AUTOMATIC AND DOMAIN-AGNOSTIC CREATIVITY MEASUREMENTS ON DIGITAL HUMAN TRACES

Towards Understanding Human Creativity at Scale

KIM-PASCAL BORCHART

Automatische und domänenagnostische Kreativitätsmessungen an digitalen menschlichen Aufzeichnungen

Universitätsmasterarbeit
zur Erlangung des akademischen Grades

Master of Science
(M.Sc.)

im Studiengang
IT-Systems Engineering

Chair of Internet Technologies and Systems
Digital Engineering Faculty
University of Potsdam

Betreuer	Prof. Dr. Christoph Meinel
	Prof. Dr. Robert Hirschfeld
Co-Betreuer	Dr. Julia v. Thienen

Eingereicht am 2. November 2022.

Kim-Pascal Borchart: *Automatic and Domain-Agnostic Creativity Measurements on Digital Human Traces*, Towards Understanding Human Creativity at Scale, © November 2, 2022

Creativity is intelligence
having *fun*.

— Albert Einstein

Dedicated to Sophie.

ABSTRACT

Most creativity studies assess only a limited number of people and creative behaviors. Traditional creativity assessments require evaluations by human expert judges, rendering assessments of creative behavior time-intensive and subjective. In my master's thesis, I apply definitions of creativity to digital behavior traces, to make algorithmic evaluations possible. I introduce C-Tracer, a data analysis tool that automatically generates various creativity scores from human event log data. Additionally, I report empirical studies in different content domains and with varying groups of participants, where C-Tracer has been used to score creativity in human behavior automatically. Empirical validation in early studies shows that C-Tracer scores can correlate significantly with creativity scores from human expert raters. Thus, it can become feasible to study large numbers of people and creative behaviors, with evaluation approaches that are less subjective and more time-efficient than human expert judgments.

ZUSAMMENFASSUNG

Die meisten Kreativitätsstudien untersuchen nur eine begrenzte Anzahl von Personen und kreativen Verhaltensweisen. Herkömmliche Kreativitätsbewertungen benötigen Experten, sodass das Bewerten von kreativem Verhalten zeitintensiv und subjektiv ist. In meiner Masterarbeit wende ich Kreativitätsdefinitionen auf digitale Verhaltensspuren an, um eine algorithmische Bewertung möglich zu machen. Ich stelle den C-Tracer vor, ein Datenanalysetool welches aus menschlichen Ereignisprotokolldaten automatisch Bewertungen von verschiedenen Kreativitätsmetriken generiert. Außerdem beschreibe ich empirische Studien aus verschiedenen Domänen und mit unterschiedlichen Teilnehmergruppen, in denen der C-Tracer verwendet wurde, um Kreativität in menschlichem Verhalten automatisch zu bewerten. Empirische Validierungen in frühen Studien zeigen, dass die C-Tracer-Bewertungen signifikant mit den Kreativitätsbewertungen von menschlichen Experten korrelieren können. So wird es möglich, eine große Anzahl von Menschen und kreativen Verhaltensweisen mit Bewertungsansätzen zu untersuchen, die weniger subjektiv und dabei zeitsparender als menschliche Expertenurteile sind.

PUBLICATIONS

Some ideas and figures have appeared previously in the following publications and presentations:

[1] Kim-Pascal Borchart. "Automated Creativity Measurement." In: *HPDTRP Community Building Workshop*. HPDTRP Community Building Workshop. Potsdam, Germany, 2021.

[2] Kim-Pascal Borchart, Julia von Thienen, and Christoph Meinel. "C-Tracer: Automatic Creativity Measurement for any Goal-Directed Behaviour that Leaves Digital Traces." In: *MIC conference of creativity*. MIC conference of creativity. Bologna, Italy, Sept. 8, 2021.

[3] Corinna Jaschek, Kim-Pascal Borchart, and Eva Krebs. "Immune Defense. A Video Game to Measure Creativity." In: *MIC conference of creativity*. MIC conference of creativity. Bologna, Italy, 2020.

[4] Corinna Jaschek, Kim-Pascal Borchart, and Eva Krebs. "Improving Creative Team Performance and Togetherness in Remote Interaction via Motion-Based Games." In: *MIC conference of creativity*. MIC conference of creativity. Bologna, Italy, Sept. 8, 2021.

[5] Corinna Jaschek, Kim-Pascal Borchart, Julia von Thienen, and Christoph Meinel. *The CollaboUse Test for Automated Creativity Measurement in Individuals and Teams: A Construct Validation Study*.

[6] Eva Krebs, Corinna Jaschek, Julia von Thienen, Kim-Pascal Borchart, Christoph Meinel, and Oren Kolodny. "Designing a Video Game to Measure Creativity." In: *2020 IEEE Conference on Games (CoG)*. 2020 IEEE Conference on Games (CoG). Osaka, Japan: IEEE, Aug. 2020, pp. 407–414. ISBN: 978-1-72814-533-4. DOI: 10.1109/CoG47356.2020.9231672. URL: https://ieeexplore.ieee.org/document/9231672/ (visited on 07/01/2022).

[7] Holly A. McKee. "Automatic Detection of the Flow Mental State in the Context of Creative Collaboration." Master Thesis. Potsdam, Germany: University of Potsdam, 2022.

[8] Julia von Thienen, Kim-Pascal Borchart, Corinna Jaschek, Eva Krebs, Justus Hildebrand, Hendrik Ratz, and Christoph Meinel. "Leveraging Video Games to Improve IT-Solutions for Remote Work." In: *2021 IEEE Conference on Games (CoG)*. 2021 IEEE Conference on Games (CoG). Copenhagen, Denmark: IEEE, Aug. 17, 2021, pp. 01–08. ISBN: 978-1-66543-886-5. DOI: 10.1109/CoG52621.2021.9618986. URL: https://ieeexplore.ieee.org/document/9618986/ (visited on 07/01/2022).

CONTENTS

LIST OF FIGURES

LIST OF TABLES

LISTINGS

ACRONYMS

AUT Alternative Uses Task

API Application Programming Interface

CAT Consensual Assessment Technique

CBI Creative Behavior Inventory

CDQ-R Revised Creativity Domain Questionnaire

COG IEEE Conference on Games

CRJ Creativity Research Journal

CSV Comma-Separated Values

CT Convergent Thinking

DT Divergent Thinking, Design Thinking

ICIT Ice Cream Imagination Task

K-DOCS Kaufmann's Domains of Creativity Scale

LSA Latent Semantic Analysis

MTurk Amazon Mechanical Turk

MVC Model-View-Controller

RAT Remote Associates Test

TTCT Torrance Test of Creative Thinking

UI User Interface

UX User Experience

Part I

THE PROBLEM SPACE

1 INTRODUCTION

Are you creative?

"Creativity pervades human life. It is the mark of individuality, the vehicle of self-expression, and the engine of progress in every human endeavor." [48] The impact of creativity on our lives is tangible. Innovation [18, 26, 56], public and personal achievement [55, 66], teacher success [50], intelligence [20, 59], even success in tourism development [53], are all connected to creativity. Being creative is one of the fundamentally *human* traits, that lets us navigate the nuanced and difficult challenges we face. Understanding creativity - what it is, what its consequences are, and how we can facilitate it - are questions that, while investigated for a long time, can still not be answered succinctly by creativity experts.

"It is audacious and ambitious to attempt to measure a construct such as creativity", says Bonnie Cramond [14], but answering questions about creativity to learn not just about oneself but also about groups of people has fascinated societies for millennia and scientists, officially, for about a century. J.P. Guilford is often credited for kickstarting creativity research in his 1950 presidential address [21], where he posed two questions: How can we find creative potential in young people? How can we promote the development of creative personalities? Now, 72 years later, researchers are still asking the same questions. Understanding creativity conceptually - as a personality trait and philosophical concept - as well as empirically - as a detectable and measurable fact - has become a focus of research that remains as dynamic as the concept of creativity itself: Methods of quantifying a person's creativity have constantly evolved since their inception, as they mirror our evolving understanding of the creative concept itself.

Confounding Factors in Creativity Measurement

Creativity is the driver of progress. Creative solutions to evolving problems are as important as ever. Against this backdrop, the fact that we as a society are on a trend towards becoming less creative is worrying [34]. As we look to support creativity and creative thinking, we still struggle to scientifically measure it. Creativity assessments are both time-consuming and unreliable, which are traits that make advocating for investing in large-scale creativity research more difficult. Unlike some other quantitative population testing methods, the concept of creativity remains just elusive enough to struggle with quick assessments, and quantitative creativity assessments often use the Consensual Assessment Technique (CAT) [5], which requires mul-

tiple raters to each evaluate every data point. Even sub-facets of creativity such as fluency, which is often measured by counting ideas, either require manual human oversight or the acceptance that joke answers, such as writing down all numbers from one to ten instead of one's creative ideas, are muddying the data analysis.
On the reliability side, the most common approach to achieve measurable ratings that are also acceptably reliable is also the CAT, which has been used many times in as many different ways. Intended to provide the reliability that creativity research had been missing, it now suffers from being used in studies without a standard that is being followed by CAT researchers. The CAT's requirements are to have a number larger than 1 of self-chosen raters rate answers from creativity tests on a response level and then report the inter-rater reliability, also on the response level. What makes an expert an expert, or whether expertise is even needed for good ratings is not discussed, and may contribute significantly to the difficulty of replicating experiments. Expert raters may have lower inter-rater reliability than non-experts, with an increased effect when comparing experts with non-expert ratings [31]. All of this means that the exact same study responses may not generate the same test scores if evaluated twice.

Sample size can have a strong effect on rater reliability [19], which calls into question the relation between sample size, rating validity, and rating reliability: As sample sizes increase, the time spent evaluating responses is proportionally increased for each rater, which can lead to fatigue, further influencing the ratings' inherent validity [15]. Then, the raters' opinions on the sample size may influence their ratings. For example, preschool children's responses have to be judged differently from the same children's responses twenty years later [15]. All of these factors together lead to reproducibility rates as low as between 25 and 62 % [10, 35, 46]. Similarly, some of the most-used creativity tests have shown to have a low retest reliability [6, 17]. How then can we design comparable studies, unless population, sample size, raters, and study setup are the same?

Designing Studies for a Dynamic Construct

Many historical discussions on creativity were less interested in the fundamental understanding of creativity but in defining what kind of people could be creative in the first place. In ancient Greece for example, poetry was considered the only creative endeavor. Other disciplines, such as painting or sculpting, were considered to be merely discovering what can be observed about a natural, lawful world [63]. Answering if one was creative was a simple matter: It only required asking if one was a poet. Over time, societies deemed more and more disciplines creative, from the Romans considering many artistic efforts creative, to discussions on creativity in nature and sciences at the beginning of the 20th century [63]. Today, discourse on creativity largely considers creativity a ubiquitous part of our human existence [3, 6, 14, 22, 23, 48].

The *science* of creativity has just emerged in the last century [7] and Guilford is often credited with the first scientific approaches of testing, quantifying, and evaluating creativity. This laid the groundwork for creativity research, and many of Guilford's concepts, introduced in the 1950s [22, 23], are still used or built upon today.

Most people even now have differing concepts of what exactly constitutes being creative [52], so when researchers are asked to design studies to evaluate creativity and, just as importantly, its effects on our lives, they may need to define specific study conditions to measure the specific aspect of creativity they are interested in. Not all do [51], which contributes to one of the fundamental difficulties of creativity research: Agreeing that two different studies measured the "same" creativity.

If creativity evolves with the societies defining it, then creativity assessments need to evolve with it. Imagine conducting a large-scale creativity study in ancient Greece. It is a self-report questionnaire with only one question: "Are you a poet?" If *Yes*: You are creative. If *No*: You are not creative. Because of the simple definition of what the ancient Greeks considered to be creative, the study design would be rather simple. Because nowadays definitions of what constitutes creativity are much more intricate, creativity studies struggle to capture these intricacies comprehensively.

To achieve some comprehensiveness, today's creativity studies often use large testing batteries to evaluate not just *if* participants are creative, but in which of the many ways their creativity manifests. Conducting those studies large-scale is time-consuming (and because of rater fatigue maybe even less reliable), as many tests within the testing batteries require not just a robust holistic analysis, but an interpretation of the results on a response level. This leads to researchers evaluating thousands of responses by hand [60].

In an increasingly automated world, "large-scale" has become less of a challenge to fear but a challenge to be solved. For example, automated and large-scale intelligence tests have been conducted, either in-person or online, for many years [24, 38]. It's easy to see that data generated by these types of intelligence tests is easy to be evaluated by a machine: All answers given by participants are either right or wrong.

No such dichotomy exists for creativity, which is often defined as the combination of *originality* and *success* [22]. That originality may be hard to grasp by a machine seems rather clear, but even success can be difficult to define in creative contexts. When is a painting successful? The evaluation of originality and success is still mostly deferred to human judges to score a participant's answers, as automatic creativity measurement approaches struggle.

Automatic Creativity Testing and Randomness Resistance

"Originality can be found in the word salad of a psychotic", write Runco and Jaeger [54]. "A truly random process will often generate something that is original."

Consider an automated test that is based on some creative prompt, such as "How might we tame Dinosaurs?". How are fully random and meaningless answers evaluated by an algorithm? Without a way to understand the meaning of answers, it couldn't distinguish between the answers *I don't know* and *Dinosaur Necromancy*. They both are a response, made out of words. Both answers have similar lengths. But the algorithm can neither understand that one answer is not really a solution nor could it distinguish differing originality between two solutions. In particular, the evaluation of an effectiveness measure (i.e., did the response solve the task in some way) is often ignored or glossed over by researchers of manual and automated study methods alike. Similarly, randomness resistance, that is being able to understand that *blabla 12 12* is not a valid response to the prompt, is rarely discussed when talking about the validity and robustness of automated approaches, maybe because it is rarely a problem in controlled in-person study environments.

Outcome-based vs. Process-based Evaluation

Creative behavior can be seen as a process that may end up in a creative production or outcome. In general, quantitative creativity assessments only consider the productions of study participants, i.e. the finished responses, though some steps towards mixed process-outcome-based scoring mechanisms are being made. For example, Silvia et. al have investigated the effect of judging not individual responses but the entire sheet of paper on which responses were written for an alternative uses task [60], which may give some additional insight on the process with which participants come up with their answers.

Still, generating single measures by analyzing the creative processes is rare, which means that process-based analysis methods have largely remained unexplored as creativity operationalizations.

Software that Understands Creativity

Taking into account the field's current difficulties in measuring creativity objectively and in cost-effective ways, turning towards software solutions that promise to be quick and objective seems to be an obvious solution. If creating such software was an easy task, however, it would be done much more frequently already. There are two areas for creativity assessments in which software can be used: The creativity test or task, and the data evaluation.

One of the challenges of software-based data evaluation is to define data characteristics that correlate with creativity. Though the definition of creativity is constantly evolving, some basic and broad defi-

nitions like Guildford's "novel and effective" (originally called novel and "acceptable") are more or less settled, as they transfer the onus of definition onto the novelty and originality constructs [54]. Just like manual creativity assessments, software-based creativity assessments would have to take into account study contexts such as the population sample and study setup. If written for the purpose of automatically evaluating creativity, it may need to be tailored to the study design and research questions, just like any other part of the study. If that means writing and re-writing software for every single study, the net benefit against manual evaluation will quickly diminish. Once created, such software tools could alleviate the cost of judging data of creative or not-creative behavior, but it may not be more cost-efficient than the manual evaluations researchers are currently doing, as the cost-intensive work of the expert judges merely gets passed on to other cost-intensive work of developing and maintaining software.

On the other hand, software solutions may open up the way for objective and reproducible experiments: Setup, setting, instructions, tangibles, raters, etc. could be enabled to remain consistent across studies and eliminate these circumstantial confounders.

In this thesis, I introduce a generalizable method for reasoning about process-based data generated from creative behavior. This method can be used for many different creativity tests as well as recorded natural behavior. It is randomness-resistant and enables process-based creativity assessments. I show its application in different creativity domains and present software support that evaluates any such creativity data in an automatic, quick, and objective way.

2

RELATED WORK: MANUAL AND AUTOMATED CREATIVITY ASSESSMENTS

There are, broadly, three main categories of quantitative measures in creativity research: Self-report questionnaires, productive thinking tests, and consensual assessment measures [18]. Measures from this trichotomy are not exclusive to one another: Many studies use multiple quantitative measures for their data analysis, combine measures, and create new quantitative measures specifically created to answer a research question. This chapter outlines some of the most used types of creativity assessments for each category, which together give a good understanding of the status quo.

2.1 CREATIVE SELF ASSESSMENTS

Creative self assessments may be the most used type of creativity measure across papers covering creativity and innovation [18]. Approximately 40% of all papers in the creativity and innovation domain use some type of self-report measure. Self-report measures may ask participants to judge their creative qualities or eminent creative outcomes in their lives. Kaufmann divides creativity self-report tests into the categories Activities, Evaluation, Process, and Beliefs [28]. The most commonly-used tests fall into the categories Activities and Evaluation, where the Activities category asks to report on general behaviors and activities that may be considered creative, such as drawing paintings or playing music, and the Evaluation category asks to report measurable creative outcomes, such as having received an award in a creative domain.

In general, creative self-assessments are used most often to support or validate other creative outcomes, though they can be used as primary outcomes as well. Because they rely on the goodwill of the participant to respond honestly, they are not used in high-stakes situations where people may be more prone to misrepresent themselves in a more positively perceived light.

Creative Achievement Questionnaire

The Creative Achievement Questionnaire (CAQ) was first published in 2005 by Carson et. al [11]. Its validation study found an interaction between creative behavior and creative achievement. The CAQ prompts participants with a series of statements relating to real-world achievements in 10 creative domains such as Visual Arts, Creative Writing, or Scientific Discovery. Examples of such prompts are "I have composed an original piece of music.", or "I have received a Scholarship based on my work in science or medicine." Answers are ranked

from more easily achievable (e.g., "I have written an original short work.") to extremely difficult to achieve (e.g., "My work has been reviewed in national publications."). Some questions record not just if something has been achieved, but how often it has been achieved (e.g., "I have won a national prize in the field of science or medicine."). Answers are then scored based on their difficulty. Overall scores can range from 0 to a theoretically unlimited upper bound, depending on how often the participant reports to have fulfilled the achievements on the multiply achievable items.

The CAQ is a well-known self-report test to gain insight into the creativity of a population, as it is easy to administer and report. Though originally reported to have two or three underlying factors that explain the data points generally produced by this test, most researchers now consider it to represent a single underlying factor - which is a decision made from a desire for simplicity [61]. Consequently, data generated by the CAQ may require a more in-depth data analysis on the queried population than a simple correlation analysis of the CAQ's sum score and whatever research item it is being tested against. As such, much of the CAQ's perceived simplicity stems from a choice to not look into missing statistical requisites of the data or what creative constructs are actually being measured.

Creative Behavior Inventory

The Creative Behavior Inventory (CBI) was one of the first developed self-report tests for creativity [25]. In its current iteration it is a short test that focuses less on rare creative behaviors or experiences (such as the ones tested by the CAQ), but more on everyday creativity such as painting a picture or preparing an original flower arrangement [16]. Dollinger's version of the CBI is simpler to interpret than the CAQ, as every question is categorically ranked from 0 (I have never done this in my life) to 3 (I have done this more than 5 times). The revised version is consistently showing moderate correlations (between r=0.3 and r=0.55) with other creativity, personality, and divergent thinking tests [61], but it is less widely adopted than the CAQ, which makes inter-study comparisons more difficult.

