Computer Science & Information Technology 168

Computer Science & Technology

David C. Wyld,
Dhinaharan Nagamalai (Eds)

Computer Science & Information Technology

- 10th International Conference on Foundations of Computer Science & Technology (FCST 2022), May 21~22, 2022, Zurich, Switzerland
- 10th International International Conference of Managing Information Technology (CMIT 2022)
- 10th International Conference on Software Engineering & Trends (SE 2022)
- 10th International Conference on Signal Image Processing and Multimedia (SIPM 2022)
- 3rd International Conference on Soft Computing, Artificial Intelligence and Machine Learning (SAIM 2022)
- 3rd International Conference on Semantic & Natural Language Processing (SNLP 2022)

Published By

AIRCC Publishing Corporation

Volume Editors

David C. Wyld,
Southeastern Louisiana University, USA
E-mail: David.Wyld@selu.edu

Dhinaharan Nagamalai (Eds),
Wireilla Net Solutions, Australia
E-mail: dhinthia@yahoo.com

ISSN: 2231 - 5403
ISBN: 978-1-925953-67-1
DOI: 10.5121/csit.2022.120801 - 10.5121/csit.2022.120809

Typesetting: Camera-ready by author, data conversion by NnN Net Solutions Private Ltd., Chennai, India

Preface

10th International Conference on Foundations of Computer Science & Technology (FCST 2022), May 21~22, 2022, Zurich, Switzerland, 10th International International Conference of Managing Information Technology (CMIT 2022), 10th International Conference on Software Engineering & Trends (SE 2022), 10th International Conference on Signal Image Processing and Multimedia (SIPM 2022), 3rd International Conference on Soft Computing, Artificial Intelligence and Machine Learning (SAIM 2022), 3rd International Conference on Semantic & Natural Language Processing (SNLP 2022) was collocated with 10th International Conference on Foundations of Computer Science & Technology (FCST 2022). The conferences attracted many local and international delegates, presenting a balanced mixture of intellect from the East and from the West.

The goal of this conference series is to bring together researchers and practitioners from academia and industry to focus on understanding computer science and information technology and to establish new collaborations in these areas. Authors are invited to contribute to the conference by submitting articles that illustrate research results, projects, survey work and industrial experiences describing significant advances in all areas of computer science and information technology.

The FCST 2022, CMIT 2022, SE 2022, SIPM 2022, SAIM 2022 and SNLP 2022. Committees rigorously invited submissions for many months from researchers, scientists, engineers, students and practitioners related to the relevant themes and tracks of the workshop. This effort guaranteed submissions from an unparalleled number of internationally recognized top-level researchers. All the submissions underwent a strenuous peer review process which comprised expert reviewers. These reviewers were selected from a talented pool of Technical Committee members and external reviewers on the basis of their expertise. The papers were then reviewed based on their contributions, technical content, originality and clarity. The entire process, which includes the submission, review and acceptance processes, was done electronically.

In closing, FCST 2022, CMIT 2022, SE 2022, SIPM 2022, SAIM 2022 and SNLP 2022 brought together researchers, scientists, engineers, students and practitioners to exchange and share their experiences, new ideas and research results in all aspects of the main workshop themes and tracks, and to discuss the practical challenges encountered and the solutions adopted. The book is organized as a collection of papers from the FCST 2022, CMIT 2022, SE 2022, SIPM 2022, SAIM 2022 and SNLP 2022.

We would like to thank the General and Program Chairs, organization staff, the members of the Technical Program Committees and external reviewers for their excellent and tireless work. We sincerely wish that all attendees benefited scientifically from the conference and wish them every success in their research. It is the humble wish of the conference organizers that the professional dialogue among the researchers, scientists, engineers, students and educators continues beyond the event and that the friendships and collaborations forged will linger and prosper for many years to come.

David C. Wyld,
Dhinaharan Nagamalai (Eds)

General Chair	Organization
David C. Wyld,	Southeastern Louisiana University, USA
Dhinaharan Nagamalai (Eds)	Wireilla Net Solutions, Australia

Program Committee Members

A. S. M. Sanwar Hosen,	Jeonbuk National University, South Korea
A.Azhagu Jaisudhan Pazhani,	Ramco Institute of Technology, India
Abdalhossein Rezai,	University of Science and Culture, Iran
Abdel-Badeeh M. Salem,	Ain Shams University, Egypt
Abdelhadi Assir,	Hassan 1st University, Morocco
Abdellatif I. Moustafa,	Umm AL-Qura University, Saudi Arabia
Abderrahim Siam,	University of Khenchela, Algeria
Abderrahmane Ez-zahout,	Mohammed V University, Morocco
Abdessamad Belangour,	University Hassan II Casablanca, Morocco
Abhishek Chakraborty,	University of Calcutta, India
Adrian Olaru,	University Politehnica of Bucharest, Romania
Ajit Singh,	Patna University, India
Akhil Gupta,	Lovely Professional University, India
Akhilesh A. Waoo,	AKS University, India
Albert Bakhtizin,	Institute of the Russian Academy of Sciences, Russia
Alessio Ishizaka,	NEOMA Business School, France
Alex Mathew,	Bethany College, USA
Alexander Gelbukh,	Instituto Politécnico Nacional, Mexico
Ali Asif,	Harbin Engineering University, China
Ali Hussein Wheeb,	University of Baghdad, Iraq
Ana Luísa Varani Leal,	University of Macau, China
Anand Nayyar,	Duy Tan University,Viet Nam
Anchit Bijalwan,	Arba Minch University, Ethiopia
Ankur Singh Bist,	Signy Advanced Technology, India
Ann Zeki Ablahd,	Northern Technical University, Iraq
Anouar Abtoy,	Abdelmalek Essaadi University, Morocco
António Abreu,	ISEL, Portugal
Antonios Andreatos,	Hellenic Air Force Academy, Greece
Aridj Mohamed,	Hassiba Benbouali University Chlef, Algeria
Arnold Kwofie,	University for Development Studies, Ghana
Arun Malik,	Lovely Professional University, India
Assem Abdel Hamied Moussa,	Chief Eng, Egypt
Auwal Salisu Yunusa,	Kano State Polytechnic, Nigeria
Azah Kamilah Muda,	UTeM, Malaysia
B Nandini,	Telangana University, Nizamabad
Balram Yadav,	Mahakal Institute of Technology, India
Benyettou Mohammed,	University center of Relizane, Algeria
Beshair Alsiddiq,	Riyad Bank, Saudi Arabia
Bouchra Marzak,	Hassan II University, Morocco
Boukari nassim,	Skikda univerity, Algeria
Brahim Lejdel,	University of El-Oued, Algeria
Cagdas Hakan Aladag,	Hacettepe University, Turkey
Chahinez Mérièm Bentaouza,	Mostaganem University, Algeria
Cheman Shaik,	Collabera Software Solutions, United States

Ke-Lin Du,	Concordia University, Canada
Kenjiro T. Miura,	Shizuoka University, Japan
khaled Osama Elzoghaly,	Alexandria University, Egypt
Khurram Hameed,	Edith Cowan University, Australia
Kiril Alexiev,	Bulgarian Academy of Sciences, Bulgaria
Koh You Beng,	University of Malaya, Malaysia
Kolla Bhanu Prakash,	KL University, India
Li Yan,	Xi’an Polytechnic University, China
Loc Nguyen,	Independent scholar, Vietnam
Loc Nguyen,	Loc Nguyen's Academic Network, Vietnam
Luisa Maria Arvide Cambra,	University of Almeria, Spain
M V Ramana Murthy (R),	Osmania University, India
M. Zakaria Kurdi,	University of Lynchburg, VA, USA
M.Suresh,	Kongu Engineering College, India
MA.Jabbar,	Vardhaman College of Engg. Hyderabad, India
Mabroukah A. M. Amarif,	Sebha University Of Libya, Libya
Mabroukah Amarif,	Sebha University, Libya
Mahdi Sabri,	Islamic Azad University, Iran
Mallikharjuna Rao K,	IIIT Naya Raipur, India
Manish Kumar Mishra,	University of Gondar, Ethiopia
Marcin Paprzycki,	Polish Academy of Sciences, Poland
Mario Versaci,	DICEAM - Univ. Mediterranea, Reggio Calabria
Masoomeh Mirrashid,	Semnan University, Iran
Mayssa Frikha,	University of Sfax, Tunisia
Mehdi Nezhadnaderi,	Islamic Azad University, Iran
Mihai Carabas,	University Politehnica of Bucharest, Romania
Mihai Horia Zaharia,	"Gheorghe Asachi" Technical University, Romania
Ming-An Chung,	National Taipei University of Technology, Taiwan
Mohamed ali el sayed fahim,	benha university, Egypt
Mohamed Anis Bach Tobji,	University of Manouba, Tunisia
Mohammad Siraj,	King Saud University, Saudi Arabia
Mohammed Akour,	Prince Sultan University,Saudi Arabia
Mohammed Benyettou,	University Center of Relizane, Algeria
Mohammed Bouhorma,	Abdelmalek Essaadi University, Morocco
Mohd Rafi Adzman,	Universiti Malaysia Perlis, Malaysia
Mueen Uddin,	Universiti Brunei Darussalam, Brunei Darussalam
Muhammad Mursil,	Northeastern University, China
Mu-Song Chen,	Da-Yeh University, Taiwan
MV Ramana Murthy,	Osmania University, India
Nadia Abd-Alsabour,	Cairo University, Egypt
Nikola Ivković,	University of Zagreb, Croatia
Nour El Houda Golea,	Batna 2 University, Algeria
Oleksii K. Tyshchenko,	University of Ostrava, Czech Republic
Otilia Manta,	Romanian –American University, Romania
Ouided Sekhri,	Constantine 1 University, Algeria
P. Susheelkumar S,	University of Mumbai, India
P.Gunasekaran,	Ramco Institute of Technology, India
P.V.Siva Kumar,	VNR VJIET, India
Piotr Malak,	University of Wroclaw, Poland
R.Arthi,	SRM Institute of Science and Technology, India
Rabah Mohammed Amin,	Algerian Space Agency, Algeria
Radu Vasiu,	Politehnica University of Timisoara, Romania

Rajeev Kanth,	University of Turku, Finland
Ramadan Elaiess,	University of Benghazi, Libya
Ramgopal Kashyap,	Amity University Chhattisgarh, India
Rana Mukherji,	The ICFAI University, India
Richa Purohit,	DY Patil International University,India
Ruksar Fatima,	Khaja Bandanawaz University, India
S.Thenmalar,	SRM Institute of Science and Technology, India
Saad Aljanabi,	Alhikma College University, Iraq
Sabyasachi Pramanik,	Haldia Institute of Technology, India
Sachin Kumar,	Kyungpook National University, South Korea
Saeed Iranmanesh,	Shahid Bahounar University of Kerman, Iran
Samarendra Nath Sur,	Sikkim Manipal Institute of Technology, India
Samrat Kumar Dey,	Dhaka International University, Bangladesh
Satish Gajawada,	IIT Roorkee, India
Sayali Kulkarni,	IIT Bombay, India
Sebastian Floerecke,	University of Passau, Germany
Seema Verma,	Banasthali University, India
Shahram Babaie,	Islamic Azad University, Iran
Shahzad Ashraf,	Hohai University, P.R China
Shashikant Patil,	ViMEET Khalapur Raigad MS India, India
Shi Dong,	Zhoukou Normal University, China
Shivanand Gornale,	Rani Channamma University Belagavi, India
Shufeng Li,	Communication University of China, China
Siarry Patrick,	Universite Paris-Est Creteil, France
Siddhartha Bhattacharyya,	Rajnagar Mahavidyalaya, India
Sidi Mohammed Meriah,	University of Tlemcen, Algeria
Smain Femmam,	UHA University, France
Smaranda Belciug,	University of Craiova, Romania
Solomiia Fedushko,	Lviv Polytechnic National University, Ukraine
Stelios Krinidis,	International Hellenic University, Greece
Suhad Faisal,	University of Baghdad, Iraq
Taha Mohammed Hasan,	University of Diyala, Iraq
Taleb zouggar souad,	Oran 2 university, Algeria
Tan Tse Guan,	Universiti Malaysia Kelantan, Malaysia
Usman Naseem,	University of Sydney, Australia
Varun Shukla,	Pranveer Singh Institute of Technology, India
Venkata Duvvuri,	Oracle Corp & Purdue University, USA
Wahbi Azeddine,	Hassan II University, Morocco
Wenwu Wang,	University of Surrey, UK
William R. Simpson,	Institute for Defense Analyses, USA
WU Yung Gi,	Chang Jung Christian University, Taiwan
Yew Kee Wong,	Huang Huai University, China
Yi Lou,	Harbin Engineering University, China
Yousef Farhaoui,	Moulay Ismail University, Morocco
Youye Xie,	Colorado School of Mines, USA
Yu-Chen Hu,	Providence University, Taiwan
Yuping Fan,	Illinois Institute of Technology, USA
Yuping Yan,	ELTE, Hungary
Zakaria Kurdi,	University of Lynchburg, USA
Zhilong Wang,	The Pennsylvania State University, USA
Zhou RouGang,	HangZhou DianZi University, China
Zoran Bojkovic,	University of Belgrade, Serbia

Technically Sponsored by

Computer Science & Information Technology Community (CSITC)

Artificial Intelligence Community (AIC)

Soft Computing Community (SCC)

Digital Signal & Image Processing Community (DSIPC)

10th International Conference on Foundations of Computer Science & Technology (FCST 2022)

10th International International Conference of Managing Information Technology (CMIT 2022)

10th International Conference on Software Engineering & Trends (SE 2022)

10th International Conference on Signal Image Processing and Multimedia (SIPM 2022)

3rd International Conference on Soft Computing, Artificial Intelligence and Machine Learning (SAIM 2022)

3rd International Conference on Semantic & Natural Language Processing (SNLP 2022)

THE PROBLEM OF ERROR FREQUENCY DISTRIBUTION IN THE MILLER-RABIN TEST FOR TRIPLEPRIME NUMBERS

Alisher Zhumaniezov

Kazan Federal University, Kazan, Russian Federation

ABSTRACT

This article investigates the error distribution of the Miller-Rabin test for the class of tripleprime numbers. At first the current results on the class of semiprimes are presented. Further, a theoretical estimation of the average frequency for triple prime numbers on an interval is derived, and a comparative analysis with a practical result is demonstrated. Graphs and intermediate conclusions accompany all comparisons. A conclusion is also made about a possible direction for improving this estimation.

KEYWORDS

Miller-Rabin test, strong pseudoprime, number theory, frequency distribution.

1. INTRODUCTION

Prime numbers are a key element in the design of cryptographic protocols. Therefore, it is very necessary to find a sufficiently large prime number quickly and efficiently. The easiest way is to iterate the numbers in a given interval and check for primality. Thus, the question arises of checking an arbitrary number for primality.

There are several approaches to checking whether a number is prime:

1. Search of divisors[1] – a deterministic algorithm that gives an answer in a limited time. The main disadvantage is a long running time, exponential dependence on the length of the number.

2. Fermat's test[2] – a probabilistic test based on Fermat's little theorem. The main disadvantage is the Carmichael numbers passing the test with an falsely primality verdict, and their infinite number.[3]

3. The Miller-Rabin test[4][5] – the most widely used probabilistic test. Is an improvement on the Fermat test.

There are many other primality tests, but they have not been considered in this paper.

The Miller-Rabin test is a probabilistic test, which means that this test may falsely conclude that a number is prime. Therefore, the problem arises of estimating the probability of such an error in order to evaluate the efficiency of the algorithm.

David C. Wyld et al. (Eds): FCST, CMIT, SE, SIPM, SAIM, SNLP - 2022
pp. 01-12, 2022. CS & IT - CSCP 2022 DOI: 10.5121/csit.2022.120801

For example, the article [6] provides an algorithm for additional verification of numbers that have passed the Miller-Rabin test with base 2. To correctly estimate its running time, it is necessary to know the distribution of strong pseudoprimes over this base.

Also, article [7] provides an algorithm for finding prime numbers by pattern. To estimate its running time, knowledge about the distribution of strong pseudoprimes was also needed.

In this paper, we present theoretical calculations for the average error of the Miller-Rabin test on the interval $[1, X]$. As part of the work, current estimation for numbers enclosed in parentheses and set on the right margin. For example, we present theoretical upper bound for semiprimes

$$n = pq \tag{1}$$

and calculate theoretical upper bound for tripleprimes

$$n = pqr \tag{2}$$

After the derivation of the upper bound, we perform practical calculations, according to which make conclusions about the closeness of the upper estimation to practical results and the need to improve the estimation.

This paper contains the following Sections: Section 2 present problem statement, all necessary definitions and current results of research. Sections 3 present obtaining of theoretical estimation for upper bound of average error on interval. Section 4 demonstrate visual comparison theoretical function and practical values on several parameters and made intermediate conclusions. Section 5 present final conclusion about the results.

2. METHODS

2.1. Miller-Rabin Test

There are many algorithms for checking the primality of a number. As part of this work, the Miller-Rabin probabilistic test will be analyzed. This test was developed in 1976 by G. Miller in the article [4]. Its modification was presented in 1980 by M. Rabin in the article [5]. This algorithm is based on Euler's theorem [8]:

$$a^{n-1} \equiv 1 (\bmod n), where\, n\, is\, a\, prime\, number \tag{3}$$

To describe the algorithm, we introduce the following definitions:

Definition 1. Let n be an arbitrary natural number, then we define the functions $bin(n)$ and $odd(n)$ as follows:

$$\begin{gathered} n = 2^s u, where\, u\, is\, odd \\ bin(n) = s \\ odd(n) = u \end{gathered} \tag{4}$$

Definition 2. Let n is an arbitrary natural number, a is a natural number belonging to the interval $[1, n-1]$. Call a is witness of primality if one of the following conditions is met:

$$\begin{gathered} a^{odd(n-1)} \equiv 1 (\bmod n) \\ \exists (0 \le i < bin(n-1)) \Big| (a^{odd(n-1)})^{2^i} \equiv -1 (\bmod n) \end{gathered} \qquad (5)$$

Thus, the final algorithm for checking for the primality of a number is as follows [5]:

```
Perform k iterations of the test:
  Choose a random number a.
  Find d = gcd(a,n). if d ≠ 1, then n is composite.
Check whether conditions from (5) are satisfied:
    If none of the conditions is met, then n is a composite.
Otherwise - probably prime.
```

The verdict "probably prime" means that there is a composite number that will pass the test as a prime number. Such numbers are called strong pseudoprime. The study of the distribution of strong pseudoprimes is important for evaluating the efficiency of the algorithm.

One of the approaches to estimating the distribution of strictly pseudoprimes is sequence ψ_n – smallest strong pseudoprime to n first prime numbers as bases. Nowadays known values are for $1 \le n \le 13$ [9][10]. There is also conjecture for values $14 \le n \le 19$ [11]. However, in present work we use another approach.

2.2. Number of witnesses to the primality of the number

An important characteristic for estimating the error of the Miller-Rabin test is the number of witnesses to the primality of an arbitrary number. This value allows us to estimate the probability of choosing a witness of primality for a composite number, and, consequently, a falsely conclusion.

In the article [12], a necessary and sufficient condition was presented that allows finding the exact number of primality witnesses for an arbitrary number:

$$\begin{gathered} n = uv, where\, u\, and\, v\, are\, coprime \\ ord_u(a) \,|\, GCD(\varphi(u), (u - \varphi(u))v - 1) \\ ord_v(a) \,|\, GCD(\varphi(v), (v - \varphi(v))u - 1) \\ bin(ord_u(a)) = bin(ord_v(a)) \end{gathered} \qquad (6)$$

We define by $W(n)$ the number of witnesses to the primality of n. The first formula for $W(n)$ was also presented in [12], but only for semiprime numbers n from (1).

$$W(n) = odd(d)^2 \frac{4^{bin(d)}+2}{3}, where\, d = GCD(p-1, q-1) \tag{7}$$

In [13], a finite formula was presented for an arbitrary number by its decomposition into prime factors:

$$\begin{gathered} n = p_1^{r_1} * p_1^{r_1} * ... * p_k^{r_k} \\ d_i = GCD(p_i - 1, \frac{n}{p_i^{r_i}} - 1) \\ s = \min(bin(d_i)) \\ W(n) = \prod_{i=1}^{k} (odd(d_i)) * (1 + \sum_{j=0}^{s-1} 2^{kj}) \end{gathered} \tag{8}$$

2.3. Average frequency distribution

After introducing utility definitions and formulas, we define the function for calculations. Since in the Miller-Rabin test one of the checks is the calculation of the GCD of a number and base, then only numbers coprime with n can be witnesses of primality. Hence:

$$W(n) \le \varphi(n) \tag{9}$$

Thus, as the frequency of witnesses, we will define:

$$Fr(n) = \frac{W(n)}{\varphi(n)} \tag{10}$$

The maximum value 1 is reached if and only if n is prime. This follows from Rabin's theorem presented in [5]:

$$W(n) \le \frac{\varphi(n)}{4}, where\, n\, is\, a\, composite\, number \tag{11}$$

The value of the $Fr(n)$ function can also be interpreted as the probability of successfully passing one iteration of the Miller-Rabin test. In this case, the probability that the composite number n is probably prime after k iterations is $\frac{1}{4^k}$. Subsequently, this estimation was improved in [12] to $\frac{1}{16^k}$.

Since the function $Fr(n)$ has no limit and reaches a maximum of $\frac{1}{4}$ on an infinite number of composite numbers, we will estimate the distribution for $Avg(Fr(n))$ – the average frequency of witnesses on the interval $[1, X]$.

Article [14] presents estimation:

$$Avg(Fr(n)) < \frac{1}{\sqrt{X}} \tag{12}$$

Article [5] presents estimation:

1) for the case n from (1) and

$$\begin{gathered} q = (p-1)k+1 \\ Avg(Fr(n)) = \frac{p^2}{2X}\ln(\ln(X))\ln(X) \end{gathered} \tag{13}$$

2) for the case n from (1) and

$$\begin{gathered} q = 2k+1, where\, 2k \bmod (p-1) \neq 0 \\ Avg(Fr(n)) = \frac{2}{X}\ln(\ln(X))\ln(X) \end{gathered} \tag{14}$$

3) for the general case from (1)

$$E(Avg(Fr(n))) = \frac{2p}{X}\ln(\ln(X))\ln(X) \tag{15}$$

2.4. Information content

Since the average probabilities in practice are extremely small, we use the information content from [15] for the "probably prime" event to compare the theoretical and practical estimations:

$$I = \log(\frac{1}{p}), where\ p - event\ probability \tag{16}$$

3. RESULTS AND DISCUSSION

3.1. Estimating the number of witnesses

At first, we define d_1, d_2, d_3 and s for n from (2) where p, q, r are distinct prime numbers:

$$\begin{gathered} d_1 = GCD(p-1, qr-1) \\ d_2 = GCD(q-1, pr-1) \\ d_3 = GCD(r-1, pq-1) \\ s = \min(bin(d_1), bin(d_2), bin(d_3)) \end{gathered} \tag{17}$$

After that, we get the formula for the number of witnesses of the primality of number n from (2) where p, q, r are distinct prime numbers. Next, we calculate the inner sum in (8) through a geometric progression and get:

$$W(n) = odd(d_1) * odd(d_2) * odd(d_3) * \frac{8^s + 6}{7} \tag{18}$$

From the definition of s and the properties of the GCD it follows:

$$odd(d_1) = \frac{d_1}{2^{bin(d_1)}} \le \frac{d_1}{2^s} \le \frac{p-1}{2^s} \tag{19}$$

$$odd(d_2) = \frac{d_2}{2^{bin(d_2)}} \le \frac{d_2}{2^s} \le \frac{q-1}{2^s} \tag{20}$$

$$odd(d_3) = \frac{d_3}{2^{bin(d_3)}} \le \frac{d_3}{2^s} \le \frac{pq-1}{2^s} \tag{21}$$

But we can improve multiplication of the inequalities from (19), (20) and (21) by this theorem: For any d_1, d_2, d_3 and s, defined as in (17) this inequality is satisfied:

$$odd(d_1) * odd(d_2) * odd(d_3) \le \frac{(p-1)(q-1)(pq-1)}{2^{3s+1}} \tag{22}$$

Since the numbers $p-1$, $q-1$ and $pq-1$ are always even, then $s \ge 1$. Substituting this property and inequality from (22) into (18) we obtain:

$$W(n) \le \frac{(p-1)(q-1)(pq-1)(1+\frac{6}{8^1})}{14} \le \frac{(p-1)(q-1)(pq-1)}{8} \tag{23}$$

Thus, we obtain an estimation for the number of witnesses of the primality.

3.2. Estimating the frequency of witnesses

Substituting the inequality from (23) into (10) we get:

$$Fr(n) = \frac{W(n)}{\varphi(pqr)} \le \frac{(p-1)(q-1)(pq-1)}{8(p-1)(q-1)(r-1)} = \frac{pq-1}{8(r-1)} \tag{24}$$

Thus, we obtain an estimation for the frequency of witnesses to the primality of an arbitrary tripleprime number n from (2).

3.3. Estimating the average frequency of witnesses

To calculate the average frequency of witnesses, we fix prime numbers p and q and introduce the parameter y. We define by $S_{pq}(y)$ the sum of the frequencies of witnesses for all numbers n from (2), where r is a prime number on the interval $[\max(p,q)+1, y]$, and by $\pi_p(y)$ the number of prime numbers on the interval $[p+1, y]$.

Then we found the average frequency of witnesses on the segment $[1, pqy]$ by the formula:

$$Avg(Fr(n)) = \frac{S_{pq}(y)}{\pi_{\max(p,q)}(y)} \tag{25}$$

At first, we estimate $S_{pq}(y)$:

$$S_{pq}(y) \leq \sum_{r \leq y} \frac{pq-1}{8(r-1)} \square \frac{pq-1}{8} \sum_{r \leq y} \frac{1}{r} \square \frac{pq-1}{8} \ln(\ln(y)) \tag{26}$$

After that, we estimate $\pi_p(y)$:

$$\pi_p(y) \square \frac{y}{\ln(y)} - \frac{p}{\ln(p)} \square \frac{y}{\ln(y)} \tag{27}$$

We denote the upper bound by X , then:

$$y \leq \frac{X}{pq} \tag{28}$$

Substituting inequalities from (26), (28) and estimation from (27) into (25) we obtain an upper bound for $Avg(Fr(n))$. Then:

$$Avg(Fr(n)) \leq \frac{(pq)^2 \ln(X)\ln(\ln(X))}{8X} \tag{29}$$

Thus, we obtained an estimation for the average frequency of witnesses to the primality of an arbitrary tripleprime number n from (2) on the interval $[1, X]$:

$$Avg(Fr(n)) \leq C_{pq} \frac{\ln(X)\ln(\ln(X))}{X} \tag{30}$$

3.4. Convergence to upper bound

Since the inequalities in (19), (20) and (21) can be strengthened for an infinite number of numbers, the final upper bound is inaccurate and needs to be improved. However, we will try to consider the subproblem of finding $Avg'(Fr(n))$ – the average frequency of witnesses on the interval $[1, X]$ of tripleprimes number from (2), which satisfy the following conditions:

$$p-1 = 2p', where\ p' - prime\ number \tag{31}$$

$$q-1 = 2q', where\ q' - prime\ number \tag{32}$$

$$bin(pq-1) = 2 \tag{33}$$

$$\begin{gathered} rq \equiv 1 (\bmod\ p') \\ rp \equiv 1 (\bmod\ q') \\ r \equiv 1 (\bmod\ pq-1) \end{gathered} \tag{34}$$

There are infinitely many such triples p, q and r, since the equation (32) has exactly one solution r_0 on the segment $[1, p'*q'*(pq-1)]$. For example, if $p=7$, $q=11$ then $r_0=533$. Then all solutions of (32) will be $r=r_0+p'*q'*(pq-1)k$, where k – any integer. Thus, we can calculate $Avg'(Fr(n))$ over an arbitrarily large segment. In this case upper bound from (29) can be improved to close enough result:

$$Avg'(Fr(n)) \le \frac{pq(pq-1)}{8}\frac{\ln(X)(\ln(\ln(X))-\ln(\ln(y_0)))}{X} \qquad (35)$$

, where y_0 - minimum value r for fixed p and q. For example, if $p=7$, $q=11$ then $y_0=8513$.

4. Summary

4.1. Comparison of theoretical and practical evaluation

After finding the theoretical estimates, we compare the result with practice for several points:

1. The value of the coefficient C_{pq} through the formula (30).
2. The ratio of both sides of the inequality in (29).
3. Comparison of both sides of the inequality in (29).
4. Convergence to upper bound from inequality in (35).
5. General conclusions.

4.2. Value of the coefficient C$_{pq}$

Figure 1 shows that the dependence of the coefficient is non-linear and not even monotonous. However, it can be noted that the type of dependence itself does not depend on the boundary of the segment, but only on pq. Also, it does not reach the theoretical value, so we can conclude that it is possible to improve the value of the coefficient C_{pq}.

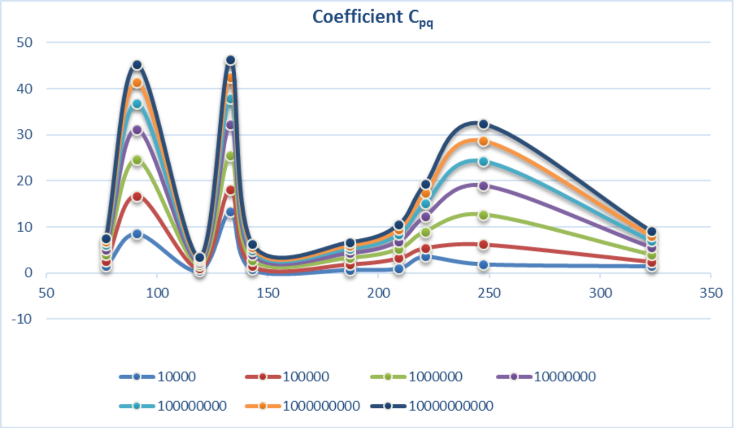


Figure 1. Estimation of the coefficient value (dependence on *pq*).

Figure 2 shows that the value of the coefficient increases monotonically, but does not exceed the theoretical value. So, we can make an assumption about approaching the theoretical value. However, due to the fact that there is no obvious slowdown in growth, it is difficult to conclude that there is better upper bound.

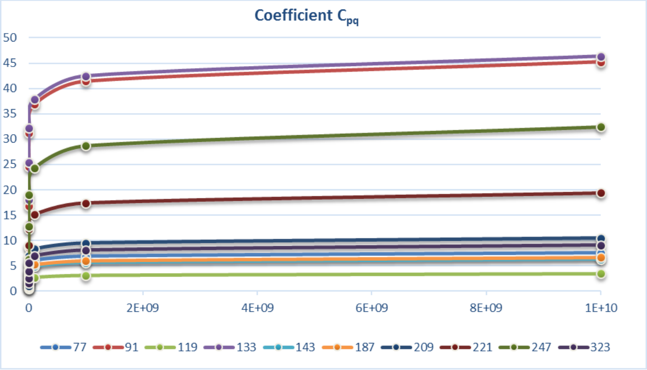


Figure 2. Estimation of the coefficient value (dependence on the boundary *X*).

Figure 3 shows that the value of the coefficient increases monotonically, but the growth rate gradually slows down. From this, we can make an assumption about the existence of an upper bound.

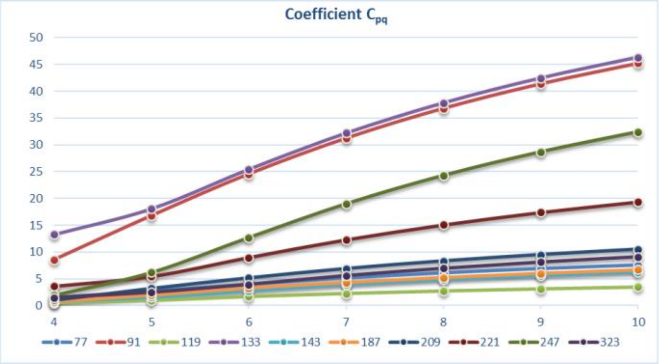


Figure 3. Estimation of the coefficient value (dependence on the logarithm of boundary *X*).

4.3. Function ratio

As we can see from the Figure 4 and Figure 5, the ratio between the functions very slowly approaches value 1, but does not reach. Also, it seems the limit of the sequence is not 1. It means that the resulting estimation is upper bound but not accurate and needs improvement.

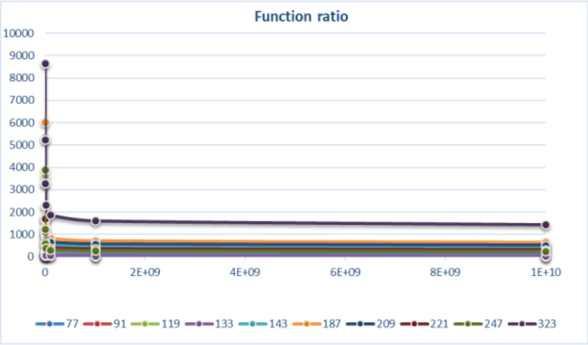


Figure 4. Function ratio (dependence on the boundary *X*, theoretical/practical).

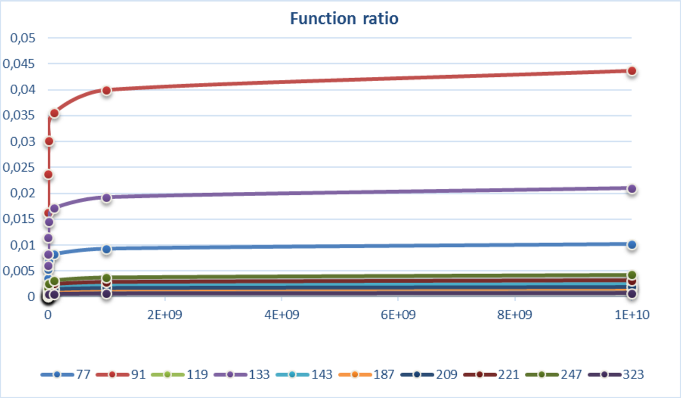


Figure 5. Function ratio (dependence on the boundary *X*, practical/theoretical).

4.4. Function comparison

We can see from the Fig. 6, which present function comparison for $pq = 77$, that the theoretical estimate is an upper bound for $Fr(n)$. We can notice that distance between the practical values and theoretical ones doesn't change much, so it can be assumed that the dependency type for the upper bound was found correctly. However, the theoretical function quite far from the practical one, which indicates an inaccurate finding of the coefficient C_{pq}.

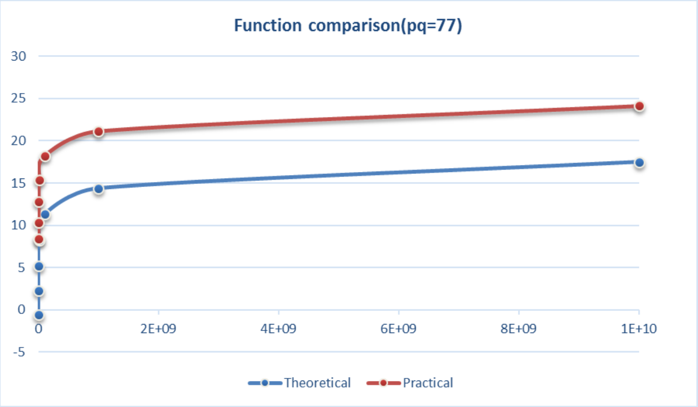


Figure 6. Comparison of theoretical and practical evaluation (dependence on the boundary *X*, *pq=77*).

4.5. Convergence to upper bound

We can see from the Fig. 7, that ratio very quickly reach value 1. It means that the resulting function is very accurate approximation of practical value. Also, we can see that after $\lg(X) = 7$ ratio become less than 1. It means the resulting function is upper bound for practical value.

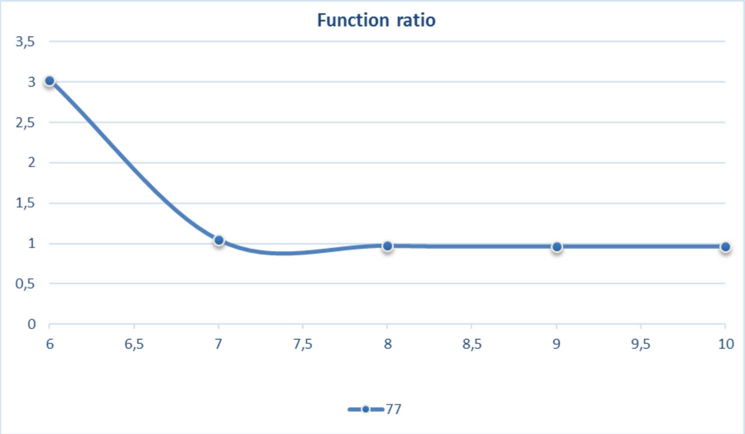


Figure 7. Function ratio (dependence on the logarithm of boundary *X*, practical/theoretical).

