.

[20] P. Gervás. An expert system for the composition of formal Spanish poetry. *Journal of Knowledge-Based Systems*, 14(3–4):181–188, 2001.

[21] P. Gervás. Exploring quantitative evaluations of the creativity of automatic poets. In *Workshop on Creative Systems, Approaches to*

Creativity in Artificial Intelligence and Cognitive Science, 15th European Conference on Artificial Intelligence, Amsterdam, Netherlands, 2002.

[22] P. Gervás. Engineering linguistic creativity: Bird flight and jet planes. In *Proceedings of the North American Chapter of the Association for Computational Linguistics Human Language Technologies (NAACL HLT 2010) Second Workshop on Computational Approaches to Linguistic Creativity*, pages 23–30, Los Angeles, California, USA, 2010.

[23] P. Gervás. Computational modelling of poetry generation. In *Artificial Intelligence and Poetry Symposium, The Society for the Study of Artificial intelligence and Simulation of Behaviour (AISB) Convention*, Exeter, UK, 2013.

[24] P. Gervás, B. Díaz-Agudo, F. Peinado, and R. Hervás. Story plot generation based on CBR. *Knowledge-Based Systems*, 18(4):235–242, 2005.

[25] J. Goldenberg, D. Mazursky, and S. Solomon. Creative sparks. *Science*, 285(5433):1495–1496, 1999.

[26] O. Gross, A. Doucet, and H. Toivonen. Document summarization based on word associations. In *Proceedings of the 37th international ACM SIGIR conference on Research and Development in Information Retrieval*, pages 1023–1026, Gold Coast, Australia, 2014.

[27] O. Gross, J. M. Toivanen, H. Toivonen, and S. Lääne. Arts, news, and poetry – the art of framing. In *Proceedings of the Fifth International Conference on Computational Creativity*, pages 336–339, Ljubljana, Slovenia, 2014.

[28] K. Hamada, T. Baba, and M. Yufu. Hybridizing a genetic algorithm with rule-based reasoning for production planning. *IEEE Intelligent Systems*, 10(5):60–67, 1995.

[29] J. Haugeland. *Artificial intelligence: The very idea.* MIT press, 1989.

[30] J. Jaffar and M. J. Maher. Constraint logic programming: A survey. *The journal of logic programming*, 19:503–581, 1994.

[31] K. E. Jennings. Developing creativity: Artificial barriers in artificial intelligence. *Minds and Machines*, 20(4):489–501, 2010.

[32] P. N. Johnson-Laird. Jazz improvisation: A theory at the computational level. In *Representing musical structure*, pages 291–325. Academic Press, London, 1991.

[33] M. I. Jordan and T. M. Mitchell. Machine learning: Trends, perspectives, and prospects. *Science*, 349(6245):255–260, 2015.

[34] A. Jordanous. A standardised procedure for evaluating creative systems: Computational creativity evaluation based on what it is to be creative. *Cognitive Computation*, 4(3):246–279, 2012.

[35] A. Kantosalo, J. M. Toivanen, P. Xiao, and H. Toivonen. From isolation to involvement: Adapting machine creativity software to support human-computer co-creation. In *Proceedings of the Fifth International Conference on Computational Creativity*, pages 1–7, Ljubljana, Slovenia, 2014.

[36] A. Kantosalo, J. M. Toivanen, and H. Toivonen. Interaction evaluation for human-computer co-creativity: A case study. In *Proceedings of the Sixth International Conference on Computational Creativity*, pages 276–283, Park City, Utah, USA, 2015.

[37] M. Karlgren and J. Sahlgren. From words to understanding. In Y. Uesaka, P Kanerva, and H. Asoh, editors, *Foundations of Real-World Intelligence*, pages 294–311. Stanford: CSLI Publications, 2001.

[38] R. Kell. Content and form in poetry. *British Journal of Aesthetics*, 5 (4):382–385, 1965.

[39] R. D. King, J. Rowland, S. G. Oliver, M. Young, W. Aubrey, E. Byrne, M. Liakata, M. Markham, P. Pir, L. N. Soldatova, A. Sparkes, K. E. Whelan, and A. Clare. The automation of science. *Science*, 324(5923):85–89, 2009.

[40] W. Klimesch. EEG alpha and theta oscillations reflect cognitive and memory performance: a review and analysis. *Brain research reviews*, 29(2):169–195, 1999.

[41] W. Klimesch, P. Sauseng, and S. Hanslmayr. EEG alpha oscillations: the inhibition–timing hypothesis. *Brain research reviews*, 53(1):63–88, 2007.

[42] I. Langkilde and K. Knight. The practical value of n-grams in generation. In *Proceedings of the International Natural Language Generation Workshop*, pages 248–255, Niagara-on-the-Lake, Ontario, Canada, 1998.