Creative Domains Questionnaire

Kaufmann's Domains of Creativity Scale (K-DOCS) is a self-assessment test for creativity that asks participants to rate themselves from 1 (Much less creative [than average]) to 5 (Much more creative [than average]) across 50 different situations and prompts [29]. K-DOCS contains prompts regarding traditionally creative domains and behaviors (e.g., composing an original song, acting in a play) as well as emergent everyday situations and difficulties (e.g., understanding how to make myself happy, teaching someone how to do something). The original test has since been revised by Kaufmann in 2009 [32]. The Revised Creativity Domain Questionnaire (CDQ-R) focuses on

the four creative factors drama, science, arts, and interaction (interaction being a notable domain that is not queried either by the CAQ or the CBI). The CDQ-R has only 21 items, making it comparatively quick to administer, and though its validity is promising it also suffers from low adoption rates compared to the CAQ.

Big 5 Personality Test / Five Factor Model

Developed not for creativity research but personality trait theory, this test places people on 5 different spectra, one of which is *openness to experience*. An example of items of the Big5 openness scale is "(I am someone who) is original, comes up with new ideas". Openness to experience has a consistently strong association with creativity as measured by other tests or questionnaires [52]. As a result, different versions of this personality test - or subsets of them that only regard openness to experience - are often used as a stand-in for other creative self-assessments.

2.2 CONSENSUAL ASSESSMENT TECHNIQUE

The Consensual Assessment Technique (CAT) was first introduced in 1983 by Amabile et. al [5]. Its main purpose is to give researchers a robust method of judging responses to questions that have no definitive answers and where the quality of problem solutions is to some extent subjective. It achieves this robustness by providing the same responses to multiple "raters", who then independently assign scores to each response. Once responses have been rated by each rater, inter-rater reliability can be computed, which is an objective measure of how much the raters agree or disagree with their ratings. Many current creativity tests rely on the CAT to produce results, and could not report objective, quantitative metrics without it. In particular, productive thinking tests as discussed in the next section all require their responses to be evaluated through the CAT to generate their results.

Though the concept of the CAT is simple, creativity researchers have not agreed on its specifics, as Amabile does give guidelines on some specifics which are prone to interpretation. The number of raters, expert level of raters, how high the inter-rater reliability must be for it to be considered reliable, the effect of rater fatigue on inter-rater reliability, which rating scale to use, which statistical method to use for which rating scale, and how ratings are to be aggregated all remain subject for discussion or are being turned a blind eye to [15, 30, 39, 62].

Even against this backdrop, the CAT may be the only method of judging creative responses or products objectively (or, as objective as possible), as there is no obvious better alternative. At the same time, it highlights the theoretical utility of truly objective approaches, such as automatically computed scores.

2.3 PRODUCTIVE THINKING TESTS

Productive Thinking Tests can include divergent and convergent creative thinking tests [22]. Divergent thinking tests provide participants with tasks that are in some way open-ended. No "correct" solutions to the tasks exist. Instead, participants may come up with creative - or not-creative - solutions and answers, which then lead to a number of scores. These tests often try to find tasks that let participants express their aptitude for specific facets of creativity, such as fluency or originality, though some tests have outcomes of a more general nature. The type of creativity that is measured has a strong influence on how difficult (i.e., resource-intensive) task results are to assess once responses have been collected.

Alternative Uses Task

Created by Guilford more than half a century ago, the Alternative Uses Task (AUT), also called Uncommon Uses Task, remains one of the most widely-used creativity tests today. Participants are asked to consider a simple object, such as a brick or a needle. They are then challenged to find creative and atypical ways for the object to be used. Participants are given limited time, often 2-3 minutes, to come up with as many ideas as possible. Commonly, responses of each participant are evaluated for fluency, flexibility, originality, and elaboration, though a general "creativity" measure is reported as well [2, 4, 5].

Fluency and elaboration are objective metrics, that are calculated by counting responses a participant has given and the length of a participant's answers, respectively. Flexibility and originality are subjective measures that require human judgments [5]. As discussed previously, there is only a general agreement on how to use the CAT; specifics are often left to the researchers, which makes comparing AUT-based results across studies difficult.

Though the AUT is used prevalently, it, like many other divergent thinking tests, suffers from conceptual, design, and psychometric shortcomings [6, 17]. One example is the reliance on a single item for which alternative uses are to be ideated, as "inter-item fluency" can have correlations as low as .2 [6]. In other words, results generated by the AUT may be entirely different based on whether participants are asked to think of alternative uses for a paperclip or a brick. The AUT has been in use for over 50 years, but still lacks a single accepted way of analyzing the data and consolidating all responses of a participant into singles scores. For example, sometimes originality scores are generated by considering all answers of a participant and averaging their originality, as opposed to considering only their most original response.

AUT response scoring is time-intensive, which has led to the creation of some alternative scoring methods: "Top Two" scoring lets each participant choose their two perceived most creative responses,

which become the only responses that they are judged on [58]. "Snapshot scoring" provides raters with the original sheet of paper on which answers are written, which are then scored for creativity [60].

Top Two and Snapshot scoring lessen the effort of judging all responses manually. They are on the one hand important steps towards larger-scale creativity assessments and on the other hand, a symptom of a larger problem: Even the most-used productive thinking test, which appears to exist in a sweet spot between efficiency and efficacy, is still very time-consuming to conduct.

Torrance Test of Creative Thinking

The Torrance Test of Creative Thinking (TTCT) is a widely used collection of creative tasks that originally aimed to nurture and enhance creativity among students [65]. It was then used to identify highly gifted children in classrooms and is now widely used as a general productive thinking test [33]. It is well-known for being used in a 50-year longitudinal study, which showed strong correlations between the original creativity score from the TTCT at a young student age and personal achievement (but not public achievement) later in life [55]. The TTCT has been updated multiple times and consists of a verbal and a figural form (TTCT-Verbal and TTCT-Figural). Both forms measure fluency, flexibility, originality, and elaboration as described by Guilford, and consist of five tasks (ask-and-guess, product improvement, unusual uses, unusual questions, just suppose) and three tasks (picture construction, picture completion, and repeated figures), respectively. Similar to the AUT, which is indeed a sub-task within the TTCT, the TTCT is subject to criticisms of rating procedures, as it makes use of the CAT. Its test-retest reliability, which has self-reported coefficients between .5 and .93, has been called into question. Still, the TTCT is one of the most comprehensive creativity/productive thinking tests used today.

Remote Associates Test

The Remote Associates Test (RAT) is a task developed as a general creativity test [44]. It prompts participants with groups of three words that are in some way related. The task is to find a fourth word that connects to the other three. For example, the three words "cream", "skate", and "water" may be connected by "ice". Answers to the RAT items are either right or wrong, which means its evaluation is very easy and possible to automate, but its place in creativity research is debated: The RAT has been criticized for measuring language proficiency to a higher degree than other similar creativity tests [67]. Its scores seem to correlate with other tests that measure general creative thinking, convergent thinking, and analytical processing, which leads to an ongoing struggle to understand what its outcomes actually measure [37].

2.4 AUTOMATICALLY ASSESSED PRODUCTIVE THINKING TASKS

Advances in the algorithmic understanding of language have led to advancements in the automation of language-based creativity assessments as well.

SemDis

SemDis is an automatic creativity test that you can take in your browser and get a creativity score within minutes [8]. It combines the AUT with a new concept, semantic distance. The semantic distance of two words or concepts expresses how closely they are being used together in some given semantic space. With advances in such semantic spaces, expressing the distance between words has been reduced to a mathematical or graph theory problem. For example, the two words "brick" and "masonry" would exhibit a low distance in most semantic spaces, as they are contextually similar and therefore used often together. By comparison, "brick" and "bark carver" are more distant concepts. Here, the latter example would get a higher creativity score.

SemDis prompts users with an item typical for the AUT, such as "brick" or "paperclip", and asks them to come up with "six creative uses" for that object. It then cleans the responses of filler words and generates an average semantic distance. There are five different semantic spaces to compare the word distances against, which is a choice that can influence the results strongly. Preliminary studies have shown good internal validity of data generated by SemDis studies in particular, should the claimed correlation of $= 0.91$ between AUT results and a SemDis model that combines their five semantic spaces hold true across different studies.

Divergent Association Task

The Divergent Association Task (DAT) [45] is a web-based divergent thinking test that has a very simple and open-ended prompt: *"Enter 10 words that are as different from each other as possible, in all meanings and uses of the words."* It then takes back some of that simplicity and introduces some rules for what kind of words to use (nouns) and what kind of words to not use (specialized vocabulary or technical terms). Using semantic distance, the scores are the average pairwise semantic distance of the first seven valid responses. Using only the first seven valid responses is a clever way to introduce redundancy in the response data, which allows the test to generate valid scores in the face of typos and unknown neologisms which otherwise would lead to an impossible-to-calculate average. In its early studies, the DAT has shown good convergent and internal validity, with good correlations to the AUT and other creativity tests. With its open-ended prompt it addresses the stimulus dependency problem of other divergent thinking tasks, and with its response redundancy, it circumvents a classic

pitfall of automated language analysis without much programmatic effort.

2.5 PROCESS ANALYSIS

Shah et al. measure novelty in ideation processes by having expert raters decompose given ideas into functions that the ideas satisfy within their "design space" [57]. For example, the "thrust method" may be a function assigned to the design space of vehicles. Each function is weighted based on perceived importance and potentially re-evaluated when the idea changes during the ideation process. To rate these functions, raters create a subjective "universe of ideas" of existing ideas or concepts that satisfy the functions to compare the new functions against. This manual and extremely time-intensive approach approximates the automatic approaches described in this thesis the closest: Both approaches record a creative process ("ideation" for Shah et al.), and both approaches compare a list of recorded productions to some created corpus ("universe") of productions to measure novelty. Note, that Shah et al. differentiate between "novelty" and "variety" - that is, unexpectedness and exploration of a solution space - as opposed to the definitions in this thesis, where this distinction is not made.

Peeters et al. introduced a further stratification of this novelty analysis without the creation of some universe of similar ideas, citing that this creation is too time-intensive [49]. Instead, they weigh all ideas differently depending on the type of novelty ("original, adaptive, or variant") that was observed.

Part II

THE SOLUTION SPACE

3

OPERATIONALIZING CREATIVITY: CREATIVITY IN DATA

There are many different methods of operationalizing creativity or facets of creativity. Creating creativity measures with good validity, i. e., measures that measure the creative facet intended to be measured reasonably well, requires first a good understanding of what kinds of creative constructs exist. Listing creative constructs can never be exhaustive, but this chapter will describe some of the most common definitions of creative constructs, definitions of measures intending to capture these constructs, and some of their common operationalization methods. Then, a novel operationalization method for creativity in creative processes will be presented, which is based on the standard definition of creativity and is valid for human behavior data.

3.1 CREATIVITY: DEFINITIONS AND OPERATIONALIZATIONS

Guilford, in his famous 1950 presidential address, defined the concept of creativity as follows: "Creativity refers to the abilities that are most characteristic of creative people" [21]. This tautology may be the earliest "modern" definition of creativity, which remains fittingly simple - and impossible to operationalize outside of ancient Greece with its similarly simple creativity definition (cf. Chapter 1). Operationalizing this tautological creativity definition is impossible, so other definitions for creativity are needed.

The standard definition of creativity can be expressed as *novelty* plus *effectiveness* [7, 13]. This is the most general contemporary creativity definition. From it emerged other creativity definitions.

For example, *Little-* and *Big-C* creativity are often used to describe the effects that being creative has on our lives [32]. Self-report questionnaires as described in Chapter 2 try to measure this operationalization of creativity. Little-C (everyday creativity) and Big-C (rare great creativity) are often mentioned in research that measures how human lives are affected by creativity and creative action. Little-C is defined as everyday creativity. Though these creativity definitions are firmly established as an important part of the creativity landscape, they will be mostly ignored in this thesis, which focuses on the way one can measure not the effects of creative action, but creative action itself.

For *in the moment creativity*, the creativity community often refers to two different modes of thinking, which we employ during creative problem-solving: Divergent Thinking (DT) and Convergent Thinking (CT). DT describes the ability to expand an action space with ideas that may or may not relate to the problem space and may or may not lead to effective solutions. CT describes the ability to select ef-

fective strategies or ideas within that action space. Intuitively, divergent thinking can be described as thinking of ideas, while convergent thinking can be described as choosing the best idea(s). In reality, there is likely not a two-phase process but a constant interplay of both modes of thinking: Convergent thinking steers divergent thinking, and divergent thinking may further be influenced by disrupting previous ideas that were "converged on". Productive thinking tasks (cf. Section 2.3) are prompting participants to come up with answers which require divergent and convergent thinking, but both these facets are difficult to measure with the typical productive thinking tests that assess only the outcome of creative thought. Though divergent and convergent thinking are important to describe creative thinking, they are generally not measured either quantitatively or qualitatively. Instead, the creative outcomes or products (i.e., the responses) of the participants are assessed by using other measures that capture facets of creativity and together may give a good indicator for the creative potential of a person. The three most common measures in such productive thinking tests are *fluency*, *flexibility*, and *originality* [23]. This ubiquitous "triad" of creative measures is sometimes expanded by the *elaboration* measure and sometimes condensed to a simple measure of *creativity*.

Fluency

Of the fluency-flexibility-originality triad, fluency is the only measure that is objective. It describes the number of ideas that a person could come up with in some amount of time. Being able to come up with many ideas can be helpful when trying to come up with creative solutions, but fluency alone, that is, the countable number of ideas, does not mean that any highly creative idea has been produced, as ideas could be repetitive, obvious, or even nonsensical. Since fluency is a simple count of all answers, this measure does not need expert ratings, but it may require some oversight to weed out joke or nonsensical answers.

Flexibility

Flexibility is domain fluency. It describes the number of topic domains that a participant's responses traversed. Thus, it offers some granularity that fluency doesn't offer: For example, a person's responses for uncommon uses for a paperclip might be "earring, ring, nose piercing", which are all broadly in the same category, jewelry. These responses would get the same flexibility rating as the single response "ring". Judging flexibility requires some method of separating topic domains that are broad enough to encapsulate every possible answer but specific enough to not group topically distinct answers together. Since domain definitions may vary or be subjective, this measure requires expert ratings.

Originality

Originality is part of the standard definition of creativity (e.g., creativity = originality + effectiveness) and is often used synonymously with creativity.

As originality remains difficult to define, many different ways of evaluating it - that is, methods of translating distinct test responses to a single value representing this characteristic - have been developed. The most common ways to generate originality scores are 1) the CAT, 2) rarity analysis, and 3) semantic distance analyses. The CAT (1) is a method that employs multiple human raters to subjectively rate all responses and generate average ratings for them (cf. Section 2.2). Rarity analysis (2) is a method that counts responses across the entire response body (i.e., not per person but per population), then scores rare responses higher than responses that have been given by some or many other participants. It is a mathematically objective measure of originality that has been criticized for being highly dependent on the number of participants, as fewer participants will automatically lead to higher originality scores [19]. Semantic distance (3) is a way to describe the closeness of words with regard to their meaning or their use. Semantic networks such as wordnet[1] are one way of calculating semantic distance. They are networks that group words based on their semantic similarity. For example, *teacher* and *educator* would generate a low distance score on most semantic networks. Another way to score semantic distance is Latent Semantic Analysis (LSA), which scores word distance based on how often words are being used together, which means that two very different words that are often used in similar contexts may generate low scores. For example, *teacher* and *grade* have a high semantic relatedness, and would thus generate a low score. Semantic distance has been used as a promising tool for automatically generating creativity scores [8, 9, 41, 45]. Semantic distance scoring is strongly influenced by the corpus on which the network is based. For example, a network generated by using Facebook comments as a corpus may generate different scores for word pairs than a network generated from ancient philosophical texts. Selecting the right corpus for use in any creativity test is an important step in ending up with valid data.

All constructs mentioned above use the productions of creative behavior as input to generate ratings, but none evaluate the creative process leading up to those productions. In the following chapters, I will introduce a novel method that explores this gap.

3.2 CREATIVE BEHAVIOR IN DATA

This section defines terms for understanding the creative process from a data perspective, particularly to distinguish the process as it is defined in creativity theory from the process as it is defined in computer science. Processes and data in computer science are intimately

1 http://wordnetweb.princeton.edu/perl/webwn

connected to general computer science. As creativity theory has developed independently from computer science, there exists some semantic overlap between definitions of the creative process in creativity and a computer science domain: Process mining. This means that naming conventions suggested in this chapter will have some semantic overlap with naming conventions from process mining, the most important semantic distinction being the *process*, which has a very different definition in creativity and process mining.

Process Mining vs. Creative Process

The creative process can be reconstructed as a sequence of mental and/or behavioral actions. It is successful if it results in at least one novel and effective production or outcome. This resulting production, in terms of creativity theory, can be any solution to a problem or design challenge. It may be tangible or intangible and can take on a variety of shapes, such as a physical good, a one-time behavior, a strategy, a technique, a service, or a process. By contrast, the process mining process is a holistic view of task execution logic across a full workflow log, which may consist of temporally and causatively disjunct actions and action sequences [1]. Intuitively, a creative process is a single series of related actions, while the process in the context of process mining is a model or summary of many similar or identical series of actions in some given data record. Process mining's pendant to the *creative* process would generally be called a "trace" or, specifically within a workflow log context, a "case". The creative "action" and process mining's "activity" represent the atomic elements of the respective processes or cases, see Table 1.

Table 1: Semantic differences and similarities of the creative process and process mining domains

CREATIVE PROCESS	PROCESS MINING
-	Process
Process	Case / Trace
Action	Activity
Event Log	Event Log / Workflow Log

Though my approach to data analysis uses some data structures that appear similar to structures used in process mining, it uses terminology as it relates to creativity and the creative process, unless otherwise specified. Most notably, a "process" refers to one instance of a *creative* process of one participant or one team of participants.

Creativity and its Representation in Data

Since processes are comprised of sequences of actions, they lend themselves well to being recorded. Using such records, I build on

the standard definition of creativity to operationalize *novelty* and *effectiveness* in creative processes.

The first step is to measure novelty (automatically) within data generated by human behavior. Any creative production would clearly have to be "different" from established productions to be considered novel. In other words, every novel production must exhibit some level of *differentness* to some known set of previous ideas. Differentness can then represent how different any production is compared to another production or some corpus of productions. Note, that though every novel production has to be different from established productions in order to be considered novel, differentness would not always indicate novelty: For example, random behavior would generate processes of high differentness, but the processes would not be considered "novel" by a human rater in the same way that a *production of novelty* would. Unlike human raters, an automatic differentness measure would thus lack the ability to differentiate novelty from randomness.

Therefore, I include in the language for tracing novelty not just how differentness is defined in data, but also how to determine when this differentness translates to actual novelty. A solution to this problem may be found in the standard definition of creativity as well: By considering not just the novelty of an idea but also filtering ideas by their effectiveness, automatic rejection of most random productions could be possible. This "randomness resistance" will have implications for the automatic and large-scale creativity assessments discussed in later chapters.

3.3 DEFINING A LANGUAGE TO REASON ABOUT CREATIVE BEHAVIOURAL DATA

This chapter explains *string distances*, and builds on them to define the *creative process*, *process differentness*, the *average process differentness*, and the *C-Score*. The latter will be the main metric for quantifying creativity based on human behavior data.