4.4. General conclusions

As we can see from comparison of theoretical and practical values, upper bound from (29) doesn't approximate $Avg(Fr(n))$ and needs improvement. However, dependency type for the upper bound as in (30) was found correctly. It means that the main improvement must be decrease of value C_{pq}.

But for subclass of tripleprime numbers with properties (31), (32), (33) and (34) we found very accurate approximation of $Avg'(Fr(n))$. One use of this estimation can be to improve the estimation for the whole class of tripleprime numbers.

5. Conclusions

In this article, we review current results for an upper bound for the average probability of error of the Miller-Rabin test. Also, we calculate new estimation for class tripleprimes numbers and made a comparison with practical results.

Conclusions were drawn about the correctness of the type of distribution of the theoretical estimation. However, it was found that the value of the coefficient C_{pq} is too high. All conclusions were accompanied by graphs for visual demonstration.

Also, we found very accurate approximation of the average probability of error of the Miller-Rabin test for some subclass of tripleprime numbers.

All our conclusions accompanied with graphs for more clarity.

Therefore, a further direction for investigation may be to attempt to decrease the value of the C_{pq} coefficient in order to obtain a more accurate upper bound.

ACKNOWLEDGEMENTS

The authors would like to thank everyone, just everyone!

REFERENCES

[1] Childs, N. Lindsay (2009) *A Concrete Introduction to Higher Algebra*, 3rd edition, Springer Publisher.

[2] Cormen, H. Thomas & Leiserson, E. Charles & Rivest, L. Ronald & Stein, Clifford (2001) *Introduction to Algorithms*, 2nd edition, MIT Press.

[3] Alford, W. R. & Granville, Andrew & Pomerance, Carl (1994) “There are Infinitely Many Carmichael Numbers”, Princeton University *Annals of Mathematics*, Vol. 140, No. 3, pp703-722.

[4] Miller, G. (1976) “Riemann’s hypothesis and tests for primality”, Elsevier *Journal of Computer and System Sciences*, Vol. 13, pp300-317.

[5] Rabin, M. O. (1980) “Probabilistic algorithm for testing primality”, Elsevier *Journal of Number Theory*, Vol. 12, No. 1, pp128–138.

[6] Nari, Kubra & Ozdemir, Enver & Ozkirisci, Neslihan Aysen (2019) *Strong pseudo primes to base 2*, arXiv Publisher, arXiv:1905.06447.

[7] Sorenson, P. Jonathan & Webster, Jonathan (2019) *Two Algorithms to Find Primes in Patterns*, arXiv Publisher, arXiv:1807.08777.

[8] Ribenboim, Paulo (1995) *The New Book of Prime Number Records*, 3rd edition, Springer Publisher.

[9] Zhang, Zhenxiang & Tang, Min (2003) “Finding strong pseudoprimes to several bases”, American Mathematical Society *Mathematics of Computation*, Vol. 72, No. 244, pp2085-2097.

[10] Sorenson, Jonathan & Webster, Jonathan (2015) “Strong Pseudoprimes to Twelve Prime Bases”, American Mathematical Society *Mathematics of Computation*, Vol. 86, No. 304, pp985-1003.

[11] Zhang, Zhenxiang (2007) “Two kinds of strong pseudoprimes up to 10^36”, American Mathematical Society *Mathematics of Computation*, Vol. 76, No. 260, pp2095-2107.

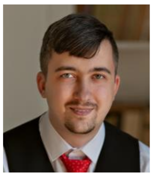
[12] Ishmukhametov, S. T. & Mubarakov, B G. & Rubtsova, R G. (2020) “On the Number of Witnesses in the Miller–Rabin Primality Test”, MDPI *Symmetry*, Vol. 12, No. 6, p890.

[13] Mubarakov, B G. (2020) “Efficient evaluation of the Miller-Rabin primality test of natural numbers”, Kazan Mathematical Society *Proceedings of the N.I. Lobachevsky Mathematical Center*, Vol. 59, pp106-109.

[14] Mubarakov, B G. (2021) “On the Number of Primality Witnesses of Composite Integers”, Springer *Russian Mathematics*, Vol. 65, No. 9, pp73–77.

[15] Hartley, R.V.L. (1928) “Transmission of information”, IEEE The Bell System Technical Journal, Vol. 7, No. 3, pp535-563

AUTHORS



Zhumaniezov Alisher Assistant professor at Kazan Federal University. Graduated from Kazan Federal University and Czech State University in 2018. Has secondary job in the Laboratory of Medical Cybernetics and Machine Vision. Preferred areas: Cryptography, Machine vision.

AN INTELLIGENT MOBILE APPLICATION TO AUTOMATE THE CONVERSATION OF EMAILS TO TASK MANAGEMENT USING AI AND MACHINE LEARNING

Yi Li[1] and Yu Sun[2]

[1]Seattle Academy, 1201 E Union St, Seattle, WA 98122
[2]California State Polytechnic University,
Pomona, CA, 91768, Irvine, CA 92620

ABSTRACT

It is no secret that a large portion of our population struggle with task management [1]. According to a 2021 research, twenty five percent of people do not employ a task management system and simply work on "whatever seems the most important at the moment". Among the population that do use some sort of task management, the most popular form of managing personal tasks is through to do lists (33%) followed by using their email inbox (24%). Thus, I thought to combine these two most common methods by creating a to do list automatically generated from the email. We used the open sourced software natural language processing (NLP) to pick out the important sentences in the text and convert them into tasks for the to-do list. We used keywords such as "tomorrow", "next", "month", etc combined with the date the email was sent to determine the "due date" of the to do list. Because the to-do list extracts information directly from the inbox without any participation from a human, unlike many other apps, this could be effectively used by those that do not check their email.

KEYWORDS

Mobile platform, Machine Learning, NLP.

1. INTRODUCTION

According to surveys, self reported reading either their personal or work email regularly. This makes preexisting lists and calendars all but ineffective as they require a human to add task to the task management system. A second problem that is also quite prevalent is that there is the chance that even if you have read your email, you neglect to add it to your calendar. As someone that has dealt with these issues personally, I wanted to find a way to reduce the likelihood that such things happen because the cost of missing opportunities is too high. In order to solve this problem, we designed and created an App that will automatically generate a to do list from your inbox [2].

This topic is very important because it will help people make the most of their opportunities -- by letting them know that they have the opportunity.

Currently there is no shortage of apps that try to address this problem. The most popular Include google calendar, smart sheets, notion, and various digital to do lists [3]. These apps are all very well developed and used with many advanced features. However, these platforms share a weakness---they cannot automatically extract tasks directly from an email and add it to the to do

David C. Wyld et al. (Eds): FCST, CMIT, SE, SIPM, SAIM, SNLP - 2022
pp. 13-21, 2022. CS & IT - CSCP 2022 DOI: 10.5121/csit.2022.120802

lists unless the senders of the email send invites [4]. The issue is that it is completely impractical to expect such consideration from everyone that emails you, and this consideration would be impossible if the email is a school announcement, or any sort of advertisement meant for many people as they do not know you are coming. Another issue, as mentioned before, that is more commonplace than people would admit is that many emails are left unread. 55% of all email users admit that they don't open either business or personal emails regularly. This means that 55% of the population does not even get the chance to use these other calendars or to-do lists effectively as they all require human involvement. The benefit of this app is that it boils down emails to the tasks you need to complete so you can quickly skim over all of them and be reminded of the things you need to do or you can do. It is important to note that this app is not strictly a to do list in the traditional sense, it is more like a reminder list of the things that you can do. Overall, there are some differences in the intent of this app compared with preexisting ones. Instead of focusing on creating an impeccably tailored list of the things you will do, we chose to focus on making sure that you are remembering that certain events exist so you don't accidentally miss an event because you did not see it when perusing your email or failing to do so.

Using NLTK, we understand every word within the sentences in the email [5]. There are two rough parts to generating a task. Understanding the task and extracting the date and time. For the actual task content, our algorithm uses NLTK to parse the sentences and tag each word with the category it falls under. I.e. nouns, verbs, numbers etc. Using this information, our program especially targets action verbs to understand what the email is asking the user to do. On the other hand, in order to obtain the date, the program uses a variety of approaches. NLTK parses the sentences and gives us a "tag" of what type of word it is (or if it is a number). This narrows down the text we need to search for keywords. For example, we only need to search for proper nouns and numbers when trying to find the date. The algorithm checks for key words such as "January" and variations such as "Jan", "jan" for every month, day of the week. In addition to this, it also checks for relative time such as "tomorrow", "next month" "next week" and etc to determine the date if it is not explicitly listed. It also detects whether a few strings of numbers are in fact critical dates or times. Once the task and the date is determined, we export our results to our front end. The back and front ends are connected through the web framework flask. Our front end is built with dirt. We chose this as it allowed us to create beautiful interfaces relatively easily.

The road map for the rest of the paper is as follows: section two will describe the road bumps and challenges we ran into during the project. Section three details how we resolved the problems outlined in section two. In Section four we will explain our testing process and the subsequent results about the app's performance. Section five presents related works done by others to make a side by side comparison of our app. Lastly, section six includes concluding remarks as well as areas needing further research.

2. CHALLENGES

In order to build the project, a few challenges have been identified as follows.

2.1. How to determine the due date

A challenge we encountered was determining when an action is "due" when no specific date and time were given in the email. For example, the program needs to understand what day "tomorrow" is in the sentence "please arrive by tomorrow." This is more complex than one might initially think. As humans use various forms of speech to describe the same thing, the computer needs to be able to understand all forms of communication [6][7]. It needs to understand the words such as "minutes" and "hours", all of the days of the week, all the months of the day when spelled out such as “January” “as well as that “1/19” is also in January. Beyond these basics, it also needs to understand things like “next”, “tomorrow”, and “yesterday”. Furthermore, the program needs to be able to make smart assumptions about what the “due date” is when it does not have definitive information. For example, it needs to be understood that unless stated otherwise, the phrase “see you on Tuesday” usually means the upcoming Tuesday.

2.2. How to make sure the computer is intelligent enough

Another major issue we faced when writing the algorithm is making sure that the computer is intelligent enough to be useful. Which in this case means the ability to discern what information is relevant and ought to be extracted and what is irrelevant noise. This is no easy feat for a computer. Although we have access to the nltk library, it can only go so far as to help us analyze the sentence structure-- which, although is necessary, it is far from being sufficient. The computer needs to know either what you need to do, what event is happening, what registration is closing, as well as what day that is happening, at what time, and at what place. Simply an analysis of what is a date, or what is time is not going to cut it. Furthermore, as this is the core of the app, it is incredibly important that it functions without a hitch.

2.3. How to publish APP

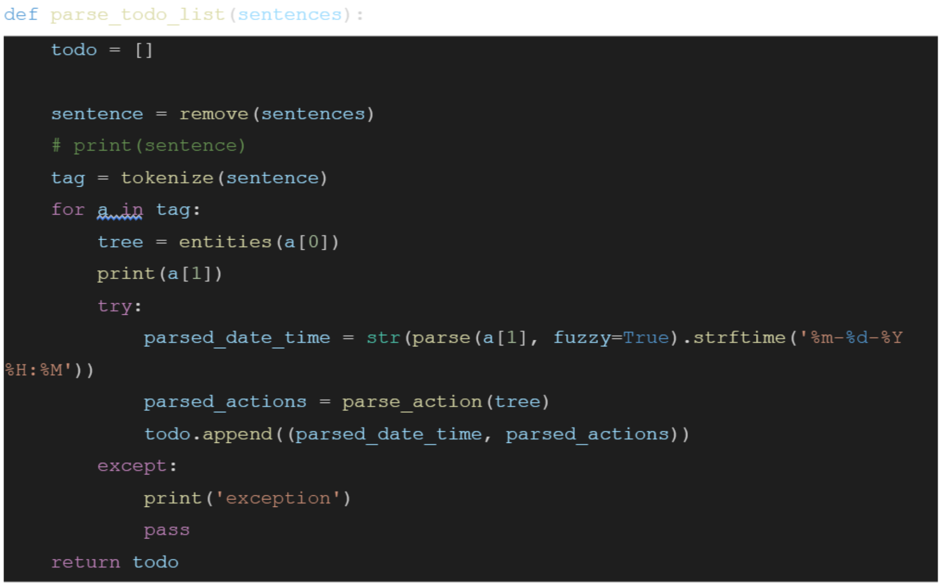
Another road bump we faced occurred during app publication [8]. Things did go smoothly with the app login and required multiple adjustments in Firebase for it to function properly [9]. Furthermore, during our testing period (after we fixed the previous issue), for some inexplicable reason, our app stopped loading the login that it was capable of just a few days before. We eventually were able to find a potential fix for the issue, but the app will need to be re-approved before use ---which stalled the testing process even more.

3. SOLUTION

TODO is a smart automatic to do list generator that uses your email inbox to generate a list of things happening each day that might be easily forgotten otherwise. It is capable of finding and extracting concrete things to do from an email. Or in other words, it is not only a to-do list, but it can also function as an email summarizer even though that is not its intended purpose. The process of its operation are as follows. First, whenever you receive an email, TODO will be able to access it and analyze the sentences using an open sourced library called nltk (natural language toolkit). Next the program goes through each sentence to see if a “due date” is stated within the sentence. If it is, this sentence is deemed the “action” of the “task”. We chose to use date as an indication of an “action” because in virtually every task, there will be a time when you need to be finished. Furthermore, this will likely be reflected in practically all sentences with actions. For example: “Let’s talk on 9/30/21”. First of all, it would only make sense if there is a date there, second of all, it would not be a pressing matter to do if no time was specified. Although this is an assumption, we believe it is reasonable [10]. After we have obtained the due date and the action, we will stitch them together as one task and send them to the app to be displayed.

The first step is to access the email of the user. This is quite a simple task; we simply have to set up the correct email verification. After we ’ ve done this, access to all of the text within the g mail inbox is at our program’ s disposal.

Our second step is to iterate through every sentence of each new email tokenize (split text into chunks for the ease of analysis) and analyze them. For the analysis, We used labels provided by the nltk open sourced library. Examples of the labels we employed are “DATE” , “TIME” , “PLACE”. The nltk library is able to automatically determine if a text is a date, time, place and etcetera and subsequently label it through their trained algorithm.



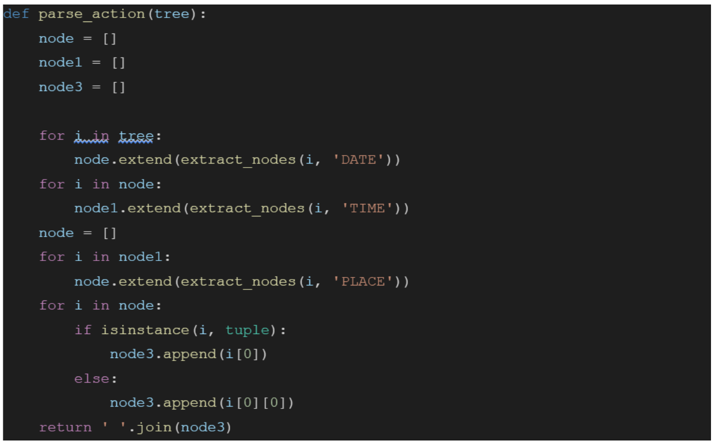
```
def parse_todo_list(sentences):
    todo = []

    sentence = remove(sentences)
    # print(sentence)
    tag = tokenize(sentence)
    for a in tag:
        tree = entities(a[0])
        print(a[1])
        try:
            parsed_date_time = str(parse(a[1], fuzzy=True).strftime('%m-%d-%Y
%H:%M'))
            parsed_actions = parse_action(tree)
            todo.append((parsed_date_time, parsed_actions))
        except:
            print('exception')
            pass
    return todo
```

Figure 1. Screenshot of to do list code

Our next step is to obtain the “due date” corresponding with that email [11]. This is harder than it seems. Humans tend to write dates in very nonuniform ways. Some write it as “dd/mm/yyyy”, some simply write “tomorrow”, some say “next Tuesday”, and much more. We had originally written code to be able to recognize this, but later we found a library that handled this much more efficiently than ours so we adopted their system. We set whatever date and time we find to Parsed_date_time. The code for this is the first line in the “try” block in the figure above.

After this, we called the parse_action function and passed in the tree (sentences labeled with grammar, and other labels such as date) to find the task we were to do. As can be seen in the code below, we iterated through every node in the tree and all appended sentences with dates to “node”. Following this, we checked if the nodes with dates also have a specified time. All nodes that do will be stored in node1 and the previous node returned to an empty state. We then checked to see if a location was specified. After this, we try to see if the node also has a specified place. The last if statement checks that if the node is not empty (has date), the node (sentence) will be the action of this email’s task. It is important that all actions within the to-do list have a due date for the sake of prioritization and organization.

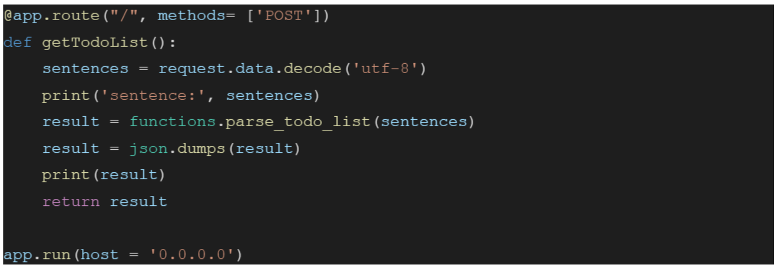
After completing this, we appended the parsed_date_time (from the previous image) with the action we just obtained and appended them together. Which becomes a “task” ready to be uploaded to the to do list.



```python
def parse_action(tree):
    node = []
    node1 = []
    node3 = []

    for i in tree:
        node.extend(extract_nodes(i, 'DATE'))
    for i in node:
        node1.extend(extract_nodes(i, 'TIME'))
    node = []
    for i in node1:
        node.extend(extract_nodes(i, 'PLACE'))
    for i in node:
        if isinstance(i, tuple):
            node3.append(i[0])
        else:
            node3.append(i[0][0])
    return ' '.join(node3)
```

Figure 2. Screenshot of parse section code

Our final step is to upload the “task” onto the app. We did this through a very simple function as can be seen in the code below. We called the functions mentioned before and sent it to the app via json where it was displayed.



```python
@app.route("/", methods= ['POST'])
def getTodoList():
    sentences = request.data.decode('utf-8')
    print('sentence:', sentences)
    result = functions.parse_todo_list(sentences)
    result = json.dumps(result)
    print(result)
    return result

app.run(host = '0.0.0.0')
```

Figure 3. Screenshot of app route code

4. EXPERIMENT

Experiment 1

In order to verify that our solution can effectively solve problems at different levels and have good user feedback, we decided to select multiple experimental groups and comparison groups for several experiments.

For the first experiment, we want to prove that our solution works stable and continuously, so we choose a group size of 40 different Tasks in 4 different types. The 5 different types of tasks are Course Schedule, Volunteer Activity, Driving Activity, and Sports Activity. The goal of the first experiment is to verify if the Task Manage System works well for different types of tasksThrough sampling 4 groups of tasks of different types and asking the same person to finish all these tasks with the schedule of our app. Results are collected by statistics if the app helps the user save time by scheduling all the tasks automatically than not using the app. Experiments have shown that all tasks in different types have a high rate of saving time. Class Schedule has the most high saving rates, which means our AI works much better in class scheduling field. This experiment could explain that the task types do have an obvious impact on the arrange results. The average saving time (in minutes) of 4 different types of the task shows below:

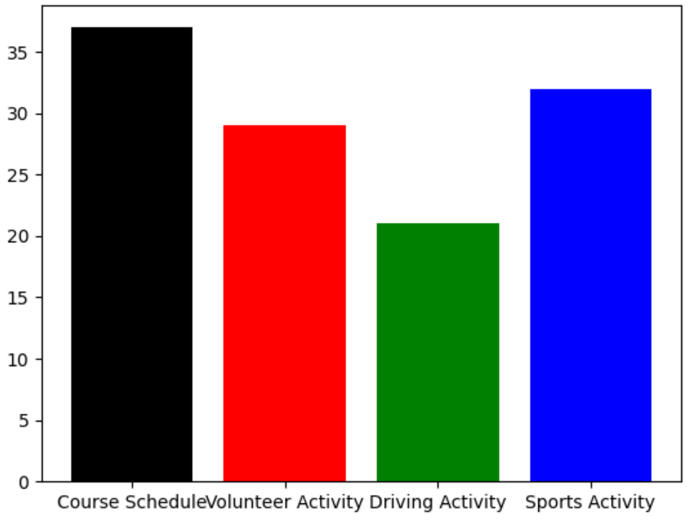


Figure 4. Result of Experiment 1

Experiment 2

A good user experience is as important as a good product. So a perfect solution should have excellent user experience feedback. In order to prove that our solution has the best user feedback, we specially designed a user experience questionnaire. We statistics the feedback result from 100 users, we divide those users into Five different groups. The first group of users ages from 10 - 20, the second group of users ages from 20 - 30, the third group of users ages from 30 - 40, the fourth group of users ages from 40 - 50, the fifth group of users ages from 50 - 60. The goal of the first experiment is to verify high feedback scores shows high performance We collected the feedback scores form these 5 different groups of users and analyze it. Experiments have shown that users who ages from 10 - 20 give the highest result feedback to our app. Which may because of paywall link appears more in the game searching The experiment graph shows below:

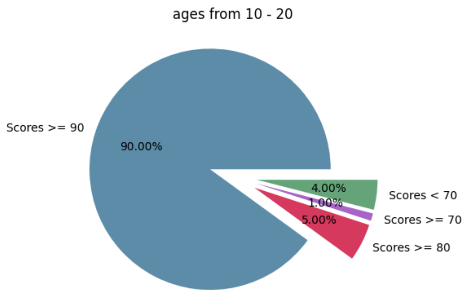


Figure 7. Result of age 10-20

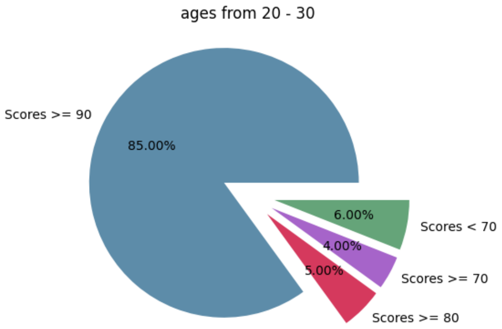


Figure 5. Result of age 20-30

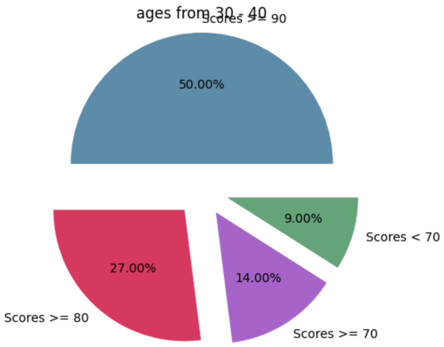


Figure 6. Result of age 30-40

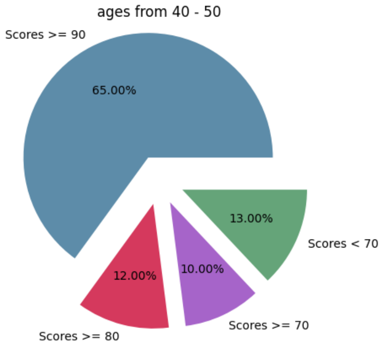


Figure 9. Result of age 40-50

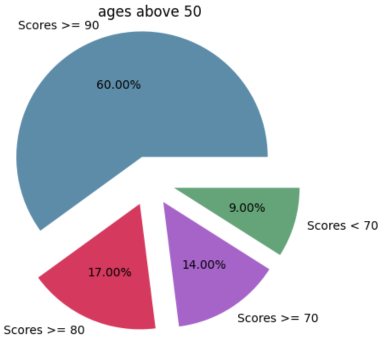


Figure 8. Result of age above 50

5. Related Work

Talking diary is an app for android that takes audio notes to process and sort into tasks [12]. It has three main components: Classifier, Scheduler, and a working hour calculator. Note: in this case, the classifier simply determines the type of tasks (ie. work, home, school, etc). When comparing this app with TODO, there are many differences. For one, although both use NLP, the point of TODO is to detach all human involvement in the generation of the to -do list while Talking Diary wants people to leave verbal notes. As for the organization and scheduling portion of the app, it is quite clear that Talking Diary has the advantage. It is able to know how much time you have, how much work you have to do, etc.

Pyhop1 is a hierarchical task network planner written in Python sharing similarities with SHOP [13]. It uses no loop detection to accomplish its tasks. With in this platform, there are three different categories of items one can enter: tasks, goals, and action. The main difference between SPyhop and TODO is that Pyhop focuses on task management while TODO's key point is task acquisition. Their algorithm in task organization is without a doubt, far superior and more complex than that of TODO's but these two platforms are meant to accomplish different things. One is meant for prioritizing and organization while the second is to simply remind you of the miscellaneous things you should do based on your inbox.

Schedule Me has four main components: Data engineering, Intelligent task breakdown and Scheduling using constraint programming, Reinforcement Learning for Personalized Task Scheduling, and User Centered Interaction Design. In data engineering, Schedule Me is able to scrape information from all relevant websites (such as school websites) and store them into a database [14]. It is also able to automatically break down tasks for you using constraint programming. Furthermore, it will organize your tasks to optimise information retention. Lastly, it makes it easy to use. In comparison with TODO, this is a very advanced scheduling system that both organizes tasks as well as acquire tasks. The difference between the two apps is that TODO is more targeted towards the email while Schedule ME is all encompassing. Furthermore, TODO does not schedule your time, it simply reminds you of the things happening today.

6. Conclusions

Task management is an important and complex issue whose consequences are significant. A missed email could be a missed opportunity. By using NLP to automatically compile a to-do list, it makes it more difficult to forget or miss something. That it can extract tasks completely autonomously means that It does not require people to already have good habits to use the app effectively. Experiments show that the app is most capable of recognizing and extracting tasks related to school work. User feedback indicates general positive reactions to the app, especially among the 10-20 age group.

There are still great limitations on the rate at which the app is identifying and saving the task from the user email–especially tasks that are non-coursework related. This impacts the reliability and dependability of the app. In addition, there is room for improvement in the user experience — especially for populations older than 30.

Further research will involve using sample emails with a wider array of topics during App development which should increase its adaptability and dependability [15]. In addition, we need to collect more user feedback, especially written user feedback so that not only do we know which age group is enjoying the app the most, we can understand why that is. This information would be important for future improvements.

REFERENCES

[1] Bellotti, Victoria, et al. "What a to-do: studies of task management towards the design of a personal task list manager." Proceedings of the SIGCHI conference on Human factors in computing systems. 2004.

[2] Joorabchi, Mona Erfani, Ali Mesbah, and Philippe Kruchten. "Real challenges in mobile app development." 2013 ACM/IEEE International Symposium on Empirical Software Engineering and Measurement. IEEE, 2013.

[3] Feddern-Bekcan, Tanya. "Google calendar." Journal of the Medical Library Association: JMLA 96.4 (2008): 394.

[4] Gillespie, Tarleton. "The politics of ‘platforms’." New media & society 12.3 (2010): 347-364.

[5] Loper, Edward, and Steven Bird. "Nltk: The natural language toolkit." arXiv preprint cs/0205028 (2002).

[6] Lüders, Marika. "Converging forms of communication?." Ambivalence towards Convergence: Digitalization and Media Change, Gothenburg: Nordicom (2007): 179-98.

[7] Janoschka, Anja. Web advertising: new forms of communication on the Internet. Vol. 131. John Benjamins Publishing, 2004.

[8] Ithnin, Muslimah, et al. "Mobile app design, development, and publication for adverse drug reaction assessments of causality, severity, and preventability." JMIR mHealth and uHealth 5.5 (2017): e6261.

[9] Khawas, Chunnu, and Pritam Shah. "Application of firebase in android app development-a study." International Journal of Computer Applications 179.46 (2018): 49-53.

[10] Lawrence, W. Gordon, Alaistair Bain, and Laurence Gould. "The fifth basic assumption." FREE ASSOCIATIONS-LONDON- (1996): 28-56.

[11] Keskinocak, Pinar, and Sridhar Tayur. "Due date management policies." Handbook of quantitative supply chain analysis. Springer, Boston, MA, 2004. 485-554.

[12] Munir, A. Hani, Abubakar Manzoor, and Utba Aziz. "Talking Diary: A Novel Approach for Automatic Audio Note Categorization and Event Scheduling for Android Application." (2020).

[13] Stanton, Neville A. "Hierarchical task analysis: Developments, applications, and extensions." Applied ergonomics 37.1 (2006): 55-79.

[14] Liyanage, A. N., et al. "ScheduleME-Smart Digital Personal Assistant for Automatic Priority Based Task Scheduling and Time Management." 2021 2nd Global Conference for Advancement in Technology (GCAT). IEEE, 2021.

[15] Mota, José Miguel, et al. "Augmented reality mobile app development for all." Computers & Electrical Engineering 65 (2018): 250-260.

THE EFFECT OF EMPLOYEES ' MARITAL SATISFACTION ON JOB PERFORMANCE : BASED ON THE PERSPECTIVE OF CONSERVATION OF RESOURCE THEORY

Lijun Sun, Zhefei Mao and Jie Zhou

Institute of Psychology, Chinese Academy of Sciences P. R. China, 100101
Department of Psychology,
University of Chinese Academy of Sciences P. R. China, 100049

ABSTRACT

The study linking the marriage with work explores the mechanism of action of employees' marital satisfaction and job performance through establishing a moderated mediating effect model. The results of the correlation and regression analyses conducted by collecting questionnaires from 290 employees indicated that: (1) Emotional exhaustion and work engagement play a chain mediating role in the positive relationship between marital satisfaction and job performance. (2) Work meaningfulness and work engagement play a chain mediating role in the positive relationship between marital satisfaction and job performance. (3) The need to support a family moderates the relationship between marital satisfaction and work meaningfulness, as well as the mediating effect of work meaningfulness and work engagement on the relationship between marital satisfaction and job performance. (4) The need to support a family moderates the relationship between marital satisfaction and emotional exhaustion, as well as the mediating effect between emotional exhaustion and work engagement on marital satisfaction and job performance. (5) Self-efficacy moderates the relationship between marital satisfaction and work meaningfulness, as well as the mediating effect between work meaningfulness and work engagement on marital satisfaction and job performance. This study provides a new perspective of family as resources for improving employees' job performance in management.

KEYWORDS

Marital satisfaction, job performance, emotional exhaustion, work engagement, work meaningfulness.

1. INTRODUCTION

Nowadays, family and work are undoubtedly the two most indispensable parts of every employee's life, hence the issue how to balance the relationship between the two has raised increasingly attention. The marital status of an employee is the core of family relationship, while marital satisfaction is the subjective or objective measure of an individual's overall marital status which affects not only individual and family well-being but also the performance in other areas such as work in the enterprise [1]. Most previous studies focus on two issues of work-family conflict due to role conflict [2] and work-family enrichment due to the contribution of one role to another [3]. In terms of the work-family enrichment, the close family relationship could contribute to the work

David C. Wyld et al. (Eds): FCST, CMIT, SE, SIPM, SAIM, SNLP - 2022
pp. 23-34, 2022. CS & IT - CSCP 2022 DOI: 10.5121/csit.2022.120803

because it not only brings positive emotions and reduce stress and anxiety, but also may serve as a kind of psychological resource that enables employees to maintain a good state of mind and positive self-evaluation and stimulates employees' desire for self-growth and self-development, thus enhancing the job performance in the enterprise [4]. Within the framework of work-family enrichment, the study investigates the process mechanism and boundary conditions of the impact of marital satisfaction as a psychological resource on job performance from the perspective of conservation of resource theory.

2. QUESTIONS AND HYPOTHESIS

According to the conservation of resource theory, an individual has the tendency to acquire, preserve and maintain resources [5]. A perfect family as a resource triggers positive emotions and cognition when sufficient, which would carry over to work and thus positively influence the psychological states and behaviors in the work [6]. In contrast, employees with low marital satisfaction are less resourceful and would experience negative emotions and cognition, which in turn affect their motivation to immerse in work (i.e., work engagement) and job performance [7]. It is inferred that employees' marital satisfaction may influence the work engagement and ultimate job performance through both emotional and cognitive paths.

The related studies on conservation of resource theory also show that emotions caused by resource sufficiency or scarcity mainly feature the emotional exhaustion [6], while the cognition includes evaluation of self and the outside world [8]. In terms of emotions, considering that marital satisfaction is a psychological resource, the satisfied marriage could bring positive emotional experiences and a steady flow of energy to employees [9] which can effectively reduce the emotional exhaustion at work. Employees can therefore put more energy and positive emotions into their working roles and contribute more effort and time to their work tasks, while employees with high work engagement are also more enthusiastic and committed to their work so as to improve the job performance [10]. On the contrary, an unhappy marriage leads to negative emotions for employees day after day [11] and individuals will bring these negative emotions from their family life to work so as to cause the further loss of individual resources, thus resulting in the emotional exhaustion [12]. In other words, a disharmonious marriage can not alleviate the emotional exhaustion at work but exacerbate it, which will result in employees not having enough resources to cope with their work tasks and affect their work engagement so as to cause the decline in the job performance [13]. Therefore, it is hypothesized that:

H1 Marital satisfaction affects employees' job performance through the chain mediating effect of emotional exhaustion and work engagement.

In terms of cognition, an individual satisfied with the marriage is more likely to have the positive self-evaluation, whereas one with low marital satisfaction tends to have the negative self-evaluation [14] and even doubts and denial of self [11]. Employees with high marital satisfaction may have positive evaluation of both self and work so that they can recognize themselves, develop a sense of purpose, or believe that they can create greater value at work through their own efforts, and then experience a higher work meaningfulness [15]. The work meaningfulness is an individual's personal subjective experience of the meaning and purpose of the work performed [16]. Even when the work is challenging, the positive beliefs generated by a high-quality marriage could promote hardworking engagement and perseverance of employees [17]. Fairlie (2011) surveyed 574 employees in North America and found that the work meaningfulness still explains 16% of the variation in work engagement after controlling other job characteristic variables, indicating that work meaningfulness is a strong predictor of work engagement [18]. Furthermore, it has been demonstrated that employees' work behaviors and attitudes are influenced by their cognition of work meaningfulness, and that the work meaningfulness positively predicts employees' passion and engagement [19], which in turn has a significant positive effect on job performance. The results of Schaufeli et al. pointed out the positive effect of work engagement on job performance in different contexts [20]. Therefore, it is hypothesized that:

H2 Marital satisfaction affects employees' job performance through the chain mediating effect of work meaningfulness and job engagement.

In summary, on the basis of the conservation of resource theory, employees' marital satisfaction may influence their work engagement and ultimate job performance through two paths of emotional exhaustion (emotions) and work meaningfulness (cognition). Moreover, it is necessary to consider the individual internal characteristics and external pressure which are important for psychological resources for their moderating role in the influencing process.

The self-efficacy, which refers to an individual's subjective assessment of his or her likelihood of performing and completing an activity and completing [22], is one of the individual characteristic variables that has received the most attention in studies related to conservation of resource theory [21]. Employees with low self-efficacy have fewer psychological resources and lower self-evaluation which leads to a lower work meaningfulness regardless of the marital satisfaction [23], which in turn negatively affects their job motivation and performance. It is therefore hypothesized as follows:

H3 Self-efficacy moderates the relationship between marital satisfaction and work meaningfulness. The higher self-efficacy contributes to strengthen this positive relationship while the lower self-efficacy to weaken this positive relationship.

H4 Self-efficacy moderates the effect of marital satisfaction on job performance through the chain mediating roles of work meaningfulness and work engagement. Specifically, the lower self-efficacy leads to the weaker indirect relationship.