[43] P. Langley, H. A. Simon, G. L. Bradshaw, and J. M. Zytkow. *Scientific discovery: Computational explorations of the creative processes*. MIT press, 1987.

[44] D. B. Lenat. On automated scientific theory formation: a case study using the AM program. *Machine intelligence*, 9:251–286, 1979.

[45] D. Leung and J. Romagnoli. Dynamic probabilistic model-based expert system for fault diagnosis. *Computers & Chemical Engineering*, 24(11):2473–2492, 2000.

[46] R. P. Levy. A computational model of poetic creativity with neural network as measure of adaptive fitness. In *Proceedings of the First Workshop on Creative Systems, International Conference of Case-Based Reasoning*, pages 165–170, Vancouver, Canada, 2001.

[47] D. Lin, K. W. Church, H. Ji, S. Sekine, D. Yarowsky, S. Bergsma, K. Patil, E. Pitler, R. Lathbury, V. Rao, K. Dalwani, and S. Narsale. New tools for web-scale N-grams. In *Proceedings of the Seventh International Conference on Language Resources and Evaluation*, pages 2221–2227, Malta, 2010.

[48] K. Lindén, M. Silfverberg, and T. Pirinen. HFST tools for morphology – an efficient open-source package for construction of morphological analyzers. In *Proceedings of the Workshop on Systems and Frameworks for Computational Morphology*, pages 28–47, Zurich, Switzerland, 2009.

[49] T. I. Lubart. Creativity. In R. J. Sternberg, editor, *Thinking and problem solving*, volume 2. Academic Press, USA, 1994.

[50] H. M. Manurung. *An Evolutionary Algorithm Approach to Poetry Generation*. PhD thesis, University of Edinburgh, Edinburgh, UK, 2003.

[51] H. M. Manurung, G. Ritchie, and H. Thompson. Towards a computational model of poetry generation. In *Proceedings of AISB Symposium on Creative and Cultural Aspects and Applications of AI and Cognitive Science*, pages 79–86, Birmingham, UK, 2000.

[52] J. M. Martins, F. C. Pereira, E. R. Miranda, and A. Cardoso. Enhancing sound design with conceptual blending of sound descriptors. In *Proceedings of the First Joint Workshop on Computational Creativity*, pages 65–77, Madrid, Spain, 2004.

[53] H. Mathews, A. Brotchie, and R. Queneau. *Oulipo compendium*. Atlas Press, London, UK, 2005.

[54] P. McCorduck. *AARON's Code: Meta-Art, Artificial Intelligence, and the Work of Harold Cohen*. WH Freeman & Co, New York, USA, 1990.

[55] M. S. Mccoy and R. R. Levary. A rule-based pilot performance model. *International Journal of Systems Science*, 31(6):713–729, 2000.

[56] M. S. McGlone and J. Tofighbakhsh. Birds of a feather flock conjointly (?): Rhyme as reason in aphorisms. *Psychological Science*, 11(5): 424–428, 2000.

[57] J.-B. Michel, Y. Kui Shen, A. P. Aiden, A. Veres, M. K. Gray, J. P. Pickett, D. Hoiberg, D. Clancy, P. Norvig, J. Orwant, S. Pinker, M. A. Nowak, and E. Lieberman Aiden. Quantitative analysis of culture using millions of digitized books. *Science*, 331(6014):176–182, 2011.

[58] G. Minnen, J. Carroll, and D. Pearce. Applied morphological processing of English. *Natural Language Engineering*, 7(03):207–223, 2001.

[59] M. L. Minsky. Why people think computers can't. *AI Magazine*, 3(4): 3–15, 1982.

[60] K. Monteith, V. Francisco, T. Martinez, P. Gervás, and D. Ventura. Automatic generation of emotionally-targeted soundtracks. In *Proceedings of the Second International Conference on Computational Creativity*, pages 60–62, Mexico City, Mexico, 2011.

[61] K. Monteith, T. Martinez, and D. Ventura. Automatic generation of melodic accompaniments for lyrics. In *Proceedings of the Third International Conference on Computational Creativity*, pages 87–94, Dublin, Ireland, 2012.

[62] R. A. Morelli, J. D. Bronzino, and J. W. Goethe. Expert systems in psychiatry. *Journal of medical systems*, 11(2-3):157–168, 1987.

[63] Y. Netzer, D. Gabay, Y. Goldberg, and M. Elhadad. Gaiku: Generating haiku with word associations norms. In *Proceedings of the*

Conference of the North American Chapter of the Association for Computational Linguistics Workshop on Computational Approaches to Linguistic Creativity, pages 32–39, Boulder, Colorado, USA, 2009.

[64] I. Niemelä. Logic programs with stable model semantics as a constraint programming paradigm. *Annals of Mathematics and Artificial Intelligence*, 25(3-4):241–273, 1999.