The Levenshtein Distance

The Levenshtein distance is a string metric that can express the difference between two strings, a concept that is also called the edit distance. It is defined on strings of arbitrary length and is a way to measure the number of operations required to transform one string into the other. For example, the Levenshtein distance between the words "house" and "mouse" is 1. This can be expressed as such:

$$\mathrm{lev}(\mathrm{house}, \mathrm{mouse}) = 1$$

The set of operations allowed in this metric is comprised of substitutions, insertions, and deletions of single characters. In the previous example, a single substitution is needed to transform house into mouse (or vice versa), which results in a Levenshtein distance of 1. Similarly, "house" and "houses" require a single deletion or insertion (it's possible to delete the s of houses or insert an s after house). In order

to transform "mouse" into "mice", three transformations are needed: Substitute o with i (miuse), substitute u with c (micse), and delete the s (mice):

$$\mathrm{lev}(\mathrm{mouse}, \mathrm{mice}) = 3.$$

Though the Levenshtein distance is formally defined on strings, it is not difficult to understand it not as a string distance metric that compares characters for equality, but as a generalized distance metric that compares any elements for equality. I create and use such a generalized Levenshtein distance metric to calculate the element-level (i. e., action-level) distance of two creative processes (see Section 5.6 for implementation details).

Processes and Process Distance

The main creative structure used for the proposed analysis is the *process*. Whenever people engage in creative thinking they also undergo a creative process. Guilford described the creative process as any sequence of thoughts and actions that leads to novel, adaptive productions [21]. There is no good way to track sequences of thought as per Guilford but a good ability to track observable actions. In the following process definition, I focus on a *recordable* process. Note also, that no assertions about a process being creative or not are made yet. Thus, the process definition is simply any sequence of actions that leads to a production or outcome.

Definition 1. *(Creative Process) A process* p *is any sequence of actions* $(\mathsf{a}_1..\mathsf{a}_n)$ *that lead to a production or outcome.*

This very broad definition may encompass rather many potential creative processes, so instead of staying abstract, I introduce a novel simple productive thinking task. The task instructions are:

> Create a delicious, creative ice cream combination that you could enjoy in a cone.

I call this task the Ice Cream Imagination Task (ICIT). This imaginary task will provide some more concrete examples for the remainder of this chapter and thesis. When performing it, participants will likely express their actions as flavors or scoops, see Figure 1.

Figure 1: Example of a process emerging in the ICIT, comprised of three actions (scoops).

Following the standard definition of creativity, I map the novelty construct to processes by formally defining the previously mentioned

differentness, for which I use the special element-level Levenshtein distance metric. This metric can be calculated as if calculating string distances but is now redefined as a function that works on processes.

Process differentness is defined on process pairs, see Figure 2. For example, the ICIT processes p1 = (Chocolate, Vanilla, Pecan) and p2 = (Chocolate, Vanilla, Lemon) have a distance of 1, and the processes p3 = (Lemon, Lemon, Lemon) and p4 = (Lemon) have a distance of 2.

Definition 2. *(Differentness of Process Pairs) Given two processes* p1 *and* p2*, their differentness* $\delta_p(p1, p2)$ *is the action-level Levenshtein Distance of processes* p1 *and* p2 *so that* $\delta_p(p1, p2) = lev_p(p1, p2)$.

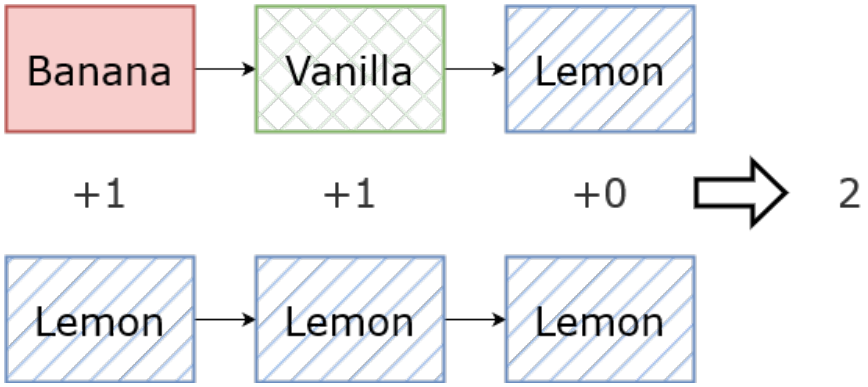

Figure 2: A distance calculation for two simple processes.

Once persons (i. e., study participants) generate some list of processes P, such as a few ideas for ice cream flavor combinations, each person's average process differentness can be evaluated by determining the pairwise action-level distances of all processes generated by that person. Then, the arithmetic mean of those distances is calculated. The resulting scores equal the average process differentness of all process pairs generated per person, see Figure 3.

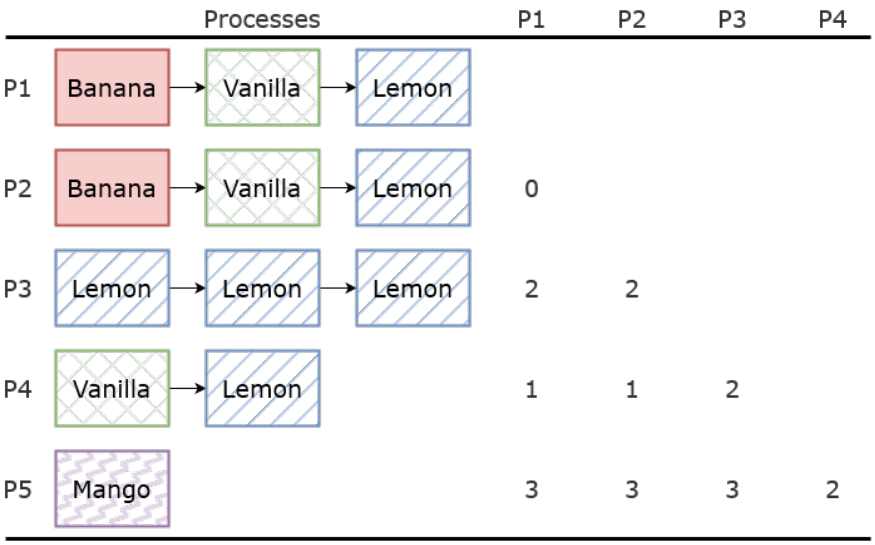

Figure 3: A pairwise distance calculation for 5 processes.

Definition 3. *(Average Process Differentness) Given a list of processes* P *that has a length of* N*, the average process differentness* $\overline{P}(P)$ *is the average differentness of all process pairs in P so that*

$$\overline{P}(P) = \frac{\sum_{1 \leqslant i < j < n} lev_p(P[i], P[j])}{N(N-1)/2}$$

A high average differentness may indicate that creative behavior was observed, though any validity of such an observation is of course dependent on the underlying task and action measurements. In the context of creating a construct for measuring creativity based on its basic definition, a measure of success is still needed; so far, the average process differentness $\overline{P}(P)$ is only a measure for "average novelty". A way to differentiate processes that were effective from those that were not effective is needed in order to capture creativity's "success" criterion. I do this by introducing such a success parameter, which must be defined within the space in which the processes are performed. While differentness as defined here is an objective metric, the definition of success depends on its context, for which domain expertise is needed. An in-depth discussion on designing problem-, action-, and solution spaces for real-life studies can be found in Chapter 6.

When conducting the ICIT, scoop flavor combinations may be considered successful if the resulting ice cream cone is fully consumed by its recipient, but it could also be considered successful if it ends up fitting a specific color scheme when melted, if it can be sold for more money than it cost to make, or if it is judged delicious by three expert raters. An effectiveness definition needs the knowledge of domain experts, without which valid results cannot be obtained. After effectiveness is defined within the context, the average process differentness of all effective (i.e., successful) processes of a person or group of persons can be calculated. I introduce this measure as the C-Score.

3.4 THE C-SCORE

The previous sections laid the foundation for the definition of the C-Score. The C-Score is the average process differentness of all effective processes within a list of processes. It operationalizes effectiveness by only considering effective processes out of a list of processes, and novelty by calculating the average process differentness within that list. Its bipartite definition mirrors the bipartite creativity definition of *effectiveness* plus *novelty*.

Definition 4. *(C-Score) Given a list of effective processes* $P_e \; 2 \; P$ *the C-Score is the average process differentness in* P_e *so that* $C\text{-}Score = \overline{P}(P_e)$.

The C-Score's operationalization of effectiveness is binary: A process is either successful or not. No degrees of success exist or are considered, which is why it is important to clearly define what success means in different contexts. Its operationalization of novelty is not binary: It is expressed as a continuous variable similar to metrics from other established creativity tests (see Chapter 2 and Section 3.1).

3.5 REPRESENTING CREATIVE BEHAVIOR IN DATA

With these definitions in place, a data structure is needed that can represent the actions multiple people are performing toward goals. This data structure must be able to record for each action four things: 1) a way to distinguish who performed the action, 2) a way to record within which process the action was performed, 3) what type of action was performed, and 4) when the action was performed. Thus, I define actions in the context of the creative process as tuples of, at least, the following four attributes:

- person id. Identifier for the person (i.e., participant) that induced the event.
- process id. Identifier for the process to which the action belongs.
- action. Name or type of action.
- order. Counting variable (such as a timestamp) that records when the action took place.

Actions and events are conceptually the same thing considered from different perspectives, and a record of many of such actions will be equivalent in structure to typical event logs, such as event logs used in process management or software profiling.

Definition 5. *(Event) An event* e *is a record of some occurrence, represented as a tuple of event attributes (person id, process id, action, order, . . .), with . . . representing any number of optional meta-attributes.*

Definition 6. *(Event Log) An event log is a data log* L *which records events* $e_1..e_n$.

Event logs are records of occurrences that let us reconstruct the (creative) processes that have been performed at some point. They are valuable tools in many different areas of information technology and information systems. Though the specific format of event logs can be as varied as their use cases, generally all event logs will contain some event names and the time of event occurrence. Then, depending on their purpose, event logs will have additional information such as who is responsible for an event and some context during which the event occurred. The definition presented here is closely aligned with event log definitions from the business processes and process mining fields, which record at least process id, action, and order, though there they are generally named case id, activity, and timestamp, respectively [1]. Most of the time, a person id (*resource* in business processes) is recorded as well, which means that most business process event logs could be analyzed for process differentness as described in the previous chapter. However, generating not just average process differentness, but a C-Score (i.e., average *effective* processes differentness) would require defining which, if any, business traces and activities represent creative behaviors. This would have to be determined through creative space definitions within the action

space of the business process, but could theoretically be done on any such event log. Event logs are the data input for the algorithm calculating the C-Score, the Delta algorithm.

3.6 THE DELTA ALGORITHM

Delta is a novel algorithm for creativity scoring in behavioral data. Based on an event log that contains information about people's creative (or not creative) processes, Delta will generate the average differentness of the processes generated by a person or group of persons (see also Section 3.3). Delta does *not* encompass the filtering for effective processes, which means depending on the input event log, its output is either the average process differentness in the case of an unfiltered list of all processes or the C-Score in the case of a filtered list of all successful (i.e., effective) processes.

Holistically, Delta splits an event log by participant, then calculates each participant's average process distance, i.e., the average differentness of each participant's processes.
Input: Any event log L of events recording at least person id, process id, event name (action), and some ordering.
Output: A table that shows for each distinct person id a number *2* R, showing their average process differentness.

Listing 1: Delta algorithm

```
Input: Event Log L = Table(PersonID, ProcessID, EventName, Oder)

OUT = Table(PersonID, AverageDifferentness)
ParticipantEvents = split L by PersonID
For Events in ParticipantEvents:
    Processes = extract processes from Events by ProcessID and
        Order
    N = Number of Processes
    Distances = Empty array
    While 1 \leq i < j < N:
        Distances[i] = lev\_p(Processes[i], Processes[j])
    TotalDistances = Sum of Distances
    AverageDistance = TotalDistance / (N(N - 1) / 2)
    Row = Current PersonId, AverageDistance
    Add Row to OUT
Return OUT
```

In Chapter 5, an implementation of Delta will be elucidated, and extensions to it are described. One such extension is the filtering of processes for effective processes (thereby enabling C-Score calculations), which will complete the mapping of the previous definitions to a real-world implementation.

4

ALGORITHMS FOR CREATIVITY MEASURES IN BEHAVIORAL DATA

This chapter details ways to reason algorithmically about different creative constructs in the context of recorded creative processes as defined in Chapter 3. The C-Score itself was designed to encompass the two specific aspects of the standard definition of creativity, as discussed in Section 3.2. But there are many other constructs that aim to capture facets of the creative space, most of them being measured on productions (i.e., not on processes). This chapter will outline how these constructs may be measured in process data.

4.1 CREATIVITY: C-SCORE

The average effective differentness maps to the standard definition of creativity. The C-Score is explained and defined in Chapter 3 and is included in this chapter for the sake of getting a more comprehensive picture.

4.2 NOVELTY: "D-SCORE"

Average process differentness is the main outcome of the Delta algorithm, as defined in Section 3.6. As opposed to the C-Score, it lacks a "success" component in its definition, which is another way of saying that it strongly maps to novelty: Participants that have many novel - successful or unsuccessful - ideas, can be expected to try many different strategies in a given situation, which would be reflected in a high D-Score.

4.3 FLUENCY: NUMBER OF SUCCESSFUL PROCESSES

Fluency is the construct of quantity. It is probably the only such construct that one could natively automate for the productions of classic language-based productive thinking tests, such as the AUT. A simple version of such a fluency algorithm would count all answers a participant has given and from that correctly get the participant's fluency score. The only thing one may miss from that algorithm is a validation step that checks all answers for joke answers or random words and letters - which suddenly makes the automation non-trivial. Before describing a process-based pendant to this algorithm it may be reiterated here, that "success" in this context does not equal creativity but the successful completion of a process. Automating fluency scores on well-defined processes does not have a randomness problem, as long as an effectiveness measure exists: As the success of processes is defined by the task, it can be constructed to have randomness resis-

tance. That is, random behavior will not result in high fluency scores because it can be filtered out by the success definition.

4.4 ELABORATION: PROCESS LENGTH

Elaboration is a construct that does not consider the idea itself, but the detail in which the idea is presented. Two identical ice cream combinations from a conversational version of the ICIT may have entirely different elaboration scores, depending on how participants may describe their creations. For example, "Chocolate-chip vanilla." and "chocolate-chip would taste really good with vanilla and also give you a little crunch!". In language-based creativity assessments, elaboration is most often simply the number of words used to convey an idea. Then, mapping words of an idea to actions of a process comes naturally: By considering not failure or success, nor process differentness, but instead process length as the process-based metric for elaboration. Note, that process length could be expressed in multiple ways, such as the average process length, or maximum process length. Currently, C-Tracer computes both.

4.5 VARIETY: NUMBER OF DISTINCT ACTIONS TAKEN

The variety measure explains the breadth of explored space within a domain, and was used as a metric in the process analysis concept of Shah et al., see Section 2.5. It could be seen as a process-compatible version of flexibility. Flexibility expresses across how many topic domains answers given by a participant are spread. Is an often-used metric in production-based creativity tests, but it may be a bad fit for process-based creativity, as a process will generally be *within a domain* already. Instead, variety measures the amount of different elements used to solve problems within the domain. That makes it easy to compute: For the Ice Cream Imagination Task, it would simply be the number of different flavors used by each participant during cone ideation.

4.6 ORIGINALITY: CONTEXT-AWARE PROCESS DIFFERENTNESS

Originality is a metric that, unlike the other "traditional" creativity metrics, requires context to be evaluated correctly. Expert raters of originality regard the entire response body as well as the study population to generate their scores. One approach that attempts to make this measure more objective is rarity analysis, which needs to count distinct responses of all participants as well as those responses' frequencies. Another such approach uses semantic distance, which requires some corpus in which distances have been defined. Both of these approaches can technically be automated, with a trivial algorithm for rarity analysis, or more sophisticated algorithms for semantic distance analysis, see Beaty et al. [8]. Neither approach solves the problem of response body and study population biases.

For example, using a brick as a meat tenderizer could be considered rather original when asking school children for uncommon uses, but less original when asking cooks or butchers, as the latter group may have familiarity with the conceptual space of treating meat. In other words, the originality of any creative product has little meaning without consideration of the population. This means, to compute some originality score, the whole response body (e.g., all answers given by the queried population of study participants) may be needed as context to be regarded by an automated algorithm.

Regarding process-based data, such a context consideration may be possible by using elements already introduced in this thesis. Two such algorithms are outlined below.

Context-Aware Process Originality 1

This is a two-step algorithm: First, compute the average process for each participant, so that each participant gets assigned one action sequence that reflects their behavior the best. This could be done by calculating process distances for all a participant's processes and taking the process with the lowest distance to all their other processes, or by using the mode of actions for each possible action index. An example for the latter: The three processes P1 = (Vanilla, Chocolate), P2 = (Vanilla, Chocolate, Chocolate), and P3 = (Chocolate, Vanilla) would result in an "average" (mode) process P_{mode} = (Vanilla, Chocolate), as index 1 of the processes was assigned vanilla twice and chocolate once, index 2 was chocolate twice and vanilla once, and index 3 was empty twice, and chocolate once. After an average process has been assigned to each participant, the average process distance to all other participants' average processes can be computed. Intuitively, this approach would compare how original each participant's most common strategy was compared to the population.

Context-Aware Process Originality 2

This version of the algorithm is simpler but computationally much more intensive. Instead of just calculating average successful process distances within the processes of each participant, compare all of each participant's processes to all other processes of all other participants. The runtime of this approach would increase multiplicatively by the number of study participants, which in the real-world study of immune defense on more than 100 participants (see Chapter 6) would take up to ten hours on a simple web server. It may however provide much more nuanced originality scores than its counterpart above.

Part III

THE ACTION SPACE

5 C-TRACER

C-Tracer is a process analysis tool that implements the Delta Algorithm and extends it to perform various process analyses on event logs generated from human behavior. It extends the landscape of creativity analysis methods with automatic and fast process-based assessments of creative behavior. By providing an interface that allows the configuration of Delta without writing code or manually executing scripts, creativity experts with little scripting or programming knowledge may make use of the Delta algorithm and C-Tracer's additional measures. Given an event log, C-Tracer calculates for each participant recorded 1) the C-Score as defined in Section 3.3; 2) the D-Score, a value for the average process distance of all processes; 3) process fluency, the total number of effective processes; 4) process elaboration, the average and longest process lengths; and 5) the total number of actions (see Chapter 4).

5.1 REQUIREMENTS

Holistically, the functional requirements for C-Tracer are:

1. Calculate various creativity-related measures on process-based behavior data.
2. Validate scores from 1. with an auxiliary data source.
3. Be data source agnostic. I. e., it should work on data from many different kinds of processes.
4. Preserve the privacy and confidentiality of participants.