The stress is an interactive process between an individual and the environment, as well as an individual's response to various stimuli in life after the subjective assessment. Considering that an individual needs to consume resources so as to cope with stress, the individual is more likely to experience anxiety and anger and needs to devote more emotions, time, energy, cognition and other resources for emotion management once the resource consumption occurs. The resources will be further consumed if the stress is not improved, leading to the loss spiral of resources [24]. Therefore, stress is undoubtedly one of the most indispensable external factors in conservation of resource theory. From the realistic and economic perspectives, with the need to support a family as a pressure, employees have to work hard to improve their ability to support the families when the need is high. Even if their marital satisfaction is not high, employees need to work to support the families with work regarded by them as a financial source to support their families, thus giving a stable sense of meaning to the work [25]. From the emotional perspective, furthermore, employees with high need to support the families consume a great deal of psychological resources even if the marital satisfaction is high, and the excessive consumption of employees' psychological and emotional resources leads to the emotional exhaustion, which in turn reduces work engagement and performance[26]. From the cognitive perspective, employees with high need to support the families would transform their work into a way to achieve their personal values for the need even though their marital satisfaction is low, and the energy gained is put into their work so as to improve their job performance [27]. It is therefore hypothesized as follows:

H5 The need to support a family moderates the relationship between marital satisfaction and emotional exhaustion. The higher need to support a family leads to the weaker negative relationship while the lower need to support a family to the stronger negative relationship.

H6 The need to support a family moderates the relationship between marital satisfaction and work meaningfulness. The higher need to support a family leads to the weaker positive relationship while the lower need to support a family to the stronger positive relationship.

H7 The need to support a family moderates the effect of marital satisfaction on job performance through the chain mediating roles of emotional exhaustion and work engagement. Specifically, the higher need to support a family leads to weaken this indirect relationship.

H8 The need to support a family moderates the effect of marital satisfaction on job performance through the chain mediating roles of the work meaningfulness and work engagement. Specifically, the higher need to support a family leads to weaken this indirect relationship.

In summary, the study proposes the following moderated mediation model (e.g., Figure 1) from the perspective of conservation of resource theory to investigate the influencing mechanism and boundary conditions of marital satisfaction for job performance.

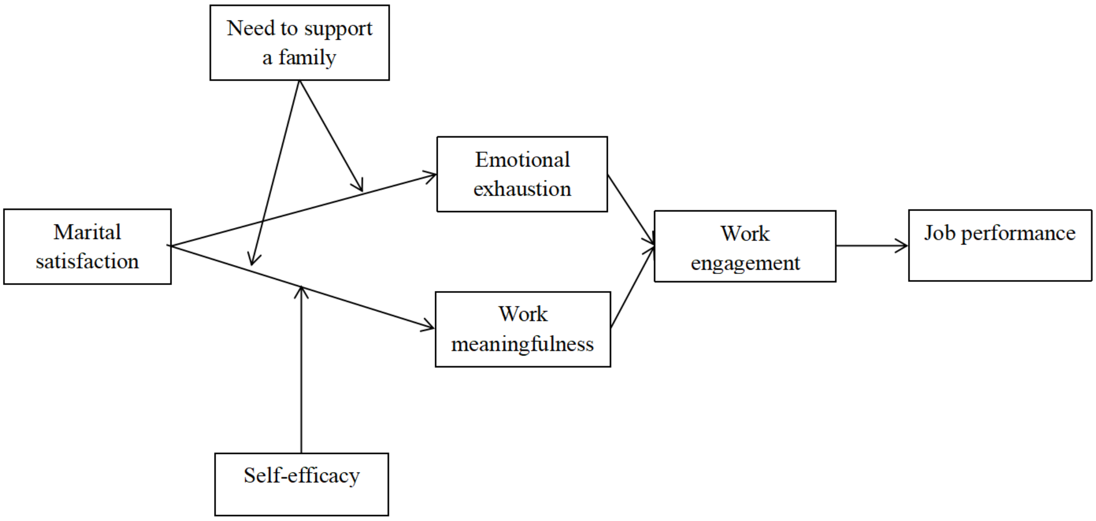


Figure 1

3. Methods

3.1. Sample and Procedures

In the study, a total of 300 questionnaires were distributed online to collect data, with 290 valid questionnaires collected excluding 10 questionnaires that were not completed carefully. The average age of the 290 valid participants is 33 years old, among which 40% are male and 67% have a bachelor’s degree. 33% of participants have worked in their current companies for 1 to 5 years, 48% for 5 to 10 years and 19% for more than 10 years. Their average income is 156,378 yuan per year. 32% of participants are ordinary employees, 37% front-line managers, 24% middle managers, and the rest senior managers. The average number of children owned is 1.14.

3.2. Variables Measurement

All variables in the study were measured through the Likert scale, with 1 representing strong disagreement and 5 strong agreement. The specific question items are as follows:

Marital satisfaction: The marital satisfaction scale based on the measurements from Fowers and Olson (1993) [28] was used with a total of 12 questions. The typical question item is as follows: “I do not like my spouse’s personality and personal habits”. The Cronbach’s α coefficient in the study was 0.662.

Emotional exhaustion: The emotional exhaustion was measured through the emotional exhaustion subscale of Maslach Burnout Inventory (MBI) classic scales by Maslach [29] et al. with 5 questions. The typical question item is as follows: “Work makes me feel fairly exhausted both physically and mentally”. The Cronbach’s α coefficient in the study was 0.846.

Work meaningfulness: The work meaningfulness was measured through three question items[30] from Workplace Spirituality scale developed by Ashmos and Duchon in 2000 and one question item from the scale developed by Bunderson & Thompson (2009) [31]. The typical question item is as follows: “My work is meaningful to me”. The Cronbach’s α coefficient in the study was 0.694.

Work engagement: The study resorted to a short version of nine questions from the Utrecht work engagement scale developed by Schaufeli’s team [32]. The typical question item is as follows: “When I am working, I am enthusiastic.” The Cronbach’s α coefficient in the study was 0.839.

Job performance: The measurement of job performance in this study with 12 questions referred to Xiaotong Shen’s master thesis [33], Juan Lu’s master thesis [34] and the job performance questionnaire developed by Motowidlo and Van Scotter (1996) [35]. The typical question item is as follows: “I always complete the work tasks assigned to me on time”. The Cronbach’s α coefficient in the study was 0.812.

Need to support a family: The study adopted a scale with 3 question items from Grant’s scale by Menges et al. [36]. The typical question item is as follows: “I have to work for money to make the family living expenses”. The Cronbach’s α coefficient in the study was 0.892.

Self-efficacy: The study referred to Schwarrzer’s general self-efficacy scale [37] and selected seven question items from it. The typical question item is as follows: “I still have the means to get what I want even if others oppose me”. The Cronbach’s α coefficient in the study was 0.782.

Control variables: According to previous studies, an employee’s gender, years, position, children quantity, age, education background, salary income and work stress would affect his or her job performance [38], hence these variables were treated as control variables in the study.

3.3. Data Analysis

The study resorted to SPSS 26, SPSS macro program PROCESS (3.5) line for data analysis. When the model provided by PROCESS could not meet the structural equation model of the study, the user-defined model was used for analysis.

4. RESULTS

4.1. Results of Descriptive and Correlation Analyses of Variables

The correlation analysis results between variables through SPSS are shown in Table 1. Marital satisfaction is significantly positively correlated with job performance (r=0.441,p<0.01), i.e., the more satisfied the employees are with their marriages, the higher their job performance. Marital satisfaction is significantly negatively correlated with emotional exhaustion (r=-0.386, p<0.01), i.e. the more satisfied the employees are with their marriages, the lower the probability of emotional exhaustion. Emotional exhaustion is significantly negatively correlated with work engagement (r=-0.657, p<0.01), i.e., the more serious the emotional exhaustion, the more difficult it is for employees to devote themselves to their work. Marital satisfaction is significantly and positively correlated with work meaningfulness (r=0.235,p<0.01), i.e. the more satisfied the employees are with their marriages, the more meaningful they feel their work is. Work meaningfulness is significantly and positively correlated with work engagement (r=0.706,p<0.01), i.e. the more meaningful employees think their work is, the more willing they are to devote their time and energy to it. Work engagement is significantly and positively correlated with job performance (r=0.723,p<0.01), i.e. the more employees are engaged in their work, the better their job performance will be. The above simple correlation results are basically consistent with the relationship between variables hypothesized in the study. According to the correlation results, the variables of an employee's gender, years, children quantity, age are not correlated with job performance, which are not consider in the regression analysis. the variables of an employee's position, education background, salary income and work stress are significantly correlated with job performance, which are controlled in the regression analysis.

Table 1. Average Values and Standard Deviations of Main study Variables

	Average Values	Standard Deviations	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
1.Gender	-	-	1														
2.Years	-	-	-.190**	1													
3.Pst	-	-	-.176**	.206**	1												
4. Qtt	1.14	0.489	-0.103	0.071	.151**	1											
5. Salary	156378	207991	-.172**	0.105	.459**	0.089	1										
6. Age	33.01	5.761	-.131*	.662**	0.087	.172**	0.015	1									
7. Edu	-	-	-0.037	-0.021	.284**	0.021	.341**	-0.112	1								
8. WS	2.9414	0.88432	-118*	-0.025	0.021	-0.102	-0.088	-0.037	-0.006	1							
9. MS	3.5578	0.53505	-.124*	0.104	0.085	-0.04	.174**	0.013	0.09	-118*	1						
10. NSF	3.6678	1.00345	-.367**	0.091	0.013	0.095	0.093	0.02	0.044	.172**	.149*	1					
11. WM	4.1897	0.52471	-0.06	.148*	.248**	0.11	.253**	0.053	0.1	-228**	.235**	.227**	1				
12. WE	4.1391	0.4656	-0.061	0.104	.192**	.156**	.235**	0.048	.186**	-.362**	.362**	.157**	.706**	1			
13. SE	3.9833	0.49928	-0.015	0.092	.176**	0.09	.324**	-0.053	.117*	-.405**	.309**	0.046	.519**	.621**	1		
14. EE	1.9766	0.68195	0.034	-0.112	-.143*	-0.059	-.260**	-0.042	-.134*	.520**	-.386**	0.008	-.516**	-.657**	-.526**	1	
15. JP	4.2782	0.36548	-0.011	0.075	.176**	0.108	.252**	-0.01	.165**	-224**	.411**	.163**	.586**	.723**	.655**	-.527**	1

Note:* *p*<0.05 (two-tailed)， ** *p*<0.01 (two-tailed)， *** *p*<0.001 (two-tailed)

Pst Position, Qtt Quantity, Edu Education ,WS Work Stress, MS Marital Satisfaction, NSF Need to Support a Family, WM Work meaningfulness, WE work engagement, SE Self-efficacy, EE Emotional Exhaustion, JP Job Performance

4.2. Hypothesis Test

4.2.1. Mediating Effect

Based on the user-defined model of SPSS macro program PROCESS, the study tested both the chain mediating effect of emotional exhaustion and work engagement on marital satisfaction and job performance, and that of work meaningfulness and work engagement on marital satisfaction and job performance.

The analysis results of the chain mediating effect of emotional exhaustion and work engagement showed the significant sequential mediating effect between emotional exhaustion and work engagement on marital satisfaction and job performance (ab=0.0590, BootSE=0.0135) with 95% confidence interval of [0.0353, 0.0876]. Hypothesis 1 was supported.

The analysis results of the chain mediating effect of work meaningfulness and work engagement showed the significant sequential mediating effect between work meaningfulness and work engagement on marital satisfaction and job performance (ab=0.0581, BootSE=0.0217) with 95% confidence interval of [0.0216,0.1058]. Hypothesis 2 was supported too.

4.2.2. Moderating Effect

The custom program analysis of SPSS macro program PROCESS showed that marital satisfaction and self-efficacy were significantly positively correlated with work meaningfulness (γ=0.4461 , p<0.01) with 95% confidence interval of [0.3301, 0.5621] not including 0. It indicated that self-efficacy played a moderating role between marital satisfaction and work meaningfulness (as shown in Figure 2). The positive effect of marital satisfaction on work meaningfulness was stronger when self-efficacy of employees was high and would be weaker when self-efficacy was low. Therefore, the Hypothesis 3 was supported. The interaction between marital satisfaction and need to support a family on emotional exhaustion was not significant (γ=0.0445, p>0.05), indicating that there was no moderating effect of need to support a family on marital satisfaction and emotional exhaustion. Thus, the Hypothesis 5 was not supported. Marital satisfaction and need to support a family were significantly negatively correlated with work meaningfulness (γ= -0.1015, p<0.05) with 95% confidence interval of [-0.1977, -0.0053] also not including 0, which indicated that need to support a family played a moderating role between marital satisfaction and work meaningfulness (as shown in Figure 3). The positive effect of marital satisfaction on work meaningfulness was weaker when employees' need to support families was high, and would be stronger when the need was low. The Hypothesis 6 was supported.

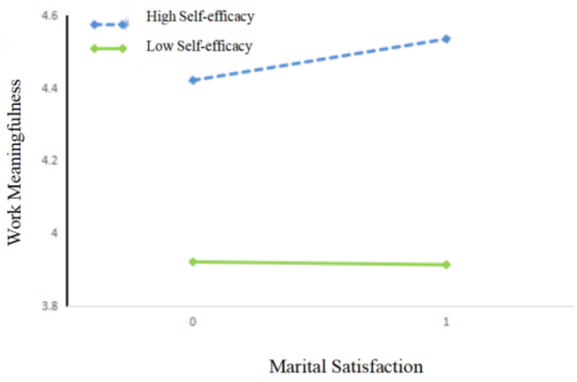


Figure 2

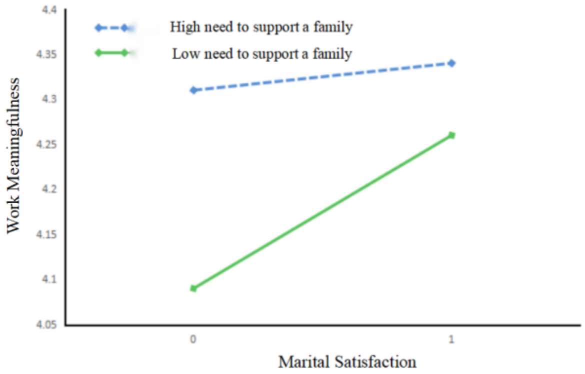


Figure 3

4.2.3. Moderated Mediating Effect

Considering the aforementioned analysis showed that the moderating effect of need to support a family on marital satisfaction and emotional exhaustion was not significant, the moderating role of need to support a family in the chain mediating effect of emotional exhaustion and work engagement also did not exist. The Hypothesis 7 was not supported. According to the sequential test of moderated mediating effect [39], the effect of marital satisfaction on work meaningfulness was significant (p<0.05),the effect of the interaction term of marital satisfaction and need to support a family on work meaningfulness was significant (p<0.05), the effect of the interaction term of marital satisfaction and self-efficacy on work meaningfulness was significant (p<0.05), the effect of work meaningfulness on work engagement was significant (p<0.001), and the effect of work engagement on job performance was significant (p<0.001). The results of the sequential test indicated that self-efficacy and need to support a family significantly moderated the chain mediating effect of work meaningfulness and work engagement respectively. The mediating effect values and 95% confidence intervals with the moderating variables at different levels were obtained through the analysis of SPSS macro program PROCESS (as shown in Table 2). When the need to support a family was low, the confidence interval did not include 0 and the mediating effect was significant with the mediating effect value of 0.0850. When the need to support a family was middle, the confidence interval did not include 0 and the mediating effect was significant with the mediating effect value of 0.0484. However, when the need to support a family was high, the confidence interval included 0 and the mediating effect was not significant. Thus, it was obvious that the higher the need to support a family was, the less significant was the chain mediating effect of work meaningfulness and work engagement on the relationship between marital satisfaction and job performance. The Hypothesis 8 was supported. Contrarily, when self-efficacy was low, the confidence interval included 0 and the mediating effect was not significant. When self-efficacy was middle, the confidence interval also included 0 and the mediating effect was not significant either. But when self-efficacy was high, the confidence interval dis not include 0 and the mediating effect was significant with the mediating effect value of 0.0694.Therefore, it was obvious that the higher the self-efficacy was, the more significant the chain mediating effect of work meaningfulness and work engagement on the relationship between marital satisfaction and job performance. The Hypothesis 4 was supported.

Table 2. Analysis of Moderated Mediating Effect

Adjustment Variables	Effect	BootSE	BootLLCI	BootULCI
Marital Satisfaction → Work meaningfulness → Work engagement → Job Performance				
Low need to support a family	0.0850	0 .0369	0.0222	0.1648
Moderate need to support a family	0.0484	0.0271	0 .0034	0.1072
High need to support a family	0 .0117	0.0334	-0.0447	0.086
Low self-efficacy	-0.0023	0.0342	-0.0673	0.0748
Moderate self-efficacy	0.0335	0.0228	-0.0102	0.0826
High self-efficacy	0.0694	0.0262	0.0176	0.1212

5. DISCUSSION

It has been demonstrated that close family relationships can provide employees with resources such as time, emotions and experience to support their tasks and performance in the work domain [40]. Conducted within the framework of conservation of resource theory, the study combined family with work and regarded marital satisfaction in the family as a resource to reveal its influence on job performance through the chain mediating effect of emotional exhaustion (emotional pathway) and work meaningfulness (cognitive pathway) to work engagement, and examined the moderating effect of self-efficacy and need to support a family on this model. It is found that marital satisfaction indirectly influenced employees' job performance through the chain mediating effect of emotional exhaustion and work engagement. The hypothesis that the need to support a family played a moderating role in this mediating process was not supported by the data. Moreover, marital satisfaction indirectly influenced employees' performance through the chain mediating effect of work meaningfulness and work engagement, while self-efficacy and need to support a family moderated respectively the mediation process of marital satisfaction - work meaningfulness - work engagement - job performance, because the relationship between marital satisfaction and work meaningfulness was moderated by self-efficacy and the need to support a family. The higher self-efficacy of employees led to the greater effect of marital satisfaction on job performance, while the increase in the need to support a family weakened the effect of marital satisfaction on job performance.

5.1. Theoretical and Practical Significance

Theoretically, previous studies mainly explained the relationship between family and work based on the perspective of work-family enrichment and work-family conflict. The study extended the perspective of work-family relationship and examined the influencing mechanism of family factors on job performance based on conservation of resource theory, by including variables of emotional exhaustion and work meaningfulness related to psychological resources as mediating variables with self-efficacy and need to support a family as moderating variables.

From the realistic perspective, conclusions from this study could enable enterprises to understand the influencing mechanism of employees' marital status on their job performance. In the management, enterprises can carry out trainings related to family relationship management, support employees to participate in family life to improve their marital satisfaction, and encourage employees to reduce unnecessary work stress and enhance their work meaningfulness through improving the work efficiency. All these can lead employees to access to more psychological resources so as to achieve the purpose of improving employees' work engagement and job performance.

5.2. Shortcomings and Future Directions

This study also has the following shortcomings: First and foremost, all the measurements in the study were based on the subjective reporting method and susceptible to social desirability. The following studies can resort to various forms of measurement such as evaluations by significant others and objective data to validate the moderated mediation model of marital satisfaction influencing job performance. Secondly, considering that the study was a cross-sectional study, the future study can adopt longitudinal study or even diary method to examine the effect of marital satisfaction on job performance and the time effect of its mediating mechanism, so as to further reveal the cross-lagged effects or causal relationships among variables. Thirdly, the future study may also hypothesize and verify more complex relationships among variables, such as the relationship between marital satisfaction and work meaningfulness which was found to be positively correlated in the present study. However, it is possible that the very relationship is true for participants with higher marital satisfaction, while the lower marital satisfaction may lead to the higher work meaningfulness for those with very low marital satisfaction since the marriage may break up at any time. It means that the relationship between marital satisfaction and work meaningfulness may be non-linear and can be explored in the future study.

6. CONCLUSION

The study demonstrated that marital satisfaction had a significant positive effect on job performance, and that emotional exhaustion and work engagement as well as work meaningfulness and work engagement functioned as the chain mediators in the process as dual paths respectively. On the mediation path of marital satisfaction - work meaningfulness - work engagement, self-efficacy and the need to support a family moderated the positive relationship between marital satisfaction and work meaningfulness.

ACKNOWLEDGEMENTS

We would like to thank reviewers and editors!

REFERENCES

[1] Greenhaus, J. H., Powell, G. N. When Work and Family Are AI—lies: a Theory of Work-Family Enrichment. academy of Management Review, 2006, 31 (1): 72-92.

[2] Clark S C. Work/Family Border Theory: A New Theory of Work/Family Balance[J]. Human Relations, 2000, 53(6): 747-770.

[3] Voydanoff P. Social Integration, Work-Family Conflict and Facilitation, and Job and Marital Quality[J]. Journal of Marriage and Family, 2005, 67(3): 666-679.

[4] Sieber, S. D., Toward a theory of role accumulation. American Sociological Review, 1974, 39: 567-568.

[5] Hobfoll, S. E. (2011). Conservation of resource caravans and engaged settings. journal of Occupational ＆Organizational Psychology, 84(1), 116-122.

[6] Hobfoll, S. E. Conservation of resource: A New Attempt at Conceptualizing Stress[J].

[7] Buric' I, Macuka I. Self-Efficacy, Emotions and work engagement Among Teachers: a Two Wave Cross-Lagged Analysis [J]. Journal of Happiness Studies, 2018, 19(7):1917-1933.

[8] Jinyun Duan, Jing Yang, Yuelong Zhu. Conservation of Resource Theory: Content, Theory Comparison and Research Perspectives[J]. Psychological study, 2020, 13(1): 49-57.

[9] Hanying Tang. Work-Family Balance of Corporate Employees: An Organizational Informal Work-Family Support Perspective. Master' s thesis, Huazhong Normal University, 2008.

[10] Rich, B. R. , Lepine, J. A. , Crawford, E. R. Job engagement: Antecedent and effects on job performance [J]. Academy of Management Journal, 2010, 53(3): 617 - 635.

[11] Wensheng Sun, Jianliang Zhong. (2003). On the role of economic factors in marital relationships.

Economics and Management, 62-63.

[12] Halbesleben, X R.,&Bolion, M. C. Too engaged?A Conservation of resource view of the relationship between work engagement and work interference with family. Journal of Applied Psychchology, 2009, 94(6), 1452-1465.

[13] GrzywaczJ. G "Marks, N. E,. Reconceptualizing the Work~family Interface: An Ecological Perspective on the Correlates of Positive and Negative Spillover between Work and Family. - Journal of Occupational Health Psychology. 2000, 5: 111-126.

[14] Lewis, R. A., & Spanier, G. B. (1979).Theorizing about the quality and stability of marriage. contemporary theories about the family: study- based theories/edited by Wesley R. Burr ... [et al.]

[15] Lysova E I, Allan B A, Dik B J, et al. Fostering meaningful work in organizations: a multi-level review and integration [J]. Journal of Vocational Behavior, 2019, 110: 374- 389.

[16] Lips-Wiersma, M., & Wright, S. (2012). Measuring the meaning of meaningful work: Development and validation of the comprehensive meaningful work scale. group & organization management , 37(5), 655-685.

[17] Di Paula A, Campbell J D. Self-esteem and persistence in the face of failure [J]. Journal of Personality and Social Psychology, 2002, 83(3), 711-724.

[18] Fairlie, P. (2011). Meaningful work, employee engagement, and other key employee outcomes: Implications for human resource development. Resources, 13(4), 504-521.

[19] Woods, Stephen A., and Juilitta A. Sofat. Personality and engagement at work: the mediating role of psychological meaningfulness [J]. Applied Social Psychology, 2013, 43(11): 2203-2210.

[20] Hakanen J J, Bakker A B, Schaufeli W B. Burnout and work engagement among teachers[J]. Journal of School Psychology, 2006,43(6):495-513.

[21] Xia Cao, Jiaojiao Qu. Traceability of Conversation of Resource Theory, Exploration of Main Contents and Implications [J]. China Human Resource Development, 2014 (15): 75 - 80.

[22] Bandura A. Self-efficacy: Toward a unifying theory of behavioral change [J]. Psychological Review, 1977,84(2): 191-215.

[23] Woods S. A. & Sofat J. A., Personality and engagement at work: The mediating role of psychological meaningfulness, Journal of Applied Social Psychology, 2013, 43(11): 2203-2210.

[24] Hobfoll S E. Social and psychological resources and adaptation [J]. Review of general psychology, 2002, 6(4): 307.

[25] Wrzesniewski A, Dutton J E. Crafting a Job: Revisioning Employees as Active Crafters of Their Work [J].Academy of Management Review, 2001, 26(2): 179 -201.

[26] Wu, L. Z., Yim, F. H., Kwan, H. K., & Zhang, X. (2012). Coping with workplace ostracism: The role of ingratiation and political skill in employee psychological distress. Journal of Management Studies, 49(1), 179-197.

[27] Menges, J.I., Tussing, D.V., Wihler, A., Grant, A.M.. When Job Performance is all Relative: How Family Motivation Energizes Effort and Compensates for Intrinsic Motivation.Academy of Management Journal. 2017, 60(2): 695 -719.

[28] Fowers B J & Olson D H. Enrich Marital Satisfaction Scale: a brief study and clinical tool [J]. Journal of Family Psychology, 1993, 7(2): 176-185.

[29] Christina, Maslach, Susan, et al. The Measurement of Experienced Burnout[J]. Journal of Organizational Behavior, 1981, 2(2): 99-113.

[30] Zhe Zhang. Organizational Citizenship Behavior of NGO Employees Under Payback Imbalance — The Moderating Role of Work Meaningfulness [D], 2017: 31-32.

[31] Bunderson J. S. & Thompson J. A., The call of the wild: Zookeepers, callings, and the double-edged sword of deeply meaningful work, Administrative Science Quarterly, 2009, 54(1):32-57.

[32] Schaufeli W B, Bakker A B, Salanova M. The measurement of work engagement with a short questionnaire: a cross-national study [J]. Educational and psychological measurement, 2006, 66(4): 701-716.

[33] Xiaotong Shen. Employee Satisfaction, Organizational Commitment and Job Performance—An Empirical Study of the Relationship[D], 2014:27-28.

[34] Juan Lu. A study on the relationship among transformational leadership, organizational identity and employee performance[D], 2013: 40-41.

[35] Van Scotter, J R, Motowidlo S J. Interpersonal facilitation and job dedication as separate facets of contextual performance. journal of Applied Psychology, 1996, 8: 525-531.

[36] Menges, J. I., Tussing, D. V., Wihler, A., & Grant, A. M. (2017). When job performance is all relative: How family motivation energizes effort and compensates for intrinsic motivation. Academy

of Management Journal, 60(2), 695-719.

[37] Schwarzer, R., & Born, A. (1997). Optimistic self-beliefs: Assessment of general perceived self-efficacy in thirteen cultures. World Psychology, 3(2) , 177 - 190.

[38] Jinmeng Ma. An empirical study on the factors influencing employee performance in Southern Cisco Human Resources Co. Master' s thesis, Nanjing University of Technology, 2013.

[39] Zhonglin Wen, Baojuan Ye (2014). Test Methods of Moderated Mediation Model: Competition or Substitution? Journal of Psychology, 46(5), 714 - 726.

[40] Carlson, D. S., Kacmar, K. M., Zivnuska, S., Ferguson, M., Whitten, D-Work-Family Enrichment and Job Performance: A Constructive Replication of Affective Events Theory. Journal of Occupational Health Psychology, 2011, 16(3): 297-312.

AUTHORS

Lijun Sun, Institute of Psychology, Chinese Academy of Sciences P. R. China, master candidate, Applied Psychology.

TEST AUTOMATION FOR QUALITY ASSURANCE: A RANDOM APPROACH

Paul Court and Omar Al-Azzam

Computer Science and Information Technology Department (CSIT),
Saint Cloud State University (SCSU), Saint Cloud, MN, USA

ABSTRACT

Testing is a necessary, but sometimes tedious chore for finding faults in software. Finding faults is essential for ensuring quality and reliability of software for industry. These are valuable traits consumers consider when investing capital and therefore essential to the reputation and financial well-being of a business. This research involves an ongoing trade-off between time and computational resources when testing via random selection of test cases versus complex logical means to find faults. More time will be devoted to an analysis of random test case selection and whether the amount of extra test cases run due to random selection is a viable alternative to the potential time spent fully evaluating the logic for coverage of a generic predicate. The reader will gain knowledge about the expectations for the increase in test cases if randomized selection is employed at some point in the process of testing.

KEYWORDS

Predicate testing, Fault detection, Simulation, Random selection, Logic coverage.

1. INTRODUCTION

Since the very first "bug" was found in a computer system and Grace Murry of Harvard University coined the phrase, fault detection in programming has been evolving [6]. The progression of software testing has been well chronicled and studied by some of the best minds in recent history. Its maturity can be correlated with the diversification in programming methodology [14]. There is no doubt that for essential, high-level functions, quality assurance is a must, but the dilemma arises when less critical software must be produced under the pressure of limited resources and tight time constraints, so efforts need to be focused on the critical risk areas [9]. In some cases, the logic necessary to cover all fault detection is extremely demanding. The correct balance between risk, cost, and quality assurance may be difficult to obtain. Randomness may play a role in test selection if the possible risks are known. There is a point where logical evaluation of predicates and clauses is so cumbersome that random test selection becomes viable as an option for a tester.

To be clear, a tester must have access/knowledge of the developer's decision-making logic and be able to translate the code of a predicate into Boolean statements suitable for their evaluation using advanced logical coverage techniques. This research will focus on one of the more powerful logic coverages, Restrictive Active Clause Coverage (RACC) [8]. Simulation will demonstrate how random selection of test cases compares to common quality assurance methods as well as weigh the pros and cons of partial uses of logical approaches and random chance both economically and for fault detection [21, 22, 24].

David C. Wyld et al. (Eds): FCST, CMIT, SE, SIPM, SAIM, SNLP - 2022
pp. 35-47, 2022. CS & IT - CSCP 2022 DOI: 10.5121/csit.2022.120804

2. BACKGROUND

When software is designed, input is necessary for the execution of the logic in a program. The potential values the input can take needs to be evaluated. The domain of these values can be partitioned into ranges that will determine the execution necessary blocks of code. Each time an input domain is partitioned, thought should be given to the input values to better understand what issues the implementation may have with the values entered [1]. These partitions evolve into Boolean values of true or false called clauses.

For example, a software may need to execute a block of code if an integer is between a minimum (x) and a maximum (y). The input domain includes the values of I that make the statement (x < I < y) either true or false. This can be considered a clause that has a Boolean value true or false. Clauses can be combined using logical operators to form more complicated structures called predicates, which determine code execution.

Testing predicates for proper performance uncovers potential faults in software. There are several logic coverages that testers employ to detect faults in predicates. One of the easier and weaker logic coverages is Predicate Coverage (PC). To achieve predicate coverage, any test case that has a predicate value of true can be selected along with any test case that evaluates the predicate to be false. Predicate coverage can usually be achieved with two test cases [1]. The most powerful logic coverage is Combinatorial Coverage. It occurs when every possible truth value combination is explored for each clause in a predicate. This is essentially exhaustive coverage and is achieved with 2n test cases, where n is the number of unique clauses. These are the extremes of logic testing. Active clause coverages such as RACC, are proven to detect faults efficiently, while minimizing the number of tests necessary [8]. To achieve RACC, each major clause must be evaluated to true or false as the minor clauses are held constant. Not an easily accomplished assessment for complex predicates. It is sometimes impossible to achieve.

Fewer faults in released software is the goal of testing and a measure of customer satisfaction, but so is reducing cost [16]. Logical evaluation of predicates can be time consuming and expensive. There is merit to using randomization to help with test case selection.
This research will consider random selection analysis for logic coverages discussed above.

H0: Randomizing test selection yields no advantage in achieving logic coverage in testing. Running unnecessary tests is expensive and consumes resources. Random selection of test cases is not a viable strategy in testing.

H1: A balance between randomness and intentional strategy will be the best method testers can employ to find a high degree of faults when logic testing becomes difficult or impossible under tight timelines.

Keeping in mind that adding value to the software while maximizing return on investment is the goal, we will show that a combination of random chance and logic will be most effective.

3. LITERATURE REVIEW

Often in testing as with businesses and industries, decisions can be automated and randomized. Data driven decision making has proliferated recently [18]. Predictive analytics use a combination of artificial intelligence, machine learning, statistical algorithms, data mining, and modelling to drive decision making and automation [2, 12]. Automation in testing has adopted these principles. Industry testing methods and tools are constantly under scrutiny for improvement. The demand for better, faster, more effective designs can lead to better return on investment [11]. Test automation coupled with regression analysis and predictive modelling can lead to more effective test sets [23]. Tests that need to be repeated often or necessitate varied inputs work well with automated test framework (ATF) formats such as Selenium or Robot Framework. Random selection of inputs from a partitioned input domain can yield diverse and effective test cases. However, if test case selection is randomized, their selection should be prioritized. Some test cases available to a tester will be more involved in fault detection than others [5]. Therefore, random selection of test cases should be evaluated with due diligence. Unfortunately, automated, and randomized testing practices usually culminate with testers evaluating the processes to decide if fault detection via these methods accomplishes its goal.

4. METHODOLOGY

Deciding the input domain for testing will help define boundaries for possible variable values to determine clause Boolean. If input associations are established, a systematic method for random assignment can be automated. Rather than having testers take the time decide static input values, parameters can be set for automation. This process can be controlled by the system and once the guidelines are set, the system dynamically decides the test values for the test set. Logic coverage for test cases in this research will focus on restrictive active clause coverage (RACC). The number of tests required for partial or complete logic coverage will be studied.

First, we will consider completely random selection of input values given the input domain, directly determining clause Boolean values. The clauses will be combined using logical operators to form predicate Boolean necessary for test cases. The test case selection will be completely randomized, with test case repetition prevented to avoid redundancy. RACC will be evaluated studying the number of tests required to achieve coverage. Second, the idea of influential clauses and test cases will be considered. A superficial logical evaluation can uncover an influential RACC pair of test cases, thus determining two of the test cases in the test set. An influential test case is one that is shared by other clauses, contributing to coverage for several clauses simultaneously. Simulation will be employed to determine the impact of removing the known pair from the total cases while randomizing the selection of the remaining test cases. The number of test cases necessary to achieve RACC will be determined, with the number of excess cases necessary as the metric of evaluation. Predicates with three, four, and five clauses will be analysed. Predictive modelling practices will be used to provide estimates for larger predicates. This will give a tester the information necessary to weigh the resource of time spent to analyse clauses logically against the inefficiency of extra tests required due to random selection.

A demonstration can be used to help explain these ideas more clearly. For example, to decide if an integer input into a method is prime or a perfect square and in a certain range, the logic necessary to ensure that the correct values are entered requires testing. If this method is to be logically tested using Restrictive Active Clause Coverage some time would have to be devoted to these analyses. To determine whether the input is appropriate, it is necessary to check to see if the value falls between the minimum and maximum and is an integer. Then the integer can be evaluated to determine if it is either prime or a perfect square. This decision-making process can be represented with a three-clause predicate $p = a \wedge (b \vee c)$ where the clauses are determined using these variables:

a: the input is valid (integer and in range)
b: the input is prime
c: the input is a perfect square
$\wedge$: the standard logical operator “and”

$\vee$: the standard logical operator “or”.

Table 1 shows the main tool in logic coverage analysis, the truth table. When truth values for each clause are evaluated logically using the predicate operations, the truth value for the predicate is recorded to assist with test design.

To achieve predicate coverage, the weakest form or logic coverage, any test case that has a predicate value of true can be selected along with any test case that evaluates the predicate to be false. For example, the test set {1, 4} would be sufficient.

Table 1. Truth table for the predicate: $p = a \wedge (b \vee c)$

Test	***a***	***b***	***c***	***p***
1	T	T	T	T
2	T	T	F	T
3	T	F	T	T
4	T	F	F	F
5	F	T	T	F
6	F	T	F	F
7	F	F	T	F
8	F	F	F	F

Several other combinations of tests could be used, but predicate coverage could be attained with two well-chosen tests. Combinatorial coverage would require 23 = 8 test cases.

For a restrictive active clause coverage (RACC), each major clause must be evaluated to true or false as the minor clauses are held constant, as mentioned earlier [1]. One example for a RACC scenario would be for test cases 1 and 5. To activate clause a, notice in test case 1 clauses b and c have the value true while the value of clause a is true in test 1 and false in test 5. The predicate values are opposing. These test cases are RACC partners. Similar pairs of test cases need to be found in the table with each clause isolated as the major clause. Each clause’s active state must be considered and how its state affects the value of the predicate. To have complete restrictive active clause coverage, four strategically chosen tests must be run; either {2, 3, 4, 6} or {2, 3, 4, 7}. Note that test cases 2, 3, and 4 are more influential test cases with test case 6 and 7, while still vital, are less influential in the outcome. Note also that test cases 1 and 5 could play a role in RACC coverage, however if these cases are selected randomly, there will have to be at least six test cases in a viable RACC test set.