[65] H. W. Nienhuys and J. Nieuwenhuizen. Lilypond, a system for automated music engraving. In *Colloquium on Musical Informatics (XIV CIM 2003)*, Firenze, Italy, 2003.

[66] R. Ochse. *Before the gates of excellence: The determinants of creative genius*. Cambridge University Press, UK, 1990.

[67] H. R. G. Oliveira, F. A. Cardoso, and F. C. Pereira. Tra-la-lyrics: An approach to generate text based on rhythm. In *Proceedings of the Fourth International Joint Workshop on Computational Creativity*, pages 47–55, Goldsmiths, London, UK, 2007.

[68] H. Paakkanen and P. O. Eticus. *Tee se kotona, kiitos!* AAARGH! hyväluoja, 2012. ISBN 9789519770840.

[69] M. T. Pearce and G. A. Wiggins. Evaluating cognitive models of musical composition. In *Proceedings of the Fourth international joint workshop on computational creativity*, pages 73–80, Goldsmiths, London, UK, 2007.

[70] A. Pease and S. Colton. On impact and evaluation in computational creativity: A discussion of the Turing test and an alternative proposal. In *Proceedings of the Society for the Study of Artificial Intelligence and Simulation of Behaviour (AISB) Symposium on AI and Philosophy*, pages 15–22, York, UK, 2011.

[71] F. C. Pereira and A. Cardoso. Experiments with free concept generation in Divago. *Knowledge-Based Systems*, 19(7):459–470, 2006.

[72] R. E. Plant and M. P. Vayssières. Combining expert system and GIS technology to implement a state-transition model of oak woodlands. *Computers and Electronics in Agriculture*, 27(1):71–93, 2000.

[73] A. Ramakrishnan, S. Kuppan, and S. L. Devi. Automatic generation of Tamil lyrics for melodies. In *Proceedings of the Workshop on Computational Approaches to Linguistic Creativity (CALC'09)*, pages 40–46, Boulder, Colorado, USA, 2009.

[74] F. Rashel and R. Manurung. Pemuisi: a constraint satisfaction-based generator of topical Indonesian poetry. In *Proceedings of the Fifth International Conference on Computational Creativity*, pages 82–90, Ljubljana, Slovenia, 2014.

[75] C. Roads. *The Computer Music Tutorial*. The MIT Press, 1996.

[76] R. Schank and A. Kass. Explanations, machine learning, and creativity. In Y. Kodratoff and S. Michalski, editors, *Machine learning: an artificial intelligence approach, Volume III*, pages 31–60. Morgan Kaufmann, 2014.

[77] B. Schneiderman. Creativity support tools: Accelerating discovery and innovation. *Communications of the ACM*, 50(12):20–32, 2007.

[78] M. Scirea, G. A. B. Barros, N. Shaker, and J. Togelius. SMUG: Scientific music generator. In *Proceedings of the Sixth International Conference on Computational Creativity*, pages 204–211, Park City, Utah, USA, 2015.

[79] M. S. Seidenberg. Language acquisition and use: Learning and applying probabilistic constraints. *Science*, 275(5306):1599–1603, 1997.

[80] M. S. Seidenberg and M. C. MacDonald. A probabilistic constraints approach to language acquisition and processing. *Cognitive science*, 23 (4):569–588, 1999.

[81] P. Simons, I. Niemelä, and T. Soininen. Extending and implementing the stable model semantics. *Artificial Intelligence*, 138(1-2):181–234, 2002.

[82] D. K. Simonton. Scientific creativity as constrained stochastic behavior: the integration of product, person, and process perspectives. *Psychological bulletin*, 129(4):475–494, 2003.

[83] R. J. Sternberg and T. I. Lubart. An investment theory of creativity and its development. *Human development*, 34(1):1–31, 1991.

[84] R. J. Sternberg and T. I. Lubart. The concept of creativity: Prospects and paradigms. In R. J. Sternberg, editor, *Handbook of creativity*, volume 1, pages 3–15. Cambridge University Press, UK, 1999.

[85] B. C. L. Tobing and R. Manurung. A chart generation system for topical metrical poetry. In *Proceedings of the Sixth International Conference on Computational Creativity*, pages 308–314, Park City, Utah, USA, 2015.

[86] J. M. Toivanen. Elot sai karkelojen teitä – koneet, luovuus ja runous. *Särö – kirjallisuus ja kulttuurilehti*, 23–24:68–73, 2014.

[87] J. M. Toivanen and P. Toivanen. Konedadaa vai keskeisloruja? *Parnasso*, 62(4):22–25, 2012.

[88] J. M. Toivanen, H. Toivonen, and A. Valitutti. Automatical composition of lyrical songs. In *Proceedings of the Fourth International Conference on Computational Creativity*, pages 87–91, Sydney, Australia, 2013.