On 1: The theoretical background on this requirement is discussed in Section 3.3. This requirement led to the creation of C-Tracer's algorithmic core, the Delta algorithm, and is discussed in-depth in Section 3.6.
On 2: Many data sources could theoretically be used as input for C-Tracer and get mathematically correct results. Whether these results have any relevance regarding the measurement of creativity depends on data definitions as well as data and construct validation. To aid in the data validation process, C-Tracer should provide not only generate scores, but help researchers validate these scores with auxiliary data. This is a soft requirement, as data validation processes are supported by other software as well. However, the existence of the data validation step has an inherent value: It communicates with the user that C-Tracer's results are only meaningful when used correctly (i. e., when they are validated).
On 3: Data source agnosticism is needed in order for C-Tracer to work

with, theoretically, any process data. In practice, this means that C-Tracer does not depend on specific keywords, data sizes, or data domains. Instead, a well-defined data format is needed to which data sources can be made to fit. This data format is discussed in Section 3.3 and is the only input format for C-Tracer. Many data sources can be transformed into such a format. Some of those transformations are shown in Chapter 6, as well as guidelines for creating new creativity tests intended to be used with C-Tracer.
On 4: Data privacy (i.e., that data cannot be accessed by secondary parties) and confidentiality (i.e., that no identification of persons is possible from the data) should, ideally, not be part of these functional requirements, as they depend on the actual data being used. With correctly anonymized data, even a hypothetical data leak would not infringe on the privacy of any person whose behavior is recorded in the data. Since not every user of C-Tracer will have the expertise to anonymize data or be able to accurately assess when not only privacy but also confidentiality is being infringed upon, C-Tracer should handle only ephemeral data and never record any data itself.
Conceptually, these four functional requirements could be fulfilled by a local script that takes data sources and a configuration file or string as input and then outputs creativity scores as well as optional validation data. However, there are additional needs regarding accessibility and usability, leading to the following non-functional requirements:

5. Usable by people with no programming background.
6. Easily accessible.
7. As fast as possible.

On 5: An early user test that focused on usability and accessibility showed that users without explicit knowledge of executing script-level applications struggled to execute early versions of C-Tracer, which at the time had been built as a command-line tool. User feedback mentioned the lack of UI elements as problematic, to the point of being unable to execute C-Tracer at all. Meant to be a tool for creativity researchers who may not be used to software without a UI, a non-functional requirement of C-Tracer is its presentation and how its underlying concepts are communicated to users.
On 6: C-Tracer should be easily accessible and have no or a limited installation process so that the hurdle to use it is decreased and its ease of use is increased. One way of achieving this is making C-Tracer browser-based since browser-based apps are accessible from most computing platforms and require no installation.
On 7: C-Tracer should be as fast as possible so as to make it as responsive as possible and create the possibility for rapid testing and analysis turnarounds.

5.2 DEVELOPMENT

C-Tracer was developed using iterative needs-based development, diverse and rough prototyping, rapid testing, and interdisciplinary

team collaboration. Thus, some of its parts that are discussed in this and later chapters may not have existed at times during development and while C-Tracer was already in use. One such example is the number of measures that C-Tracer calculates now, whereas it calculated only the C-Score during early development. In part, this development occurred semi-concurrently with the video game "Immune Defense" [36], which is designed to record in-game events and output cohesive event logs (cf. Section 6.5). Immune Defense provided valuable learnings. Most notably, it provided a realistic upper limit for data complexity, which means its data was regularly used for testing new features. Many of the features C-Tracer has now, were inspired by the data provided by Immune Defense, as it was used across the studies described in Chapter 6. Once C-Tracer was in a robust state (i.e., fully data-agnostic and with a usable UI), it found use in other projects, such as text-, and divergent thinking-based data records. The use of C-Tracer for these different data sources is detailed in Chapter 6. While it was in use, incremental changes to its functionality, as well as improvements to its run-time were continuously made.

5.3 ARCHITECTURE

To make the Delta algorithm accessible to end users without knowledge of executing scripts, its configuration needs to be easily achievable through a user interface. Conceptually, this means C-Tracer's functionality is as follows: Take a valid comma-separated values (CSV) file as input (see Section 3.3), and then allow users to adjust relevant configuration parameters needed for executing Delta and C-Tracer's other analyses. After the analysis is complete, show a result table to the user, which can optionally be downloaded. Then, allow the upload of a secondary validation CSV file, and quickly produce a second table with a statistical analysis of the relationship between C-Tracer's creativity measures and the measures contained in the secondary validation file.

Figure 4 shows conceptual interactions between user, UI, and server. From a user perspective, using C-Tracer is a three-step interaction: 1) Upload a CSV file, 2) configure parameters, and 3) start the analysis. Optionally, a fourth step, downloading the result, can be performed.

During typical usage, C-Tracer and users traverse six logical states, see Figure 5: C-Tracer starts in a ready state which only allows the uploading of a CSV file. Once a CSV file has been uploaded, C-Tracer changes states and allows the user to configure all additional parameters needed for execution. This mainly means mapping column names to their function (e.g., selecting the "participant" column as the column which contains the participants' IDs) and selecting which analyses C-Tracer should perform. Next, C-Tracer enters a "running" state, during which a loading bar is shown to the user, showing which participants and process pairs are currently being evaluated. After completion, C-Tracer enters the "results" state, in which a results table

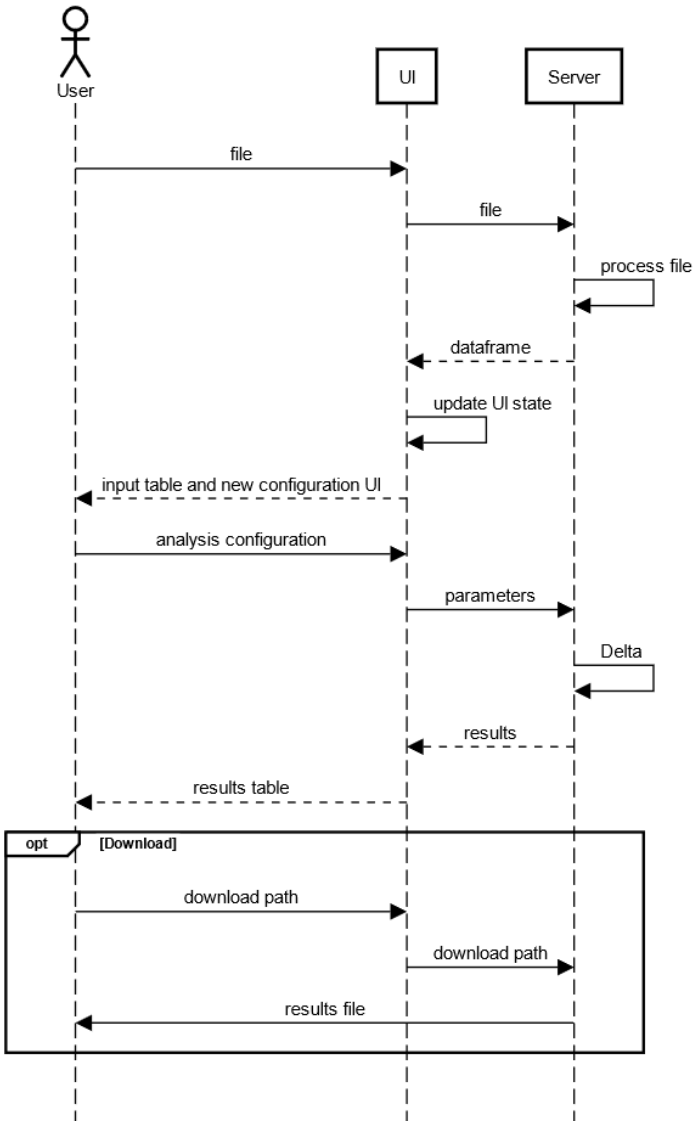

Figure 4: Interaction between user, UI, and server of C-Tracer

is shown. Users can download this table, re-configure and re-execute the analysis, or upload a different CSV file from this state. Two additional states are traversed should the user want to validate the analysis results with a auxiliary CSV file: The first allows configuration of the new file, the second visualizes correlations in a long-format table.

5.4 EXTENSIONS TO THE DELTA ALGORITHM

Most event-based data, should it be in a format that is incompatible with Delta, can easily be preprocessed to make it compatible. During development, some common log characteristics that needed such preprocessing crystallized:
1) Event logs can contain either both effective and non-effective processes or only effective processes. This means one of two things, respectively: If all processes are already filtered to be effective, the Delta algorithm will natively calculate the processes' C-Score. Otherwise, if not all processes in the event log are effective, the removal of all non-

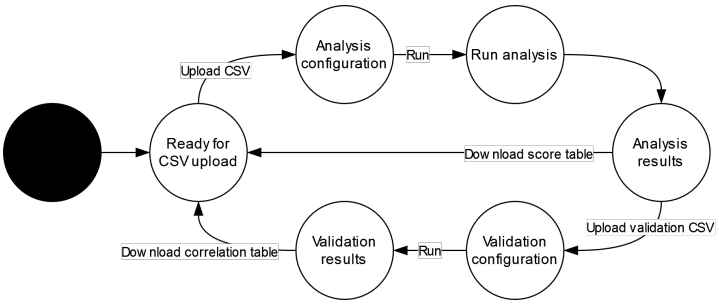

Figure 5: Conceptual states of C-Tracer during typical use.

effective processes would be necessary in order for Delta to produce the C-Score. Both types of event logs are common, so users may need to be able to configure C-Tracer to work with either type. Therefore, functionality to filter unsuccessful processes was added to C-Tracer as a preprocessing step.
2) Depending on the way participants whose actions are recorded in an event log are allowed to explore an action space, multiple processes might be related and lead to solving one singular problem. In the recurring example of the ICIT (i. e., creatively combining ice cream flavors), a participant might start to make two ice cream cones and then decide that both cones should be combined to create an even better ice cream creation. Logically, even though each cone might originally be expressed as a single process, later they turn out to be parts of the same process. In the Immune Defense game, processes are often connected in this way and may build "trees" of connected processes, which result in a single logical process, see Figure 6. In event logs, this might be represented by using an extra attribute field that refers to another process relating to the current process. Enabling the analysis of such process combinations by combining processes before performing the Delta algorithm has been of value in analyzing the more complicated event logs created by software such as the Immune Defense game. Therefore, C-Tracer natively combines related processes if provided with the data field which contains these "connected process IDs".

One test domain in which this type of process joining might occur frequently is games in which the exploration of the action space is often accompanied by the combination of different strategies. Each of these strategies may originally be recorded by the game as a single process since it can't predict an eventual combination of these processes.

5.5 FULFILLING C-TRACER'S REQUIREMENTS WITH R AND SHINY

C-Tracer was designed as a web app to increase its reach and ease of access beyond a simple executable file or script. Early prototypes used an R-Server backend with a Flutter Web front end. Though R remained the language for the implementation of C-Tracer, the front

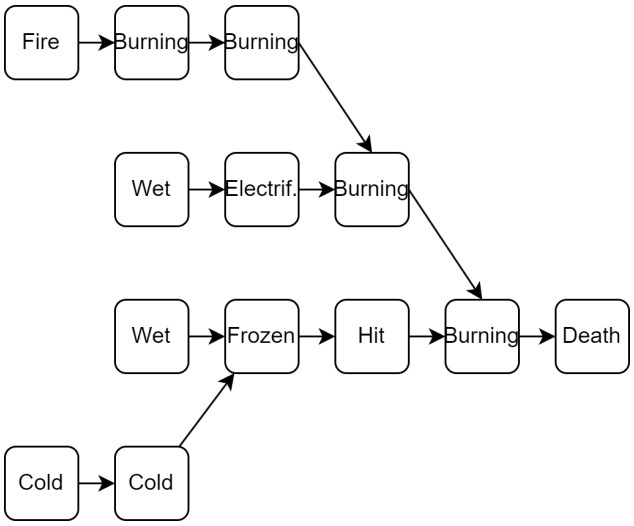

Figure 6: Example of a "process tree" from the Immune Defense game.

end was soon implemented as a shiny application [1]. Shiny is an R package that can build reactive web-based UIs for R code, and it has particular value because of its related cloud service, shinyapps.io [2], which allowed for quick deployment of C-Tracer updates that could be tested by others. Shiny uses a declarative programming style to define how a front end should look. Like the well-known React framework or Facebook's Flux, it effectively hides the controller from the programmer and thus separates itself from the more traditional Model-View-Controller (MVC) pattern approaches. Unlike such a typical MVC implementation, it has a one-way data binding between the model and the view. From a programmatic perspective, this makes exposing algorithmic results to a user very easy. When the user is happy with their C-tracer configuration, its results are calculated and automatically shown. C-Tracer's user workflow mirrors the one-way data-binding paradigm of Shiny. From the upload of the unprocessed data to the eventual download of C-Tracer's results: Steps that can be taken are linear and one-way.

Creating a simple yet responsive UI/UX for users

C-Tracer is conceptually a state-based application, where the three main states, "Configuration", "Calculation", and "Outcome Analysis", are logically simple: There are no unexpected dependencies between the states, the states follow each other, and each state is fully dependent on its previous state (unless it's the initial state). Additionally, as long as the configuration and data are valid, there are no side effects that can disrupt future states. "Under the hood", C-Tracer needs users to perform configuration steps, upload and preprocess data and execute the algorithm. A faulty configuration or bad data are the main ways C-Tracer can fail to produce valuable results, which means the UI should ideally only expose "correct" (i.e., logi-

1 https://cran.r-project.org/web/packages/shiny/index.html
2 https://www.shinyapps.io/

cal) actions to a user. There is somewhat of an assumption that any user of C-Tracer has expertise in definitions and semantics of the creative process as well as the data that can represent creative processes. However, user tests have shown that the specifics of C-Tracer, the Delta algorithm, and the other creativity measures were being treated more as a black box when being used. Together with C-Tracer's runtime, which depending on the data ranges from less than a second to theoretically hours, a focus of C-Tracer is using Shiny's reactivity to provide users with logical user experience (UX) constraints as well as visualizations of what is currently C-Tracer's focus.

5.6 AN ANY-ELEMENT LEVENSHTEIN DISTANCE IMPLEMENTATION

The Delta algorithm depends on a distance function that can take two processes as input to generate the action-level distance between the processes as an output, see Chapter 3. This section will describe an implementation of this any-element Levenshtein algorithm. R has an extensive library (i. e., package) support for string comparison, but no existing package offers a Levenshtein-like algorithm on a word level or list-element level. Thus, a custom function was written that would output any-element distances for any given list pair. That is, instead of comparing characters of a string for equality, it tests elements of a list for equality. To test the implementation, the dataset from the first Immune Defense study was used (see Section 6.5). This dataset has 15445 observations created by 17 participants that played the Immune Defense game for 5 minutes. Most processes in the dataset have a length of 5 to 20 actions, with very few processes being much longer, with up to 130 actions. The dataset proved a useful benchmarking tool: It provided a large enough amount of simple distance calculations to evaluate time constraints of space allocation, function calls, and other programming decisions, as well as some process distances long enough to likely represent the upper end of complexity that the any-element Levenshtein algorithm could be expected to be used for. This assumption seems to be confirmed so far; all other datasets tested with C-Tracer at the time of this writing have had far fewer observations, far fewer process comparisons, and much shorter processes than those generated by Immune Defense. In total, evaluating the Immune Defense dataset would lead to the pairwise comparison of 123.211 process pairs.

Listing 2 shows the naive R implementation of the element-level Levenshtein function.

Listing 2: Naïve any-element Levenshtein distance implementation

```
levenshtein <- function(list1, list2, len1=NULL, len2=NULL,
    offset1 = 0, offset2 = 0) {
if (is.null(len1)) len1 <- length(list1)
if (is.null(len2)) len2 <- length(list2)
if (len1 == 0) return(len2)
if (len2 == 0) return(len1)
cost <- 0
if (list1[[offset1 + 1]] != list2[[offset2 + 1]]) cost <- 1
dist <- min(
levenshtein(list1, list2, len1 - 1, len2, offset1 + 1, offset2) +
    1,
levenshtein(list1, list2, len1, len2 - 1, offset1, offset2 + 1) +
    1,
levenshtein(list1, list2, len1 - 1, len2 - 1, offset1 + 1,
    offset2 + 1) + cost)
return(dist)
}
```

It outputs element-level Levenshtein distances for any two lists, which in practice means any two processes. This version is unoptimized, specifically in the context of being used in native R, which does not optimize tail-recursive functions. Since this function is tail recursive, it is both slow and unable to take on inputs of larger sizes: With its three recursive function calls and the potential for zero matching elements (the worst case for Levenshtein distance runtime), the call tree of this implementation has $(3^{min(m,n)})$ nodes for a single process pair, where m and n are the two process lengths, respectively. It is not surprising that this exponential runtime did not perform well in the Immune Defense dataset. Up to the point of this writing, it would still have sufficed for all datasets that were not generated by Immune Defense because those other datasets used with C-Tracer generated process lengths of generally under 10. 3^{10} (i.e., maximally 59049) recursive calls are quick on any modern processor. This function's runtime for the Immune Defense dataset is difficult to know exactly, as it had to be extrapolated from observed partial execution. After 15 hours, only the first approx. 4000 (of 123211) process pairs had been evaluated. Note that most of those belonged to a participant with very low average process lengths, and a further non-linear time increase of the algorithm for the longer process pairs of later participants could be expected. In benchmarks of later algorithm iterations, some of the more complex process pair calculations would take as long as the first 4000 process pairs combined, which means the potential runtime for this naïve implementation could have taken longer than a year just to evaluate a dataset of 17 participants.

The unoptimized Levenshtein function was soon replaced by a memoized version, see Listing 3.

Listing 3: Memoized any-element Levenshtein distance implementation

```
levenshtein <- local({
  memo <- list()
  function(list1, list2, len1 = NULL, len2 = NULL, offset1 = 0,
      offset2 = 0) {
    if (is.null(len1)) len1 <- length(list1)
    if (is.null(len2)) len2 <- length(list2)

    key <- paste(c(
        toString(offset1),
        toString(len1),
        toString(offset2),
        toString(len2)),
        collapse = ",")
    if (!is.null(memo[[key]])) return(memo[[key]])

    if (len1 == 0) return(len2)
    if (len2 == 0) return(len1)
    cost <- 0
    if (list1[[offset1 + 1]] != list2[[offset2 + 1]]) cost <- 1
    dist <- min(
      levenshtein(list1, list2, len1 - 1, len2, offset1 + 1,
          offset2) + 1,
      levenshtein(list1, list2, len1, len2 - 1, offset1, offset2
          + 1) + 1,
      levenshtein(list1, list2, len1 - 1, len2 - 1, offset1 + 1,
          offset2 + 1) + cost
    )
    memo[key] <<- dist
    return(dist)
}})
```

Adding memoization eliminates having to evaluate sub-problems that have already been solved by previous iterations. The time complexity of evaluating two processes, while still not trivial, is reduced to $O(m \quad n)$ where the upper bound is given by the complexity of calculating the distance for two processes that share no common action, and every suffix of process m is compared to every suffix of process n. The trade-off for this is the added space requirement of $O(m \quad n)$, where the bound is given by the space required to memoize the results of all distance calculations of two processes that share no common action, meaning that all suffixes of both words are also memoized.

In benchmarks, the added space complexity had no tangible effect on the otherwise improved runtime. Every key-value pair costs about 100 Byte in memory, which means the memoization variable grew to around 1 MB (100B * m * n = 100B * 100 * 100) for the longest process pairs in the Immune Defense dataset. The runtime of the memoized version of the algorithm was unsurprisingly faster than the non-memoized version, but surprisingly not as much as expected. At around 15-17 hours for the evaluation of the 123211 process pairs of the Immune Defense dataset, the runtime was slower than what

was hoped for. Notably, this function was as quick as expected for any single process-pair comparison: The most complicated process pair in the Immune Defense dataset took around one second to evaluate. However, when repeated multiple times, the function became slower: The same process pair took minutes to solve when called as part of the repeated comparisons made in a loop, which pointed towards some other slowing side effect in the implementation. This behavior was not only unexpected but also difficult to pinpoint, as it only occurred during the "real-life" runtime of the whole C-Tracer application, never when benchmarking the any-element Levenshtein function manually, where it remained quick. A deeper exploration of the R-language was needed to understand why the function slowed down - seemingly only when nobody was looking.

R Internals: Call-by-Value, Environments, Reference Counting, Garbage Collection

Consider the code in Listing 4.