These evaluations are tedious and require accurate representations of the logic for a predicate. They need to be considered to create test sets that are comprehensive with the goal of finding the maximum number of faults present in a code segment. Imagine if there were several more clauses involved. The analyses can become burdensome, time-consuming, and counterproductive cost wise. If a tester is underqualified or unmotivated to perform these analyses or the logic criterion are so cumbersome that they cannot be efficiently evaluated, can random selection of tests be a viable alternative. What are the risks?

Using probability to evaluate the example above, randomly selected test cases for predicate coverage where one test needs to have a predicate value of true and one test must evaluate to false, there is a 0.375 (3 out of 8) probability of selecting a test case with a predicate value of true. If the test cases are selected at random, to have the minimum number of cases of two, the scenario becomes: the predicate true is selected first, followed by the false or the predicate false test can be first followed by the true. The probability of predicate coverage happening in the first two random selections is 0.46875. This implies that in 0.53125 proportion of random cases, more than the logically evaluated minimum number of cases would be necessary. This leads to running extra tests. However, if the situation is analysed differently and three tests are run randomly, the only result that would NOT produce predicate coverage is when all three tests are true or all three tests are false. An extra randomly selected test case increases the probability of satisfying predicate coverage by adding one extra test case to 0.703. These calculations are done allowing the test cases to be independently selected. If repetition is not allowed (the trials become dependent), the results above become slightly better.

5. EXPERIMENTAL RESULTS

Using the predicate in the example given, for predicate coverage, random selection of test cases with no repetition of test cases chosen shows that, on average, 2.776 tests would be necessary (standard deviation of 0.976) with 90% of trials achieving predicate coverage with four tests chosen at random. Unfortunately, four tests double the number of tests necessary in a logical evaluation. Figure 1 displays the results of 10,000 randomly selected test cases without repetition of selection. The selection is discontinued when predicate coverage is achieved, and the number of test cases is counted.

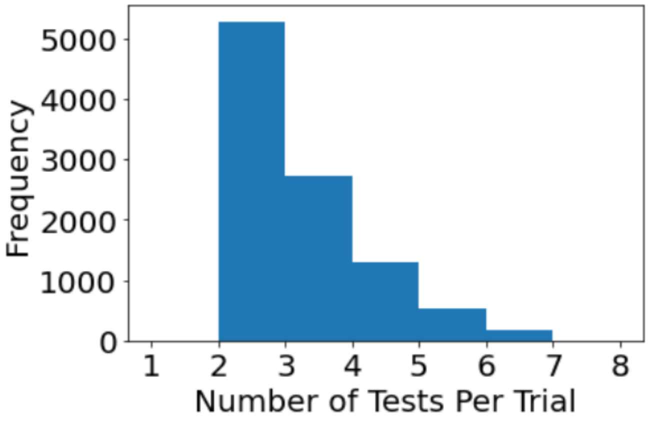


Figure 1. Predicate Coverage Random Simulation. The number of test cases necessary to achieve predicate coverage via random selection as demonstrated with 10,000 trials using the predicate in the example from Table 1.

In the case of the given example, a quick analysis of the logic would yield that test case 1 will have a predicate value of true. Another assumption would be that the predicate will be false more often than it is true. If we decide to choose test case 1 to be one of our predicate coverage test cases but let randomness decide the other, the total number of test cases necessary will obviously be larger than the minimum number of two and fewer than the four previously determined to be necessary for 90% fault detection by pure randomness. Figure 2 displays the results of 10,000 trials of this scenario. Test case 1 is assumed to be true and the other seven test cases are randomly selected. When a test case is found to have predicate value of false, the trial ceases and the number of extra cases is noted.

The average number of tests necessary under these conditions becomes 1.324 with standard deviation 0.553 tests, bringing the total number of tests necessary to 2.324, on average with 90% predicate coverage in 2 additional tests. A total of three tests will achieve predicate coverage for the scenario described. When one test case is chosen logically and one is chosen at random, there is a gain of one test case from purely random selection.

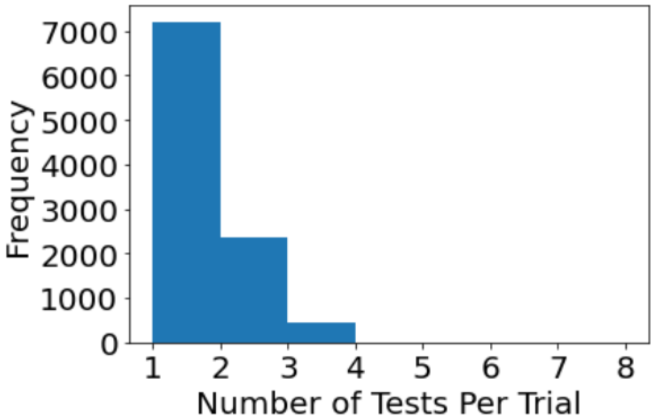


Figure 2. Predicate Coverage Simulated with One Test Case Removed. The results of 10,000 random trials evaluating predicate coverage with one test case determined logically and the other determined randomly.

This is very similar to a hypergeometric distribution (Equation 1) given that two tests are selected, one is required to have predicate value true and the other false. As the number of tests increases, the necessity is finding a predicate with opposing values to those already selected.

$$p(X = k) = \frac{\binom{K}{k}\binom{N-K}{n-k}}{\binom{N}{n}}$$

Equation 1: Hypergeometric probability where N is the population size, *K* is the number of successes in the population, *n* is the number of draws and *k* is the number of successes in the trial. There must be mutually exclusive categories and the trials have probabilities that are dependent according to the previous trial from a finite population.

Probability analysis of these events is extremely interesting but not the direction of this paper. Simulation and predictive modelling will be used instead.

Predicate coverage is not very powerful in finding faults. Our focus will be on the more effective Restrictive Active Clause Coverage. Purely random selection of RACC test cases shows that, on average 6.840 tests would be necessary to achieve RACC with 90% coverage at 8 tests or exhaustive testing. The results are depicted in Figure 3. As noted, a logical analysis indicates that four would suffice. This would, again double the number necessary. However, a superficial analysis of the predicate can yield knowledge that clause c largely determines the predicate.

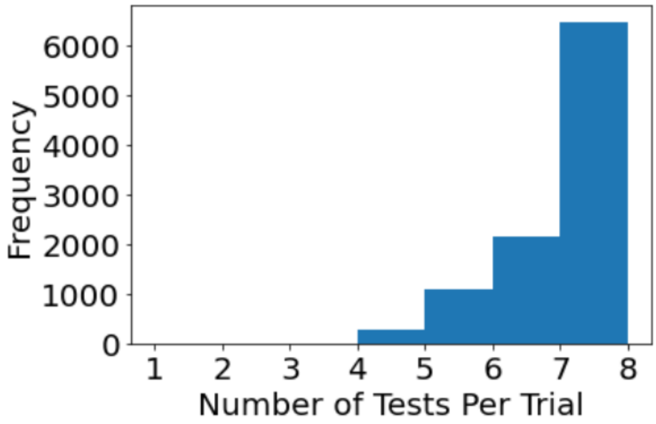


Figure 3. RACC Randomized. The results of 10,000 random trials to achieve restrictive active clause coverage in a three-clause predicate. The tests are randomly chosen until RACC is achieved.

If clause c is activated in test cases 3 and 4, these cases can be eliminated from the random selection and the process re-evaluated. Since RACC can be achieved with either of these minimum test cases {2, 3, 4, 6} or {2, 3, 4, 7}, test cases 2, 3, and 4 are higher priority with test cases 1, 5 helping to activate clause a, but not in the minimum test set. Test 8 is irrelevant in RACC coverage. If the logic to determine two high priority tests can be done easily, we can eliminate those tests from the random selection. Figure 4 shows a simulation of 10,000 trials randomly selected after test cases 3 and 4 were established to be part of the test set necessary to achieve RACC.

Other tests were randomly chosen until RACC was satisfied. The simulated results determined that an average of 3.964 extra tests need to be run, bringing the total number of tests in the test set to just under six. Coverage of 90% would take 6 more tests for a total of eight.

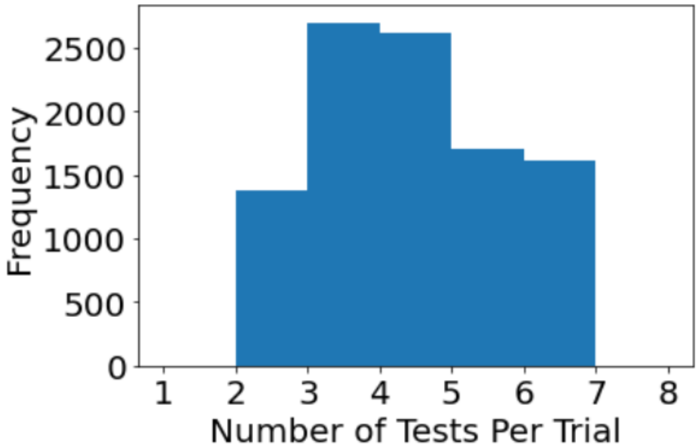


Figure 4. RACC Randomized with One Pair of Tests Removed. The results of 10,000 random trials to achieve restrictive active clause coverage for a three-clause predicate with an influential pair of tests removed and the remaining tests randomly chosen.

Further evaluating the scenario for many predicates involving three clauses it can be found that it is rare to find a predicate in which RACC is NOT achieved with four test cases.

Now consider a four-clause predicate such as p = (a ∨ (b ∧ c)) ∧ d. A truth table for this clause can be found in Table 2. Purely random test case selection to determine RACC will not be considered as there is little gain from the 2n = 16 total cases.

Table 2. A truth table for the four-clause predicate p = (a ∨ (b ∧ c)) ∧ d.

	a	b	c	d	p
1	T	T	T	T	T
2	T	T	T	F	F
3	T	T	F	T	T
4	T	T	F	F	F
5	T	F	T	T	T
6	T	F	T	F	F
7	T	F	F	T	T
8	T	F	F	F	F
9	F	T	T	T	T
10	F	T	T	F	F
11	F	T	F	T	F
12	F	T	F	F	F
13	F	F	T	T	F
14	F	F	T	F	F
15	F	F	F	T	F
16	F	F	F	F	F

To logically achieve RACC for this scenario, the minimum number of test cases would be five. A test set of either {3, 4, 9, 11, 13} or {3, 5, 6, 9, 11, 13} or {5, 6, 9, 11, 13} would provide RACC coverage.

If a RACC pair such as {9, 11} can be determined easily leaving the other three test cases to random selection, an average of 8.878 extra tests are needed. Bringing the total number of tests to about 11 instead of the necessary minimum of five. Figure 5 depicts the number of extra test cases randomly selected to achieve RACC for a four-clause predicate when an influential pair of tests is removed from the sixteen possible tests. The remaining tests are selected randomly until RACC is achieved.

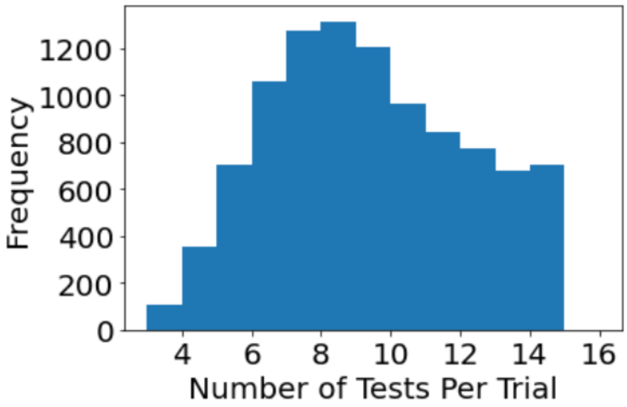


Figure 5. RACC Randomized with One Pair of Tests Removed. In a four-clause predicate with one fixed pair of RACC tests, the number of extra test cases necessary to achieve RACC via random selection of the remaining test cases.

Again, analysing many four-clause predicates, it is rare to find a predicate in which RACC is NOT achieved with five test cases.

Performing the same analyses on a five clause predicate the findings were that a minimum of six test cases were necessary to achieve RACC. If an influential RACC pair of tests are determined and the rest of the necessary case selections are left to chance, an average of 14.917 extra tests would be necessary for a total of 17 tests out of the possible 32. If RACC could be achieved with the minimum test cases of six, a tester would have to run 11 extra tests on average or almost double the number of tests.

Again, the probability analytics for this would be interesting and akin to the analysis of the game "Craps", however predictive modelling and simulation will provide our estimates for these predictions.

6. Discussion

There is good evidence that RACC can be achieved by logically determining a minimum test set which if carefully selected can usually done with n + 1 test cases, where n is the number of unique clauses. It has also been shown that random test selection has little advantage over exhaustive testing of predicates. If a RACC pair of test cases can logically be determined, there becomes two fewer tests necessary to achieve coverage. Examining many predicates of with clause sizes 3, 4, and 5, a generic scenario was developed to determine test sets. Simulation of 10,000 trials for each clause size was run, showing the proportion of tests necessary to achieve RACC. The number of test cases necessary to achieve coverage was subtracted from the minimum necessary given the clause size and divided by the minimum, making a comparison possible. Figure 6 displays the results of the 30,000 simulations depicting the proportion of extra tests necessary.

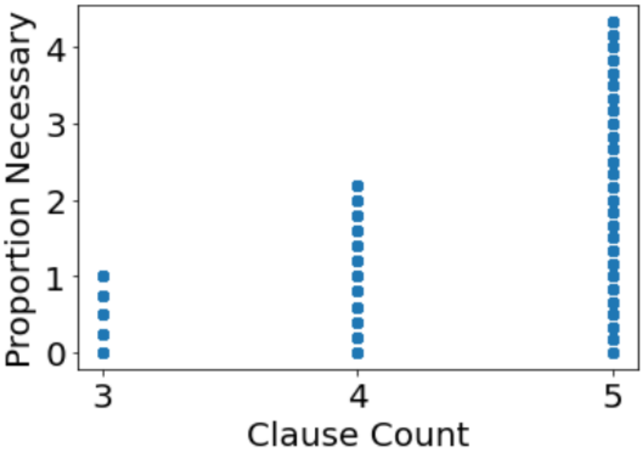


Figure 6. Proportion of Extra Tests per Clause. The proportion of extra test cases necessary to achieve RACC for predicates of clause size 3, 4, and 5 as determined by 30,000 simulated random test cases.

Linear, quadratic, cubic, quartic, and exponential regressions were run on the data to develop models to predict the performance of lager predicates under these conditions. There was little difference in the goodness of fit for the polynomial models. The linear model will not serve as a good choice, consistently under predicting the results. Similarly, the exponential model quickly overpredicting the proportion of test cases, making predictions higher than the total possible number of cases. Extrapolation of clause values outside the frame of reference studied can be dangerous and after a clause size of eight, the models started to disagree. Figure 7 shows the linear, quadratic, cubic, and quartic models graphed over the simulated data. (Note: exponential regression is not pictured.)

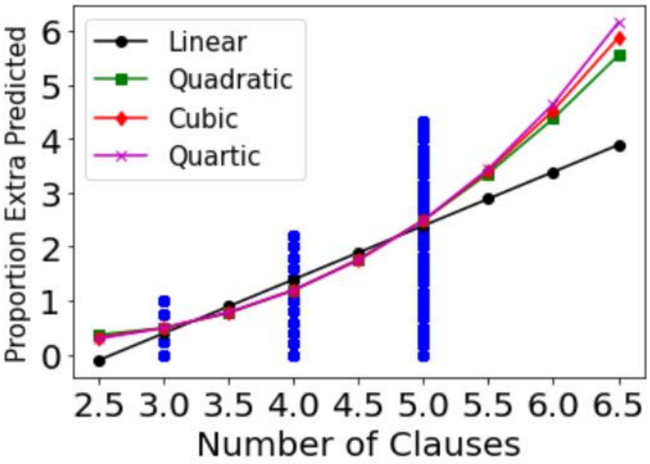


Figure 7. Predictions. Regression equations developed to help predict the proportion of extra test cases necessary to achieve RACC, using the number of clauses as the independent variable.

$$y = -2.600 + 0.996x \qquad \ldots (2)$$
$$y = 2.070 - 1.441x + 0.304x^2 \qquad \ldots (3)$$
$$y = 0.026 - 0.007x - 0.221x^2 + 0.513x^3 \qquad \ldots (4)$$
$$y = 0.002 + 0.006x + 0.010x^2 - 0.0061 + 0.058x^4 \qquad \ldots (5)$$
$$y = 10^{-4.148}(10^{1.0039})^x \qquad \ldots (6)$$

Equations 2 – 6: Predictive equations for the proportion of extra tests run to achieve RACC for a predicate with *x* distinct clauses. Equation (2) is a linear model, equation (3) is a quadratic model, equation (4) is a cubic model, equation (5) is a quartic model, and equation (6) is an exponential model.

Table 3. Predictions for the proportion of extra tests necessary to achieve RACC given the number of distinct clauses in a predicate.

	6	7	8	9
Linear	3.38	4.37	5.37	6.37
Quadratic	4.39	6.91	10.04	13.78
Cubic	4.55	7.53	11.60	16.89
Quartic	4.68	8.10	13.15	20.27
Exponential	75.08	757.66	7645.62	77152.81

To summarize the findings from Table 3. It is possible to predict the average proportion of extra tests necessary to achieve RACC when two influential tests are removed from the random selection with the remaining tests necessary chosen by chance. Each predictive model was evaluated with the clause size six through nine entered as the independent variable. Table 4 summarizes the predictions in terms of raw number of extra tests given the number of clauses, n. The minimum logically evaluated number of test sets is n + 1. The maximum number of tests by exhaustion 2n. The predicted number of extra tests approximated by the models. The total number of tests run including the initial two chosen. The number of extra tests was computed, and the prediction is given as the next largest integer value. The prediction for quadratic, cubic, and quartic were considered. Linear and exponential models were disregarded as under and overestimates, respectively. When the models disagreed, the most conservative (highest) predicted number of tests was considered.

Table 4. Summary of finding in terms of the average number of tests necessary to achieve RACC given the number of clauses.

n	$n + 1$	2^n	Prediction	Total
3	4	8	2	4
4	5	16	6	8
5	6	32	15	17
6	7	64	33	35
7	8	128	65	67
8	9	256	118	120
9	10	512	200	202

7. CONCLUSIONS

The null hypothesis that randomizing test selection yields no advantage in achieving logic coverage in testing was shown to be largely true. There is a slight advantage to randomization of testing, but it is difficult to determine the number of tests run randomly to achieve any specified degree of RACC coverage. Running unnecessary tests is expensive and consumes resources. A balance between randomness and intentional strategy will be the best method testers can employ to find a high degree of faults when logic testing becomes difficult or impossible under tight timelines. Using this predictive analysis may enlighten those to make informed decisions about the potential time necessary to evaluate a predicate logically and the time/expense of running extra test cases to detect the faults in a software. Understanding the risk consequences involved in is also a factor in the decision-making process. Also of note, it has been determined that very few predicates (0.65%) have four or more clauses [1]. However, if higher clause predicates need testing, the tester now can weigh the risks of random selection extra test cases against the time necessary evaluate the scenario logically.

Simulations for all trials employed python programming using Jupyter Notebook from Anaconda.Navigator. Random selection of test cases via input domain partitioning to determine truth values generated nonredundant test cases until the coverage criterion was met. The number of excess test cases necessary was determined. After much study of RACC coverage behaviours, a generic predicate was developed and used for simulations.

8. FUTURE WORKS

This analysis took into consideration that a tester would be able to determine an influential restrictive active clause coverage test pair with nominal time invested. More research would be necessary to determine the ramifications of selecting a test pair that was less influential. Another extension that would have merit for study would be incrementally increasing the test cases chosen logically to see the impact on the proportion of extra test cases necessary for coverage via random selection of the remaining test set. Similar ideas could be employed with different logic coverage choices, such as restrictive inactive clause coverage.

REFERENCES

[1] Ammann, Paul and Offutt, Jeff, *Introduction to Software Testing*, Second Edition, Cambridge University Press, 2016

[2] Chowdhury, Arnab Roy, (2020, March 21) *Test Analytics: What You Should Be Measuring in Your QA*, TestIM, https://www.testim.io/blog/test-analytics-qa/

[3] Clayton, Erna, (2020, January 27) *The Impact of Automated Software Testing on Native Manual Testing*, Developer Tip, Tricks & Resources, https://www.mabl.com/blog/machine-learning-in-testing-bots-vs-humans

[4] Elgabry, Omar, (March 17, 2017) *Software Engineering – Software Process and Software Process Models (part 2)*, Software Engineering — Software Process and Software Process Models (Part 2) | by Omar Elgabry | OmarElgabry's Blog | Medium

[5] Fang, C, Chen, Z, Xu, B. (2012) Comparing Logic Coverage Criteria on Test Case Prioritization, researchgate.net, https://www.researchgate.net/profile/Zhenyu-Chen-5/publication/257686390_Comparing_logic_coverage_criteria_on_test_case_prioritization/links/55d0150108ae6a881385e066/Comparing-logic-coverage-criteria-on-test-case-prioritization.pdf

[6] Hernandez, David Amrani (Dec 3, 2019) *History of Software Testing*, https://medium.com/@davidmoremad/history-of-software-testing-cfa461c4ae0a

[7] Hughs, Troy Martin, (2016), *SAS Data Analytic Development: Dimension of Software Quality*, John Wiley and Sons Inc.

[8] Kaminski, G, Ammann, P, Offutt, J. (2011), Better Predicate Testing, cs.gmu.edu, https://cs.gmu.edu/~offutt/rsrch/papers/ror-logic.pdf

[9] Kenett, Ron S, Fabrizio, Ruggeri, Faltin, Fredrick W, (2018) *Analytic Methods in Systems and Software Testing*, John Wiley and Sons Inc., First Edition.

[10] Kinsbruner, Eran, (2019, August 13) *Manual Testing vs. Automated Testing*, https://www.perfecto.io/blog/automated-testing-vs-manual-testing-vs-continuous-testing

[11] Lee, Jihyun, Kang, Sungwon, Lee, Danhyung, () *A Survey on Software Testing Practices*, https://www.researchgate.net/profile/Sungwon-Kang/publication/260649940_Survey_on_software_testing_practices/links/54b7af070cf2e68eb2803d04/Survey-on-software-testing-practices.pdf

[12] Mackerras, Claire, (2020, January 2) *You Need Predictive Analytics for Your Software Testing: Here's Why, My Tech Decisions,* https://mytechdecisions.com/it-infrastructure/predictive-analytics-software-testing/#:~:text=Predictive%20analytics%20helps%20the%20testing,to%20drive%20better%20application%20efficiencies.

[13] Martinez-Fernandez, Silverio, *Continuously Assessing and Improving Software Quality with Software Analytics Tools: A Case Study*, IEEE Xplore, Continuously Assessing and Improving Software Quality With Software Analytics Tools: A Case Study - IEEE Journals & Magazine

[14] Nicola, (March 25, 2019), *Agile Testing vs. Waterfall Testing*, EuroSTAR, https://huddle.eurostarsoftwaretesting.com/agile-testing-vs-waterfall-testing/

[15] Nederkoorn, Cordny (2016, January). *Data Science from a Software Tester's Perspective*, Sweetcode, https://sweetcode.io/data-science-from-a-software-testers-perspective/

[16] Page, Alan, (), *The Cost of Software Testing?*, CIOReview, The Cost of Software Testing? (cioreview.com)

[17 faker] Sakinala, Krishna, (May 16, 2019) *Test Data Generation for Automation Testing,* Evoke Technologies, Test Data Generation for Automation Testing | Evoke Technlogies (evoketechnologies.com)

[18] Sarro, Federica (2018, May). *Predictive Analytics for Software Testing: Keynote Paper,* SBST'18, https://dl.acm.org/doi/abs/10.1145/3194718.3194730

[19] Scheier, Robert L., *How Predictive Analytics Will Disrupt Software Development*, TechBeacon, How predictive analytics will speed software development, improve quality (techbeacon.com)

[20] Sharma, Sudney, (2020, September 2) *Big Data – Testing Strategy,* https://www.loginradius.com/blog/async/big-data-testing-strategy/

[21] TestIM, (April 15, 2020) *Test Automation ROI: How to Quantify and Measure it.* TestIM, Test Automation ROI: How to Quantify and Measure It (testim.io)

[22] testIM Group (2020, March 21). *Test Analytics: What You Should be Measuring in Your QA*, testIM, https://www.testim.io/blog/test-analytics-qa/

[23] Vardhan, Harsh (2019, December 30). *Appling Data Analytics to Test Automation,* STICKYMINDS, https://www.stickyminds.com/article/applying-data-analytics-test-automation

[24] (December 27, 2011) *How to Calculate ROI for Test Automation*, TestingWhiz, How to Calculate ROI for Test Automation (testing-whiz.com)

AUTHORS

Dr Omar Al-Azzam is an Associate Professor of Software Engineering in the Department of Computer Science and Information Technology (CSIT) at Saint Cloud State University (SCSU). Dr Al-Azzam earned his BSc and MSc from Yarmouk University, Jordan and PhD from North Dakota State University (NDSU). Dr Al-Azzam main research interests are big data analytics, bioinformatics and data mining.

Paul Court is a graduate student in the Professional Science Master of Software Engineering (PSMSE) program at Saint Cloud State University (SCSU) in the Department of Computer Science and Information Technology (CSIT). Mr. Court earned a MEd in Mathematics from the University of Minnesota and a BA in Mathematics from the University of Minnesota, Morris.

TEXT-TO-FACE GENERATION WITH STYLEGAN2

D. M. A. Ayanthi and Sarasi Munasinghe

Department of Computer Science, Faculty of Science,
University of Ruhuna, Wellamadama, Matara, Sri Lanka

ABSTRACT

Synthesizing images from text descriptions has become an active research area with the advent of Generative Adversarial Networks. The main goal here is to generate photo-realistic images that are aligned with the input descriptions. Text-to-Face generation(T2F) is a sub-domain of Text-to-Image generation(T2I) that is more challenging due to the complexity and variation of facial attributes. It has a number of applications mainly in the domain of public safety. Even though several models are available for T2F, there is still the need to improve the image quality and the semantic alignment. In this research, we propose a novel framework, to generate facial images that are well-aligned with the input descriptions. Our framework utilizes the high-resolution face generator, StyleGAN2, and explores the possibility of using it in T2F. Here, we embed text in the input latent space of StyleGAN2 using BERT embeddings and oversee the generation of facial images using text descriptions. We trained our framework on attribute-based descriptions to generate images of 1024x1024 in resolution. The images generated exhibit a 57% similarity to the ground truth images, with a face semantic distance of 0.92, outperforming state-of-the-artwork. The generated images have a FID score of 118.097 and the experimental results show that our model generates promising images.

KEYWORDS

Text-to-Face Generation, StyleGAN2, High-Resolution, Semantic Alignment, Perceptual Loss.

1. INTRODUCTION

With the advent of Generative Adversarial Networks (GANs)[1], image generation has achieved ground-breaking results because of its ability to generate high-quality images that show a close resemblance to real images. However, the original GAN did not have the ability to control the images that it was trained to generate and render images that meet a given specification. In order to overcome this potential issue, various conditional GAN models were proposed over time for different tasks. Text-to-Image generation (TTI) is one such application. It refers to the generation of an image that meets the context specified in a natural language description. We can also call this as the inverse of image captioning as it tries to learn a mapping from the text space to the image space. This emerging research topic is less explored mainly because of its complexity and challenging nature. However, it has a huge number of interesting applications including image processing tasks like image editing, computer-aided design, and computer game development.

David C. Wyld et al. (Eds): FCST, CMIT, SE, SIPM, SAIM, SNLP - 2022
pp. 49-64, 2022. CS & IT - CSCP 2022 DOI: 10.5121/csit.2022.120805

Table 1. Samples from Text2FaceGAN dataset.

 The man has a chubby face. He sports a goatee with sideburns. His hair is black in color. He has narrow eyes and a slightly open mouth. The man looks young.	 The woman has oval face and high cheekbones. She has big lips with arched eyebrows and a slightly open mouth. The smiling, young attractive woman has heavy makeup. She's wearing earrings, necklace and lipstick.

The traditional methods for TTI, mostly use similar frameworks consisting of a pretrained text encoder to encode the text descriptions to a semantic vector and a conditional GAN model as the image decoder. These setups were mostly limited to simpler images like birds and flowers due to the complexity of learning the mapping between the text and the image spaces and the availability of datasets like the CUB bird dataset [2], Oxford Flower bird dataset [3], and MS-COCO dataset [4].

Text-to-Face generation (TTF) is a sub-topic coming under TTI which is particularly focused on generating human faces from natural language descriptions. Similar to TTI, TTF has two main targets; 1) to generate realistic, high-resolution facial images, 2) to generate images that are well aligned with the input descriptions. Compared to TTI, TTF has more value considering the public safety domain. This can be used mainly in criminal investigation and in the preparation of datasets for bio-metric research involving face data like face recognition and age estimation. Realistic faces can be generated to assist identifying criminals by automating the task of a sketch artist and in place of a sketch, this could be used to generate an image that is favourable in the investigation. The face datasets that are used in research like face detection and age estimation are typically compiled by scraping images from the internet, mostly, without the consent of the owner and this raises ethical concerns. Another major issue with these datasets is racial biases like minorities not being represented properly. Using datasets composed of synthetically generated images is a possible solution to this.

Most of the existing T2F frameworks have only been able to produce low-resolution images that have poor consistency with the input descriptions. It is important to be able to generate high-resolution images to use TTF in the above-mentioned applications. Recent progress on GANs has established a remarkable paradigm on image generation in terms of quality, fidelity, and realism.
StyleGAN2[5] is one of the most significant GAN frameworks that has been introduced and with its style-based generator architecture it is able to produce high-resolution images with unmatched photorealism. It has not only been trained to generate human faces but also other images. However, when comparing the images generated through TTF frameworks and those generated through GAN models specialized for face generation, like the StyleGAN2 there is a clear difference. The traditional multi-staged architectures and the progressive training of TTF frameworks have not been able to generate quality images. Also, it can be seen that the prospect of using a high-resolution generator for TTF has not been considered. Taking this into consideration we propose a novel framework for TTF using the StyleGAN2 generator. Here we aim to represent text descriptions in the latent space of the StyleGAN2 generator and thereby generate facial images. We propose a simpler model that can find a mapping between the text

space and the image space and use this as a means to generate images from natural language descriptions. We use BERT [6] sentence encoder as the language model and the StyleGAN2 generator as the image generator. We use the sentence level embeddings obtained from the sentence encoder to learn a text-to-latent model, that maps the descriptions to the input space of the state-of-the-art generator, StyleGAN2 to generate images.

Our main contributions are as follows,

- Propose a novel TTF framework using StyleGAN2 as the image generator and BERT sentence encoder as the language model
- Generate high-resolution images that are properly aligned with the input descriptions.

The rest of the paper is organized as follows. In Section 2, we discuss the related work and in Section 3 we discuss the proposed methodology including the datasets and the other preliminary frameworks used in this work. The experimental analysis, results obtained and the evaluation is presented in Section 4 and lastly, in Section 5 we discuss the conclusion and the future work.

2. RELATED WORK

Text-to-Face generation has been greatly influenced by the work done in Text-to-Image generation. Apart from that, image generation (Face generation) is another domain to be considered. This section discusses the important work that was done in these three fields.

2.1. Text-to-Image generation

The main goals of these models are two-fold. One is to generate high-quality images and the other is to generate images that are properly aligned with the input description. During the early stage of the development of these models, it was focused on generating high-quality images. The first model to take advantage of a GAN was presented in 2016 by Reed et al [7]. They contributed with two models for text-to-image generation using conditional GANs. They used a pretrained Char-CNN-RNN network [8] as the text encoder, a model similar to the DCGAN [9] as the image decoder, and produced images of 64x64 and 128x128. This model was unable to find a good mapping between the keywords and the image features due to the direct concatenation of the text embeddings with noise inputs. To overcome this issue, StackGAN [10] was proposed. It was a GAN model with two stages. The first stage captures some information in the description and generates an initial, low-resolution image. During the second stage, the image is refined with the description to produce images of 256x256. This network, with hierarchically nested generators, is used in most of the later approaches as well [10][11][12]. Even so, generating high-resolution images like 1024x1024 is very expensive using this kind of architecture.

When it was possible to generate realistic images, the next target was to generate images to improve the similarity between the generated image and the input description. Among the various proposed approaches, the attention mechanism is significant. This was first proposed in the context of image generation by Xan in AttnGAN. They introduced a word-level attention mechanism that enabled the generation of fine-grained images of 256x256. However, the word level attention alone was not enough and lead to the generation of unrealistic images. Another issue with the model was that as this model was trained on descriptions with mostly one sentence, it was unable to handle longer sentences. In spite of these shortcomings, this attention mechanism has been the base model for most of the work that was conducted later [13][14].

2.2. Image generation (Face generation)

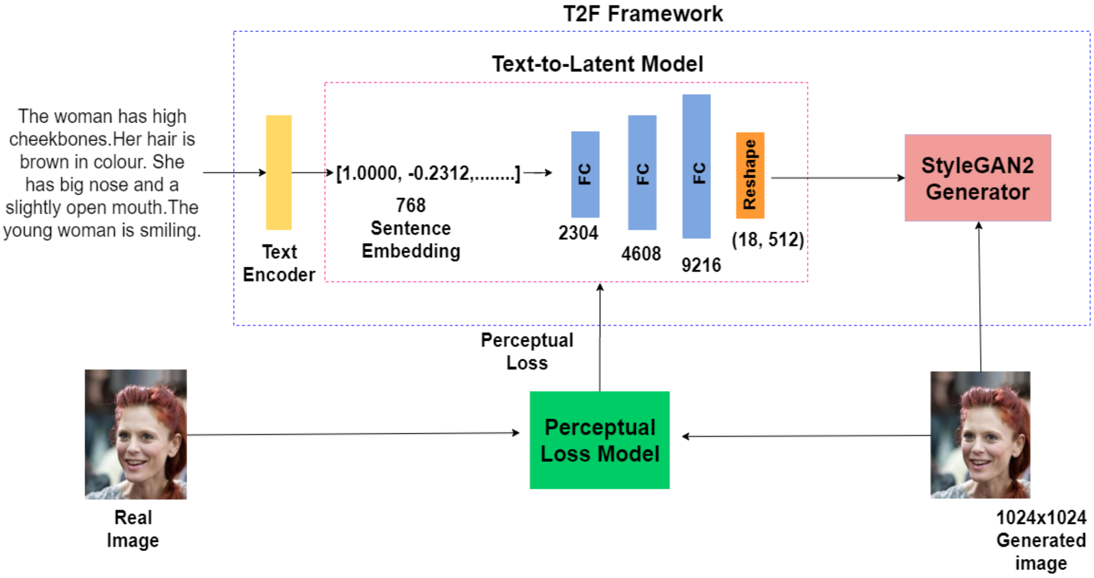


Figure 1. Proposed Model Architecture

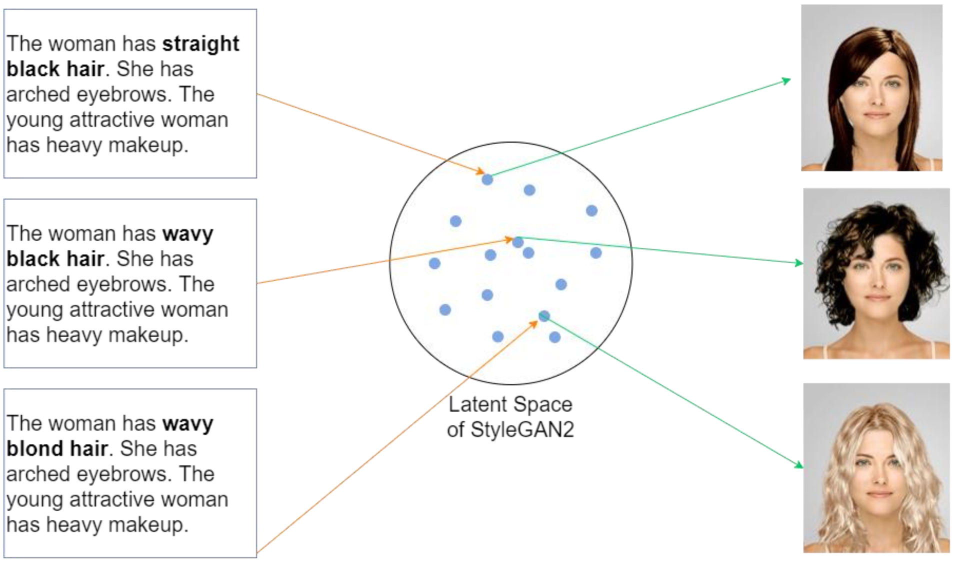
Image generation models using GANS, learn a mapping from a noise vector to the normal distribution corresponding to real images. With the availability of two large datasets, LFW [15] and CelebA[16], face synthesis is also a trending research area. Most of these models are based on conditional GANs. Some of the popular models are ProGAN [17], StyleGAN [18], and StyleGAN2. Instead of attempting to train all the layers of the generator and the discriminator at once, in ProGAN they have gradually grown the GAN one layer at a time, to generate high-resolution images gradually. Using this approach, they have been able to start from images of 4x4 and gradually increase up to 1024x1024. StyleGAN and StyleGAN2 are also built on top of ProGAN, but they have control over the style factors, unlike ProGAN. The images generated by StyleGAN2 are very realistic that they cannot be easily recognized even by humans as fake, generated images. The only disfunction here is that there is no control over the generated images. Therefore, it cannot generate a particular face at our request.