[89] J. M. Toivanen, H. Toivonen, A. Valitutti, and O. Gross. Generování obsahu a formy v poezii na základé korpusu. *Psí Víno*, (68):40–57, 2014.

[90] S. C. Tornay. *Ockham: Studies and selections*. The Open Court Pub. Co., USA, 1938.

[91] K. Toutanova, D. Klein, C. D. Manning, and Y. Singer. Feature-rich part-of-speech tagging with a cyclic dependency network. In *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology-Volume 1*, pages 173–180, Edmonton, Canada, 2003.

[92] A. M. Turing. Computing machinery and intelligence. *Mind*, pages 433–460, 1950.

[93] A. Valitutti, H. Toivonen, O. Gross, and J. M. Toivanen. Decomposition and distribution of humorous effect in interactive systems. In *Artificial Intelligence of Humor, AAAI Fall Symposium Series*, pages 96–100, Arlington, Virginia, USA, November 2012.

[94] G. A. Wiggins. Searching for computational creativity. *New Generation Computing*, 24(3):209–222, 2006.

[95] M. T. Wong and A. H. W. Chun. Automatic haiku generation using VSM. In *Proceedings of the Seventh International Conference on Applied Computer and Applied Computational Science (ACACOS'08)*, pages 318–323, Hangzhou, China, 2008.

Appendices

Appendix A

Computationally produced poetry by P.O. Eticus and the Brain Poetry installation have been featured in the media widely. This is a listing of the most relevant media coverage by November 2015.

1. Radioshow Ajantasa in YLE Suomi, November 2011. A radioshow on computational poetry by P. O. Eticus. Poetry critique by Jukka Virtanen.
2. Karjalainen newspaper: "Koneen muisti riittää jo runoihin" February 2012.
3. Helsingin Sanomat newspaper: "Kuvia katujyrällä, runoja tietokoneella", September 2012.
4. Yliopisto magazine: "Vinksahtaneita haikuja", October 2012.
5. TV show YLE Strada, September 2013.
6. TV show Hyvät ja huonot uutiset, Nelonen, May 2014.
7. Suomen Kuvalehti magazine: "Messujen jälkeen hiljaisuus", October 2014.
8. ScienceBlogs: "Buchmesse 1: Luftfahrt. Gehirnwellen. Mond. Autoren (auch)", October 2014.
9. Rhein-Zeitung newspaper: "Brain Poetry: Finnen wandeln Hirnströme in Versmasse um", October 2014.
10. NZW Online newspaper: "Ehrengast Finnland zeigt sich skandinavisch offen", October 2014.

11. Frankfurter Allgemeine newspaper: “Die klare, wahre, unverfälschte Natur”, October 2014.
12. Echo Online newspaper: “Die Frankfurter Buchmesse startet - In Frankfurt wappnet sich die Branche gegen Amazongs Algorithmen, und die Finnen entsaften unsere Hirne”, October 2014.
13. Focus Online: “Ehrengast Finnland zeigt sich skandinavisch offen”, October 2014.
14. Tekniikka & Talous magazine: “Rautaista runoutta”, November 2014.
15. TV show Prisma Studio, February 2015.

Appendix B

Examples of Replacement Poems in Finnish and English with General Word Associations

On the topic “sielu” (soul):

Laula se ukkoseksi suurten sielujen
luo onnien, jumalin
hoiloa sielu!
Kirkkaus.
Tuo korkea herra huus:
“Pyhä hehkuko meiltä soinut ois?”

On the topic “death”:

Principal parade of skeletons,
two more positions
paint out sacred plastics of corporation,
undergoing my skeleton.

On the topic “sadness”:

For me emotions are sadness.
I sacrifice my meditation, living.
And now the lowest unhappiness
is nourished across the purposelessness of the guilt.

Examples of Constraint Poems in Finnish and English

On the topic “aurinko” (the sun) with exactly ten syllables on every line:

Nousi kuu laskien tytön luota
armaan laskit kohta kahta lasket
kevään varhaisen soi neiollensa
elo nurjamieli kalliolla
kolmeen suveen virtensä äärelle.

A lipogram omitting the letter o and with a rhyme pattern ABAB on the topic “music”:

Music swells, accent practises, theatre hears!
Her delighted epiphanies bent in her universe:
– And then, singing directly a universe she disappears!
An anthem in the judgements after verse!

Examples of Replacement Poems in English with Document-Specific Word Associations

Based on a news story about the US states reconsidering execution methods:

I die;
perhaps I have began;
this is a doubt;
this is a prisoner;
and there is state....

Based on a news story about Justin Bieber drinking and driving:

Is it the entourage, the sport,
the singer of later lamborghinis,
and the early thursdays of our singers?
These are but justins

Examples of Songs