Listing 4: A memoization example

```
my_memoized_func <- local({
        memo <- list()
        function(input) {
                result <- ### Some Calculation
                memo[input] <<- result
                memo_recursive(input - 1)
        }
})

for (i in 1:100) {
        memo_recursive(i)
}
```

The function and recursion shown in Listing 4 are an abstracted and condensed version of the memoized any-element Levenshtein function and how it is called within a loop. Its purpose is to illustrate the behavior of the variable `memo`. First, `memo` is bound (i. e., initialized) outside of the function call. Then, the function performs some calculations, memoizes the calculation result, and calls itself recursively.

The `memo` variable will, unexpectedly to some, already be bound before `my_memoized_func()` is ever called. The variable has a binding, even though the function in which it is expected to be bound has never been executed. Additionally, when calling `my_memoized_func()` multiple times in a loop, the memo variable persists across all calls. It grows across the function's repeated calls as opposed to being bound (and subsequently unbound and garbage collected) every time `my_memoized_func()` is called. Depending on the use case, this behavior could indeed be preferable. For example, a repeated call to a memoized version of Fibonacci that memoizes across function calls would result in a much faster total execution time than just memoizing within each separate call. But for the use case of Delta, this behavior

has two problems: First, the keys used to populate the memo variable are generated from current offsets and suffix lengths, which means key-result pairs are not the same for different list (i.e., process) inputs. Second, the memo variable is growing far beyond the expected size of O(m n) to a size of O((m n) k(k - 1)=2 where k is the number of processes. This is causing the slow speed of that Delta version and shows why the single benchmarking tests remained quick: k was low.

Why is `memo` not ephemeral like all other variables in the function?

Understanding this behavior requires knowledge of how the R language passes objects to functions, how R environments relate to objects and functions (i.e., function objects), and when R's garbage collector removes objects from memory.

Environments

Environments are R's scoping mechanism. They function like named lists with some additional constraints: Every environment has a parent environment, elements in an environment must have unique names and are unordered, and environments never get copied when modified. Unlike most other objects in R, environments in R follow the call-by-reference paradigm, so they are never passed to some other function or environment. Instead, a reference to the environment is passed implicitly. Environments also bind other environments, which can be done explicitly

```
env2 <- new.env(),
parent.env(env2)
# <environment: R_GlobalEnv>
```

or implicitly, for example by executing a function, which binds an ephemeral execution environment to the calling environment until the function terminates. When an environment is created, objects can be bound to it (i. e., objects are added to the list that is the environment).

```
env2$obj <- "in env2"
where(obj)
# <environment: env2>
```

The most intricate - but implicit - environment-related behavior is that of function objects (i. e., "functions"). Function objects are interacting with four different environments:

1) The binding environment is where the function's name is bound. It's where you "find" the function.

```
my_fun <- function() obj
where(my_fun)
# <environment: R_GlobalEnv>
```

2) The enclosing environment is where the function *exists*. This is typically the same as the binding environment, but a function can exist in an enclosing environment - with the enclosing environment's objects

- and be bound to another environment, the binding environment, where the function's name is found.

```
environment(my_fun) <- env2
identical(environment(my_fun), env2)
# TRUE
where("my_fun")
# <environment: R_GlobalEnv>
obj <- "in global"
my_fun()
# "in env2"
```

In the above example, `my_fun()` is bound to the global environment. This means it can be called from the global environment. But its enclosing environment has been changed to `env2`, which means it is being executed in the enclosing environment.

3) The execution environment (also called local environment) is the ephemeral environment of the function's arguments and operations while the function is running.

```
my_fun <- function() print(obj <- "in exec")
my_fun()
# "in exec"
obj
# "in global"
```

4) The calling environment (also called parent frame) is the environment from which the function is called.

```
env2$my_fun <- function() parent.frame()
env2$my_fun()
# <environment: R_GlobalEnv>
```

In the above example, `my_fun()` gets bound to `env2` and will also be enclosed by `env2`, but it is called from the global environment.

While a function is executing, it looks for named objects first in its execution environment, then in its enclosing environment. For function arguments, R considers only the calling environment.

Reference Counting and Garbage Collection

Objects in R have a reference counter, which counts how many other objects they are referenced by. Environments are objects which are bound to another environment when they are created, so they are each at least referenced once by their enclosing environment. In the same manner, objects currently bound (referenced) by an environment are referenced by at least one object: Their binding environment. An object's reference count can become zero when its environment is removed normally (e. g., it's an execution environment and the execution is finished), or when its references are explicitly removed.

```
rm(obj)
obj
# Error: object 'obj' not found
```

When an object's reference counter becomes zero, it can be collected by the garbage collector. For example, `x <- 1; rm(x)` initializes the object x by assigning `1` to it, then removes all references to x. After, the garbage collector will free the memory address and allocated space of x. Consider initializing x in some environment, such as a function environment `my_fun<-function()x<-1` where x is assigned `1` in the execution environment of `my_fun`. Its reference counter becomes `1` while `my_fun` is executing, then `0` after `my_fun` has terminated and the execution environment has stopped existing.

Memoization and Implicit Variable Bindings

The environments' call-by-reference semantics are the exceptions for R objects, as they mostly follow the call-by-value paradigm instead. In this regard, R is maximally lazy and often operates on promises until an object is explicitly needed. Whenever an object (such as a memoization variable) is passed as a parameter to a function, it is passed as a promise until it is needed: The value of the object can be read within the function's execution environment this way as if the object had been passed by reference. Exactly when the object is modified within the execution environment, it is copied to that environment, the promise is fulfilled, and the reference to the original object is discarded. In the case of the memoization variable, which is modified whenever a new input is calculated by its function, the variable gets copied on every function call. Clearly, this is not intended, which is why memoization in R is generally achieved by creating a factory function: The `local` keyword used in the memoized version of the Levenshtein implementation helps R to avoid copying the memoization variable on every recursive function call. It is creating a single non-ephemeral enclosing environment for all future execution environments. This enclosing environment is different from the calling environment and accessible from the function's execution environment. Without this, the function's enclosing and calling environments would be the same, and the memoization variable would have to be maintained outside of its own function. Once the memoization variable is bound to the function's enclosing environment, the "super assignment operator" `<<-` is used to grow the variable in that environment as opposed to having it be copied to every new execution environment of every recursive call.

With the specifics of environments, function environments, reference counting, and garbage collection in place, a single detail explains why the memoized implementation as shown in Listing 3 becomes slower when used repeatedly: *All functions keep a reference to their enclosing environment.* Thus, the enclosing environment which contains the `memo` variable and was intended to be ephemeral, binds the `function` on line 3, but is also bound in turn *by* that function. The enclosing environment, even though it is an *execution environment*, is not ephemeral because it binds *and* encloses a function. The memoiza-

tion variable will never lose its binding: It will persist across function calls to lev().

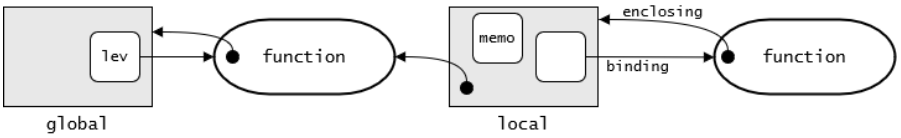

Figure 7: Environments and bindings for a memoized function in R

In the end, the `memo` variable is bound by the `local` environment. `local` and the functions defined in it have a mutual binding, which means neither will ever be garbage collected as their reference counter is at least 1, see Figure 7. Because the `local` environment will never be garbage collected and `memo` is bound by it, `memo` will never be garbage collected. Because `memo` will never be garbage collected, it persists across any calls to `lev()`. And, finally, because `memo` persists across calls to `lev()`, it grows forever and slows `lev()` during real-life execution.

Calling the enclosing environment of the custom Levenshtein function during runtime shows this behavior: The enclosing environment exists immediately after creating the function, but remains different from the binding and execution environments. Even though `memo` was created in the function's enclosing environment to prevent having to maintain it from the binding and calling environment, it has to be maintained (manually) from there after all. In this case, this means explicitly resetting it within the enclosing environment after every call to lev() by assigning an empty list to it.

```
assign("memo", list(), envir = environment(lev))
```

This stops it from growing infinitely and reduces the total evaluation time of the Immune Defense dataset from 15-17 hours to around 17 minutes.

After this journey through R internals, readers may find themselves saddened that a different approach to decrease runtime was ultimately developed and used, as even the improved runtime remained too slow for a larger study conducted later in the development process. This approach is elaborated on in the following section.

5.7 PROBLEM REDUCTION: ANY-ELEMENT LISTS TO WORDS

Originally not a necessity because the existing datasets that were to be used with Delta were small enough, more recent studies that are using Delta have been generating much larger amounts of data to be processed. With more than 100 participants generating play data by playing Immune Defense (see Section 6.5), Delta's execution time would reach around two hours when using the previously discussed any-element Levenshtein function. Compared to any manual evaluation of the thousands of processes that are generated by over 100 participants this might be acceptable, but of course less preferable than finding a way to further reduce runtime. With the R-native options

of memoization and recursion seemingly exhausted, a different approach was developed: The Levenshtein distance is well-known and well-defined, with many implementations for solving it in different languages, including R. The existing implementations could not be used to calculate the any-element distances needed by Delta since they only accept character strings as input. After exhausting optimization options from R's algorithmic side, the new approach consists of just two steps: 1) reduce the existing problem to a problem solvable by a faster library, and 2) use that library.

The StringDist package for R can calculate string-level Levenshtein distances in one of the fastest ways possible: a dynamic programming approach written in native C [40]. Though asymptotically bound by the same constraints as the self-written element-level Levenshtein function, the real-life runtime of this implementation is much faster. StringDist only operates on strings, which is why the need for a custom Levenshtein function surfaced originally. In order to use any-element lists with StringDist, the lists would have to be converted to some character vectors (i.e., strings) that generate the same Levenshtein distances by StringDist as the processes would when evaluated per action. In other words, action-level lists need to be transformed into "character-level strings" (which is of course just what strings are). To do this, all distinct actions of a given dataset are mapped to single characters, so that each process is represented by a string, see Section A.1. The strings are then compatible with StringDist and the calculated string distances will result in the same values as the calculated any-element process distances would have, see Figure 8.

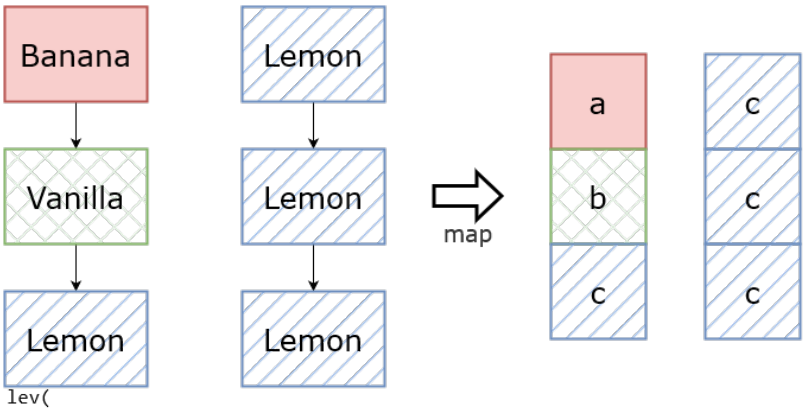

Figure 8: Reduction of two ICIT processes to two strings.

Once transformed into characters, the process (now string) pairs can be used as valid input for StringDist. The runtime of evaluating the original Immune Defense dataset was reduced to around one minute, as opposed to the 17 minutes when using the custom element-level Levenshtein function. Evaluation of the data produced by 103 participants of a later study (cf. Section 6.5) could thus be analyzed in around 15 minutes.

5.8 PROCESSES EXTRACTION

Before C-Scores or other metrics can be calculated on participants' sets of respective processes, the processes need to be extracted from the event log, see Figure 9.

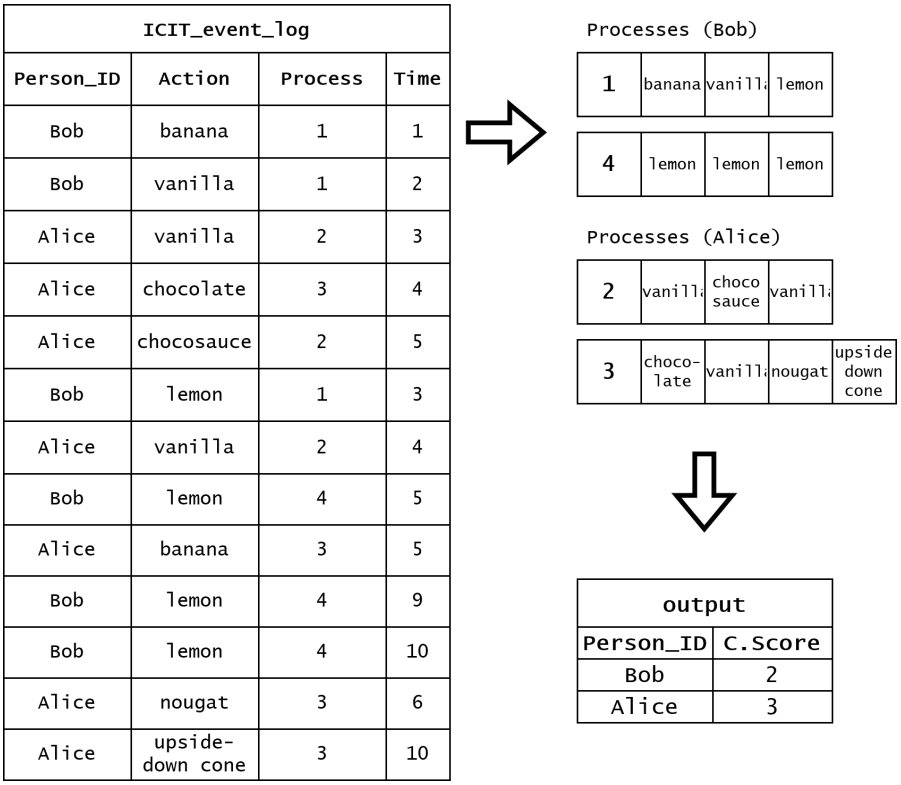

Figure 9: Generating distinct sets of processes from an ICIT event log.

When ignoring the possibility of connected processes, this operation would be a simple `group` aggregation that groups by process ID. When connected processes are indicated by the optional attribute, an additional step of combining processes is performed.

5.9 AUTOMATED METRICS BEYOND THE C-SCORE

In addition to the C-Score, C-Tracer computes other measures which relate to other creative constructs. Since C-Tracer already extracts processes, these measures may increase the utility of process-based creativity evaluations, such as the ones discussed in Chapter 4.

NUMBER OF TOTAL PROCESSES. The number of total processes is the number of processes (successful or not) that a person generated. In the case of connected processes existing in the event log, there exists some ambiguity on whether each unsuccessful process that is also connected to other processes should be evaluated as its own process or as part of a larger process comprised of the other sub-processes. In C-Tracer's implementation, the amount of unique process IDs in a person's event log is measured through the number of processes. This means that this measure resembles more closely the "number of

total approaches" than the number of finished processes, as without a definition of success the difference between an unsuccessful process and an unfinished one cannot be determined.

NUMBER OF SUCCESSFUL PROCESSES. The number of successful processes is self-explanatory and doesn't suffer from ambiguity such as the definition of the total number of processes. This measure is calculated from the length of the list of processes of each person after the successful processes have been extracted. Figure 9 gives an example where both participants' number of successful processes is 2.

LONGEST SUCCESSFUL PROCESS LENGTH. Shows the longest process among all successful processes generated by each participant, see Section A.2.

AVERAGE SUCCESSFUL PROCESS LENGTH. Shows the average length of successful processes generated by each participant, based on the arithmetic mean of all the participants' successful process lengths.

NUMBER OF ACTIONS. Shows how many actions each participant has undertaken in the event log, which is also the number of rows in the event log for each participant.

5.10 SERIAL-ORDER-EFFECT ANALYSIS FOR TEST-RETEST VALIDITY

The serial-order-effect can occur during tests in which responses start out being low in creativity, but increase in creativity over time [12, 47]. The lack of this effect in tests is an indicator of high validity, as it means that the quality of responses is not influenced by task expertise. At best, a test will generate similar measurements for a participant even if the participant retests the same test. As requested by testers early during the development of C-Tracer, a way for enabling this type of analysis within a single event log was added, which splits the log into sub-logs. This had an extensive code-level impact on the rest of the implementation: 1) Each person's event log needs to be dynamically split into parts that reflect the time during which actions were performed. Since processes may be started in one phase but ended in another phase, processes are assigned to phases based on the last process action's timestamp, see Figure 10.

2) The lists containing each person's processes are getting another layer and become lists of lists, with the outer lists representing the different phases, and the inner lists containing the processes. 3) Each analysis step needs to perform its analyses "for each phase". Since the C-Score does not have the distributive property, that is (C-Score(Log1) + C-Score(Log2))/2 is not equivalent to C-Score(Log1+Log2), splitting the data into phases forced a decision: Either, calculate the total C-Score followed by the C-Scores of each phase and accept the increase

ICIT_event_log			
Person_ID	Action	Process	Time
Alice	vanilla	2	3
Alice	chocolate	3	4
Alice	chocosauce	2	5
Alice	vanilla	2	4
Alice	banana	3	5
Alice	nougat	3	6
Alice	upside-down cone	3	10

Phase 1			
Person_ID	Action	Process	Time
Alice	vanilla	2	3
Alice	vanilla	2	4
Alice	chocosauce	2	5

Phase 2			
Person_ID	Action	Process	Time
Alice	chocolate	3	4
Alice	banana	3	5
Alice	nougat	3	6
Alice	upside-down cone	3	10

Figure 10: Two-phase split of an ICIT event log.

in runtime. Or calculate only the C-Scores of each phase, then average those scores to get a slightly incorrect overall C-Score. The error that this way of calculation introduces proved to be rather low during the exploratory analysis of available datasets, which lead to the decision to calculate only the faster way per default, with the option of an additional full C-Score calculation later.

5.11 TOOL SUPPORT FOR RAPID VALIDATION

Since process-based creativity assessments are not widely adopted, there may be a particular need of validating their results early and quickly. This is why C-Tracer allows for an additional automatic correlation analysis of its results with a secondary dataset, such as creativity measures of the same participants from some established creativity test (see Section A.3). C-Tracer will match IDs from the secondary dataset to the IDs in the C-Score results and output a long-format correlation table that may give an early indicator of data and measurement validity.

6

USING C-TRACER IN THE REAL WORLD

This chapter details experiments and studies in which C-Tracer and/or the Delta algorithm were used. They differ in setup, that is experimental settings, data collection tools, the data itself, participants, etc., which makes them a good means to demonstrate the steps needed to make each of them compatible with C-Tracer. The first section in this chapter will outline a generalized checklist for C-Tracer compatibility, which will then be referenced by the experiment and study reports in later sections.

6.1 EXPERIMENT AND STUDY DESIGN CHECKLIST FOR C-TRACER

Experiments in studies intending to analyze their results with C-Tracer need to fulfill four requirements in order to generate valid results:

1. Define a creative task (Problem Space).
2. Define what makes an action (Action Space).
3. Define process success (Solution Space).
4. Mapping these spaces to records of actions, as detailed in Chapter 3.

Only the fourth step is necessary for C-Tracer to generate results, but all four steps are necessary for C-Tracer to generate *valid* results. These requirements are directly related to definitions of creativity as described in Chapter 3.