2.3. Text-to-Face Generation

Compared to the work done in T2I there is a far lesser amount of work done in T2F. The main reason behind this is the variety of facial attributes in terms of ethnicity, age, and the facial descriptions being vague about the attributes. Even with the said challenges, there is a smaller number of inspiring work done in text-to-face generation. In the project T2F by Akanimax [19], he proposed to encode text descriptions into a summary vector using an LSTM and use ProGAN as the generator. As the image quality was poor, they used MSG-GAN [20] as the generator and improved the image quality. Text2FaceGAN [21] was based on the GAN-INT-CLS architecture by Reed et al [7]. The Text2Face dataset was also introduced using the attributes of the CelebA dataset and an algorithm for caption generation. They could only produce images of resolution 64x64. Here they also showed howInception Score [22] which is a generally used metric in GAN evaluation is not suitable to be used with facial images.

FTGAN [23], proposed an architecture that combined the training of the text encoder and the image decoder that was done separately so far. The main idea was that to generate quality images the text encoder and the image decoder needs to be trained together because when using a pretrained text encoder, the input to the image decoder is too dependent on the output of the text encoder. However, with this proposed architecture, FTGAN produced images up to 256x256. Another similar approach is given in [24]. Instead of the BiLSTM in FTGAN here they are using a Char-CNN-RNN [8] to obtain the semantic vector and train both the text encoder and the image decoder at the same time. With this approach, they obtained slightly better images but still, they could produce images only up to 256x256. TTF-HD [25] proposed a framework consisting of a multi-label text classifier, an image label encoder, and an image generator to generate facial images of 1024x1024. With the multi-label classifier, they have been able to consider a feature disentangled latent space and focus on the diversity of the images generated other than the quality and the semantic consistency. However, utilizing the ability of StyleGAN2 to produce high-quality facial images has not been considered much in this work.

When referring to the relevant literature we could observe that most T2F models were based on architectures introduced for T2I generation and yet the image quality was far less than those naïve GAN models specialized for face generation. Generating high-resolution facial images is still a problem that has not been addressed properly. Also, the usage of existing high-resolution generators in T2F is minimal. Therefore, there is a need to explore the possibility of using face generation models like StyleGAN2 in T2F to generate high-resolution images. It was also identified that newer language models like BERT have not been used for this task. Based on these gaps, we conduct experiments in this paper to determine the possibility of using StyleGAN2 in T2F and develop a novel framework for T2F to generate high-resolution facial images that are consistent with the input descriptions.



The woman has **straight black hair**. She has arched eyebrows. The young attractive woman has heavy makeup.

The woman has **wavy black hair**. She has arched eyebrows. The young attractive woman has heavy makeup.

The woman has **wavy blond hair**. She has arched eyebrows. The young attractive woman has heavy makeup.

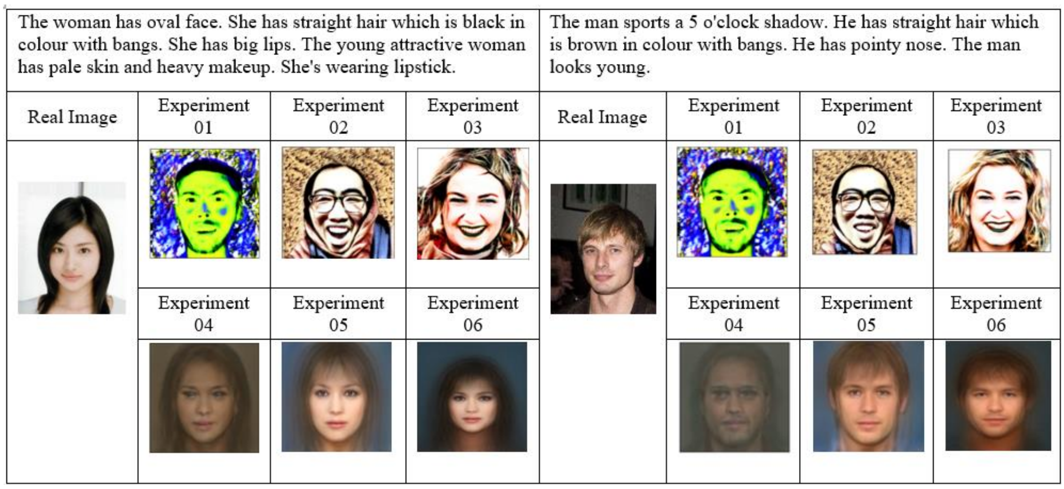
Figure 2. Representation of a description in the latent space of the StyleGAN2 and the corresponding images when attributes in the description are changed.

3. Proposed Methodology

This section provides the details of the dataset, proposed architecture for T2F and the evaluation metrics. The proposed approach is a pipeline consisting of several steps and the complete architecture is presented in Figure 1.

3.1. Dataset

There are several datasets used by T2F models like the Face2Text[26], SCU-Text2Face[23], and Text2FaceGAN[21]. Face2Text dataset consists only of 400 image-description pairs, which is insufficient and the SCU-Text2Face dataset is not publicly available. Therefore, we used the Text2FaceGAN dataset. This is a set of captions generated for the images in the CelebA dataset, using the attribute list provided. The CelebA dataset has 40 facial attributes and the Text2FaceGAN dataset uses them in the captions that have been created using an algorithm. Table 1 shows some sample records from the dataset. We chose a sample of 6000 images from the Text2FaceGAN dataset for training and testing the model due to the limited resources.



The woman has oval face. She has straight hair which is black in colour with bangs. She has big lips. The young attractive woman has pale skin and heavy makeup. She's wearing lipstick.

The man sports a 5 o'clock shadow. He has straight hair which is brown in colour with bangs. He has pointy nose. The man looks young.

Figure 3. Experimental results obtained for all experiments conducted in both Z and W latent spaces.

3.2. Proposed Model Architecture

The proposed architecture for T2F consists of four modules. 1) Text encoder 2) Text-to-Latent model 3) Image generator 4) Perceptual loss model. In the following sections we have discussed them in detail.

3.2.1. Text encoder

The text encoder is used to extract the semantic vectors of the input descriptions. This is done to bring the text space and the image space to some common grounds, to learn how to map the descriptions to the images. We use the state-of-the-art language model, BERT (Bidirectional Encoder Representations from Transformers) as the text encoder to extract the semantic vectors from the input descriptions in the form of a sentence embedding. We used the BERT-as-a-service facility to ease the task.

BERT model producing state-of-the-art results in major Natural Language Processing tasks was the main reason for selecting BERT as our language encoder. The technical innovation behind BERT is the application of bi-directional training of transformers into language processing tasks that gives the model a deeper sense of the language context and flow than single-directional models. So, unlike the other context-free word embeddings like GloVe[27] or Word2Vec[28], BERT embeddings are produced considering the context the words are used in. This was another reason for the selection of BERT as the language encoder. We chose the BERT BASE model, which has a stack of 12 transformer layers and returns an embedding vector of 768 dimensions for a given description.

3.2.2. Text-to-Latent model

To use text as an input to the StyleGAN2, we need to learn how to represent text in the latent space of the StyleGAN2 model. For this purpose, we build a model that maps the input text description in the form of a text embedding to the input latent space and uses this as the input to the StyleGAN2 generator. This in turn allows us to control the images generated through the StyleGAN2 using a text description. Figure 2. gives an idea of the representation of descriptions in the latent space.

In StyleGAN2 there are two latent spaces; the initial latent space Z and the intermediate latent space W. These latent spaces have different properties and a lot of experiments are carried out to identify the properties and get a good understanding of these latent spaces at present. In representing the text in the latent space of the StyleGAN2 generator using the text-to-latent model, we experimented on both the latent spaces. The proposed text-to-latent space model is a multi-layer perceptron consisting of fully connected layers to transform from the BERT distribution space to the input latent space of the StyleGAN2.

3.2.3. StyleGAN2 generator

The output of the text-to-latent model will be passed to the high-resolution generator, StyleGAN2. It is an improved version of StyleGAN and was released by NVIDIA in 2020. This high-resolution generator produces faces with unmatched realism. Therefore, in this paper, we explore ways to represent text in the input latent space of the generator and use it in T2F rather than training a new generator from scratch. The main reason for the selection of the StyleGAN2 generator is the state-of-the-art results it produced in face generation with the introduction of features like weight demodulation, path length regularization, and removal of progressive growing. These features have led the way to overcome problems in the StyleGAN architecture like the generation of phase artifacts and the droplet effect.

3.2.4. Perceptual Loss Model

When coming to image enhancement work like image super-resolution, colorization and style transfer the loss function intends to evaluate how far the generated/predicted output of the model is from target/ground truth image, to train the model to minimize the loss. The goal of our study is to define a model to represent text in the latent space of StyleGAN2 to control the images generated. In this case, as well, we need to visually match the generated image with the ground truth i.e., the features generated in the image should be close to the features in the real image. The most commonly used loss functions in image enhancement processes are the pixel loss based on mean squared error (MSE), root mean squared error (RMSE), or peak-signal-to-noise ratio (PSNR). However, we chose feature loss or perceptual loss[29], which is a better measure because perceptual loss allows to reconstruct finer details of images compared to per-pixel loss. We used VGG16[30] to extract features of the input images from selected layers for the perceptual loss calculation.

The generated image from the StyleGAN2 generator and the corresponding real image are fed into the perceptual loss model. The activation maps, called feature maps, capture the result of applying the filters to input images. The feature maps close to the input detect small or fine-grained detail, whereas feature maps close to the output of the model capture more general features. Our aim is to generate images representing features of the ground-truth image.

Therefore, we need a balance of layers from the top and bottom. We experimented with several layer combinations for this reason. The difference in the selected feature maps of the real image and the generated images were used to update the text-to-latent model.

3.3. Evaluation Metrics

The goal of our model is to generate realistic images that are closely aligned with the input descriptions. The alignment of the images is measured by comparing the distance between the facial features of both images and the cosine similarity of them. The distance between the features is called Face Semantic Distance (FSD) and the similarity is called Face Semantic Similarity (FSS). FSD and FSS are calculated using equations (1) and (2).

$$Face\ Semantic\ Distance = \frac{1}{N}\sum_{i=0}^{N}\left|\left(F_{G_i}\right) - \left(F_{GT_i}\right)\right| \tag{1}$$

$$Face\ Semantic\ Similarity = \frac{1}{N}\sum_{i=0}^{N} cos\left(F_{G_i} - F_{GT_i}\right) \tag{2}$$

In the above equations, F_{G_i} is the feature vector of the generated image and F_{GT_i} is the feature vector of the real image. Cos indicates the cosine similarity between the feature vectors.

Generated images will be compared against the real images to measure how far they are realistic using the FID score[31]. It summarizes how similar the real and generated images are in terms of statistics on computer vision features of the raw images calculated using the Inceptionv3 model [32] used for image classification. Lower scores indicate the two groups of images are more similar or have more similar statistics. Fréchet distance also called the Wasserstein-2 distance is calculated using the equation (3).

$$d^2 = \left||mu_1 - mu_2|\right|^2 + Tr\left(C_1 + C_2 - 2\sqrt{C_1 * C_2}\right) \tag{3}$$

The score is referred to as d2, showing that it is a distance and has squared units. The “mu1” and “mu2” refer to the feature-wise mean of the real and generated images. The C1 and C2 are the covariance matrix for the real and generated feature vectors, often referred to as sigma. The ||mu1 – mu2||2 refers to the sum squared difference between the two mean vectors. Tr refers to the trace linear algebra operation (the sum of the elements along the main diagonal of the square matrix). The sqrt is the square root of the square matrix, given as the product between the two covariance matrices.

All images were scaled to 299x299 before calculating the scores.

Table 2. Summary of all the experiments conducted.

Experiments in the initial latent space Z		**Experiments in the intermediate latent space W**	
Experiment No.	Layers of VGG16	Experiment No.	Layers of VGG16
01	Conv4_3 Conv5_3	04	Conv4_3 Conv5_3
02	Conv3_2 Conv4_2 Conv5_2	05	Conv3_3 Conv4_3 Conv5_3

03	Conv3_2 Conv4_2 Conv5_2 with Hyper-columns	06	Conv1_2 Conv2_2 Conv3_2 Conv4_3

4. EXPERIMENTS AND EVALUATION

This section discusses the extensive experimental analysis that has been carried out to evaluate the performance of the proposed model.

In this paper, we aim to represent text in the input latent space for StyleGAN2 to produce quality images that can be controlled using a text description. Therefore, the latent space of the StyleGAN2 generator is of great importance. The architecture of the StyleGAN2 generator uses two latent spaces; Z and W. The initial latent space Z is an entangled latent space and the intermediate latent space W shows more properties of disentanglement. However, the dimensionality of both the latent spaces is 512. In developing the proposed framework to project text to the latent space of the StyleGAN2 model, first, we need to determine the latent space that works best for our framework. The optimization of the Text2LatentSpace model is done using the perceptual loss model that uses a perceptual loss function between the feature maps of the generated images and the real images obtained through the VGG16 network. Different layers in the VGG16 network extract different features. Therefore, it is necessary to choose a suitable layer or layer combination to be used as the feature extractors.

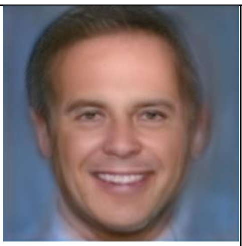
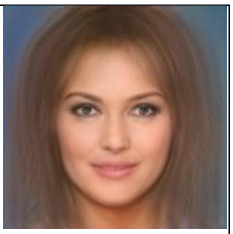
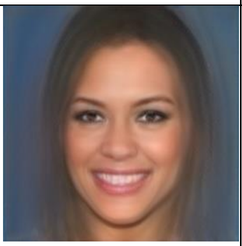
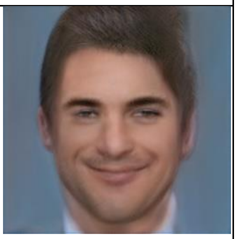
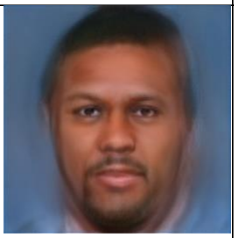
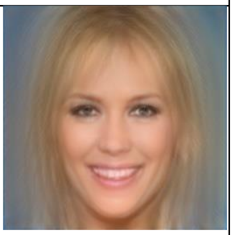
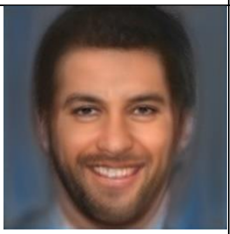
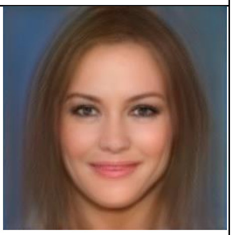
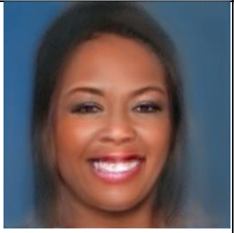
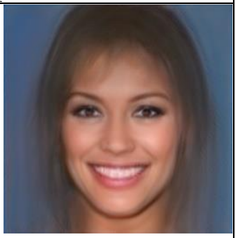
In this context, there were two sets of experiments designed to reach our objectives. One to choose the most suitable latent space and the other to choose the best layer combination for feature extraction in the perceptual loss model. Table 2 gives a summary of all the experiments conducted with different configurations. The layers here are named following the layer names used in the VGG16[30]. Figure 3. shows results obtained from each experiment.

Through the experimental analysis, we observed that experiments conducted in the initial latent space were not successful. The model was unable to recognize the features in the descriptions and led to the generation of images with very slight changes that did not correspond to the descriptions. Apart from that, the generated images were unrealistic. This was due to the entangled nature of the initial latent space. However, the experiments conducted by projecting the descriptions to the intermediate latent space of the StyleGAN2 generator produced better results. These images were aligned with the descriptions as well as more realistic. From these experiments, we observed that experiment 05, which was done in the intermediate latent space using the conv3_3, conv4_3, and conv5_3 produced the best results both in terms of realism and semantic alignment. This model was trained for 500 epochs with an initial learning rate of 0.0001 and the Adam optimizer. Table 3 shows some images generated with their descriptions.

4.1. Qualitative Evaluation

Image quality: Generating visually appealing facial images is one of the main goals of this paper. Figure 4. shows some example facial images generated with the proposed model. We can see that all facial features have been correctly rendered in the generated images and the results are visually appealing to a greater extent. It also shows the ability of the proposed model to generate various faces across different facial features like gender, hair, smile, and age.

Table 3. Sample generated images and their input descriptions.

 The man has a double chined face. He sports a 5 o'clock shadow. He has a receding hairline. He has big lips and big pointy nose and a slightly open mouth. The man is smiling. He's wearing necktie.	 The woman has oval face and high cheekbones. She has wavy hair which is brown in colour with bangs. She has big lips and pointy nose with arched eyebrows. The young attractive woman has heavy makeup. She's wearing a necklace and lipstick.
 The woman has high cheekbones. Her hair is black in colour with bangs. She has big lips with arched eyebrows and a slightly open mouth. The smiling, young attractive woman has rosy cheeks and heavy makeup. She's wearing earrings, necklace and lipstick.	 The man has high cheekbones. He sports a 5 o'clock shadow. He has straight hair. He has big nose, narrow eyes with bushy eyebrows. The young attractive man is smiling. He's wearing necktie.
 The man has a chubby face. He sports a goatee and mustache. His hair is black in colour. He has big lips and big nose. The man looks young. He's wearing necktie.	 The woman has high cheekbones. She has wavy hair which is blond in colour with bangs. She has pointy nose and a slightly open mouth. The smiling, young attractive woman has heavy makeup. She's wearing earrings and lipstick.
 The man has high cheekbones. He sports a goatee with sideburns. His hair is black in colour. He has big nose with bushy eyebrows and a slightly open mouth. The young attractive man is smiling. He's wearing necktie.	 The woman has high cheekbones. Her hair is brown in colour. She has arched eyebrows. The smiling, young attractive woman has rosy cheeks and heavy makeup. She's wearing lipstick.
 The chubby double chined woman has high cheekbones. She has a receding hairline. She has big lips and big nose, narrow eyes with arched eyebrows and a slightly open mouth. The smiling, young woman has heavy makeup. She's wearing earrings, necklace and lipstick.	 The woman has oval face and high cheekbones. Her hair is black in colour with bangs. She has big lips and pointy nose with arched eyebrows and a slightly open mouth. The smiling, young attractive woman has rosy cheeks and heavy makeup. She's wearing earrings and lipstick.

Semantic alignment: The second goal is to generate images that are consistent with a given description. Generating images that are close to the real images is the most natural way to show the semantic alignment of the images with their descriptions. Table 5 shows how close the generated images are to the real images corresponding to the description. Our model generates images that are similar to the real image to a certain extent. However, the generated images represent most of the features in the descriptions and it shows that the generated images have a good consistency with the descriptions.

Another interesting observation made during the generation of the images is the sensitivity to facial attributes in the descriptions. Table 6 shows how generated images through the proposed framework can be manipulated by changing the attributes in the description. Changes in the description are clearly shown on the generated images. This shows how well the model has learnt the facial attributes in the descriptions and thereby, the semantic consistency of the generated images.

4.2. Quantitative Evaluation

The generated images are quantitatively evaluated for the quality and the semantic consistency using the FID score, and the similarity shown to the real image corresponding to the description using the Face Semantic Similarity and the Face Semantic Distance.

Table 4. Comparison with other face generation models using the FSD and FSS criterion.

Model	**FSD**	**FSS (%)**
AttnGAN [11]	1.269	59.28
StackGAN [10]	1.310	-
FTGAN [23]	1.267	59.41
Realistic Image Generation of Face [24]	1.118	-
Ours	**0.9224**	**56.96**

As shown in the above table, our model is performing very close to the state-of-the-artwork. From the table, we interpret that our model has an FSD of 0.9224 which is lower compared to AttnGAN[11], StackGAN[10], FTGAN[23], and Realistic Image Generation of Face from Text[24] models. This tells us that our model is capable of generating images closer to the ground truth better than the other models. Our model can generate images that are almost 57% closer to the ground truth images. This is most likely due to the limited dataset we have used in the training process. We believe if a larger dataset is used the results would be even better. However, these images show high consistency with the input descriptions. Our model generates images of resolution 1024x1024, whereas the other models are only capable of generating images of resolution 256x256. This is one of the biggest contributions of our model. The images generated with our model have an FID score of 118.097. However, we cannot directly compare the FID scores of these models to ours because of the difference in the resolution of the images.

5. Conclusion and Future Work

Our principal objective was to develop a novel framework for T2F that can generate realistic, high-resolution images that are consistent with the input descriptions. The availability of high-resolution face generation models and the need to explore the use of them in T2F generation was the main motivation behind this paper. In that sense, we focused on three tasks.

- To utilize StyleGAN2 generator for T2F to produce high-resolution images.
- To generate images that are semantically aligned with the input descriptions.
- To measure the semantic consistency of the generated images against the real images.

We proposed a model, comprising the BERT language model, StyleGAN2 generator, and a text-to-latent space model to achieve those tasks. Here we embedded the descriptions in the latent space of the StyleGAN2 generator and thereby controlled the facial images generated using text descriptions.

With this approach, we have been able to achieve our goal of generating realistic facial images that are aligned with the input descriptions. Furthermore, we have made use of an existing high-resolution generator and opened up for more work on exploring the task of using the latent space of the StyleGAN2 for controlling the images generated using text descriptions. From both, the quantitative and qualitative evaluative comparisons we can see that the generated images exhibit good image quality and consistency with the input descriptions. We were able to generate images of 1024x1024 in resolution and these images showed 57% similarity to the real image, performing better than most recent work. We achieved these results using a smaller dataset, due to limitations in resources. Therefore, we can conclude that with a larger dataset and exposure to more facial attributes, this approach would produce even better results.

However, we still need to improve the image quality and the consistency with the descriptions. Appearance enhancing attributes like earrings, necklaces, caps were not visible in the images, and images generated using less frequent attributes did not show good realism. Therefore, we believe using a larger dataset would help tackle this issue when being exposed to more facial attributes. We further hope to work on focusing on the diversity of the facial images generated. So far in this model, we have only focused on generating one image per description. However, a single description corresponds to many facial images. Therefore, this too is an area that needs to be focused on.

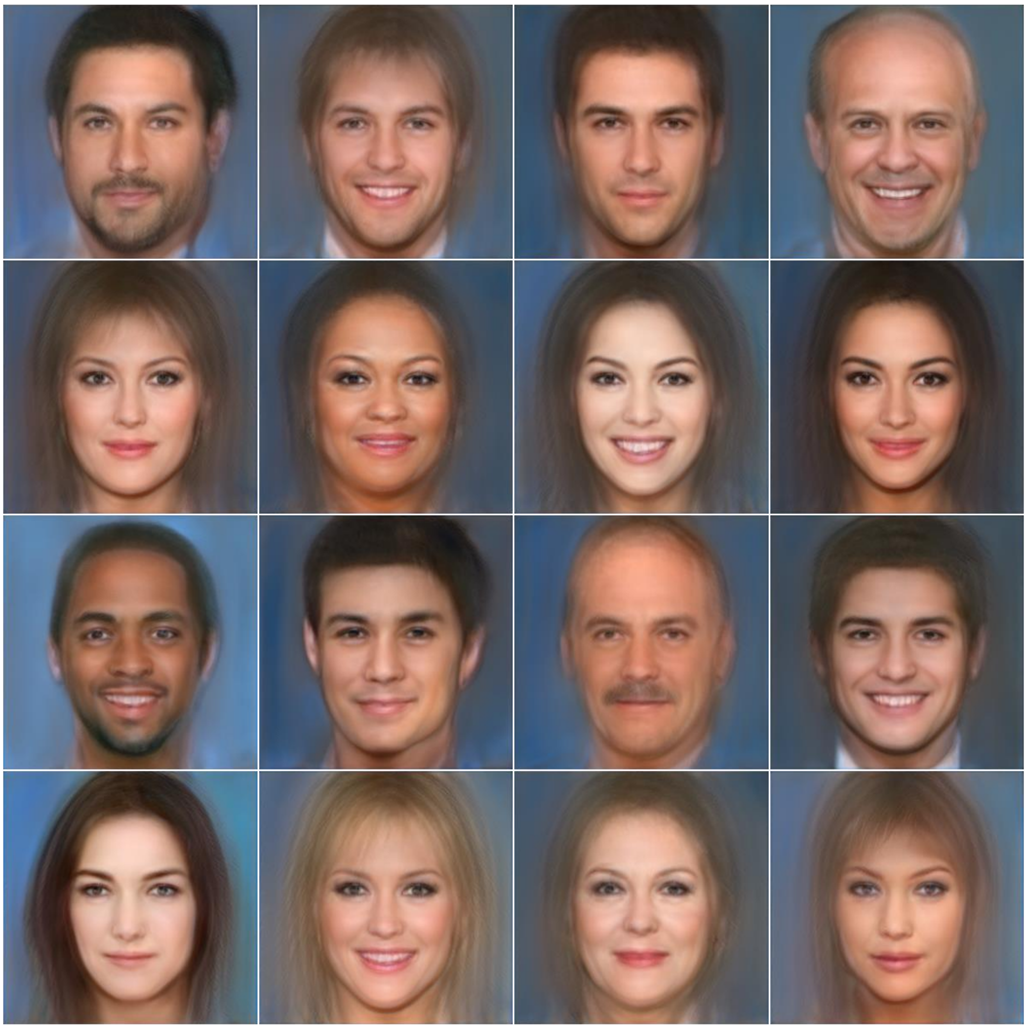
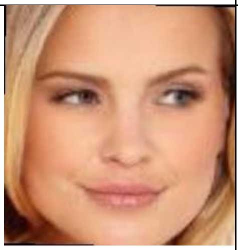
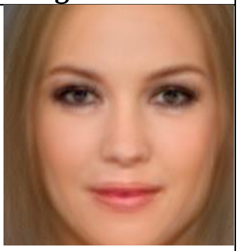
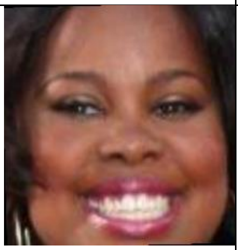
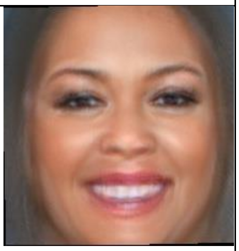


Figure 4. Sample images generated by the proposed model.

Table 5. Generated images with their corresponding real images and descriptions.

Description	**Real Image**	**Generated Image**
The woman has oval face. Her hair is blond in colour. She has arched eyebrows. The smiling, young attractive woman has heavy makeup. She's wearing lipstick.		
The chubby double chined woman has high cheekbones. She has a receding hairline. She has big lips and big nose with arched eyebrows. The smiling, young woman has heavy makeup. She's wearing earrings and lipstick.		

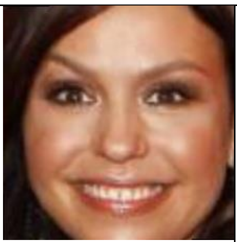
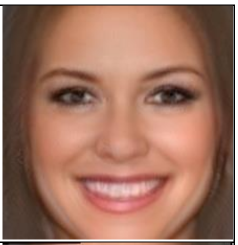
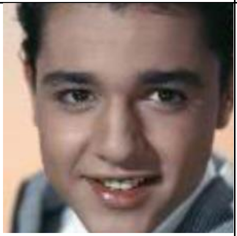
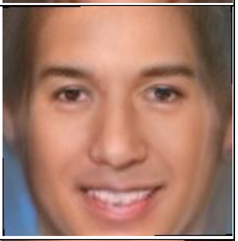
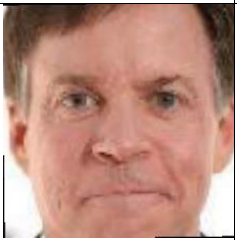
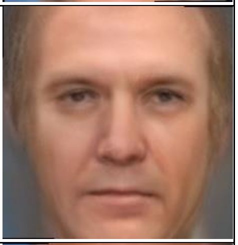
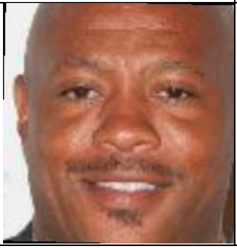
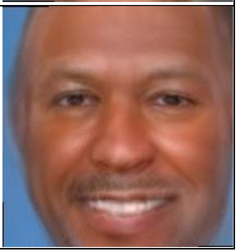
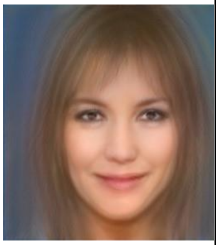
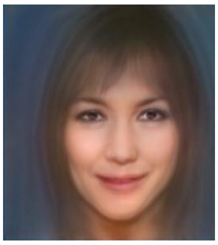
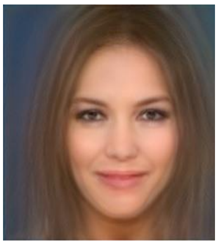
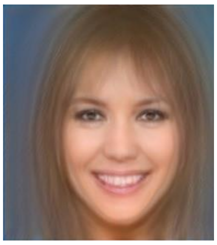
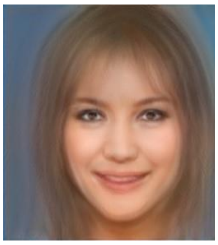
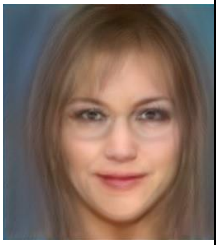
The woman has oval face and high cheekbones. She has wavy hair. She has arched eyebrows and a slightly open mouth. The smiling, young attractive woman has rosy cheeks and heavy makeup. She's wearing earrings, necklace and lipstick.		
The man has wavy hair which is black in colour. He has big lips and big nose with bushy eyebrows and a slightly open mouth. The young attractive man is smiling. He's wearing necktie.		
The man has straight hair which is brown in colour. He has big nose. He's wearing necktie.		
The chubby double chined man has oval face and high cheekbones. He sports a 5 o'clock shadow, goatee and moustache. He is bald. He has big lips and big nose and a slightly open mouth. The man is smiling. He's wearing necktie.		

Table 6. Manipulating Generated Images.

The woman has high cheekbones. She has straight hair which is brown in colour with bangs. The smiling, young attractive woman has heavy makeup. She's wearing lipstick.				
				
Black Hair	No Bangs	Open Mouth	Chubby Face	Wearing Eyeglasses

REFERENCES

[1] I. J. Goodfellow et al., "Generative adversarial nets," Adv. Neural Inf. Process. Syst., vol. 3, no. January, pp. 2672–2680, 2014.

[2] B. Englert and S. Lam, "On the use of XML for port communications," IFAC Proc. Vol., vol. 42, no. 15, pp. 50–57, 2009, doi: 10.3182/20090902-3-US-2007.0059.

[3] M. E. Nilsback and A. Zisserman, "Automated flower classification over a large number of classes," Proc. - 6th Indian Conf. Comput. Vision, Graph. Image Process. ICVGIP 2008, pp. 722–729, 2008, doi: 10.1109/ICVGIP.2008.47.

[4] T. Y. Lin et al., "Microsoft COCO: Common objects in context," Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics), vol. 8693 LNCS, no. PART 5, pp. 740–755, 2014, doi: 10.1007/978-3-319-10602-1_48.

[5] T. Karras, S. Laine, M. Aittala, J. Hellsten, J. Lehtinen, and T. Aila, "Analyzing and improving the image quality of stylegan," Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit., pp. 8107–8116, 2020, doi: 10.1109/CVPR42600.2020.00813.

[6] J. Devlin, M. W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," NAACL HLT 2019 - 2019 Conf. North Am. Chapter Assoc. Comput. Linguist. Hum. Lang. Technol. - Proc. Conf., vol. 1, no. Mlm, pp. 4171–4186, 2019.

[7] S. Reed, Z. Akata, X. Yan, L. Logeswaran, B. Schiele, and H. Lee, "Generative adversarial text to image synthesis," 33rd Int. Conf. Mach. Learn. ICML 2016, vol. 3, pp. 1681–1690, 2016.

[8] S. Reed, Z. Akata, B. Schiele, and H. Lee, "Learning Deep Representations of Fine-grained Visual Descriptions," Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit., vol. 2016-December, pp. 49–58, May 2016, Accessed: Oct. 05, 2021. [Online]. Available: https://arxiv.org/abs/1605.05395v1.

[9] A. Radford, L. Metz, and S. Chintala, "Unsupervised representation learning with deep convolutional generative adversarial networks," 4th Int. Conf. Learn. Represent. ICLR 2016 - Conf. Track Proc., pp. 1–16, 2016.

[10] H. Zhang et al., "StackGAN++: Realistic Image Synthesis with Stacked Generative Adversarial Networks," IEEE Trans. Pattern Anal. Mach. Intell., vol. 41, no. 8, pp. 1947–1962, 2019, doi: 10.1109/TPAMI.2018.2856256.

[11] T. Xu et al., "AttnGAN: Fine-Grained Text to Image Generation with Attentional Generative Adversarial Networks," Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit., pp. 1316–1324, 2018, doi: 10.1109/CVPR.2018.00143.

[12] Z. Zhang, Y. Xie, and L. Yang, "Photographic Text-to-Image Synthesis with a Hierarchically-Nested Adversarial Network," Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit., pp. 6199–6208, 2018, doi: 10.1109/CVPR.2018.00649.

[13] W. Li et al., "Object-driven text-to-image synthesis via adversarial training," arXiv, 2019.

[14] T. Qiao, J. Zhang, D. Xu, and D. Tao, "Mirrorgan: Learning text-to-image generation by redescription," Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit., vol. 2019-June, pp. 1505–1514, 2019, doi: 10.1109/CVPR.2019.00160.

[15] Z. Liu, P. Luo, X. Wang, and X. Tang, "Deep learning face attributes in the wild," Proc. IEEE Int. Conf. Comput. Vis., vol. 2015 Inter, pp. 3730–3738, 2015, doi: 10.1109/ICCV.2015.425.

[16] Z. Liu, P. Luo, X. Wang, and X. Tang, "Large-scale celebfaces attributes (celeba) dataset," Retrieved August, 2018.

[17] T. Karras, T. Aila, S. Laine, and J. Lehtinen, "Progressive Growing of GANs for Improved Quality, Stability, and Variation," 6th Int. Conf. Learn. Represent. ICLR 2018 - Conf. Track Proc., Oct. 2017, Accessed: Oct. 05, 2021. [Online]. Available: https://arxiv.org/abs/1710.10196v3.

[18] T. Karras, S. Laine, and T. Aila, "A style-based generator architecture for generative adversarial networks," Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit., vol. 2019-June, pp. 4396–4405, 2019, doi: 10.1109/CVPR.2019.00453.

[19] "akanimax/T2F: T2F: text to face generation using Deep Learning." https://github.com/akanimax/T2F (accessed Dec. 04, 2021).

[20] A. Karnewar and O. Wang, "MSG-GAN: Multi-Scale Gradients for Generative Adversarial Networks," pp. 7796–7805, 2020, doi: 10.1109/cvpr42600.2020.00782.

[21] O. R. Nasir, S. K. Jha, M. S. Grover, Y. Yu, A. Kumar, and R. R. Shah, "Text2FaceGAN: Face generation from fine grained textual descriptions," Proc. - 2019 IEEE 5th Int. Conf. Multimed. Big Data, BigMM 2019, pp. 58–67, 2019, doi: 10.1109/BigMM.2019.00-42.

[22] T. Salimans, I. Goodfellow, W. Zaremba, V. Cheung, A. Radford, and X. Chen, “Improved techniques for training GANs,” Adv. Neural Inf. Process. Syst., pp. 2234–2242, 2016.

[23] X. Chen, L. Qing, X. He, X. Luo, and Y. Xu, “FTGAN: A Fully-trained Generative Adversarial Networks for Text to Face Generation,” no. 1, 2019, [Online]. Available: http://arxiv.org/abs/1904.05729.

[24] M. Z. Khan et al., “A Realistic Image Generation of Face from Text Description using the Fully Trained Generative Adversarial Networks,” IEEE Access, pp. 1–1, 2020, doi: 10.1109/access.2020.3015656.

[25] T. Wang, T. Zhang, and B. Lovell, “Faces \`a la Carte: Text-to-Face Generation via Attribute Disentanglement,” 2020, [Online]. Available: http://arxiv.org/abs/2006.07606.

[26] A. Gatt et al., “Face2Text: Collecting an annotated image description corpus for the generation of rich face descriptions,” Lr. 2018 - 11th Int. Conf. Lang. Resour. Eval., pp. 3323–3328, 2019.

[27] J. Pennington, R. Socher, and C. D. Manning, “GloVe: Global Vectors for Word Representation,” Accessed: Oct. 05, 2021. [Online]. Available: http://nlp.

[28] T. Mikolov, K. Chen, G. Corrado, and J. Dean, “Efficient Estimation of Word Representations in Vector Space,” 1st Int. Conf. Learn. Represent. ICLR 2013 - Work. Track Proc., Jan. 2013, Accessed: Oct. 05, 2021. [Online]. Available: https://arxiv.org/abs/1301.3781v3.

[29] J. Johnson, A. Alahi, and L. Fei-Fei, “Perceptual Losses for Real-Time Style Transfer and Super-Resolution,” Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics), vol. 9906 LNCS, pp. 694–711, Mar. 2016, Accessed: Oct. 07, 2021. [Online]. Available: https://arxiv.org/abs/1603.08155v1.

[30] K. Simonyan and A. Zisserman, “Very deep convolutional networks for large-scale image recognition,” 3rd Int. Conf. Learn. Represent. ICLR 2015 - Conf. Track Proc., pp. 1–14, 2015.

[31] M. Heusel, H. Ramsauer, T. Unterthiner, B. Nessler, and S. Hochreiter, “GANs trained by a two time-scale update rule converge to a local Nash equilibrium,” Adv. Neural Inf. Process. Syst., vol. 2017-Decem, no. Nips, pp. 6627–6638, 2017.

[32] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, “Rethinking the Inception Architecture for Computer Vision,” 2016, doi: 10.1109/CVPR.2016.308.

AUTHORS

Akila Ayanthi is an undergraduate currently pursuing the BCS special degree from the University of Ruhuna, Sri Lanka. Her research interests include computer vision, image processing, deep learning and Generative Adversarial Networks (GANs).

Sarasi Munasinghe received the PhD degree in computer science and engineering from the Queensland University of Technology, Australia in 2018. She completed her BSc. in Engineering from the University of Peradeniya, Sri Lanka in 2010. She currently works at the Department of Computer Science, University of Ruhuna as a senior lecturer. Her research interests are in the fields of deep learning, computer vision and image processing.

PROMINENT DISCORD DISCOVERY WITH MATRIX PROFILE : APPLICATION TO CLIMATE DATA INSIGHTS

Hussein El Khansa, Carmen Gervet and Audrey Brouillet

Espace-Dev, Univ. Montpellier, IRD, Univ. Guyane,
Univ. La Réunion, Montpellier, France

ABSTRACT

The definition and extraction of actionable anomalous discords, i.e. pattern outliers, is a challenging problem in data analysis. It raises the crucial issue of identifying criteria that would render a discord more insightful than another one. In this paper, we propose an approach to address this by introducing the concept of prominent discord. The core idea behind this new concept is to identify dependencies among discords of varying lengths. How can we identify a discord that would be prominent? We propose an ordering relation, that ranks discords and we seek a set of prominent discords with respect to this ordering. Our contributions are 1) a formal definition, ordering relation and methods to derive prominent discords based on Matrix Profile techniques, and 2) their evaluation over large contextual climate data, covering 110 years of monthly data. The approach is generic and its pertinence shown over historical climate data.

KEYWORDS

Prominent discord discovery, Large time series, Matrix profile, Climate data.

1. INTRODUCTION

The analysis of climate data towards the extraction of global climate trends using ensemble mean approaches is receiving a wide interest. To this date, the search for pattern anomaly or outlier patterns has received less attention in climate data studies. Such data is contextual due to the geo-localization and timestamps of the data series relative for instance to soil humidity, temperature or rainfall. It comes from historical sources or complex simulation impact models (physical processes of atmosphere and ocean) such as Earth System Models ([1], [2]). Existing approaches mainly focus on seeking long-term trends, and the study of abnormal behaviour tend to focus on the search of extreme values.

An important element in climate data analysis is the observation window. For instance, a thirty years window has been commonly accepted for climate studies, and is now being reconsidered given the impact it can have on change detection (e.g. [3]). Thus to our knowledge, in climate data analysis, we note 1) a lack of robust outlier pattern detection, and 2) a need to consider very large data sets to minimize the bias induced by the limited observation window. We propose to address those issues in this paper, by introducing a novel concept of pattern outlier, and evaluating our approach to the field of climate data.

David C. Wyld et al. (Eds): FCST, CMIT, SE, SIPM, SAIM, SNLP - 2022
pp. 65-79, 2022. CS & IT - CSCP 2022 DOI: 10.5121/csit.2022.120806

In the realm of data mining, outlier detection is receiving much interest, and has shown its benefits in a wide range of applications including fraud detection [4], cybersecurity [5] and the health sector [6]. Identifying outliers through data sets contributes towards decision support, risk and impact studies. Various definitions of what constitutes an outlier have been proposed, along with associated detection methods. A survey [7] specifies twelve different interpretations of outliers from the perspective of different studies. Overall, an outlier can be commonly defined as "an observation that is significantly dissimilar to other data observations or an observation that does not behave like the expected typical behavior of the other observations" [8]. The observations can specify a single point outlier, or a shape/pattern denoting an abnormal sub-sequence over a time series data. The latter form our outliers are also called discords.

In this article, we investigate discords over contextual time series. Furthermore, we are interested in very large data series to mitigate the potential impact of the length of the data set at hand upon the results. We apply our approach to large data series coming from monthly data between 1902 and 2005. A scalable and exact approach that has proven its computational and space efficiency to detect discords is the Matrix Profile method [9]. It requires the length of the sub-sequence (window) to be set as a parameter. The chosen window has a strong impact to detect meaningful discords. In existing studies, the detection algorithm is run with different window [10], leading potentially to multiple discords. The ranking of such discords is challenging since it requires meaningful criteria to prefer a discord to another. If the data is labelled such ranking is possible since it can be related to an event, but in case of unlabelled data it usually requires expert knowledge.

We propose a novel concept and approach to identify relevant discords over different windows automatically. To do so, we introduce the concept of *prominent discord* that specifies the most significant discord as an anomalous pattern over the longest continuous period, from a shared starting position. A core benefit of the prominent discord is to gain insight onto discords that coincide over their start date, while searching for the ones with longer and subsuming anomalous pattern. For instance if a drought is found through a window of four months, we seek whether it belongs to a longer dry period that may last six months, one year, ten years. By doing so, we search for the longest span of a discord that would subsume other discords and relates to similar occurrences. In contrast to point outliers that are likely to indicate extreme events, resulting from potential long term changes, prominent discords would reveal the anomalous patterns that cause such events.

We formally define the concept of prominent discord, propose a detection algorithm and present an application to large date sets of so called climate impact runoff data. This means that they are historical data relative to surface and subsurface runoff observations, a variable that provides information about flood and drought risks depending on values being high or low.

Figure 1 gives an intuition of what a prominent discord is. We show three discords of respective lengths 13, 37 and 58; all starting at the same position in the series. The reading of the plots corresponds to daily runoff data over five years in millimetres. The Xs covers the daily timestamp while the Ys the height of the runoff in the Sahelian region. Peaks clearly indicate the rainy season. For each window length the prominent discord is highlighted in a continuous line over the time spam. We notice that when the window size changes we have three prominent discords, all starting at the same position date and the one corresponding to the window size of 58 months covers the other two. It is the most prominent one.

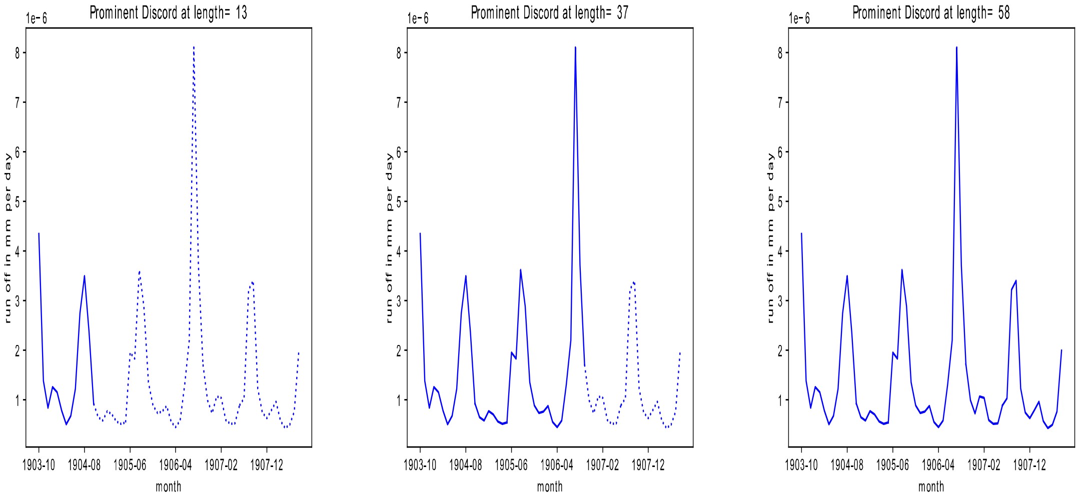


Figure 1. Prominent discord and subsequent discords

The longest one discovered is the most prominent discord: it subsumes the other two. It actually gives insight into lasting changes and anomalous behaviour that could lead towards a change of system state altogether when studying global change and climate data.

The main contributions of this paper are: 1) the novel concept, formalization and method to compute prominent discords and extract the most prominent ones applied to large-scale time series, 2) its application to climate-related data to detect and evaluate the relevance and insights of such discords from a climate point of view.

The paper is organized as follows: section 2 gives a background and related work in the field, section 3 presents our conceptual and methodological contributions, and section 4 its evaluation before concluding in section 5.

2. Background and Related Work

In this section we review previous work relative to time series discord discovery, more specifically with variable length discords and very large time series such as climate models data.

2.1. Climate data analysis

In climate data studies, long-term trends, defined as a tendency towards a climate change pattern, are often characterized by basic statistical measures, such as the average rate of variables increase/decrease over a given time period [11]. In the field of weather extreme events (e.g. droughts, floods, heatwaves, storms), a common approach consists in quantifying how a given climate indicator jumps out, against the background of former climate records (in intensity, frequency or duration; ; [12]). These approaches are conducted under an arbitrary choice of a base time window since a climate reference is inherently defined to assess whether a long-term change and extreme event occurrences can be considered as emergent and/or anomalous. This choice greatly affects the meaning and the robustness of climate studies outcome when this reference is shifted [3].

2.2. Time series discord discovery

Time series discord detection is receiving an increasing interest in data mining since its formalization ([13]–[15]). Efficient and exact methods have been proposed to discover discords in data series [9], [16]. These approaches require some parameter settings including the size of the observation window. The window size is fixed and needs to be specified as an input parameter (HOT SAX [17], QUICK MOTIF [18]). As a result, recent works have drawn on the challenges and insight limitations of a fixed set window size leading to research towards computing all possible discords within a size range, using different methods such as quadratic regression [19], dynamic time warping (DTW) [20], or a graph-based approach [21].

PanMatrix [10] and VALMOD [22] compute variable-length top k discords, using the Matrix Profile method for different window sizes, given a value k. VALMOD considers an interval of possible subsequence lengths as initial parameter; whereas PanMatrix computes exact distances for subsequences of all lengths.

These approaches address the issue of multiple discord computations, but there has been no attempt to order the discords of variable length, and seek those that would have more impact in terms of revealing a potential change of state. Also, the role played by the data series time span has received little attention with respect to its link to discord discovery. Both issues are important for climate model data towards insightful impact studies. A key element is to investigate coincident variable length discords to be able to extract actionable insight through the identification of prominent discords, longest ones sharing a starting position with smallest discords, and thus subsuming them. This is the goal of our approach that can be considered as a meta discord discovery problem over very large data series, with no a priori interpretation of outlier patterns. In other words, we seek *discords for which all the subsequences within a resulting length interval are also discords sharing their starting position*.

We use the Matrix profile approach, because it is an exact method to compute discords for a given window length. It is also computational and space efficiency. Let us now the key notions at hand that will be used to formalize our concept of prominent discord.

2.3. Matrix profile and discords

The matrix profile is a data structure computed to discover discords and motifs using similarity search algorithms [9]. Many algorithms have been proposed with different space and time performance (e.g. STOMP and GPU-STOMP [23], SCRIMP and SCRIMP++ [24]). The data structure builds on the following notions [23], recalled hereafter for further usage and completeness :

Definition 1. *A time series T is a sequence of real-valued numbers t_i: $T = [t_1, t_2, \ldots, t_n]$ where n is the length of T.*

Definition 2. *A subsequence $T_{i,m}$ of a time series T is a continuous subset of values in T, of length m and starting at position i. Formally, $T_{i,m} = [t_i, t_{i+1}, \ldots, t_{i+m-1}]$, where* $1 \le i \le n - m + 1$.

Definition 3 (Distance Profile). *A distance profile $D_{i,m}$ of time series T and length m is a vector of the z-Euclidean distances between a given query subsequence $T_{i,m}$ and all subsequences of length m in the time series T. Formally, $D_{i,m} = \left[d_{i,1}, d_{i,2}, \ldots, d_{i,n-m+1}\right]$, where $d_{i,j} (1 \le i,j \le n - m + 1)$ is the distance between $T_{i,m}$ and $T_{j,m}$ with $i \ne j$.*

Definition 4 (z-Euclidian distance). The z-normalized Euclidean distance $d_{i,j}$ between two subsequences $T_{i,m}$ and $T_{j,m}$ of length *m*, is defined by:

$$d_{i,j} = \sqrt{2m\left(1 - \frac{T_{i,m}.T_{i,m} - m\mu_i\mu_j}{m\sigma_i\sigma_j}\right)} \qquad (1)$$

Where μ_i and σ_i are respectively the mean and standard deviation of $T_{i,m}$, μ_j and σ_j the mean and standard deviation of $T_{j,m}$.

Definition 4 (Matrix Profile). *A matrix profile P_m of time series T and given length m is a meta series of the Euclidean distances vector between each subsequence $T_{i,m}$ of given length m where i varies, and its nearest neighbor (closest match) in time series T , together with the corresponding position vector for each closest neighbor associated with $min(D_{i,m})$. We denote it $P_m = [min(D_{1,m}), \ldots, min(D_{n-m+1,m})]$, where $D_{i,m}(1 \leq i \leq n - m + 1)$ is the distance profile $D_{i,m}$ of time series T for subsequences of length m.$P_m = [min(D_{1,m}),\ldots,min(D_{n-m+1,m})]$, where $D_{i,m}(1 \leq i \leq n-m+1)$ is the distance profile $D_{i,m}$ of time series T for subsequences of length m.*

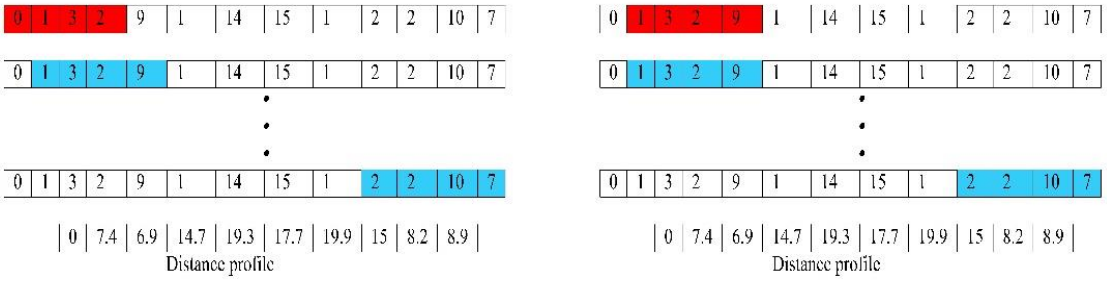
Definition 5 (Index vector). *A matrix profile index vector V_m, associated with a matrix profile P_m denotes the vector of starting position j of the subsequence corresponding to the minimal distance. It is specified by the vector: $V_m = [V_1, V_2, \ldots, V_{n-m+1}]$, such that $V_i = j$ if $d_{i,j} = min(D_{i,m})$.*

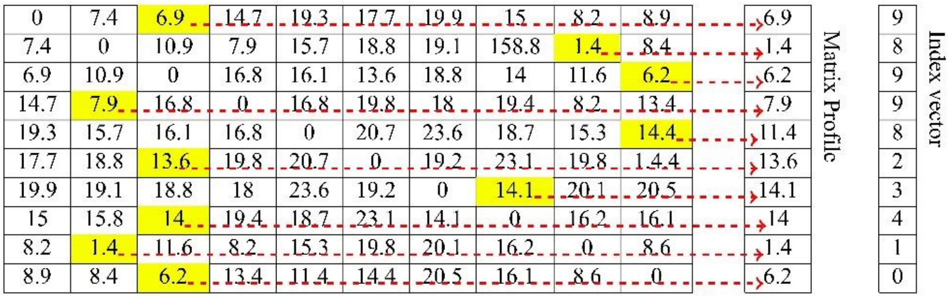
Definition 6 (Discord). *A discord denoted $\Delta_{j,m}$ is a subsequence $T_{j,m}$ of length m starting at the position j in V_m, that corresponds to the maximum distance value in P_m.*

In other words, the discord of length *m*, is the subsequence in the data series (specified by its starting position *j*), such that among the shortest Euclidean distance, it is the one with the maximal value, ie. with largest anomalous pattern among all.

The following example gives the Distance profiles for two susbsequences (in red) of window size 4 over a time series of length 13. The distance profile vector gives the z-Euclidian distance between the chosen subsequence and all the other ones (the next one is highlighted in blue). There are 10 of them, one per subsequence of size 4, sliding over the time series.

The resulting matrix profile extracts for each subsequence the smallest distance in each distance profile (yellow). Finally, we extract the discord from the matrix profile that is the subsequence corresponding to the greatest distance value in the profile. In this example, the biggest distance value is 14.1, distance between subsequence 7 and subsequence 3 ($V_7 = 3$). Thus the discord is the seventh subsequence [14,15,1,2] in the time series.





										Matrix Profile	Index vector
0	7.4	6.9	14.7	19.3	17.7	19.9	15	8.2	8.9	6.9	9
7.4	0	10.9	7.9	15.7	18.8	19.1	158.8	1.4	8.4	1.4	8
6.9	10.9	0	16.8	16.1	13.6	18.8	14	11.6	6.2	6.2	9
14.7	7.9	16.8	0	16.8	19.8	18	19.4	8.2	13.4	7.9	9
19.3	15.7	16.1	16.8	0	20.7	23.6	18.7	15.3	14.4	11.4	8
17.7	18.8	13.6	19.8	20.7	0	19.2	23.1	19.8	14.4	13.6	2
19.9	19.1	18.8	18	23.6	19.2	0	14.1	20.1	20.5	14.1	3
15	15.8	14	19.4	18.7	23.1	14.1	0	16.2	16.1	14	4
8.2	1.4	11.6	8.2	15.3	19.8	20.1	16.2	0	8.6	1.4	1
8.9	8.4	6.2	13.4	11.4	14.4	20.5	16.1	8.6	0	6.2	0

Figure 2. Distance and matrix profile, sequences of length 4

3. Prominent Discords: Definition and Algorithms

In this section we present our approach and contributions, that include the concept of prominent discord and the method we developed to extract a set of prominent discords. The overall problem we address is the following:

Problem addressed: Given a data series T, a large range of possible lengths and a unit step, compute all the variable length discords, and find the discords, such that all of their subsequences within a length interval and shared starting position in T are discords. From these prominent discords, identify a relevant ordering that relates to their length, and number of subsumed discords.

3.1. Definitions

We work at a meta level with respect to the matrix profile and the discord since we compute matrix profiles over sequences of variable lengths, and seek the longest discords that contain discords with a given ordering relation. To formalize our problem, we introduce three concepts: **discord profile**, **discord subsumption ordering** and **prominent discord**. Note that to allow a reliable reasoning over a large number of subsequences, the interval for the variable lengths is set to [4,..,*n*/2].

Definition 7 (Discord profile). *The discord profile ΔP of a time series T is a set of discords $\Delta_{j,l}$ of variable lengths l and starting positions j, such that $\Delta P = \{\Delta_{j,l} \mid j \in 1..n/2 - 1,0 < l \leq n/2\}$.*

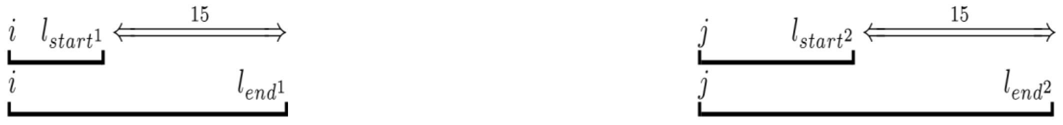
Definition 8 (Discord subsumption ordering). *A discord $\Delta_{j,m}$ subsumes a discord $\Delta_{i,l}$, specified as $\Delta_{i,l} \preccurlyeq_{\Delta} \Delta_{j,m}$ if and only if, $i = j$ and $l < m$..*

Note that this ordering relation is a partial order since we assume that two discords with different starting position in the time series are not comparable. The motivation behind this ordering is that such discords might relate to very different events, whereas discords that co-occur in their starting position are more likely to share the root event for the outlier pattern. With identical start position, two discords can relate to the same anomalous pattern. This is not necessarily true for different starting positions.

Definition 9 (Prominent discord). *A Prominent discord $\bar{\Delta}_{j,l_{start},l_{end}}$ of a time series T is the top discord $\Delta_{j,l_{end}}$ of a lattice of all discords $\Delta_{j,l}$ in ΔP such that $\Delta_{j,l_{start}} \preccurlyeq_{\Delta} \Delta_{j,l} \preccurlyeq_{\Delta} \Delta_{j,l_{end}}$.*

Our approach will compute a set of prominent discords for a given time series. We propose an ordering that accounts for 1) the number of subsumed discords (of different lengths of course), and 2) the relative length of the shortest subsumed discord. The idea behind this ordering is to exploit discords for insight studies on the anomalous patterns that can have a lasting impact, and pertained change of behavior in the time series. Intuitively, a point outlier can be the *consequence* of an existing change of behavior (e.g. rising number of extreme weather events), whereas a discord of droughts of length 4, also found in subsequences of lengths 8 and 15 for example, can indicate a first anomalous pattern, that pertains as an anomalous pattern in longer discords. The longest discord subsuming a much shorter one, can indicate a potential important weather change, and impact on soils, agriculture etc.

The ranking function sorts the prominent discords in decreasing order of $sort((l_{end} - l_{start})/l_{start})$, where the top value corresponds to the longest set of subsumed discords ($l_{end} - l_{start}$), and the lower one the length of the first subsumed discord (l_{start}). As illustrated in Figure 3 the prominent discord A will outrank prominent discord B even though they subsume the same number of discords, because A builds upon a shorter outlier (l_{start}).



Prominent discord A

Prominent discord B

Figure 3. Prominent discord ordering: A is preferred to B with higher ratio function value

3.2. Algorithms

To derive the set of prominent discords, we first derive the discord profile (Algorithm 1) over variable length discords and extract the prominent ones (Algorithm 2) through a counting method based on shared starting positions. Note that Algorithm 1 makes use of an efficient matrix profile computation algorithm (line 5), the **STOMP** algorithm [23] omitted for space reasons. This algorithm, like other matrix profile computation methods (eg, STAMP) derives the z-normalized Euclidean distance to measure efficiently the distance between subsequences.

Algorithm 1 takes as input the whole time series, a maximum subsequence length and list of variable lengths (line 2--3), called windows (from the terminology of the matrix profile approach) that specifies the subsequence lengths considered. For each window size (lines 4--7), we compute the matrix profile, extract the discord $\Delta_{j,l}$ to be stored in the Discord profile list ΔP.

Algorithm 1: Discord Profile: Compute the list of variable length discords

```
  input  : Time Series $T$
  output: Discord Profile $\Delta P$
1 initialization
2 int $m = length(T)/2$
3 list(int) $Windows = [4, 5, 6, 7, 8, \dots, m]$
4 foreach $l$ in $Windows$ do
5     $P_l \leftarrow$ STOMP $(T, l)$ // Matrix profile for window size $l$
6     $\Delta_{j,l} \leftarrow \max(P_l)$ // Discord of size $l$
7     Discord Profile $\Delta P \leftarrow \Delta P$.add($\Delta_{j,l}$)
8 end
9 return $\Delta P$
```

The main algorithm, Algorithm 2, computes the list of prominent discords ΔC and returns the sorted list of prominent discords (in decreasing value of respective $(count/l_{start})$ value. It takes as input the time series and returns the sorted list of prominent discords $\bar{\Delta}_{j,l_{start},l_{end}}$ including its starting position, first discord length and longest one. Line 3 initializes a counter of discords having identical starting positions. In line 4 the discord profile ΔP is computed from Algorithm 1 and contains all the discords Δ_{jl} one per length l considered in Algorithm 1. Line 5 and 6 define the variables used to extract the length l and starting position j of a discord in ΔP. Line 7 extracts the length of the first discord. Lines 9--11 increment the count as long as the next discord has the same starting position as the current one denoted by i. Lines 12--14 create a new prominent discord with starting position j, starting length s, last length $s + count$. It is added to the prominent discord list. Lines 15--16 re-initialize the count, and new starting s position to the one of the next discord in ΔP. Line 19 sorts the prominent discord list in decreasing order according to the proposed function $(l_{end} - l_{start})/l_{start}$; and finally line 20 returns the final sorted list ΔC.

```
Algorithm 2: Sorted list of Prominent Discords
   input  : Time Series $T$,,number of prominent discord $K$
   output: List of sorted Prominent Discords $\Delta C$
 1 initialization
 2 $\Delta C \leftarrow [\ ]$ /List of prominent discords
 3 int $count = 1$ /Increment counting of subsumed discords
 4 $\Delta P \leftarrow$ Discord_Profile($T$)
 5 Var $j$ / variable that extracts the starting position of $\Delta_{jl}$ in $\Delta P$
 6 Var $l$ / variable that extracts the length of $\Delta_{jl}$ in $\Delta P$
 7 int $s = \Delta P[0].l$
 8 for i ← 0 to i ≤ length($\Delta P$)-1 do
 9     if $\Delta P[i].j == \Delta P[i+1].j$ then
10         $count++$
11     end
12     else
13         $\bar{\Delta}_{j,s,s+count} = new\bar{\Delta}(\Delta P[i].j, s, s+count)$
14         $\Delta C \leftarrow \Delta C.add(\bar{\Delta})$
15         $count = 1$
16         $s = \Delta P[i+1].l$
17     end
18 end
19 $\Delta C \leftarrow$ Quicksort($\Delta C$, $(l_{end} - l_{start})/\,l_{start}$)
20 return $\Delta C$ // sorted list of prominent discords
```

Worst case time complexity. Algorithm 2 (calling algorithm 1) overall runs in the worst case in $O(n^3)$ where n is the length of the time series. To decompose, we have: Algorithm 1 calls $n/2$times STOMP, thus runs in the worst case in $O(n^2 \times n) = O(n^3)$. The For loop in Algorithm 2 runs in the length of the discord profile list thus in the worst case $O(n)$ since there is one discord per length ($n/2$ variable length discords), and the list of prominent discords is sorted in the worst case in $O(nlogn)$.

4. Experimental Evaluation and Comparisons

We now present an application of our method to the analysis of large datasets relative to runoff historical climate data. Runoff data correspond to measures of waters in terms of distance (in mm) above the land surface to reach a stream but also to infiltrate the soil surface. All experiments were run on an Intel(R) Xeon(R) Bronze 3106 CPU processor at 1.70GHz with 8 core with 64 GB of RAM. We also compare our proposed approach to other discord discovery methods.

The climate data. We consider observed monthly runoff data, defined in climate data science as an impact variable analyzed to quantify flood and drought risks at regional and global scales (e.g. [12], [25], [26]). These monthly runoff observations are obtained from the Global Runoff Reconstruction dataset (GRUN) that covers the 1902-2014 time period (113 years), with a 0.5° × 0.5° spatial grid resolution [27]. We focus our analysis on the Sahel region, a particularly soil water vulnerable area, and we spatially average monthly runoff over the corresponding grid box [5°W-25°E ; 10°N-18°N]. Our prominent discord approach is then applied to the obtained Sahel time series between 1902 and 2005 (i.e. 104 years, 1248 months), a period commonly considered in historical climate analysis.

4.1. Prominent discords discovery

For the dataset of monthly runoff observation data over 1248 months, we applied our approach and derived the list of all prominent discords, including their ranking based on our proposed ratio function, after calculating the discords for all variable length windows in the interval [4,..,1248⁄2], with a monthly step increment. The set of prominent discords was derived in 3.5 minutes.

Figure 4 shows for five prominent discords, including the most prominent one, their respective subsumed discords. The X-axis indicates window lengths up to 130 months, and the Y-axis the discords starting date. Each blue dot represents discord with its window length (Xs) and its starting date (Ys). The arrows show the prominent discords with all the subsumed discords of coinciding starting dates. The length of the arrows illustrates how many discords the prominent one subsumes. Here we have four prominent discords. The most prominent discord starts at the position date 1903-10-15, with a lower window size of 13 months for its first subsumed discord, and upper length of 58 months.

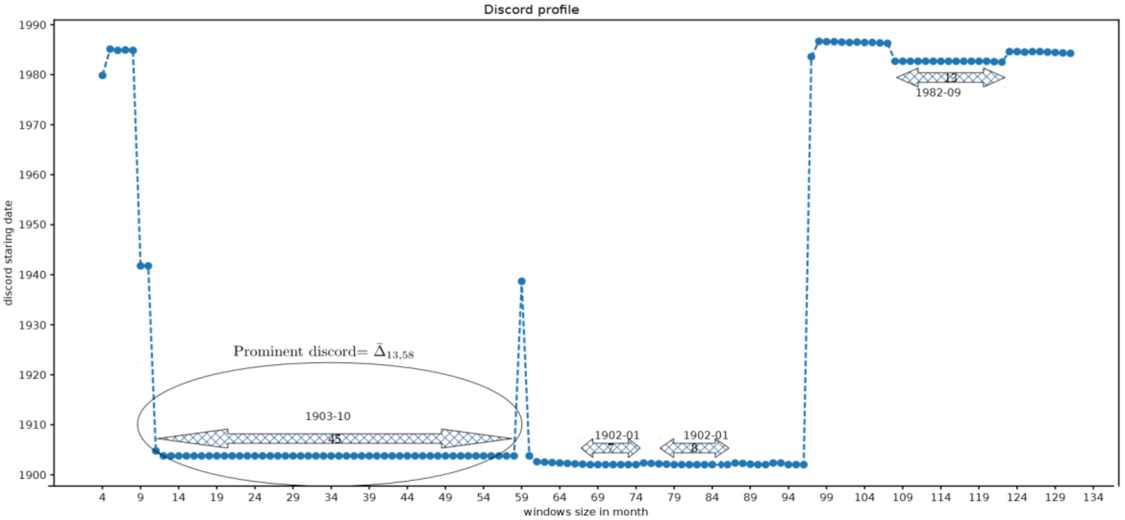


Fig.4. Prominent discords derived from observed runoff monthly data.

Table 1 shows the top five prominent discords in ΔC according to our proposed ratio ordering. The first and second ones are found in Figure 4.

Table 1. Top 5 prominent discords sorted by the ratio function

date	starting window size	ending window size	ratio
1903-10-15	13	58	3.46
1982-09-15	109	120	0.10
1902-09-15	242	264	0.09
1903-09-15	193	209	0.08
1902-01-15	80	86	0.075

4.2. Comparison with alternative approaches

We compared our approach with existing discord discovery methods, illustrating mainly the importance of considering an exact approach for the prominent discord extraction, and the need of variable length discord computation without parameterized length settings.

We considered HOT-SAX, an extension of the SAX algorithm [28]. SAX discretizes time series into words and detects motifs in time series but not discords. HOT-SAX algorithm was developed to detect discords. It builds a suffix tree that stores the words generated by SAX. A word with the least number of occurrences is a discord. A requirement is that the number of extracted discords is predefined.

Rare Rule Anomaly (RRA) [29] is an algorithm that uses grammar-based compression able to detect motifs and discords in time series. Similar to HOT-SAX, RRA uses SAX algorithm to discretize the time series. A grammatical induction algorithm (ex: Sequitur [30]) is used to generate the grammar. These generated grammars are used to detect the discords.

One of the main parameters for HOT-SAX and RRA is the window size. To be able to compare them with our approach, we use the value l_{end} of each prominent discord as the window parameter.

Table 2. HOT SAX results using windows extracted from the l_{end} of each prominent discord

HOT SAX		
window size	start	end
58	1902-01-01	1906-11-01
120	1943-08-01	1953-08-01
264	1972-02-01	1994-02-01
209	1973-06-01	1990-11-01
86	1902-07-01	1909-09-01

Table 3. RRA results using windows extracted from the l_{end} of each prominent discord

RRA		
window size	start	end
58	1902-02-01	1907-04-01
120	1944-08-01	1954-11-01
264	1953-01-01	1977-08-01
209	1953-12-01	1971-12-01
86	1943-04-01	1951-01-01

We can see that both HOT SAX and RRA lead to different results in terms of starting and end date of the prominent discord for a given window size. This comes from the fact that they are not exact methods and given an input window length, lead to a different discord.

We also compared with the PAN MATRIX algorithm. It calculates variable length discords, using the SKIMP algorithm to compute the matrix profile for all motif lengths.

Table 4 shows the top five discords with the PAN MATRIX. Compared with our results in Table 1, the top 5 discords are different, and the top 5 discord of PAN MATRIX are not relative to subsequences since they are not based on an ordering among variable length discords. PAN MATRIX is efficient to calculate variable length discord, but is not designed to order discords.

Table 4. PAN MATRIX top 5 discords

Pan Matrix	
Date	Window size
1952-07-01	619
1952-07-01	618
1952-07-01	617
1952-07-01	616
1952-07-01	615

4.3. Analysis and insights for climate data analysis

To analyze those results, as well as the relevance and insights of our prominent discord and proposed orderings, we compared with statistical analysis of those historical climate data (e.g. long-term trends, seasonal cycles, standard-deviation). It is worth noting that the prominent discord approach is unsupervised, and does not consider any climate behaviour nor known physical processes.

Figure 6 shows the annual average runoff data over the Sahel region illustrating the maximum, minimum and average annual values, providing a general idea of the global fluctuations and extremes. Figure 7 relates the monthly runoff values together with the top five prominent discords we discovered, cf Figure 5. For each color, the dotted lines indicate the starting position of a prominent discord and the vertical lines the respective l_{start} and l_{end}.