Defining a Problem Space

The problem space defines and constrains the task that participants will be asked to perform. All creativity tests (not creativity questionnaires) need to have this. Examples of problem spaces from known creativity tests are finding creative uses for a common object (AUT, Section 2.3) or finding words that are as different from one another as possible (DAT, Section 2.4). In the context of the ICIT (Chapter 3), the problem space is finding delicious ice cream flavor combinations. A good way to think about the problem space may be phrasing it like a generative design question. In this case, the question might be "How could I create the most delicious ice cream cone?"

Understanding an experiment's problem space is particularly important in the context of creative processes, as participants may perform actions that are unrelated to the problem space in which they are ideating. For example, a participant performing the ICIT may joke

that they prefer savory food and will attempt to create delicious sandwich topping combinations. Depending on the prior problem space definition, some researchers may include this part of the process in their data, as they want their problem space to be *creating delicious foods based on an ice-cream prompt*. Other researchers may want a more narrow problem space definition, which only regards creative actions pertaining to exactly the keywords in the prompt.

Defining an Action Space

The action space allows C-Tracer to map data to novelty. It is the alphabet of actions that explains the possible creative exploration of a problem space. It defines all actions that participants can perform in a task. Participants may or may not be aware of the existence of such actions. For example, consider three different action space definitions for a verbal version of the ICIT. A participant says the following sentence: "Hm, maybe just chocolate? Everyone likes chocolate. Well, maybe not dogs. Oh! Vanilla and chocolate sauce."

A first definition could state that only and exactly ice cream flavors are part of the action space. The participant will get their statement recorded as *(Vanilla)* because the final answer was "Vanilla and chocolate sauce" but only "Vanilla" is part of the defined action space. The second definition allows a wider action space, in which any single word of the solution that relates to food is part of the action space. "Vanilla and chocolate sauce" may be recorded as *(Vanilla, Chocolate sauce)*, or even *(Vanilla, Chocolate, Sauce)*. Lastly, consider a third action space definition that defines actions, not as the actual solutions, but as types of creative patterns exhibited by participants during problem-solving. For example, an action space alphabet may be *(Contemplating, Joking, Reasoning, Converging, Aha-Moment)*. The participant's sentence could be recorded as *(Contemplating, Reasoning, Joking, Aha-Moment, Converging)*.

Though the problem space of the ICIT may have stayed the same, the experiment's action space definitions will determine what is actually measured and therefore what kinds of interpretation the results may allow.

Defining a Solution Space

The solution space is needed to explain what constitutes a valid solution in the context of a problem space and action space. Such a solution definition complements the action spaces' formalization of novelty, by creating a formalization for success.

Success requires a particular understanding of the construct or construct facet intended to be measured. A "success" can be arriving at a single specific goal like making a paper plane that can fly more than 10 meters, producing a valid idea (vanilla + chocolate), getting a point in a game (scoring a point in basketball), or any other domain-specific success. Success can be explicit or implicit: Explicit success

means recording the moment of success as a special part of the action space. For example, *(Vanilla, Chocolate, Success)* could be an explicit use of success. This may be especially useful for more complicated processes where actions are recorded in the moment but the success is an uncertainty. Conversely, implicit success means that the fact that a process got recorded signifies its success. Any unsuccessful processes were never recorded, or have been discarded.

Creating a Valid Data Record

C-Tracer has some data requirements that are outlined in Chapter 3. In particular, it needs a CSV file in which each row represents one creative action with attributes of at least participant ID, process ID, action, and time. In general, any time an action in the action space is observed, one row should exist in the data record, that shows who performed that action, as part of which process, and at what time.

Checklist for generating valid data for C-Tracer analysis

1. Problem space definition
2. Action space definition
3. Solution space definition
4. Data Record

6.2 VALIDATION AND REAL-WORLD APPLICATION: REAL-WORLD DATA AND C-TRACER

C-Tracer has been used in a number of real-world use cases: 1) In a study where students were asked to play the video game "Immune Defense" [36], C-Scores were generated from the processes of defeating "bacteria" and "viruses" in the game. 2) For auxiliary analysis of a novel productive thinking test, CollaboUse [64]. 3) On a creative writing study [42]. 4) In a large-scale online study on creativity [27]. In addition to these studies, C-Tracer was experimentally applied to other data, such as an "Apple Customer Support" event log.

6.3 INTERNAL USABILITY TESTS

After the development of the Delta algorithm, early informal usability tests revealed a fundamental problem: Creativity experts were neither used to executing script-based programs nor did they have experience using programming shells or similar tools without explicit user interfaces. C-Tracer was the answer to this obstacle, providing the UI for configuring and executing Delta, see Chapter 5. Once accessible on the internet and from a browser, the general difficulty of understanding the underlying functionality of the tool became more apparent. Users were now provided with a UI, which helped non-technical

users gain access to C-Tracer, but helped little in the ways of educating them on the underlying functions, data requirements, and necessary configuration steps. One example of this is assigning columns to their respective categories (e.g., the "Timestamp" column of the event log to the "Order" category of C-Tracer). A part of this difficulty may be that C-Tracer brings with it a new domain-specific language. With creativity experts traditionally using finished productions in their creativity assessments, the evaluation of creative processes brings with itself new terminology and logic that may require further refinements or simply time for users to get familiar with it.

6.4 COLLABOUSE

C-Tracer was used to analyze the results of a novel productive thinking test, CollaboUse. CollaboUse is a collaborative browser-based test, based on the test logic of the AUT. It prompts participants or teams of participants to think of solutions to a prompt. Participants are given only a limited amount of "objects" to combine in order to think of interesting ideas for the prompt, see Figure 11. An example prompt is "science-fiction decorations for a movie set", and example objects are a rope, a pin needle, or a mirror.

Figure 11: The CollaboUse creativity test

CollaboUse: Problem Space

The problem space of CollaboUse is defined by the prompt. Participants are asked to imagine how they could combine some given items in order to fulfill a requirement.

CollaboUse: Action Space

CollaboUse's action space is exactly defined by the objects that participants are allowed to use. If participants are allowed to use ten particular objects, then that is also the action space they can traverse.

CollaboUse: Solution Space

Any idea that fulfills the requirements outlined in the prompt is defined as being part of the solution space. Every finished solution is considered successful and no additional measure of success or failure is recorded other than the success of coming up with an idea.

CollaboUse: Pilot Study and Results

CollaboUse was first used in a study on creative team performance [64] in which creative team performance in the CollaboUse was measured after different interventions. There, C-Scores and fluency correlated positively with interventions intended to promote togetherness.

6.5 IMMUNE DEFENSE

Immune Defense is a video game that can be played in the browser. It records events that happen during gameplay. The resulting event logs are compatible with C-Tracer and can be used to generate C-Scores.

Immune Defense: Problem Space

Immune Defense prompts players to defend a "heart" at the center of their screen from attacking viruses and bacteria, see Figure 12. The problem space definition asks: "How could you protect the heart as well as possible?"

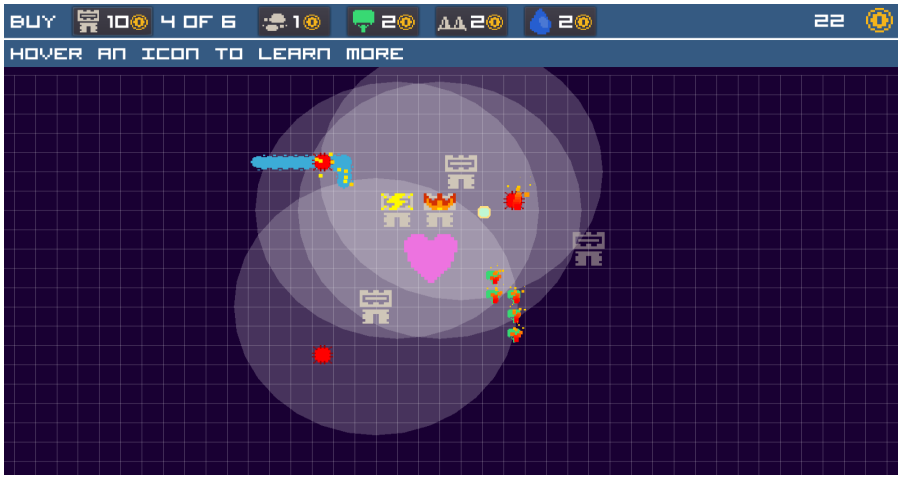

Figure 12: The Immune Defense game

Immune Defense: Action Space

Immune Defense serves as a good example of the potential different action spaces that can exist within the same problem space. For Immune Defense, two different action spaces were discussed. The first considered player actions, such as placing a tower or areas of water

in the game. The second one considered the effect of player actions on the environment and attackers. The latter was chosen to be more relevant to the study goals, based on the domain expertise of the researchers. The main reason for this is the way that effects in Immune Defense can be combined to create different effects. If a player makes the action "Place Ice Tower" and then the action "Place Water", placing both the ice tower and the water closely together, then the player may have intended to combine these effects to create ice and freeze attackers. This means that two players may undertake the same actions but have differing intentions and different effectiveness depending on where in the game world the actions took place. This is why events from the second action space were recorded, which are able to not only show that attackers were wet or cold but also frozen.

Immune Defense: Solution Space

Immune Defense's solution space is defined by the defeat of attackers. In an event log, this is marked by the "death" event that records the defeat of an attacker.

Immune Defense: Studies and Results

Published in the 2020 IEEE Conference on Games (CoG), the paper "Designing a Video Game to Measure Creativity" was a work that would later lead to the development of the Delta algorithm [36]. The aim of this study was to validate a video game that was designed to measure creativity. A particular interest of this study was the potential of using this game to generate creativity scores without human data analysis (e.g., without using the CAT as described in Section 2.2). 17 participants were asked to play the video game Immune Defense as well as to perform the Alternative Uses Task and fill out a self-report questionnaire. The AUT was evaluated by experts as per the CAT, and the C-Score was calculated automatically, though at the time a script specifically written for Immune Defense generated C-Scores, as the C-Tracer did not exist yet. Results showed moderate to high correlations between the C-Score and AUT Fluency and AUT Flexibility, and no strong correlation with the game score of the players, see Table 2. The latter was an indicator, that task expertise did not influence the C-Score.

The promising correlations of AUT scores and Immune Defense-based creativity scores lead to three fundamental questions: Could these findings be replicated in a larger scale study? Could this approach or a similar one be automated to the point of requiring almost no human input in the data analysis? Could this approach be generalized to work with outputs of other creativity tests or data sources?

In order to answer these questions, another large-scale study was conducted on Amazon Mechanical Turk.

Table 2: Correlations of AUT and Immune Defense scores

		1	2	3	4	5
1.	AUT Flexibility	1				
2.	AUT Fluency	0.95*	1			
3.	AUT Originality	0.14	0.20	1		
4.	C-Score	0.57*	0.54*	0.13	1	
5.	Game Score	0.36	0.34	0.12	0.06	1

The table shows Pearson correlations, except for AUT Originality, which is calculated with Kendall's Tau. * < .05 significance.

6.6 COMBINED IMMUNE DEFENSE AND COLLABOUSE STUDY

In the context of large-scale user studies, Amazon Mechanical Turk (MTurk) is a popular tool for reaching a large number of study participants quickly. As discussed in Chapter 1, mainstream creativity research has so far not adopted this avenue of crowd-sourcing for studies, though web-based creativity tests exist that focus on more simplistic productions such as the ones generally expected from productive thinking tasks. Most of them are variations of the Alternative Uses Task with differing ways of evaluating answers, see Chapter 2. As opposed to these setups, where random visitors may participate for a very short time, MTurk opens up the possibility of conducting longer studies on the internet where participants can be queried for auxiliary information such as demographics or personality. This enables more traditional study setups, but also requires more technical setup work in comparison to in-person studies of the same kind.

Thus, this study, which was accepted to be published in the Creativity Research Journal (CRJ), may be the first of its kind, combining prompt- or questionnaire-based creativity tests with five minutes of gameplay [27]. Building on the study design of the in-person Immune Defense study, 103 international participants took part in browser-based versions of the AUT and Immune Defense. In addition to the main interest in the results of a large-scale, fully online creativity study, a meta-interest in this study was the effort it would take for data analysis which traditionally would not scale well with such a high amount of participants.

Participants in this 30-minute study went through a battery of self-report tests as well as the AUT, CollaboUse, and Immune Defense. CollaboUse and AUT were evaluated for fluency, flexibility, and originality. CollaboUse originality, AUT flexibility, and AUT originality were scored using the CAT with three expert raters. CollaboUse fluency, CollaboUse flexibility, and AUT fluency were scored automatically. Additionally, C-Scores were calculated for Immune Defense and CollaboUse. Table 3 shows correlations between the different measures. It illustrates well, that C-Scores from different creativity tests may not represent the same constructs: Immune Defense's C-Scores are almost entirely independent from CollaboUse's C-Scores: It ap-

Table 3: Correlations between *CollaboUse* and *Immune Defense*

		1	2	3	4	5
1.	ID C-Score					
2.	CU Fluency	.244*				
3.	CU Flexibility	.203*	.845**			
4.	CU MaxOrig	.163	.443**	.412**		
5.	CU MeanOrig	.178	.246*	.232*	.903**	
6.	CU C-Score	.064	.298**	.463**	.248*	.216*

** Correlation is significant at the 0.01 level (2-tailed).
* Correlation is significant at the 0.05 level (2-tailed).
ID: Immune Defense Game; CU: CollaboUse

pears that the average differentness of successful game actions in the game Immune Defense is not related to the average differentness of creative item combinations in the CollaboUse test. At the same time, there are small to moderate correlations between Immune Defense C-Scores and CollaboUse Fluency/Flexibility, as well as moderate correlations between CollaboUse C-Scores and other CollaboUse creativity metrics.

C-Tracer's data analysis itself required less than one minute of user input for the analysis of an amount of data that would be impossible to evaluate manually. The real opportunity costs of conducting such online studies are setting up a server that can receive and record the data, configuring the server to format the data into cohesive datasets, and managing the MTurk platform. These costs, though not a complicated problem to solve from a software engineering perspective, might still deter experts of other fields from conducting similar online studies or even having the ability to do so. This could indicate that, should such study designs be rising in popularity, a specialized framework for data collection and formats might be needed.

6.7 CREATIVE WRITING

C-Tracer's C-Scores have also been used as a measure to quantify creativity in creative writing [42, 43], making it the first approach to take preprocessed creative texts as input and generating C-Scores, etc. based on those texts. The texts were written based on writing prompts, such as "Write a story about a magical window, or a window like no other". McKee used the C-Tracer as part of a wider analysis regarding self-reported creativity assessments and their connection to C-Tracer's automatically computed creativity measures.

Creative Writing: Problem Space

The problem space in McKee's study was defined by open-ended writing prompts. The problem could be "solved" by writing a creative text. The participants were aware that the texts would be judged on their creativity.

Creative Writing: Action Space

Defining the action space as "all possible words", while straightforward, would have ended up with insignificant results. Consider the following creative text:

```
I like gelato. You enjoy pasta. We love food.
```

First, an action space of all possible words would have resulted in C-Scores that mostly represent the length of sentences, as Delta's pairwise checking for equality would have rarely found word pairs that are the same across two sentences. Second, the repetitive sentence structure does not feel novel, original, or generally creative. Thus, if each sentence is a process and all possible words define the action space, the C-Score of this text would be 3, the maximal possible score for processes of length 3.

Both the problem of action space size, as well as the sentences' structural similarity evaluation were solved by redefining the action space as the space of all word types (e.g., nouns, verbs, adverbs, etc.), see Figure 13.

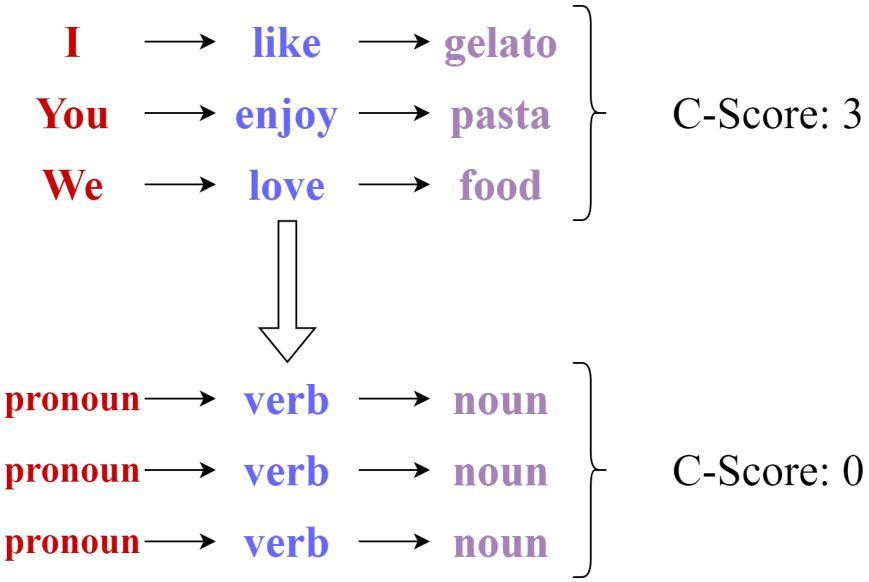

Figure 13: Effect of different action space definitions on the C-Score.

Creative Writing: Solution Space

The solution space in the creative writing task was defined as finished generations of sentences. While thinking of the next sentence to write, participants likely mentally explored different sentences before converging on the sentence they finally wrote. In creative process terms, they likely explored different creative processes until finding one that was effective.

Creative Writing: Study and Results

McKee undertook her study with 26 experienced authors. In the experiment, authors wrote creative texts both alone and in 13 pairwise teams, writing for 15 minutes in each task condition. McKee reported a positive correlation between C-Score and sentence length ($r = .461$, $p < 0.001$). It seems that within her study and its action, problem, and solution space, the C-Score captures a mix of elaboration and sentence complexity.

6.8 STANFORD LIVE DEMO

During the Spring 2022 Hasso Plattner Design Thinking Research Workshop, C-Tracer was used live to quickly generate various creativity scores for tests presented in a live demo of the different creativity tests Immune Defense, CollaboUse, and the prompt-based creative writing test. It allowed for audience participation, after which results were immediately presented to the same audience. The implications of this may be that rapid prototyping of novel creativity tests is made possible by an automated tool such as C-Tracer.

7

DISCUSSION, OUTLOOK, CONCLUSION

7.1 RANDOMNESS AND CREATIVITY ANALYSES

As creativity research slowly recognizes the opportunities that automated data collections and analyses can provide, it must adapt to the circumstances that come with these new avenues. The MTurk study discussed in Chapter 6 showed on the one hand, that creativity studies can be adapted to work with hundreds of participants from all over the world. On the other hand, it reaffirmed the difficulties of keeping the data quality high, as participants on such platforms have incentives to fulfill the requirements of the study tasks as quickly as possible to maximize monetary earnings based on their invested time. Other contemporary approaches that automatically test and evaluate creativity are not randomness-resistant. There, random answers will generally lead to high creativity values, which means that the automated approaches are forced to either accept a lower data quality or concede the need for some manual data analysis.

Process-based creativity assessments have the advantage of allowing researchers to define in which ways a process may terminate as to be considered effective. Such effective processes generally require multiple actions which are unlikely to be random *and* effective. Including such a "success" measure is a significant way in which C-Tracer's analysis distinguishes itself from other modern approaches to automatic creativity analyses. The definition of successful and unsuccessful processes, as well as the subsequent exclusion of unsuccessful attempts, means that random behavior is automatically filtered by the success definition. This resistance to random data allows for less or no human oversight, which in turn improves the scalability of such studies compared to similar studies without randomness resistance.

7.2 LIMITATIONS

Data-based outcomes can only ever be as good as the data itself. Since any recorded process in the correct data format can be used as input for C-Tracer, it could calculate values with little to no validity that look indistinguishable from highly valid ones. Should a business process dataset of Apple customer support employees be used as input for C-Tracer? The event log may contain all needed attributes required for a successfully terminating C-Tracer analysis, but persons recorded within that event log were probably not given the same tasks, their success and failure were never clearly defined, and the representation of persons in the event logs is highly unbalanced, with some employees having many more events than others. C-Tracer, in this case, may

invite drawing false conclusions. It does not make judgments on data quality and presents every result objectively. It will also not report any logical errors in data as long as a correct event log format is provided. Since the data analysis is fully automated, and therefore opaque to users, a clear understanding of the meaning and utility of C-Scores is needed when wanting to use C-Tracer correctly.

7.3 OUTLOOK

The main contributions this thesis offers to the creativity research domain are a novel language and algorithm for reasoning about creative behavior, as well as a tool - the C-Tracer - with which creativity measurements based on creative behavior can be computed automatically and quickly.

Regarding creative behavior: While reviewing contemporary creativity measurement approaches, the lack of process-based creativity assessments in the domain was noticeable. With advancements in how software can help illuminate the much noisier data that creative processes produce compared to creative productions, it now becomes feasible for creativity researchers to approach process-based creativity assessments or even analyze data generated from natural behavior.

Regarding C-Tracer: C-Tracer has shown that with the right study setup, domain definitions, and data collection, automated creativity assessments of thousands of data points are possible within seconds. During its use, the opportunities that come with the ability to evaluate creativity "on the fly" became clear. Also apparent became the potentially unexplored depth that creative processes have: Originally, C-Tracer was just the way to distribute the Delta algorithm, but added algorithms for measures such as fluency, elaboration, and the "D-Score" seem to only scratch the surface of the ways that C-Tracer could be extended now that a basic programmatic framework for extracting processes from event log data exists.

7.4 CONCLUSION

In my thesis, I introduce a new data-driven language and new definitions for reasoning about creative behavior and the creative process. Building on this language, I construct a novel algorithm, Delta, which calculates the average process differentness for a list of processes. Additionally, I introduce definitions and algorithms that allow well-known creative measures such as fluency, flexibility, and originality to be derived from creative processes. The novel process analysis tool C-Tracer is the programmatic utilization of these measures. It has been tested and validated in different studies. There, it calculated creativity scores by analyzing data from the video game Immune defense, the productive thinking task CollaboUse, and a creative writing challenge. Studies of over 100 participants were conducted, where C-Tracer calculated creative scores of all participants automatically and quickly. Since C-Tracer is domain-agnostic, I show how to define the

problem, action, and solution spaces, so that data can be made compatible with C-Tracer. This analysis is objective, randomness-resistant, and requires no human labor.

Part IV

APPENDIX

A

APPENDIX

A.1 IMPLEMENTATION OF PROBLEM REDUCTION: ACTION TO CHAR

Note: An "event" is equivalent to an "action" in a creative process.

```
map_strings_to_characters <- function(events) {
    start <- 32 # 32 is the character index for a whitespace.
    events_as_factors <- as.factor(events)
    event_factor_indices <- as.integer(events_as_factors)

    #A success event is special, so we need to know its factor.
    success_event_level <- which(
      levels(events_as_factors) == config$success_event
      )

    single_char_success <- intToUtf8(start + success_event_level)
    single_char_events <- unlist(
      lapply(event_factor_indices, function(i) intToUtf8(start +
          i))
    )
    #Return the renamed success event separately.
    list(single_char_success, single_char_events)
```

A.2 FINDING THE LONGEST PROCESS IN A LIST OF PROCESSES

```
find_longest_process <- function(processes_by_phase) {
    res <- 0
    for (phase in seq_along(processes_by_phase)) {
        longest_successful_process_in_phase <- max(
            unlist(
                lapply(
                    processes_by_phase[[phase]], function(x)
                        length(x)
                )
            )
        )
        res <- max(
            res,
            longest_successful_process_in_phase
        )
    }
    res
}
```

A.3 C-TRACER STATES

During typical use, C-Tracer will traverse different states, as discussed in Chapter 5.

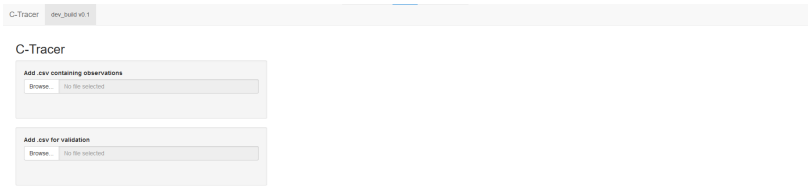

Figure 14: Initial state of C-Tracer

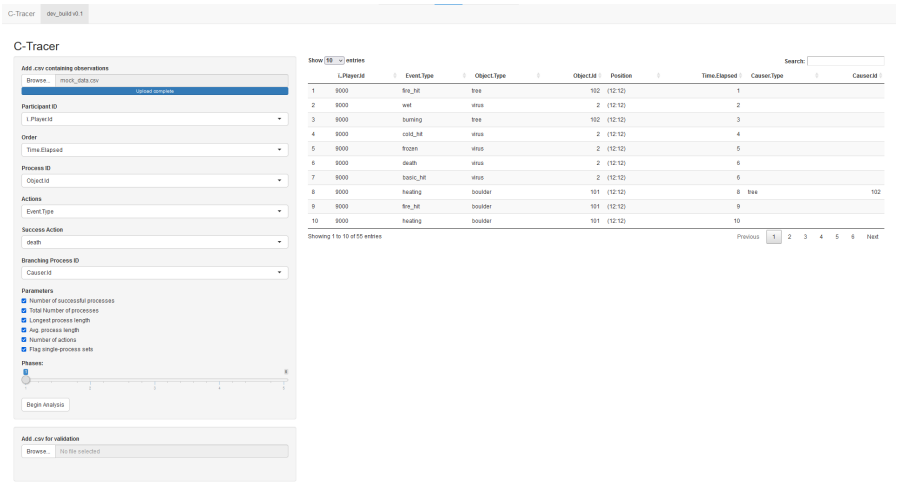

Figure 15: C-Tracer configuration of analysis

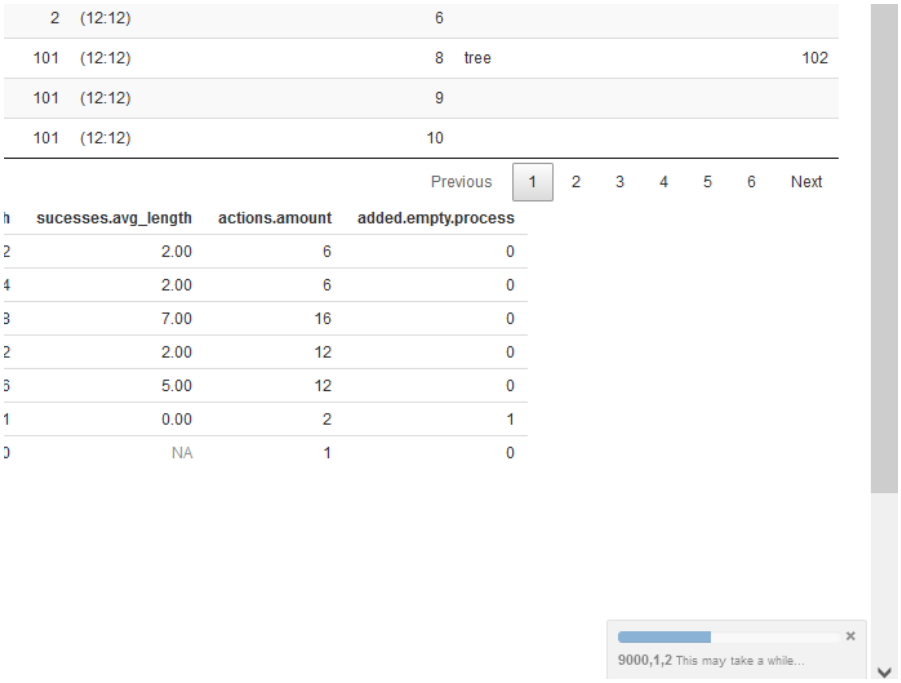

Figure 16: C-Tracer analysis progress visualization

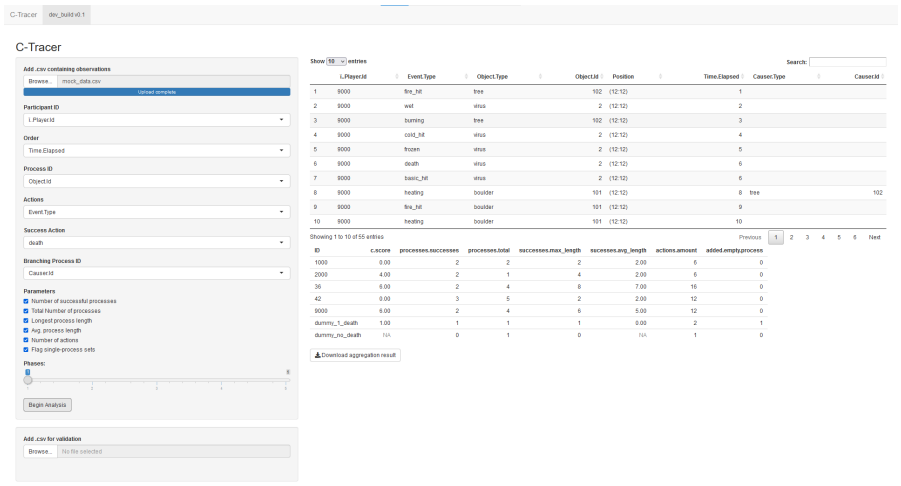

Figure 17: C-Tracer results table

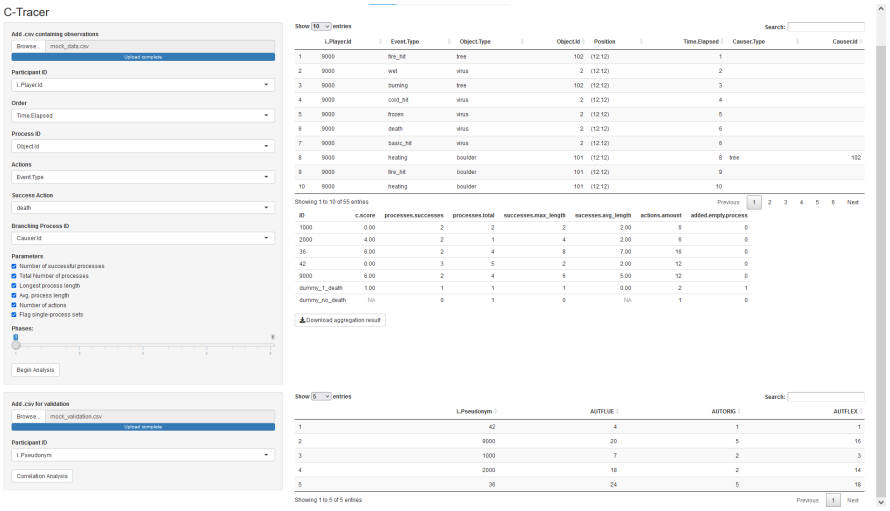

Figure 18: C-Tracer configuration of validation

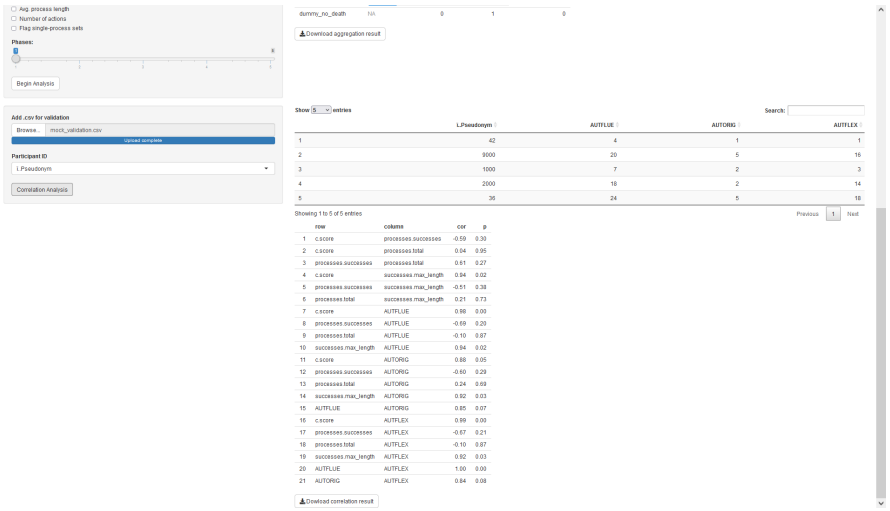

Figure 19: C-Tracer showing validation results

BIBLIOGRAPHY

[1] Wil van der Aalst. *Process Mining*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2016. ISBN: 978-3-662-49850-7 978-3-662-49851-4. DOI: 10.1007/978-3-662-49851-4. URL: http://link.springer.com/10.1007/978-3-662-49851-4 (visited on 04/16/2022).

[2] Anna Abraham. *The Neuroscience of Creativity*. 1st ed. Cambridge University Press, Oct. 25, 2018. ISBN: 978-1-316-81698-1 978-1-107-17646-1 978-1-316-62961-1. URL: https://www.cambridge.org/core/product/identifier/9781316816981/type/book (visited on 06/20/2022).

[3] Robert S. Albert and Mark A. Runco. "A history of research on creativity." In: *Handbook of creativity.* New York, NY, US: Cambridge University Press, 1999, pp. 16–31. ISBN: 0-521-57285-1 (Hardcover); 0-521-57604-0 (Paperback).

[4] Amin Alhashim, Megan Marshall, Tess Hartog, Rafal Jonczyk, Danielle Dickson, Janet van Hell, Gül Okudan-Kremer, and Zahed Siddique. "Work in Progress: Assessing Creativity of Alternative Uses Task Responses: A Detailed Procedure." In: *2020 ASEE Virtual Annual Conference Content Access Proceedings*. 2020 ASEE Virtual Annual Conference Content Access. Virtual Online: ASEE Conferences, June 2020, p. 35612. DOI: 10.18260/1-2--35612. URL: http://peer.asee.org/35612 (visited on 06/20/2022).

[5] Teresa M. Amabile. "A Consensual Technique for Creativity Assessment." In: *The Social Psychology of Creativity*. Ed. by Teresa M. Amabile. Springer Series in Social Psychology. New York, NY: Springer, 1983, pp. 37–63. ISBN: 978-1-4612-5533-8. DOI: 10.1007/978-1-4612-5533-8_3. URL: https://doi.org/10.1007/978-1-4612-5533-8_3 (visited on 05/18/2022).

[6] Baptiste Barbot. "The Dynamics of Creative Ideation: Introducing a New Assessment Paradigm." In: *Frontiers in Psychology* 9 (Dec. 11, 2018), p. 2529. ISSN: 1664-1078. DOI: 10.3389/fpsyg.2018.02529. URL: https://www.frontiersin.org/article/10.3389/fpsyg.2018.02529/full (visited on 06/21/2022).

[7] Christopher M Barlow. *Guilford's Structure of the Intellect*. URL: http://www.cocreativity.com/handouts/guilford.pdf (visited on 05/02/2022).

[8] Roger E. Beaty and Dan R. Johnson. "Automating creativity assessment with SemDis: An open platform for computing semantic distance." In: *Behavior Research Methods* 53.2 (Apr. 2021), pp. 757–780. ISSN: 1554-3528. DOI: 10.3758/s13428-020-01453-w. URL: https://link.springer.com/10.3758/s13428-020-01453-w (visited on 06/21/2022).

[9] Kenes Beketayev and Mark A. Runco. "Scoring divergent thinking tests by computer with a semantics-based algorithm." In: *Europe's Journal of Psychology* 12.2 (May 31, 2016), pp. 210–220. ISSN: 1841-0413. DOI: 10.5964/ejop.v12i2.1127. URL: http://ejop.psychopen.eu/article/view/1127 (visited on 06/22/2022).

[10] Colin F. Camerer et al. "Evaluating the replicability of social science experiments in Nature and Science between 2010 and 2015." In: *Nature Human Behaviour* 2.9 (Sept. 2018), pp. 637–644. ISSN: 2397-3374. DOI: 10.1038/s41562-018-0399-z. URL: http://www.nature.com/articles/s41562-018-0399-z (visited on 06/02/2022).

[11] Shelley H. Carson, Jordan B. Peterson, and Daniel M. Higgins. "Reliability, Validity, and Factor Structure of the Creative Achievement Questionnaire." In: *Creativity Research Journal* 17.1 (Feb. 2005), pp. 37–50. ISSN: 1040-0419, 1532-6934. DOI: 10.1207/s15326934crj1701_4. URL: http://www.tandfonline.com/doi/abs/10.1207/s15326934crj1701_4 (visited on 05/02/2022).

[12] Paul R. Christensen, J. P. Guilford, and R. C. Wilson. "Relations of creative responses to working time and instructions." In: *Journal of Experimental Psychology* 53.2 (1957), pp. 82–88. ISSN: 0022-1015. DOI: 10.1037/h0045461. URL: http://doi.apa.org/getdoi.cfm?doi=10.1037/h0045461 (visited on 10/10/2022).

[13] Giovanni Emanuele Corazza. "Potential Originality and Effectiveness: The Dynamic Definition of Creativity." In: *Creativity Research Journal* 28.3 (July 2, 2016), pp. 258–267. ISSN: 1040-0419, 1532-6934. DOI: 10.1080/10400419.2016.1195627. URL: http://www.tandfonline.com/doi/full/10.1080/10400419.2016.1195627 (visited on 03/16/2022).

[14] Bonnie Cramond. "The Audacity of Creativity Measurement." In: *ECCI XII Proceedings: The ultimate experience in collaboratio*. 1st. 2011. ISBN: 978-989-97569-0-8. (Visited on 05/03/2022).

[15] Genevieve M. Cseh and Karl K. Jeffries. "A scattered CAT: A critical evaluation of the consensual assessment technique for creativity research." In: *Psychology of Aesthetics, Creativity, and the Arts* 13.2 (May 2019), pp. 159–166. ISSN: 1931-390X, 1931-3896. DOI: 10.1037/aca0000220. URL: http://doi.apa.org/getdoi.cfm?doi=10.1037/aca0000220 (visited on 05/18/2022).

[16] Stephen J. Dollinger. "Need for Uniqueness, Need for Cognition, and Creativity." In: *The Journal of Creative Behavior* 37.2 (June 2003), pp. 99–116. ISSN: 00220175. DOI: 10.1002/j.2162-6057.2003.tb00828.x. URL: https://onlinelibrary.wiley.com/doi/10.1002/j.2162-6057.2003.tb00828.x (visited on 06/13/2022).

[17] Denis Dumas and Kevin N. Dunbar. "Understanding Fluency and Originality: A latent variable perspective." In: *Thinking Skills and Creativity* 14 (Dec. 2014), pp. 56–67. ISSN: 18711871. DOI: 10.1016/j.tsc.2014.09.003. URL: https://linkinghub.

elsevier.com/retrieve/pii/S1871187114000480 (visited on 06/27/2022).

[18] Marie J. C. Forgeard and James C. Kaufman. "Who cares about imagination, creativity, and innovation, and why? A review." In: *Psychology of Aesthetics, Creativity, and the Arts* 10.3 (Aug. 2016), pp. 250–269. ISSN: 1931-390X, 1931-3896. DOI: 10.1037/aca0000042. URL: http://doi.apa.org/getdoi.cfm?doi=10.1037/aca0000042 (visited on 05/03/2022).