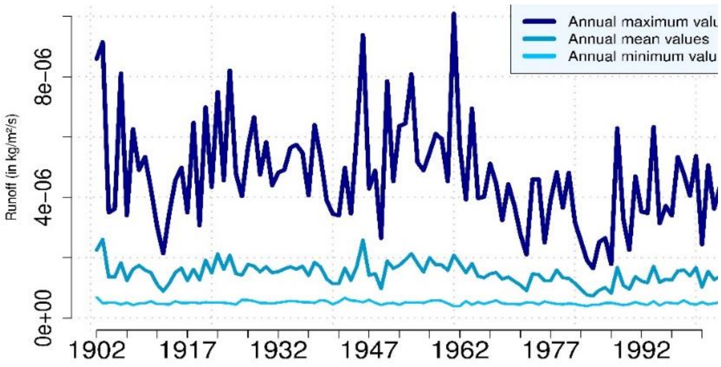


Figure 6. Annual mean runoff values

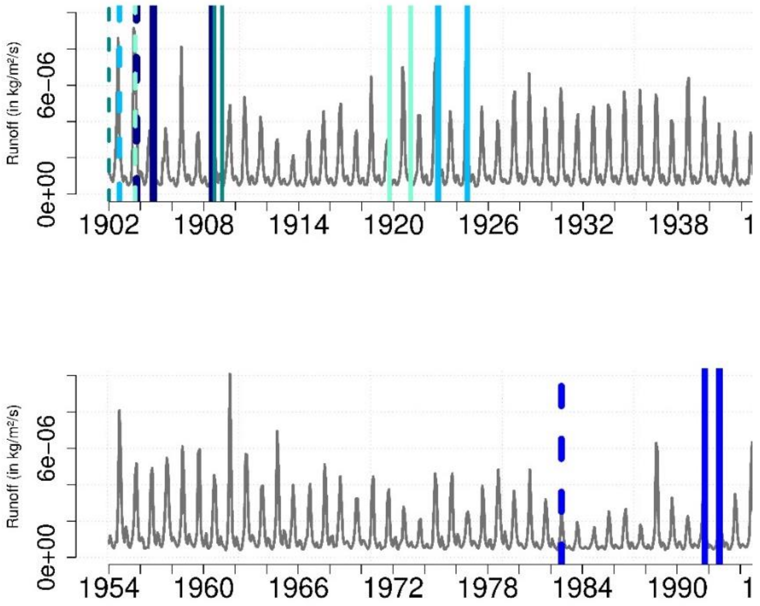


Figure 7. Monthly runoff with prominent discords between 1900 and 2005

According to climate studies, a well known soil drying trend mostly resulting from a rainfall decrease is observed between 1900 and 2013 in the Sahel ((e.g. [27])). We find consistent trends using runoff data, particularly in the annual maximum runoff time evolution between 1902 and 2005 ([26]). In our work, four of the five first prominent discords coincide with starting dates and length intervals during the first 22 years of the time series (Table 5 and Figure 7). The corresponding averaged monthly runoff over 1902-1924 also shows a mean of $15e^{-07}$ kg/m^2/s and a standard-deviation (i.e. an inter-annual variability indicator; ([31]) of 0.14 kg/m^2/s, whereas the entire 1902-2005 time series is characterized by a mean of $12e^{-07}$ kg/m^2/s and a standard deviation of 0.12 kg/m^2/s. These four prominent discords may thus illustrate the specific time pattern between 1902 and 1924 with higher soil water amount and larger inter-annual variability, before the upcoming continuous long-term drying trend observed within Sahel.

The second prominent discord is detected for a starting date at 1982-09-15 and window sizes in [109 ; 120 months]. This period corresponds to the time period with the smallest mean runoff values of the entire time series (Fig. 6 and Fig. 7). The associated 1982-1992 mean runoff is $9e^{-07}$ kg/m^2/s compared to a mean runoff of $12e^{-07}$ kg/m^2/s over 1900-2005. This prominent discord illustrates the temporal pattern resulting from intense droughts that occurred in Sahel in the 1980s. During that decade, the most severe drought ever recorded over the African continent occurred during 1983-1984 (Figure 2 in [32]).

In this study, we demonstrated two major usefulness of our proposed prominent discord approach and ordering relations, in climate data analysis in terms of real insights towards the emergence of a long-term change, and the detection of recurring anomalous events.

First, we showed that the prominent discord concept does capture a 20-30 years pattern illustrating a different (former) climate regime compared to the rest of the considered time series (*emergence*). Second, we showed that the subsumption ordering and prominent discords ranking capture the time pattern at a decade scale of the driest recorded yearly event (*recurring anomaly detection*).

5. Conclusion and Future Work

In this paper we proposed a new concept in the realm of variable length discords, the prominent discord. It focuses on identifying anomalous patterns that last through time and subsume sets of discords. The main contributions are the formalization and method to compute prominent discords and extract the most prominent ones applied to large scale time series, and the application to climate-related data.

Our ordering and results show their relevance in the field of climate data series. In particular, they show that through such ordering we gain insights on the anomalous patterns that have a lasting impact, and pertained change of behavior in the time series. This is new to our knowledge. Future works include further experimental studies on different impact climate data and other data sets, to evaluate the thematic insights of our approach, as well as some optimization of the algorithm to ensure scalability.

Acknowledgements

The authors would like to thank the Occitanie Region, who partially funded this research, and the readers of previous versions for the tremendous guidance.

References

[1] Randall, D. A.; Wood, R. A.; Bony, S.; Colman, R.; Fichefet, T.; Fyfe, J.; Kattsov, V.; Pitman, A.; Shukla, J.; Srinivasan, J.; others. (2007). Climate models and their evaluation, *Climate Change 2007: The Physical Science Basis. Contribution of Working Group I to the Fourth Assessment Report of the IPCC (FAR)*, Cambridge University Press, 589–662

[2] Knutti, R.; Masson, D.; Gettelman, A. (2013). Climate model genealogy: Generation CMIP5 and how we got there: CLIMATE MODEL GENEALOGY, *Geophysical Research Letters*, Vol. 40, No. 6, 1194–1199. doi:10.1002/grl.50256

[3] Sippel, S.; Zscheischler, J.; Heimann, M.; Otto, F. E. L.; Peters, J.; Mahecha, M. D. (2015). Quantifying changes in climate variability and extremes: Pitfalls and their overcoming, *Geophysical Research Letters*, Vol. 42, No. 22, 9990–9998. doi:10.1002/2015GL066307

[4] Hilal, W.; Gadsden, S. A.; Yawney, J. (2022). Financial Fraud: A Review of Anomaly Detection Techniques and Recent Advances, *Expert Systems with Applications*, Vol. 193, 116429. doi:10.1016/J.ESWA.2021.116429

[5] Hopkins, S.; Kalaimannan, E.; John, C. S. (2020). Sub-Erroneous Outlier Detection of Cyber Attacks in a Smart Grid State Estimation System, *2020 11th IEEE Annual Ubiquitous Computing, Electronics \& Mobile Communication Conference (UEMCON)*, 447–454

[6] Xia, H.; An, W.; Li, J.; Zhang, Z. J. (2020). Outlier knowledge management for extreme public health events: understanding public opinions about COVID-19 based on microblog data, *Socio-Economic Planning Sciences*, 100941

[7] Ayadi, A.; Ghorbel, O.; Obeid, A. M.; Abid, M. (2017). Outlier detection approaches for wireless sensor networks: A survey, *Computer Networks*, Vol. 129, 319–333

[8] Hodge, V.; Austin, J. (2004). A survey of outlier detection methodologies, *Artificial Intelligence Review*, Vol. 22, No. 2, 85–126

[9] Yeh, C.-C. M.; Zhu, Y.; Ulanova, L.; Begum, N.; Ding, Y.; Dau, H. A.; Silva, D. F.; Mueen, A.; Keogh, E. (2016). Matrix profile I: all pairs similarity joins for time series: a unifying view that includes motifs, discords and shapelets, *2016 IEEE 16th International Conference on Data Mining (ICDM)*, 1317–1322

[10] Madrid, F.; Imani, S.; Mercer, R.; Zimmerman, Z.; Shakibay, N.; Keogh, E. (2019). Matrix profile xx: Finding and visualizing time series motifs of all lengths using the matrix profile, *2019 IEEE International Conference on Big Knowledge (ICBK)*, 175–182

[11] Hartmann, D. L.; Klein Tank, A. M. G.; Rusticucci, M.; Alexander, L. V; Brö\"nnimann, S.; Charabi, Y.; Dentener, F. J.; Dlugokencky, E. J.; Easterling, D. R.; Kaplan, A.; Soden, B. J.; Thorne, P. W.;

Wild, M.; Zhai, P. M. (2013). Observations: Atmosphere and Surface, T. F. Stocker; D. Qin; G.-K. Plattner; M. Tignor; S. K. Allen; J. Boschung; A. Nauels; Y. Xia; V. Bex; P. M. Midgley (Eds.), *Climate Change 2013: The Physical Science Basis. Contribution of Working Group I to the Fifth Assessment Report of the Intergovernmental Panel on Climate Change*, Cambridge University Press, Cambridge, United Kingdom and New York, NY, USA, 159–254. doi:10.1017/CBO9781107415324.008

[12] Sillmann, J.; Kharin, V. V; Zwiers, F. W.; Zhang, X.; Bronaugh, D. (2013). Climate extremes indices in the CMIP5 multimodel ensemble: Part 2. Future climate projections: CMIP5 PROJECTIONS OF EXTREMES INDICES, *Journal of Geophysical Research: Atmospheres*, Vol. 118, No. 6, 2473–2493. doi:10.1002/jgrd.50188

[13] Fu, A. W.-C.; Leung, O. T.-W.; Keogh, E.; Lin, J. (2006). Finding time series discords based on haar transform, *International Conference on Advanced Data Mining and Applications*, 31–41

[14] Chiu, B.; Keogh, E.; Lonardi, S. (2003). Probabilistic discovery of time series motifs, *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 493–498. doi:10.1145/956750.956808

[15] Kleijnen, J. P. C. (2008). Response surface methodology for constrained simulation optimization: An overview, *Simulation Modelling Practice and Theory*, Vol. 16, No. 1, 50–64. doi:10.1016/j.simpat.2007.10.001

[16] Yankov, D.; Keogh, E.; Rebbapragada, U. (2008). Disk aware discord discovery: Finding unusual time series in terabyte sized datasets, *Knowledge and Information Systems*, Vol. 17, No. 2, 241–262

[17] Keogh, E.; Lin, J.; Fu, A. (2005). Hot sax: Efficiently finding the most unusual time series subsequence, *Fifth IEEE International Conference on Data Mining (ICDM'05)*, 8--pp

[18] Li, Y.; Leong, H. U.; Yiu, M. L.; Gong, Z. (2015). Quick-motif: An efficient and scalable framework for exact motif discovery, *2015 IEEE 31st International Conference on Data Engineering*, 579–590

[19] Leng, M.; Chen, X.; Li, L. (2008). Variable length methods for detecting anomaly patterns in time series, *2008 International Symposium on Computational Intelligence and Design* (Vol. 2), 52–56

[20] Vy, N. D. K.; Anh, D. T. (2016). Detecting variable length anomaly patterns in time series data, *International Conference on Data Mining and Big Data*, 279–287

[21] Boniol, P.; Palpanas, T. (2020). Series2graph: Graph-based subsequence anomaly detection for time series, *Proceedings of the VLDB Endowment*, Vol. 13, No. 12, 1821–1834

[22] Linardi, M.; Zhu, Y.; Palpanas, T.; Keogh, E. (2020). Matrix profile goes MAD: variable-length motif and discord discovery in data series, *Data Mining and Knowledge Discovery*, Vol. 34, No. 4, 1022–1071

[23] Zhu, Y.; Zimmerman, Z.; Senobari, N. S.; Yeh, C.-C. M.; Funning, G.; Mueen, A.; Brisk, P.; Keogh, E. (2016). Matrix profile ii: Exploiting a novel algorithm and gpus to break the one hundred million barrier for time series motifs and joins, *2016 IEEE 16th International Conference on Data Mining (ICDM)*, 739–748

[24] Zhu, Y.; Yeh, C.-C. M.; Zimmerman, Z.; Kamgar, K.; Keogh, E. (2018). Matrix profile XI: SCRIMP++: time series motif discovery at interactive speeds, *2018 IEEE International Conference on Data Mining (ICDM)*, 837–846

[25] Arnell, N. W.; Lloyd-Hughes, B. (2014). The global-scale impacts of climate change on water resources and flooding under new climate and socio-economic scenarios, *Climatic Change*, Vol. 122, Nos. 1–2, 127–140. doi:10.1007/s10584-013-0948-4

[26] Gosling, S. N.; Zaherpour, J.; Mount, N. J.; Hattermann, F. F.; Dankers, R.; Arheimer, B.; Breuer, L.; Ding, J.; Haddeland, I.; Kumar, R.; Kundu, D.; Liu, J.; van Griensven, A.; Veldkamp, T. I. E.; Vetter, T.; Wang, X.; Zhang, X. (2017). A comparison of changes in river runoff from multiple global and catchment-scale hydrological models under global warming scenarios of 1 °C, 2 °C and 3 °C, *Climatic Change*, Vol. 141, No. 3, 577–595. doi:10.1007/s10584-016-1773-3

[27] Ghiggi, G.; Humphrey, V.; Seneviratne, S. I.; Gudmundsson, L. (2019). GRUN: an observation-based global gridded runoff dataset from 1902 to 2014, *Earth System Science Data*, Vol. 11, No. 4, 1655–1674. doi:10.5194/essd-11-1655-2019

[28] Lin, J.; Keogh, E.; Lonardi, S.; Chiu, B. (2003). A symbolic representation of time series, with implications for streaming algorithms, *Proceedings of the 8th ACM SIGMOD Workshop on Research Issues in Data Mining and Knowledge Discovery, DMKD '03*, Vol. 13, 2–11. doi:10.1145/882082.882086

[29] Senin, P.; Lin, J.; Wang, X.; Oates, T.; Gandhi, S.; Boedihardjo, A. P.; Chen, C.; Frankenstein, S. (n.d.). Time series anomaly discovery with grammar-based compression., *Researchgate.Net*

[30] Nevill-Manning, C.; Research, I. W.-J. of A. I.; 1997, undefined. (1997). Identifying hierarchical structure in sequences: A linear-time algorithm, *Jair.Org*, Vol. 7, 67–82

[31] Hansen, J.; Sato, M.; Ruedy, R. (2012). Perception of climate change, *Proceedings of the National Academy of Sciences*, Vol. 109, No. 37, E2415--E2423. doi:10.1073/pnas.1205276109

[32] Masih, I.; Maskey, S.; Mussá, F. E. F.; Trambauer, P. (2014). A review of droughts on the African continent: a geospatial and long-term perspective, *Hydrology and Earth System Sciences*, Vol. 18, No. 9, 3635–3649. doi:10.5194/hess-18-3635-2014

AUTHORS

Hussein Al Khansa is a computer science PhD student from the university of Montpellier, within the Espace Dev laboratory. He carried out his Masters degree at the Lebanese International University, in Lebanon.

Carmen Gervet is professor of computer science at the university of Montpellier and director of the Espace-Dev research unit. She is specialized in Artificial Intelligence and more specifically decision support systems and constraint-based reasoning.

Audrey Brouillet is a post-doctoral researcher in the laboratory Espace-Dev. She obtained her PhD in climatology from the university Paris Saclay in 2020. She specializes in climate and impact data analysis, in the context of climate change.

AN INFORMATIONAL SPACE BASED SEMANTIC ANALYSIS FOR SCIENTIFIC TEXTS

Neslihan Suzen, Alexander N. Gorban,
Jeremy Levesley and Evgeny M. Mirkes

[1]School of Computing and Mathematical Sciences,
University of Leicester, Leicester, UK

ABSTRACT

One major problem in Natural Language Processing is the automatic analysis and representation of human language. Human language is ambiguous and deeper understanding of semantics and creating human-to-machine interaction have required an effort in creating the schemes for act of communication and building common-sense knowledge bases for the 'meaning' in texts. This paper introduces computational methods for semantic analysis and the quantifying the meaning of short scientific texts. Computational methods extracting semantic feature are used to analyse the relations between texts of messages and 'representations of situations' for a newly created large collection of scientific texts, Leicester Scientific Corpus. The representation of scientific-specific meaning is standardised by replacing the situation representations, rather than psychological properties, with the vectors of some attributes: a list of scientific subject categories that the text belongs to. First, this paper introduces 'Meaning Space' in which the informational representation of the meaning is extracted from the occurrence of the word in texts across the scientific categories, i.e., the meaning of a word is represented by a vector of Relative Information Gain about the subject categories. Then, the meaning space is statistically analysed for Leicester Scientific Dictionary-Core and we investigate 'Principal Components of the Meaning' to describe the adequate dimensions of the meaning. The research in this paper conducts the base for the geometric representation of the meaning of texts.

KEYWORDS

Natural Language Processing, Information Extraction, Scientific Corpus, Scientific Dictionary, Quantification of Meaning, Word Representation, Text Representation, Dimension Extraction, Dimensionally Reduction, Principal Component Analysis, Meaning Space.

1. INTRODUCTION

One major problem in Natural Language Processing is the automatic analysis and representation of human language. Computational methods attempt to repeat human behaviour in the processing natural languages in a world where humans have no limitations on the range of interpretation of words, and the construction of complex meaning (semantic binding). Unlike humans as a group, machines may fail to provide a rich enough set of contexts to represent and distinguish different concepts.

The 'meaning of meaning' is a topic that has been extensively discussed by philosophers, linguistics, psychologists, neuroscientists, and computer scientists, in order to build "common-sense" knowledge bases, but the consensus has yet to be reached [1-4]. Wittgenstein formulates this as follows: "Meaning is use" or, in more detail, "For a large class of cases though not for all

David C. Wyld et al. (Eds): FCST, CMIT, SE, SIPM, SAIM, SNLP - 2022
pp. 81-99, 2022. CS & IT - CSCP 2022 DOI: 10.5121/csit.2022.120807

in which we employ the word ‘meaning’ it can be defined thus: the meaning of a word is its use in the language” [5, §43].

For the world of scientific texts (abstracts or brief reports), there is a well-defined dominant communicative function: a representative function. In an idealised scheme of the act of communication (see Figure 1), two representations of the situation on the “blackboards of consciousness” exist: the sender’s representation (Representation 1) of the situation (Situation 1) and the receiver’s representation (Representation 2) of the situation (Situation 2). A text related to the first situation is generated by the sender (Translation 1). This text is transmitted to the receiver and transformed by the receiver into a representation of the situation (Translation 2). The sender’s and the receiver’s representations never coincide.

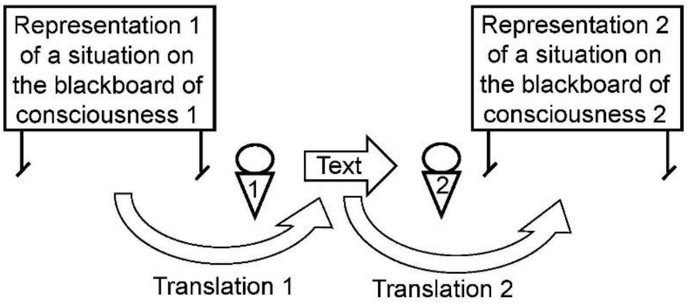

Representation 1 of a situation on the blackboard of consciousness 1
Representation 2 of a situation on the blackboard of consciousness 2
1
Text
2
Translation 1
Translation 2

Figure 1. The idealised scheme of the act of communication. There is a representation of a situation on the sender’s “blackboard of the consciousness” (a representation 1 of Situation 1). A text related to this situation is generated by the sender (Translation 1). This text is transmitted to the receiver and transformed by her into a representation of the situation (Representation 2 of Situation 2).

In this study, we consider the chain: Representation 1 → Text → Representation 2, and translations between them. Translations depend on a broad range of factors related to communication, including the experience of the sender and receiver. It is noteworthy that there may be many receivers and senders. One-to-many or even many-to-many communication add more situations and representations.

A very basic scheme is sufficient for our analysis of meaning. Meaning is hidden in the relationship between the representation of situations on the blackboard of the consciousness and the texts of the messages. The meaning of meaning can be understood if and only if the translation operations are created in the scheme of a communication act. Moreover, understanding can be represented as a reflexive game [6] with different levels (The sender prepares a message taking into account the experience of the receiver, his goals and tools, and guesses that the receiver takes into account the experience of the sender, her goals and tools, and... . Analogously, the receiver tries to understand the message taking into account..., etc.)

The relation between the text and the representation of the situation is a many-to-many correspondence. Each text corresponds to many situations and each situation can have many representative texts. At this stage, we characterise a situation “behind the text” by a set of attributes.

Despite the challenges in creating and describing plausible translation, with recent remarkable progress of machine translation, applying modern machine learning tools seems to be attractive idea for the analysis and simulation of translation operations. However, there is no generally accepted tools for working directly with representations of situations, and we cannot propose a general solution to this problem. Such a solution, perhaps, is impossible in a finite closed form despite much effort over many decades.

Our goal is more modest. We will provide computational analysis of the relations between texts of messages and representations of situations for a large collection of brief scientific texts. Such representations must be standardised, at least in part, and expressed in the form of diagrams, specially organized texts, or by other means. The simplest and universal approach is to replace the situation representations with vectors of attributes. Sentiment analysis provides many examples of such representations. We aim to provide another basic example that is specific to scientific texts: a list of scientific subject categories that the text belongs to.

In any text classification, subject categories can be chosen by humans or a computer system with an understanding of the text, but conflicts of understanding are possible and maybe inevitable. Even famous preprint servers (such as arXiv), moderators sometimes change the category selected by the authors. This is because the content of the text may differ from its meaning [7], a confusion which often occurs (just as understanding the situation behind the text is often confused with recognising the content of the text).

In our analysis of meaning, the starting point is the combination of the text with the list of the subject categories the text belongs to – definition of the attributes of the situation behind the text. The key idea of this approach goes back to the lexical approach of Sir Francis Galton, who selected the personality-descriptive terms and stated the problem of their interrelations for real persons. Following his idea, in classical psycholinguistic studies, a similar approach was used in publications [8-10]. Osgood, with co-workers, in the theory of the Semantic Differential, hypothesised a 3-dimensional semantic space to quantify connotative meanings concerning psychological and behavioural parameters [11, 12]. They used an approach for the extraction of three 'coordinates of meaning' from the evaluation of the 'affective meaning' of words (objects) by people. The semantic space was built by, in his words, 'three orthogonal bipolar dimensions': Evaluation (E), Potency (P) and Activity (A). Of course, the research started considering many different attributes and these three were extracted by factor analysis. These evaluations of a single object were related to some situations involving this single object, not just to an isolated abstract object. The people evaluated not the abstract 'terms' but psychologically meaningful situations behind these terms; these situations were the sources of 'affective meaning'.

For our world of scientific texts, we characterise the situation of use by a scientifically specific description – the research subject categories of the text. Quantifying the meaning in our research follows the road: Corpus of texts + categories → Meaning Space (MS) for words + Geometric representation of the meaning of texts.

In our analysis of meanings, the starting point is to combine the text with the list of the subject categories the text belongs to. These categories can intersect: a text can belong to several categories as texts can be assigned to more than one category. The categories evaluate the situation (the research area) related to the text as a whole, not as a result of the combination of the meaning of words. This holistic approach defines the general meaning of a word in short scientific texts as the information that the use of this word in texts carries about the categories to which these texts belong. More explicitly, we quantify meaning by using the Relative Information Gain (RIG) (see Equation 7) for a word in a category. To do this we require two attributes of text d for a given word w_j and a given category c_k, defined as:

$c_k(d)$: The text d is in the category c_k: Attribute values are Yes ($c_k(d)$ = 1) or No ($c_k(d)$ = 0);
$w_j(d)$: The word is in the text: Attribute values are Yes ($w_j(d)$ = 1) or No ($w_j(d)$ = 0).

In this approach, the corpus of scientific texts is a probabilistic sample space (the space of equally probable elementary results, each of which is a random selection of text from the corpus). $RIG(c_k, w_j)$ measures the (normalized) information about the value of $c_k(d)$, which can be extracted from the value $w_j(d)$ (i.e. from observing or not observing the word w_j in the text d) for a text d from the corpus. By this, we identify the importance of the word for the corresponding category in terms of information gained when separating the corresponding category from its complement.

To follow our road, a triad is needed: texts, dictionary and multidimensional evaluation of the situation of use presented by the categories. In this research, short scientific texts are abstracts of research articles or proceeding papers. For the first element of the triad, the whole world of abstracts is narrowed to a sample: 1,673,350 texts from the *Leicester Scientific Corpus (LSC)* [13]. The meaning of a word extracted from the corpus is represented by a 252-dimensional vector of RIGs, in which each of the texts in the LSC is assigned to at least one of these 252 Web of Science (WoS) categories [14]. Thus, we use these simple 252 binary attributes for multidimensional evaluation of the text usage situation, where the second element of the triad is the *Leicester Scientific Dictionary-Core (LScDC)* [15].

Next, a vector space to represent a word's meanings has been introduced: the *Meaning Space*. In the Meaning Space, coordinates correspond to the subject categories. Each word w_j in the dictionary is represented by the vector $\overrightarrow{RIG_j}$, of information gains for the word for each of the subject categories. These vectors are estimations of the meaning of words as to their importance in each of the research fields. hypotheses here are: if words have similar vectors, they tend to have similar meanings, and if texts have a similar distributions of word meanings – similar clouds of word vectors – then they tend to have similar meanings (often referred to as the Distributional Semantic Hypothesis). We demonstrate that RIG-based word ranking is much more useful than ranking based on raw word frequency in determining the science-specific meaning and importance of a word. The proposed model based on RIG is shown to have ability to stand out topic-specific words in subject categories.

Having represented each word in the Meaning Space, these representations can be used in many text analysis problems including the creation of a thesaurus such as the *Leicester Scientific Thesaurus (LScT)* [16]. The LScT contains the most informative 5,000 words in science; in formativeness is measured as the average RIGs of a word across categories.

This representation scheme is the basis of the computational analysis of the meaning of texts and will be used later for our holistic approach to the meaning of text: the text is considered as a collection of words, the meaning of the text is hidden in a situation of use, which is evaluated as a whole.

In this study, the hypothesis that lexical meaning in science can be represented in a lower dimensional space rather than the 252-dimensional Meaning Space is tested. Principal Component Analysis (PCA) is performed to reduce the dimensionality of the Meaning Space, in which points are the 5,000 words of LScT and dimensions are categories. We analyse the dimension of the Meaning Space and visualise words and categories in the space of principle components (PCs). We interpret the first five PCs by their coordinates. For each component, categories are divided into three groups: categories that positively and negatively correlated with the corresponding component, and categories having near zero values in the component. Topics in these groups are analysed. We then analyse the extreme topic groups at opposite ends of the PCs in order to describe the PCs. Finally, different selection criteria (Kaiser, Broken Stick, an empirical method based on multicollinearity control – PCA-CN) are used to reduce the dimensionality of the category space to 61, 16 and 13, respectively.

2. Dataset

Our new approach is applied to construct the Meaning Space on the basis of Leicester Scientific Corpus (LSC) and Leicester Scientific Dictionary-Core (LScDC) [13, 14]. The LSC is a scientific corpus of 1,673,350 abstracts and the LScDC is a scientific dictionary of 103,998 words extracted from the LSC. Each text in the LSC belongs to at least one of the 252 subject categories of Web of Science (WoS). Words in the LScDC will be represented by 252-dimensional vector in the Meaning Space.

Finally, a thesaurus of science is created by selecting the most informative words from the LScDC. The informativeness here was measured by the average RIGs in categories. We introduced the *Leicester Scientific Thesaurus (LScT)* where the most informative 5,000 words from the LScDC were included in [16]. These words are considered as the most meaningful words in science. Later we will use the LScT in the study of the representation of the meaning of texts.

3. An informational Space of Meaning

In this section, we introduce our novel vector space model developed for quantifying the meaning of words. The architecture of the approach to estimating the word meaning for a large family of natural language scientific texts has discussed. The new approach to word meaning is applied to construct the Meaning Space based on the LSC and LScDC.

We introduce the *Meaning Space*, in which the meaning of a word is represented by a vector of RIGs about the subject categories that the text belongs to. We hypothesize that words have scientifically specific meaning in categories and the meaning can be estimated by information gains from the word to the category. 252 subject categories of WoS are used in construction of vectors of information gains. This representation technique is evaluated by analysing the top-ranked words in each category. For individual categories, RIG-based word ranking is compared with ranking based on raw word frequency in determining the science-specific meaning and importance of a word.

We finally create a scientific thesaurus, LScT, in which the most informative words are selected from the LScDC by their average RIGs in categories. LScT contains the most informative 5,000 words in the corpus LSC. These words are considered as the most meaningful words in science.

3.1. Word Meaning as a Vector of RIGs Extracted for Categories

We start with measuring how informative a word is for a category in terms of its ability to separate the corresponding category from its set theoretical complement. We hypothesize that topic-specific words in categories have larger information gain than other words, and such words are expected to have less gain in most other categories. Therefore, we approach this problem by defining, for each subject category c_k, a random Boolean variable: the text belongs to the category c_k or the text does not belong to the category c_k (this class is denoted as $\overline{c}_k$). The frequencies of words in classes c_k and $\overline{c}_k$ are shown in Table 1. Let D^j denote the set of texts containing the word w_j and D_k be the set of texts in the category c_k.

For every word w_j from the dictionary (j = 1, ..., N) and every text d_i from the corpus (i = 1, ..., M) the indicator $w_j(d_i)$ is defined as follows: If the word w_j occurs in the text d_i (once or more), then $w_j(d_i)$ = 1, otherwise, $w_j(d_i)$ = 0. The frequency of the word w_j in the category c_k is

$$w_{jk} = \sum_{d_i \in D_k} w_j(d_i); \tag{1}$$

$w_{jk}$is the number of texts containing the word w_j in the category c_k.

Table 1. Representation of the word by a pair of frequencies: the number of texts containing the word $\boldsymbol{w_j}$ that belong and does not belong to the category $\boldsymbol{c_k}$

Category Word	c_k	$\bar{c}_k$
w_1	w_{1k}	$\lvert D^1 \rvert - w_{1k}$
w_2	w_{2k}	$\lvert D^2 \rvert - w_{2k}$
⋮	⋮	⋮
w_N	w_{Nk}	$\lvert D^N \rvert - w_{Nk}$

Since words are obviously not mutually exclusive (one text usually contains several different words), to evaluate the information gain of the category c_k from the word w_j it is necessary to introduce, for each word w_j, a random Boolean variable with two states: w_{jk} denotes the presence of the word in texts of the category c_k and $\overline{w_{jk}}$ denotes the absence of the word w_j in texts of the category c_k. The2 × 2 contingency table used to calculate the information gain of the category c_k from the word w_j is presented in Table 2.

Table 2. Contingency table for the category $\boldsymbol{c_k}$ and the word $\boldsymbol{w_j}$

Category Word	c_k	$\bar{c}_k$	Total
w_j	w_{jk}	$\lvert D^j \rvert - w_{jk}$	$\lvert D^j \rvert$
$\overline{w_j}$	$\lvert D_k \rvert - w_{jk}$	$M - \lvert D_k \rvert - (\lvert D^j \rvert - w_{jk})$	$M - \lvert D^j \rvert$
Total	$\lvert D_k \rvert$	$M - \lvert D_k \rvert$	M

A general concept for computing information is the Shannon entropy [17]. *Information Gain (IG)* is a common feature selection criterion in machine learning used, in particular, for evaluation of word goodness [18, 19]. Information gain is a measure of the information extracted about one random variable if the value of another random variable is known. It is closely related to the mutual information that measures the statistical dependence between two random variables. The larger the value of the gain, the stronger the relationship between the variables.

The goal of this research is to evaluate the informativeness of words for category identification, and use this informativeness for word ranking and text representations. Therefore, we will consider information gain of the category c_k from the word w_j: $IG(c_k, w_j)$. This information gain evaluates the number of bits extracted from the presence/absence of the word w_j in the text for the prediction of this text belonging to category c_k. One may expect that if a word is a very topic-specific for a category, it appears in texts belonging to this category more frequently than in texts which do not belong to this category, and the majority of texts belonging to this category contain the word.

For each category, c_k, a function is defined on texts that takes the value 1, if the text belongs to the category c_k, and 0 otherwise. For each word, w_j, a function is defined on texts that takes the value 1 if the word w_j belongs to the text, and 0 otherwise. We use for these functions the same notations c_k and w_j. Consider the corpus as a probabilistic sample space (the space of equally probable elementary outcomes). For the Boolean random variables, c_k and w_j, the joint probability distribution is defined according to Table 2. The entropy and information gains can be defined as follows.

The information gain about category c_k from the word w_j, $IG(c_k, w_j)$, is the amount of information on belonging of a text from the corpus to the category c_k from observing the word w_j in the text. It can be calculated as [17]:

$$IG\left(c_k, w_j\right) = H(c_k) - H\left(c_k \middle| w_j\right), \tag{2}$$

where $H(c_k)$ is the Shannon entropy of c_k and $H(c_k|w_j)$ is the conditional entropy of $c_k$given the observing the word w_j. Entropies $H(c_k)$ and $H(c_k|w_j)$ are computed as follows:

$$H(c_k) = -P(c_k)\log_2 P(c_k) - P(\overline{c_k})\log_2 P(\overline{c_k}). \tag{3}$$

$P(c_k)$ is the probability that the text belongs to the category c_k, $P(\overline{c_k})$is the probability that the text does not belong to the category c_k. Furthermore,

$$\begin{aligned} H(c_k\,|w_j) = P(w_j\,)\left(-\,P\left(c_k\middle|w_j\,\right)\log_2 P\left(c_k\middle|w_j\right) -\, P\left(\overline{c_k}\middle|w_j\,\right)\log_2 P\left(\overline{c_k}\middle|w_j\right)\right) \\ +\,P(\overline{w_j}\,)\left(-\,P\left(c_k\middle|\overline{w_j}\right)\log_2 P\left(c_k\middle|\overline{w_j}\right) -\, P(\overline{c_k}|\overline{w_j}\,)\,\log_2 P(\overline{c_k}|\overline{w_j}\,)\right)_{,} \end{aligned} \tag{4}$$

where

- $P\left(w_j\right)$ is the probability that the word w_j appears in a text from the corpus;
- $P(\overline{w_j})$is the probability that the word w_j does not appear in a text from the corpus;
- $P\left(c_k\middle|w_j\,\right)$ is the probability that a text belongs to the category c_k under the condition that it contains the word w_j;
- $P\left(\overline{c_k}\middle|w_j\,\right)$ is the probability that a text does not belong to the category c_k under the condition that it contains the word w_j;
- $P\left(c_k\middle|\overline{w_j}\,\right)$ is the probability that a text belongs to the category c_k under the condition that it does not contain the word w_j;
- $P(\overline{c_k}|\overline{w_j}\,)$ is the probability that a text does not belong to the category c_k under the condition that it does not contain the word w_j.

All the required probabilities, entropies and relative entropies are evaluated using the contingency Table 2:

$$H(c_k) = -\frac{|D_k|}{M}\log_2\frac{|D_k|}{M} - \frac{M - |D_k|}{M}\log_2\frac{M - |D_k|}{M}, \tag{5}$$

and

$$\begin{aligned} H(c_k|w_j) = {} & \frac{|D^j|}{M}\left(-\frac{w_{jk}}{|D^j|}\log_2\frac{w_{jk}}{|D^j|} - \frac{|D^j| - w_{jk}}{|D^j|}\log_2\frac{|D^j| - w_{jk}}{|D^j|}\right) \\ & + \frac{\mathrm{M} - |D^j|}{M}\left(-\frac{|D_k| - w_{jk}}{M - |D^j|}\log_2\frac{|D_k| - w_{jk}}{M - |D^j|}\right. \\ & \left. - \frac{M - |D_k| - (|D^j| - w_{jk})}{M - |D^j|}\log_2\frac{M - |D_k| - (|D^j| - w_{jk})}{M - |D^j|}\right). \end{aligned} \tag{6}$$

A high value of the informational gain $IG(c_k, w_j)$ (2) does not mean, in general, that the large proportion of information about a text belonging to the category c_k can be extracted from observing the word w_j in this text. This proportion depends on the value of the entropy $H(c_k)$ (5). The Relative Information Gain (RIG) measures this proportion directly. It provides a normalised measure of the Information Gain with regard to the entropy of c_k. RIG is defined as

$$RIG(c_k, w_j) = \frac{IG(c_k, w_j)}{H(c_k)}. \tag{7}$$

We expect higher $RIG(c_k, w_j)$ for the topic-specific words in the category c_k. For simplicity, we denote $RIG(c_k, w_j)$ =RIG_{jk}. Given the word w_j, RIG_{jk} is used to form the vector $\overrightarrow{RIG_j}$, where each component of the vector corresponds to a category. Therefore, each word is represented by a vector of RIGs. It is obvious that the dimension of vector for each word is the number of categories K (for the WoS subject categories K = 252). For the word w_j, this vector is

$$\overrightarrow{RIG_j} = (RIG_{j1}, RIG_{j2}, \dots, RIG_{jK}). \tag{8}$$

The set of vectors $\overrightarrow{RIG_j}$can be used to form the *Word-Category RIG Matrix*, in which each column corresponds to a category c_k and each row corresponds to a word w_j. Each component RIG_{jk} corresponds to a pair (c_k, w_j) and its value is the RIG from the word w_j to the category c_k.

We define the Meaning Space as the vector space of such vectors $\overrightarrow{RIG_j}$. The dimension of this space is the number of categories and each coordinate is the RIG from a word to this category.

If we choose an arbitrary category, the words can be ordered by their RIGs from the most informative word to the least informative one. We expect that the topic-specific words will appear at the top of the list.