[19] Boris Forthmann, Dorota M. Jankowska, and Maciej Karwowski. "How reliable and valid are frequency-based originality scores? Evidence from a sample of children and adolescents." In: *Thinking Skills and Creativity* 41 (Sept. 2021), p. 100851. ISSN: 18711871. DOI: 10.1016/j.tsc.2021.100851. URL: https://linkinghub.elsevier.com/retrieve/pii/S1871187121000663 (visited on 05/18/2022).

[20] Emily Frith, Michael J. Kane, Matthew S. Welhaf, Alexander P. Christensen, Paul J. Silvia, and Roger E. Beaty. "Keeping Creativity under Control: Contributions of Attention Control and Fluid Intelligence to Divergent Thinking." In: *Creativity Research Journal* 33.2 (Apr. 3, 2021), pp. 138–157. ISSN: 1040-0419, 1532-6934. DOI: 10.1080/10400419.2020.1855906. URL: https://www.tandfonline.com/doi/full/10.1080/10400419.2020.1855906 (visited on 04/26/2022).

[21] J. P. Guilford. "Creativity." In: *American Psychologist* 5.9 (1950), pp. 444–454. ISSN: 1935-990X, 0003-066X. DOI: 10.1037/h0063487. URL: http://doi.apa.org/getdoi.cfm?doi=10.1037/h0063487 (visited on 07/06/2022).

[22] J. P. Guilford. "The structure of intellect." In: *Psychological Bulletin* 53.4 (1956), pp. 267–293. ISSN: 1939-1455, 0033-2909. DOI: 10.1037/h0040755. URL: http://doi.apa.org/getdoi.cfm?doi=10.1037/h0040755 (visited on 05/02/2022).

[23] J. P. Guilford. "Three faces of intellect." In: *American Psychologist* 14.8 (Aug. 1959), pp. 469–479. ISSN: 1935-990X, 0003-066X. DOI: 10.1037/h0046827. URL: http://doi.apa.org/getdoi.cfm?doi=10.1037/h0046827 (visited on 06/20/2022).

[24] Adam Hampshire, Roger R. Highfield, Beth L. Parkin, and Adrian M. Owen. "Fractionating Human Intelligence." In: *Neuron* 76.6 (Dec. 2012), pp. 1225–1237. ISSN: 08966273. DOI: 10.1016/j.neuron.2012.06.022. URL: https://linkinghub.elsevier.com/retrieve/pii/S0896627312005843 (visited on 05/09/2022).

[25] Dennis Hocevar. "The Development of the Creative Behavior Inventory (CBI)." In: *Annual Meeting of the Rocky Mountain Psychological Association*. 1979, p. 15. URL: https://eric.ed.gov/?id=ED170350.

[26] David J. Hughes, Allan Lee, Amy Wei Tian, Alex Newman, and Alison Legood. "Leadership, creativity, and innovation: A critical review and practical recommendations." In: *The Leadership Quarterly* 29.5 (Oct. 2018), pp. 549–569. ISSN: 10489843. DOI: 10.1016/j.leaqua.2018.03.001. URL: https://linkinghub.elsevier.com/retrieve/pii/S1048984316302582 (visited on 05/03/2022).

[27] Corinna Jaschek, Kim-Pascal Borchart, Julia von Thienen, and Christoph Meinel. *The CollaboUse Test for Automated Creativity Measurement in Individuals and Teams: A Construct Validation Study*.

[28] J.C. Kaufman. "Self-assessments of creativity: Not ideal, but better than you think." In: *Psychology of Aesthetics, Creativity, and the Arts* 13 (2019), pp. 187–192.

[29] James C. Kaufman and John Baer. "Sure, I'm Creative—But Not in Mathematics!: Self-Reported Creativity in Diverse Domains." In: *Empirical Studies of the Arts* 22.2 (July 2004), pp. 143–155. ISSN: 0276-2374, 1541-4493. DOI: 10.2190/26HQ-VHE8-GTLN-BJJM. URL: http://journals.sagepub.com/doi/10.2190/26HQ-VHE8-GTLN-BJJM (visited on 06/13/2022).

[30] James C. Kaufman and John Baer. "Beyond New and Appropriate: Who Decides What Is Creative?" In: *Creativity Research Journal* 24.1 (Jan. 2012), pp. 83–91. ISSN: 1040-0419, 1532-6934. DOI: 10.1080/10400419.2012.649237. URL: http://www.tandfonline.com/doi/abs/10.1080/10400419.2012.649237 (visited on 06/14/2022).

[31] James C. Kaufman, John Baer, Jason C. Cole, and Janel D. Sexton*. "A Comparison of Expert and Nonexpert Raters Using the Consensual Assessment Technique." In: *Creativity Research Journal* 20.2 (May 7, 2008), pp. 171–178. ISSN: 1040-0419, 1532-6934. DOI: 10.1080/10400410802059929. URL: http://www.tandfonline.com/doi/abs/10.1080/10400410802059929 (visited on 05/18/2022).

[32] James C. Kaufman and Ronald A. Beghetto. "Beyond Big and Little: The Four C Model of Creativity." In: *Review of General Psychology* 13.1 (Mar. 2009), pp. 1–12. ISSN: 1089-2680, 1939-1552. DOI: 10.1037/a0013688. URL: http://journals.sagepub.com/doi/10.1037/a0013688 (visited on 06/13/2022).

[33] Kyung Hee Kim. "Can We Trust Creativity Tests? A Review of the Torrance Tests of Creative Thinking (TTCT)." In: *Creativity Research Journal* 18.1 (Jan. 1, 2006), pp. 3–14. ISSN: 1040-0419, 1532-6934. DOI: 10.1207/s15326934crj1801_2. URL: https://www.tandfonline.com/doi/full/10.1207/s15326934crj1801_2 (visited on 05/02/2022).

[34] Kyung Hee Kim. "The Creativity Crisis: The Decrease in Creative Thinking Scores on the Torrance Tests of Creative Thinking." In: *Creativity Research Journal* 23.4 (Oct. 2011), pp. 285–295. ISSN: 1040-0419, 1532-6934. DOI: 10.1080/10400419.2011.

627805. url : http://www.tandfonline.com/doi/abs/10.1080/10400419.2011.627805 (visited on 10/11/2022).

[35] Richard A. Klein et al. "Many Labs 2: Investigating Variation in Replicability Across Samples and Settings." In: Advances in Methods and Practices in Psychological Science 1.4 (Dec. 2018), pp. 443–490. issn: 2515-2459, 2515-2467. doi: 10.1177/2515245918810225. url: http://journals.sagepub.com/doi/10.1177/2515245918810225 (visited on 06/02/2022).

[36] Eva Krebs, Corinna Jaschek, Julia von Thienen, Kim-Pascal Borchart, Christoph Meinel, and Oren Kolodny. "Designing a Video Game to Measure Creativity." In: 2020 IEEE Conference on Games (CoG). 2020 IEEE Conference on Games (CoG). Osaka, Japan: IEEE, Aug. 2020, pp. 407–414. isbn: 978-1-7281-4533-4. doi: 10.1109/CoG47356.2020.9231672. url: https://ieeexplore.ieee.org/document/9231672/ (visited on 07/01/2022).

[37] Christine S. Lee, Anne Corinne Huggins, and David J. Therriault. "A measure of creativity or intelligence? Examining internal and external structure validity evidence of the Remote Associates Test." In: Psychology of Aesthetics, Creativity, and the Arts 8.4 (Nov. 2014), pp. 446–460. issn: 1931-390X, 1931-3896. doi: 10.1037/a0036773. url: http://doi.apa.org/getdoi.cfm?doi=10.1037/a0036773 (visited on 10/21/2022).

[38] Katie Logos, Neil Brewer, and Robyn L. Young. Convergent Validity of a Quick Online Self-Administered Measure of Verbal IQ for Psychology Research. preprint. PsyArXiv, Oct. 1, 2021. doi: 10.31234/osf.io/7csvm. url: https://osf.io/7csvm (visited on 05/09/2022).

[39] Haiying Long. "An Empirical Review of Research Methodologies and Methods in Creativity Studies (2003–2012)." In: Creativity Research Journal 26.4 (Oct. 2, 2014), pp. 427–438. issn: 1040-0419, 1532-6934. doi: 10.1080/10400419.2014.961781. url: http://www.tandfonline.com/doi/abs/10.1080/10400419.2014.961781 (visited on 05/09/2022).

[40] Mark P. J. van der Loo. "The stringdist Package for Approximate String Matching." In: The R Journal 6.1 (2014), pp. 111–122. doi: 10.32614/RJ-2014-011. url: https://doi.org/10.32614/RJ-2014-011.

[41] Yuval Marton, Saif Mohammad, and Philip Resnik. "Estimating semantic distance using soft semantic constraints in knowledge-source-corpus hybrid models." In: Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing Volume 2 - EMNLP '09. the 2009 Conference. Vol. 2. Singapore: Association for Computational Linguistics, 2009, p. 775. isbn: 978-1-932432-62-6. doi: 10.3115/1699571.1699614. url: http://portal.acm.org/citation.cfm?doid=1699571.1699614 (visited on 06/13/2022).

[42] Holly A. McKee. “Automatic Detection of the Flow Mental State in the Context of Creative Collaboration.” Master Thesis. Potsdam, Germany: University of Potsdam, 2022.

[43] Holly A. McKee, Julia von Thienen, Shama Rahman, and Christoph Meinel. “Comparing different forms of automated creativity measurement in the study of individual and collaborative creative writing.” In: MIC conference of creativity. MIC conference of Creativity. Bologna, Italy, Sept. 8, 2021.

[44] Sarnoff Mednick. “The associative basis of the creative process.” In: Psychological Review 69.3 (1962), pp. 220–232. issn: 0033-295X. doi: 10.1037/h0048850. url: http://content.apa.org/journals/rev/69/3/220 (visited on 10/21/2022).

[45] Jay A. Olson, Johnny Nahas, Denis Chmoulevitch, Simon J. Cropper, and Margaret E. Webb. “Naming unrelated words predicts creativity.” In: Proceedings of the National Academy of Sciences 118.25 (June 22, 2021), e2022340118. issn: 0027-8424, 1091-6490. doi: 10.1073/pnas.2022340118. url: https://pnas.org/doi/full/10.1073/pnas.2022340118 (visited on 06/21/2022).

[46] Open Science Collaboration. “Estimating the reproducibility of psychological science.” In: Science 349.6251 (Aug. 28, 2015), aac4716. issn: 0036-8075, 1095-9203. doi: 10.1126/science.aac4716. url: https://www.science.org/doi/10.1126/science.aac4716 (visited on 06/02/2022).

[47] Sidney J. Parnes. “Effects of extended effort in creative problem solving.” In: Journal of Educational Psychology 52.3 (June 1961), pp. 117–122. issn: 1939-2176, 0022-0663. doi: 10.1037/h0044650. url: http://doi.apa.org/getdoi.cfm?doi=10.1037/h0044650 (visited on 10/10/2022).

[48] Elliot Samuel Paul and Scott Barry Kaufman, eds. The Philosophy of Creativity: New Essays. Oxford University Press, May 9, 2014. isbn: 978-0-19-983696-3. doi: 10.1093/acprof:oso/9780199836963.001.0001.

[49] J Peeters. “Re ned Metrics for Measuring Novelty in Ideation.” In: Proceedings of IDMME - Virtual Concept 2010. International conference IDMME – VIRTUAL CONCEPT 2010. 2010, p. 5.

[50] Reza Pishghadam, Tahereh Ghorbani Nejad, and Shaghayegh Shayesteh. “4 Creativity and its Relationship with Teacher Success.” In: Brazilian English Language Teaching Journal 3.2 (Dec. 2012), pp. 204–216.

[51] Jonathan A. Plucker, Ronald A. Beghetto, and Gayle T. Dow. “Why Isn't Creativity More Important to Educational Psychologists? Potentials, Pitfalls, and Future Directions in Creativity Research.” In: Educational Psychologist 39.2 (June 2004), pp. 83–96. issn: 0046-1520, 1532-6985. doi: 10.1207/s15326985ep3902_1. url: http://www.tandfonline.com/doi/abs/10.1207/s15326985ep3902_1 (visited on 07/11/2022).

[52] Jeb S. Puryear, Todd Kettler, and Anne N. Rinn. “Relationships of personality to differential conceptions of creativity: A systematic review.” In: Psychology of Aesthetics, Creativity, and the Arts 11.1 (Feb. 2017), pp. 59–68. issn: 1931-390X, 1931-3896. doi: 10.1037/aca0000079. url: http://doi.apa.org/getdoi.cfm?doi=10.1037/aca0000079 (visited on 05/02/2022).

[53] Greg Richards. “Creativity and tourism.” In: Annals of Tourism Research 38.4 (Oct. 2011), pp. 1225–1253. issn: 01607383. doi: 10.1016/j.annals.2011.07.008. url: https://linkinghub.elsevier.com/retrieve/pii/S0160738311001204 (visited on 05/03/2022).

[54] Mark A. Runco and Garrett J. Jaeger. “The Standard Definition of Creativity.” In: Creativity Research Journal 24.1 (Jan. 2012), pp. 92–96. issn: 1040-0419, 1532-6934. doi: 10.1080/10400419.2012.650092. url: http://www.tandfonline.com/doi/abs/10.1080/10400419.2012.650092 (visited on 04/12/2022).

[55] Mark A. Runco, Garnet Millar, Selcuk Acar, and Bonnie Cramond. “Torrance Tests of Creative Thinking as Predictors of Personal and Public Achievement: A Fifty-Year Follow-Up.” In: Creativity Research Journal 22.4 (Nov. 10, 2010), pp. 361–368. issn: 1040-0419, 1532-6934. doi: 10.1080/10400419.2010.523393. url: http://www.tandfonline.com/doi/abs/10.1080/10400419.2010.523393 (visited on 05/02/2022).

[56] Hessamoddin Sarooghi, Dirk Libaers, and Andrew Burkemper. “Examining the relationship between creativity and innovation: A meta-analysis of organizational, cultural, and environmental factors.” In: Journal of Business Venturing 30.5 (Sept. 2015), pp. 714–731. issn: 08839026. doi: 10.1016/j.jbusvent.2014.12.003. url: https://linkinghub.elsevier.com/retrieve/pii/S0883902614001098 (visited on 05/03/2022).

[57] Jami J. Shah, Steve M. Smith, and Noe Vargas-Hernandez. “Metrics for measuring ideation effectiveness.” In: Design Studies 24.2 (Mar. 2003), pp. 111–134. issn: 0142694X. doi: 10.1016/S0142-694X(02)00034-0. url: https://linkinghub.elsevier.com/retrieve/pii/S0142694X02000340 (visited on 07/25/2022).

[58] Paul J. Silvia. “Discernment and creativity: How well can people identify their most creative ideas?” In: Psychology of Aesthetics, Creativity, and the Arts 2.3 (Aug. 2008), pp. 139–146. issn: 1931-390X, 1931-3896. doi: 10.1037/1931-3896.2.3.139. url: http://doi.apa.org/getdoi.cfm?doi=10.1037/1931-3896.2.3.139 (visited on 06/20/2022).

[59] Paul J. Silvia. “Intelligence and Creativity Are Pretty Similar After All.” In: Educational Psychology Review 27.4 (Dec. 2015), pp. 599–606. issn: 1040-726X, 1573-336X. doi: 10.1007/s10648-015-9299-1. url: http://link.springer.com/10.1007/s10648-015-9299-1 (visited on 05/03/2022).

[60] Paul J. Silvia, Christopher Martin, and Emily C. Nusbaum. "A snapshot of creativity: Evaluating a quick and simple method for assessing divergent thinking." In: *Thinking Skills and Creativity* 4.2 (Aug. 2009), pp. 79–85. ISSN: 18711871. DOI: 10.1016/j.tsc.2009.06.005. URL: https://linkinghub.elsevier.com/retrieve/pii/S1871187109000285 (visited on 06/20/2022).

[61] Paul J. Silvia, Benjamin Wigert, Roni Reiter-Palmon, and James C. Kaufman. "Assessing creativity with self-report scales: A review and empirical evaluation." In: *Psychology of Aesthetics, Creativity, and the Arts* 6.1 (Feb. 2012), pp. 19–34. ISSN: 1931-390X, 1931-3896. DOI: 10.1037/a0024071. URL: http://doi.apa.org/getdoi.cfm?doi=10.1037/a0024071 (visited on 03/16/2022).

[62] Paul J. Silvia, Beate P. Winterstein, John T. Willse, Christopher M. Barona, Joshua T. Cram, Karl I. Hess, Jenna L. Martinez, and Crystal A. Richard. "Assessing creativity with divergent thinking tasks: Exploring the reliability and validity of new subjective scoring methods." In: *Psychology of Aesthetics, Creativity, and the Arts* 2.2 (May 2008), pp. 68–85. ISSN: 1931-390X, 1931-3896. DOI: 10.1037/1931-3896.2.2.68. URL: http://doi.apa.org/getdoi.cfm?doi=10.1037/1931-3896.2.2.68 (visited on 03/16/2022).

[63] Władysław Tatarkiewicz. *A History of Six Ideas*. Dordrecht: Springer Netherlands, 1980. ISBN: 978-94-009-8807-1 978-94-009-8805-7. DOI: 10.1007/978-94-009-8805-7. URL: http://link.springer.com/10.1007/978-94-009-8805-7 (visited on 05/02/2022).

[64] Julia von Thienen, Kim-Pascal Borchart, Corinna Jaschek, Eva Krebs, Justus Hildebrand, Hendrik Ratz, and Christoph Meinel. "Leveraging Video Games to Improve IT-Solutions for Remote Work." In: *2021 IEEE Conference on Games (CoG)*. 2021 IEEE Conference on Games (CoG). Copenhagen, Denmark: IEEE, Aug. 17, 2021, pp. 01–08. ISBN: 978-1-66543-886-5. DOI: 10.1109/CoG52621.2021.9618986. URL: https://ieeexplore.ieee.org/document/9618986/ (visited on 07/01/2022).

[65] E. Paul Torrance. "Predictive Validity of the Torrance Tests of Creative Thinking*." In: *The Journal of Creative Behavior* 6.4 (Dec. 1972), pp. 236–262. ISSN: 00220175. DOI: 10.1002/j.2162-6057.1972.tb00936.x. URL: https://onlinelibrary.wiley.com/doi/10.1002/j.2162-6057.1972.tb00936.x (visited on 05/02/2022).

[66] E. Paul Torrance. "The beyonders in a thirty year longitudinal study of creative achievement." In: *Roeper Review* 15.3 (Feb. 1993), pp. 131–135. ISSN: 0278-3193, 1940-865X. DOI: 10.1080/02783199309553486. URL: http://www.tandfonline.com/doi/abs/10.1080/02783199309553486 (visited on 05/02/2022).

[67] Blaine R. Worthen and Philip M. Clark. "Toward an Improved Measure of Remote Associational Ability." In: *Journal of Educational Measurement* 8.2 (June 1971), pp. 113–123. ISSN: 0022-0655,

1745-3984. DOI: 10.1111/j.1745-3984.1971.tb00914.x. URL: https://onlinelibrary.wiley.com/doi/10.1111/j.1745-3984.1971.tb00914.x (visited on 10/21/2022).

DECLARATION

I hereby declare that this thesis is my own unaided work. All direct or indirect sources used are acknowledged as references.

Potsdam, November 2, 2022

Kim-Pascal Borchart