For a given word w_j, the sum S_j of RIGs is calculated from the Word-Category RIG Matrix as:

$$S_j = \sum_{k=1}^{K} RIG_{jk}. \tag{9}$$

The sum S_j is a measure of the average informativeness of a word (this word has the informativeness $\frac{S_j}{K}$ on average). Now, the words in the dictionary can be ordered by their S_j. For each of these ordered lists of words, the most informative (meaningful) n words for scientific texts can be selected based on this criteria.

3.2. Experimental results

Having calculated RIGs for each word and created the Word-Category RIG Matrix, we evaluate the representation model by checking words in each category. That is, we consider the list of words with their RIGs in the corresponding category. Those words that have larger RIG are more informative in the category. Being 'more informative' here allows for the interpretation of being 'more specific' to the category's topic.

To visualise the top words in each category in a convenient way, we looked at word clouds. The font size of each word in a word cloud is proportional to its RIG in the category. For each category, words are sorted by their RIGs and the top 100 words are shown in the word clouds. Intuitively, the more informative the word is, the bigger size the word appears in word cloud Word clouds for the top 100 most informative words and histograms of RIGs for the top 10 most informative words for each of 252 categories can be found in [20].

In general, the RIG-based method proves to be more sensitive than the frequency-based method in identifying topic-specific words for a category. This means that representing words in Meaning Space has the advantage of transforming words to vectors efficiently with a benefit of considerably lower dimension than the standard word representation schemes.

To illustrate this result, we choose categories 'Biochemistry & Molecular Biology' and 'Mathematics' and compare two word clouds that are formed by using raw frequencies and RIGs in categories (see Figure 2 and Figure 3). It can be seen from the figures that the majority of the most frequent words in both categories are frequent words for the entire corpus. These words are not topic-specific for categories as they appear in almost all abstracts. The frequent but non-informative words can be seen as generalised service words of Science and deserve special analysis. This proves that raw frequency is not important for identifying scientifically specific meanings of words. Therefore, by representing words as vector of RIGs, we can avoid such frequency bias. The most informative words in categories for RIG representation are topic-related in the corresponding category. We interpret these results as evidence for the usefulness of the RIG-based representation.

Words that are expected to be used together have very close values of RIGs. In " Health Care Sciences & Services", "health" and "care" are top words and RIGs for these words are so close (see Figure D.1 in [21]). Another example is "xrd" and "difract" in "Material Science, Ceramics". "XRD" is actually abbreviation of "X-ray diffraction"; therefore, they appear together as "X-ray diffraction (XRD)" for most of cases in the category (see Figure D.2 in [21]).

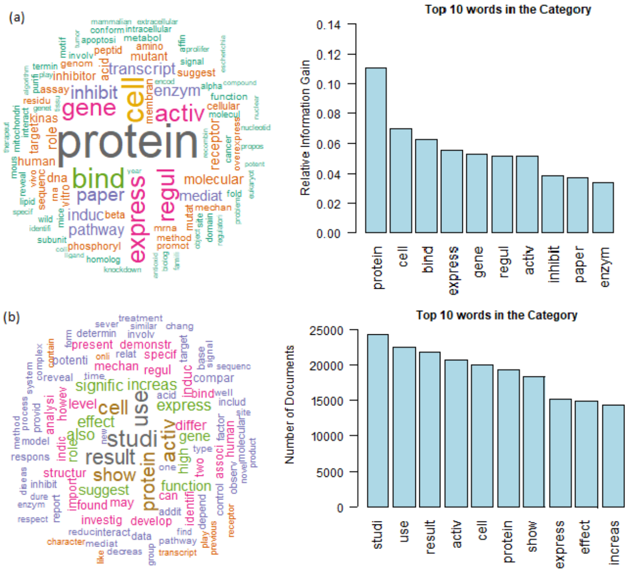


Figure 2. Category 'Biochemistry & Molecular Biology': word cloud of the top 100 most informative words and the histogram of the top 10 most informative words. The informativeness is defined by (a) RIG (b) frequency

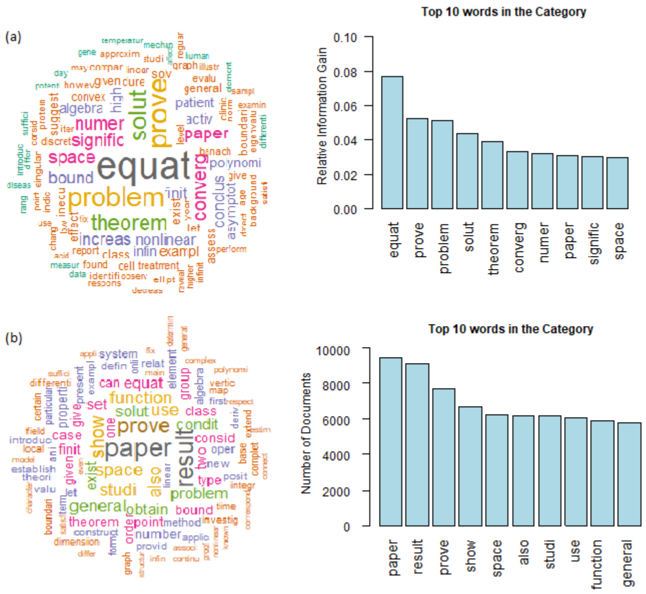


Figure 3. Category "Mathematics": word cloud of the top 100 most informative words and the histogram of the top 10 most informative words. The informativeness is defined by (a) RIG (b) frequency.

3.3. Thesaurus for Science: Leicester Scientific Thesaurus (LScT)

We have constructed the Word-Category RIG Matrix, where each entry corresponds to a pair (word, category) and its value shows the RIG for a text to belong to a category by observing this word in this text [21]. Row vectors of the matrix indicate the words' meaning in the scientific texts. A thesaurus of science was created by selecting the most informative words from the LScDC. The informativeness here was measured by the sum of RIGs in categories for this word.

We have introduced the Leicester Scientific Thesaurus (LScT): a list of 5,000 words which are created by arranging words of LScDC in their informativeness in the scientific corpus. The top 5,000 most informative words in the LScDC, where words are arranged by their S_j are considered as the most meaningful 5,000 words in scientific texts. The full list of words in the LScT with their S_j can be found in [16].

4. PRINCIPAL COMPONENTS OF MEANING

In this section, we hypothesize and test that lexical meaning in science can be represented in a lower dimensional space than 252. This space is constructed using PCA (singular value decomposition) on the matrix of word-category relative information gains. We argue that 13 dimensions is adequate to describe the meaning of scientific texts, and propose possibilities for the qualitative meaning of the principal components [22].

We apply PCA to reduce the dimensionality of the Meaning Space, in which points are 5,000 words of LScT and dimensions are categories. This section analyses the dimension of the Meaning Space and provides visualisation of words and categories in the space of PCs. In order to avoid redundant attributes in the data and identify the actual dimension of the space, we explore the Meaning Space by PCA.

We apply PCA and interpret the first five PCs by their coordinates (loadings). For each component, categories are divided into three groups defined as the main coordinates of the dimension and being unrelated attributes to the PC: categories that positively and negatively correlated with the corresponding component, and categories having near zero values in the component. We analyse the topics in these groups and visualise both categories and words on the PC axes. We also analyse the extreme topic groups at opposite ends of the PCs in order to describe the PCs based on extremely influential categories at both ends (10 categories at both ends).

Finally, by using three different selection criteria (Kaiser, Broken Stick, an empirical method based on multicollinearity control – PCA-CN), we reduce the dimensionality of the category space to 61, 16 and 13 respectively. Therefore, we argue that (lexical) meaning in science can be represented in a 13 dimension Meaning Space. We show that this reduced word set plausibly represents all texts in the corpus, so that the principal component analysis has some objective meaning with respect to the corpus. We argue that 13 dimensions is adequate to describe the meaning of scientific texts, and hypothesise about the qualitative meaning of the principal components.

4.1. Dimension of the Meaning Space

Given 252 subject categories, it is unreasonable to expect that every one of these categories is uncorrelated with all others (or distinct from them). For instance we might expect that the categories Literature and Literary Theory & Criticism will represent words in a very similar way in the Meaning Space (MS). Indeed, subcategories are likely to occur in the data and they are expected to have close values of RIGs for words. Such attributes will measure related information, and so the original 252 dimensional data contain measurements for redundant categories. Although the MS underlying the representation of word meaning has 252 dimensions, we expect that we will be able to represent words with significantly fewer dimensions in the MS.

An efficient way to represent words would be to map vectors onto a space that is constructed based on a combination of original features. Mathematically speaking, we look at a linear transformation from the original set of categories to a new space composed by new components. These new components are called *Components of the Meaning*. Two precise questions to be asked are: *how many components of meaning are there and how are these components constructed?* Thus, analysis of components (new attributes) based on the original attributes is crucial in understanding the MS. For instance, it is very important to understand which categories contribute the most and which the least to the new dimensions. Also, it is instructive to see if the new dimensions have some real semantic meaning, for instance, in distinguishing between natural and social sciences or experimental and theoretical research.

Words can be similarly represented in two or more categories. If two categories are correlated in the MS, it is possible to represent words in a reduced dimension by using a suitable linear combination of these original attributes. More specifically, if two categories are completely correlated, we would use the sum of two categories as one new attribute. The new attribute can be considered as a representative of the two original attributes. PCA provides a solution to this problem. Linear combination of weights (coefficients) is provided by PCA to create the new attribute, which we term a principal component (PC), with the aim of preserving as much variability as possible (the maximum variation in the data) [23,24]. The level of the effectiveness of PCA in explaining the data varies differently with the different sets of PCs. Therefore, in the sequel we investigate the effectiveness of PCA as a technique for determining the actual dimension of the data. Our goal is also to empirically investigate the effectiveness of the RIG-based word representation technique using PCs instead of the original attributes.

In PCA one of the crucial questions to answer is how many PCs should be selected. The Kaiser Rule is one of the methods developed to select the number of components [25, 26]. Eigenvalues of the covariance matrix are used to determine the appropriate number by taking components with eigenvalues greater than average of eigenvalues; only components explaining greater data variance than the original attributes should be kept [27].

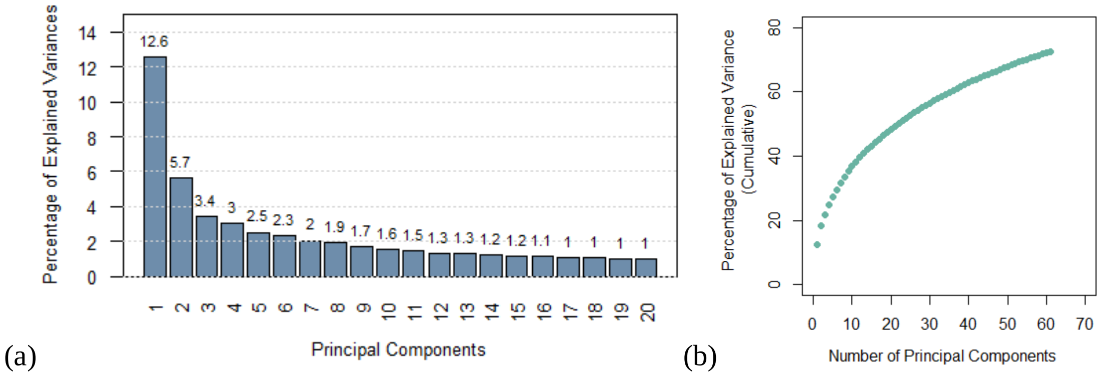


Figure 4. (a) Fraction of variance explained as a function of PCs retained for categories; (b) Cumulative fraction of variance explained as a function of PCs retained for categories (61 PCs). The mean eigenvalue is 1.

PCs were assessed sequentially from the largest eigenvalue to the smallest. All PCs having eigenvalue less than average were considered to be trivial (non-significant) by the Kaiser rule. Hence 61 PCs are included as non-trivial, that is, 61 axes summarize the meaningful variation in the entire dataset. These non-trivial PCs are retained as informative at the first stage. The cumulative percentage of variance explained is displayed in

Figure **4**. The cumulative percentage is approximately 73%, indicating the variance accounted for by the first 61 components. They explain nearly 73% of the variability in the original 252 attributes, so we can reduce the complexity of the data four times approximately, with only a 27% loss of information.

To interpret each component, the coefficients (influence) of the linear combination of the original attributes for the first five principal components are examined. The coordinates of the attribute divided by the square root of the eigenvalue gives the unit eigenvector, whose components give the cosine of the angle of rotation of the category to the PC. Furthermore, positive values indicate a positive correlation between an attribute and a PC and negative values indicate a negative correlation. Both the magnitude and direction of coefficients for the original attributes are taken into account. The larger the absolute value of the coefficient, the more important the corresponding attribute is in calculating the PC. Positive and negative scores in PCs push the overall score of a word in the meaning space to the right or left on the PC axis.

Following data reduction via PCA we then restricted the analysis of the informative categories to the non-trivial PCs; these are used to list informative attributes (categories). The importance of an attribute is determined as the maximum of the absolute values in coordinates of informative PCs for this attribute. The threshold $1/\sqrt{252}$ (threshold of importance) is used in the selection of informative attributes.

To examine the original attributes in the PCs, we introduce a threshold for categories having near zero values. The threshold used was $1/2\sqrt{252}$, which is half of the threshold of importance in selection of informative attributes. All values between $-1/2\sqrt{252}$ and $1/2\sqrt{252}$ are considered to be negligible so are in the zero interval. Hence, the initial attributes are considered as belonging to three groups: (1) positive, (2) negative, and (3) zero. We interpret the categories belonging to positive and negative groups as the main coordinates of the dimension as these categories contribute significantly to that direction. Categories belonging to the 'zero' group are seemed to be unrelated attributes to the PC. However, this information could be also useful. Hence, all categories in the three groups are meaningful and should be interpreted.

Categories in the three groups for each PC can be seen in Figures 3.3-3.7 in [22]. For the demonstration of the idea, we have displayed the second principal component in Figure 5. The zero interval is shown by a line in the figure. The number of categories in each group is presented in Table 3. The full list of categories in positive, negative and zero groups for each of five PCs can be found in Appendix C in [22].

We can see that there are no negative values for the first principal component. The first component primarily measures the magnitude of the contribution of categories to the PC. It is a weighted average of all initial attributes. The most prominent categories are 'Engineering, Multidisciplinary' and 'Engineering, Electrical & Electronic', that is, they strongly influence the component. This component explains 12.58 % of all the variation in the data. This means that more than 85% of the variation still retained in the other PCs.

Table 3. Number of categories in the groups of positive, zero and negative for the first five principal components

	PC1	**PC2**	**PC3**	**PC4**	**PC5**
Positive	221	63	60	67	55
Zero	31	131	131	129	142
Negative	0	58	61	56	55

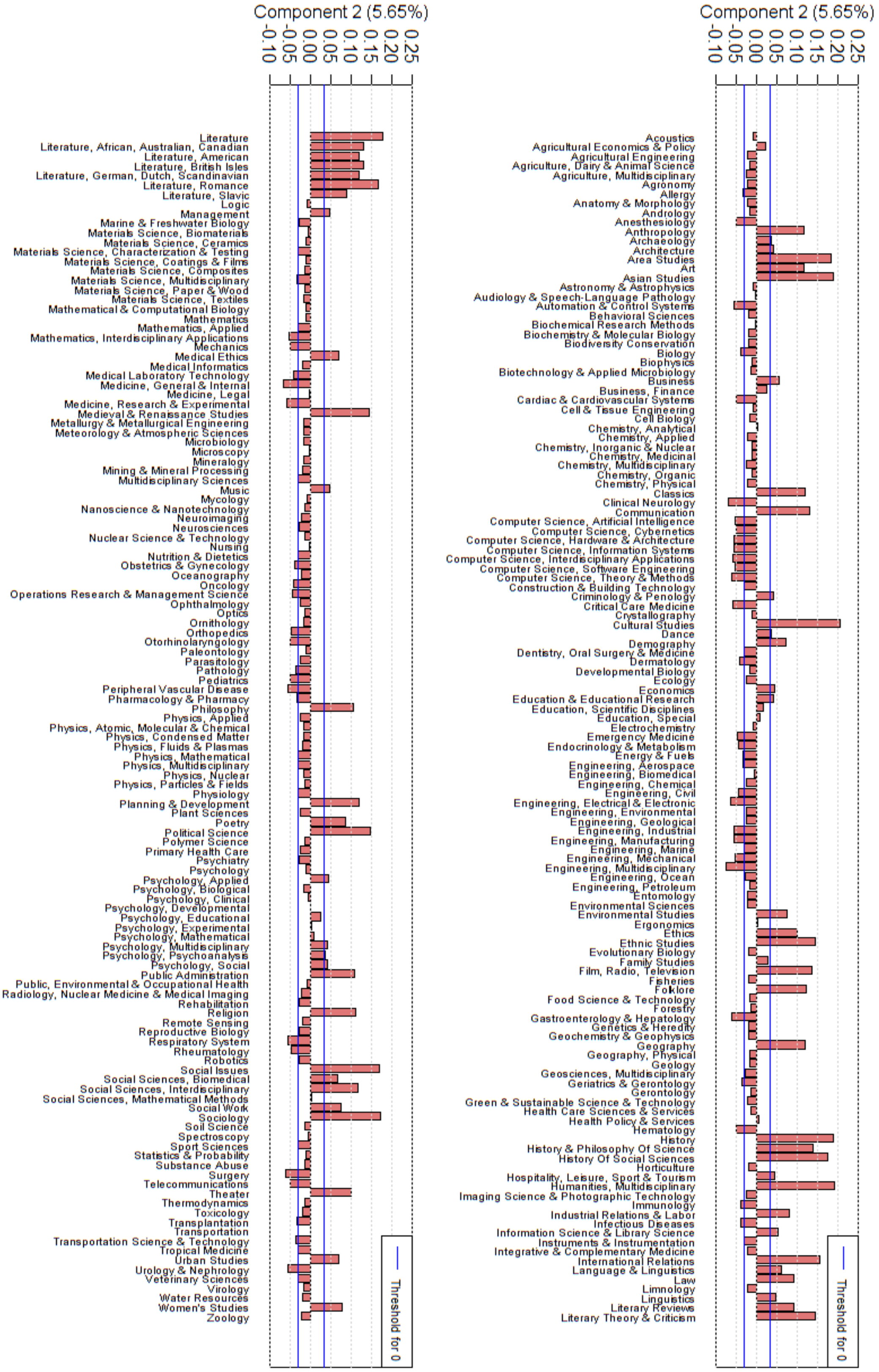


Figure 5. The second principal component of the LScT. The plot shows the contributions of original attributes (categories) on the second principal component.

The second component has positive associations with categories related to social sciences and humanities, and negative associations with categories related to engineering and natural sciences (see Figure 5). The plot shows that they are completely oppositely correlated. Hence, this component primarily measures the separation of two main branches of science. The most prominent category in the component is ‘Cultural Studies’. The largest negative contribution to the component score is from the category ‘Engineering, Multidisciplinary’, which is approximately 2.5 times smaller than the contribution of ‘Cultural Studies’. In the zero interval, extremely low values are present for attributes such as ‘Psychology, Developmental’, ‘Ergonomics’ and ‘Medicine, Legal’.

The largest positive values on the third component can be interpreted as contrasting the biological science, computer science and engineering related areas with medicine, social care and some other disciplines. We may expect words that are used in biological science, computer science and engineering will go toward the positive side of the axis on the third principal coordinate. The largest negative values suggest a strong effect of psychology, medicine-health and physics related areas.

The other two principal components can be interpreted in the same manner. In the fourth component, the most prominent categories with positive values are some of social science branches such as economics, managements, psychology, ethics, education and multidisciplinary social science. Large negative values are for categories related to literature and medicine-health science. The fifth component has large positive associations with ecological, environmental sciences and geosciences.

We then analyse the topic groups at opposite ends of the PCs (positive and negative ends) in order to describe the PCs based on extremely influential categories at both ends. As such categories have high contributions in the PC, they are the parts of the trends in PCs and so explain the general trends of the PCs. This implies that we consider positive and negative groups introduced before, select the top *n* categories with the highest component coefficients in each group and describe the grouping of categories in a way that categories at extreme ends can be distinguished from each other somehow and meaningfully described by a classification of research fields in science.

We implemented a heuristic technique. This approach starts with a search for a set of 10 categories with maximum coefficients at the two ends of the PC. The most informative 150 words are extracted in each of 10 categories, and the common words are listed. Words are analysed by human inspection to understand the meaning behind the opposite ends of the PC. The procedure is repeated for the PC2, PC3, PC4 and PC5. For the first PC1, the sign of coefficients are positive for all categories. High numbers for categories in this PC indicate that that category is well-described by words in the LScT.

The second PC seems to correspond a separation between discourse studies and experimental studies when we consider both the categories and words. For example, it is seen that three of the most informative common words are “argu”, “polit” and “discours” for the groups of categories in the positive side and three of the most informative common words are “clinic”, “treatment” and “therapi” for the groups of categories in the negative side in the PC. This is the **Nature of Science** dimension.

The third PC reflects two opposite types of research in terms of the requirement of microscopic and macroscopic instruments. At the positive end, scientific research mostly required detailed tools to work with the objects. Such tools can be instruments such as the microscope as well as programming tools used in coding. On the negative end, we are talking about human and population scale objects, but still related to humans. So this is the **Human Scale** dimension.

The fourth component appears to describe two classes of science: science of understanding the human condition through experiments and science of understanding the human condition through critical discourse studies. For instance, literary studies in the negative side are prominent and many texts from the literature are literary criticism of works. This is the **Human Condition** dimension.

Finally, the fifth component can be interpreted as contrasting natural science and intelligence. Categories related to natural science research are grouped in the positive extreme side and categories of understanding intelligence are located in the negative extreme side in this PC. ’Intelligence’ can be both human intelligence and machine intelligence. For example, the categories ’Computer Science, Artificial Intelligence’ and ’Psychology’ are two of the top 10 categories. This is the **Inner World/Outer World** dimension.

4.2. Deciding the Dimension of the Meaning Space

The number of principal components determined by the Kaiser rule was 61. However, the Kaiser rule can underestimate or overestimate the number of PCs to be retained [28]. So, we also tested the Broken-Stick rule to determine the number of PCs [29-32]. Figure 6 demonstrates the optimal number of components determined by the Broken Stick and the Kaiser rules. The Broken Stick rule suggests that the reduction to only 16 PCs is reasonable.

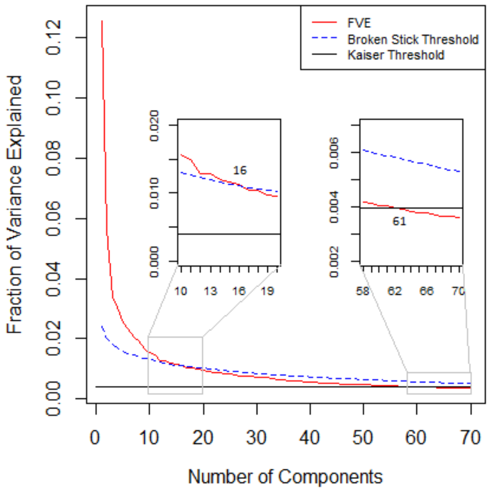


Figure 6. The number of principal components determined based on the Kaiser rule and the Broken Stick rule

Finally, we compared these two criteria of PC selection with the criterion: the ratio of the maximal and minimal retained eigenvalues ($\lambda_{max}/\lambda_{min}$) should not exceed the number of components selected (the condition number) [33-35]. This is described as multicollinearity control. To avoid the effects of multicollinearity, the conditional number of the covariance matrix after deleting the minor components should not be too large. That is, k is the number of components to be retained if k is the largest number for which $\lambda_1/\lambda_k < C$, where C is the conditional number. This method is called PCA-CN [35]. In our work, modest collinearity is defined using collinearity with C = 10 as in [34]. Therefore, the number of PCs to be retained is 13 by PCA-CN.

5. CONCLUSION AND DISCUSSION

In this work, we have initially studied the first stage of ‘quantifying of meaning’ for scientific texts: constructing the space of meaning. We have introduced the *Meaning Space* for scientific texts based on computational analysis of situations of the use of words. The situation of use of a word is described by the absence/presence of the word in the text in scientific subject categories. The meaning of the text is hidden in the situations of usage and should be extracted by evaluating the situation related to the text as a whole.

The situation of use is described by these 252 binary attributes of the text. These attributes have the form: a text is present (or not present) in a category. The meaning of a word is determined by categorising texts that contain the word and texts that do not. It is represented by the 252-dimensional vector of RIG about the categories that the text belongs to, which can be obtained from observing the word in the text.

We introduced an informational space of meaning for short scientific texts. The proposed word representation technique was developed and implemented on the basis of LSC with LScT. For concreteness, we followed the road: Corpus of texts + categories → Meaning Space for words. This involved the representation of words in the constructed Meaning Space and a detailed analysis of the Meaning Space. The proposed representation technique is evaluated by analysing the top-ranked words in each category. For individual categories, RIG-based word ranking is compared with ranking based on raw word frequency in determining the science-specific meaning and importance of a word.

We conclude that the use of informational semantics provides sizeable improvements to represent meaning in scientific texts over classical representation approaches based on raw frequencies, but how to make best use of it in different NLP tasks remains an open question that deserves further investigation.

This research has also introduced and analysed a scientific thesaurus LScT: a thesaurus of 5,000 words from the LSC. In the creation of the thesaurus, we have focused on the most informative words in science, which are the main scientific content words.

Our approach to meaning has been directed to meet some of the main challenges in extracting meaning from texts. First, it solves the problem of extracting the scientific-specific meanings because the proposed models of informational semantics characterise the situation of use by the subject categories of the text. Second, words have good representation for individual categories as well as the entire corpus because the relative importance of a word across all scientific categories is taken into account. Third, the creation of a space to represent words and texts is automated and reproducible so that it does not require a huge amount of human.

We also explore the Meaning Space by using Principal Component Analysis (PCA). We interpret the first five PCs by using their coordinates. We also suggest qualitative meanings for the first five of these dimensions. We welcome fierce debate over the meaning of these dimensions, but giving a qualitative meaning to these is a crucial step to understanding the meaning of meaning.
By exploring three different selection criteria (Kaiser, Broken Stick, PCA-CN) we reduced the dimensionality of the category space to 61, 16 and 13 respectively. If it turns out that we cannot explain some components of meaning at some time in the future with only 13 dimensions, we can increase the dimension. It remains a challenge to describe all 13 such dimensions in a way that makes some philosophical sense, but we hope that we have opened up this debate in this paper.

REFERENCES

[1] Ogden, C. K., & Richards, I. A., (1923) The Meaning of Meaning: A Study of the Influence of language upon Thought and of the Science of Symbolism, Vol. 29,. K. Paul, Trench, Trubner &Company, Limited.

[2] Putnam, H., (1975) 'The meaning of 'meaning', In Language, Mind, and Knowledge, ed. by K. Gunderson, Minnesota Studies in the Philosophy of Science, Vol. VII, 131–193.

[3] Carston, R., (2002) 'Linguistic meaning, communicated meaning and cognitive pragmatics', Mind & Language, 17(12), 127-148.

[4] Michaelis, L., (2003) 'Word meaning, sentence meaning, and syntactic meaning'. Cognitive approaches to lexical semantics, 23, 163-209.

[5] Wittgenstein, L., (2009) Philosophical Investigations, Fourth Edition, Revised, P. Hacker, J. Schulte, Eds., John Wiley & Sons

[6] V.A. Lefebvre, V.A., (2010) Lectures on the Reflexive Games Theory, Los Angeles, CA, Leaf &Oaks Publisher.

[7] Shchedrovitsky, G.P., (1974) 'Sense and meaning', In: Solntsev, V.M. (ed.) Problems of Semantics, Moscow, "Nauka", 76–111.

[8] Thurstone, L.L., (1934) "The vectors of mind." Psychol. Rev. 41(1): 1–32. https://doi.org/10.1037/h0075959

[9] McCrae, R.R., & Costa, P.T., (2004) "A contemplated revision of the NEO fivefactor inventory." Personal. Individ. Differ. 36(3): 587–596. https://doi. org/10.1016/s0191-8869(03)00118-1

[10] Fehrman, E., Egan, V., Gorban, A.N., Levesley, J., Mirkes, E.M., & Muhammad, A.K., (2019) Personality Traits and Drug Consumption. A Story Told by Data. Cham, Springer. https://arxiv.org/abs/2001.06520

[11] Osgood, C. E., Suci, G. J., & Tannenbaum, P. H., (1957) The measurement of meaning (No. 47). University of Illinois press.

[12] Osgood, C. E., (1952) "The nature and measurement of meaning." Psychological bulletin, 49(3): 197–237. https://doi.org/10.1037/h0055737

[13] Suzen, Neslihan (2019) LSC (Leicester Scientific Corpus). figshare. Dataset. https://doi.org/10.25392/leicester.data.9449639.v2 597

[14] WoS Subject Categories. (2019, October). Retrieved from https: //images.webofknowledge.com/WOKRS56B5/help/WOS/hp_subject_ category_terms_tasca.html

[15] Suzen, Neslihan (2019) LScDC (Leicester Scientific Dictionary-Core). figshare. Dataset. https://doi.org/10.25392/leicester.data.9896579.v3

[16] Suzen, Neslihan (2020): LScDC Word-Category RIG Matrix. University of Leicester. Dataset. https://doi.org/10.25392/leicester.data.12133431. v2

[17] Shannon, C. E. (1948) 'A mathematical theory of communication', Bell System Technical Journal, 27(3), 379-423.

[18] Largeron, C., Moulin, C., & Gry, M., (2011, March) 'Entropy based feature selection for text categorization', In Proceedings of the 2011 ACM symposium on applied computing, pp. 924-928.

[19] Yang, Y., & Pedersen, J. O., (1997, July) 'A comparative study on feature selection in text categorization', In Icml ,Vol. 97, No. 412-420, p. 35.

[20] Suzen, Neslihan (2020) LScDC Word Clouds and Tables to Visually Present the Most Informative Words in Subject Categories. figshare. Figure. https://doi.org/10.25392/ leicester.data.12191604.v1

[21] Suzen, N., Mirkes, E. M., & Gorban, A. N., (2020) "Informational Space of Meaning for Scientific Texts." arXiv preprint arXiv:2004.13717.

[22] Süzen, N., Gorban, A. N., Levesley, J., & Mirkes, E. M., (2020) "Principal Components of the Meaning." arXiv preprint arXiv:2009.08859.

[23] Dunteman, G. H., (1989) Principal components analysis, No. 69, Sage.

[24] Pearson, K., (1901) LIII. 'On lines and planes of closest fit to systems of points in space.' The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science, 2(11), 559-572.

[25] Guttman, L., (1954) 'Some necessary conditions for common-factor analysis', Psychometrika, 19(2), 149-161.

[26] Kaiser, H. F., (1960) 'The application of electronic computers to factor analysis', Educational and psychological measurement, 20(1), 141-151.

[27] Yeomans, K. A., & Golder, P. A., (1982) 'The Guttman-Kaiser criterion as a predictor of the number of common factors', The Statistician, 221-229.

[28] Zwick, W. R., & Velicer, W. F., (1986) ‘Comparison of five rules for determining the number of components to retain’, Psychological bulletin, 99(3), 432.

[29] Jackson, D. A., (1993) ‘Stopping rules in principal components analysis: a comparison of heuristical and statistical approaches’, Ecology, 74(8), 2204-2214.

[30] King, J. R., & Jackson, D. A., (1999) ‘Variable selection in large environmental data sets using principal components analysis’ Environmetrics: The official journal of the International Environmetrics Society, 10(1), 67-77.

[31] Peres-Neto, P. R., Jackson, D. A., & Somers, K. M., (2005) ‘How many principal components? Stopping rules for determining the number of non-trivial axes revisited’, Computational Statistics & Data Analysis, 49(4), 974-997.

[32] Cangelosi, R., & Goriely, A., (2007) ‘Component retention in principal component analysis with application to cDNA microarray data’, Biology direct, 2(1), 2.

[33] Fukunaga, K., & Olsen, D. R., (1971) ‘An algorithm for finding intrinsic dimensionality ofdata’IEEE Transactions on Computers, 100(2), 176-183.

[34] Gorban, A. N., Golubkov, A., Grechuk, B., Mirkes, E. M., & Tyukin, I. Y., (2018)‘Correction ofAI systems by linear discriminants: Probabilistic foundations’, Inf. Sci., 466, 303-322.

[35] Mirkes, E. M., Allohibi, J., & Gorban, A. N., (2020) ‘Fractional norms and quasinorms do not helpto overcome the curse of dimensionality’, arXiv preprint arXiv:2004.14230.

AUTHORS

Neslihan Suzen (PhD) is Data Analytics and AI Innovation Fellow in the University of Leicester. She holds a PhD in the field of Natural Language Processing from the University of Leicester. Her research interests are focused on data analytics, machine learning and computational linguistics. She has practical hands-on experience in data science across a variety of fields.

Alexander N. Gorban is a Professor in Applied Mathematics and the Director of the Centre for Artificial Intelligence, Data Analysis and Modelling (AIDAM) at the University of Leicester. He worked for Russian Academy of Sciences, Siberian Branch and ETH Zürich (Switzerland), was a visiting professor Clay Mathematics Institute (Cambridge, MA), IHES (Bures-sur-Yvette, France), Courant Institute of Mathematical Sciences (New York), and Isaac Newton Institute for Mathematical Sciences (Cambridge, UK). His main research interests are dynamical systems, biomathematics and machine learning.

Jeremy Levesley (PhD, FIMA) is a Professor Emeritus in Applied Mathematics in the University of Leicester. He is a Senior Data Analyst at Redshift, University Liaison at Synoptix, and Senior Research Fellow at EMPAC. His research activity includes approximation in Euclidean space and on spheres using radial basis functions, and generalisations of these procedures to locally compact manifolds. He is interested in the applications of RBFs in finance, especially practical high dimensional approximation using sparse grid methods. Prof Jeremy is also interested in Smart City and Digital Medicine.

Evgeny Mirkes (Ph.D., Sc.D.) is a Research Associate at the University of Leicester, and a leader of the Data Mining group. His main research interests are biomathematics, data mining and software engineering, neural network and artificial intelligence. He has led and supervised many projects in data analysis and the development of decision-support systems for computational diagnosis and treatment planning and has participated in applied projects in Natural Language Processing in the area of social media data analysis. Dr Mirkes has rich experience in Predictive Mathematical and Computational Modelling and in finding solutions to classification, clustering, and auto coding problems.

A WEARABLE AND INTERNET-OF-THINGS (IOT) APPLICATION FOR SLEEP DETECTION AND LIGHTING CONTROL USING AI AND MACHINE LEARNING TECHNIQUES

William Ma[1] and Yu Sun[2]

[1]Crean Lutheran High School, 12500 Sand Canyon Ave, Irvine, CA 92618
[2]California State Polytechnic University,
Pomona, CA, 91768, Irvine, CA 92620

ABSTRACT

There are many apps that let you control hardwares with the application of the internet-of-thing today, however, I have seen none that lets you customly control the hardwares, most likely you can only use the few controls the developer of the hardware gives you, and there are very few auto control options. This paper designs an application to auto control the hardwares into desired state based on personal status detected [1]. We use a smart watch to detect the heart beat of the user and determine if they are asleep, and once they are asleep, we turn off a light switch in the user's room to create a total darkness environment. Much research done on sleeping quality shows that sleeping in total darkness gives much better sleeping quality, while many, out of fear or sleeping disorders, still leave little lights on when sleeping [3]. This software helps to give these people better sleeping quality with ease [2].

KEYWORDS

IOT, AI, Machine learning.

1. INTRODUCTION

I created this project on this topic because sleeping in total darkness means better sleeping quality. But sometimes it’s really hard for people to fall asleep in total darkness. I personally was really afraid of darkness when I was small, and I still somewhat do right now. For me it is really a pain to sleep in total darkness because it creates a lot of fear and I need a long time to actually fall asleep even though I know it’s good for my health. I also know some relatives with sleeping disorders also cannot fall asleep in total darkness. I searched up on the internet and I did find many people with this problem, and there were no real good solutions other than just overcoming the fear by force, which is hard, or using sleeping pills, for the ones with sleeping disorders, which is bad for your body and creates a reliance if used too much. I really want to solve this problem not only for myself, but also for others with similar problems around the world. This application can be used by anyone who cannot fall asleep in total darkness to increase their sleeping quality, whether it’s because of fear, sleeping disorders, or any other reasons. This application probably will not be used by the majority, but it will be very helpful for the ones struggling with sleeping problems [4]. One side benefit of using it is that it also saves electricity to not have one’s light on the whole night, and saving electricity means saving some money.

David C. Wyld et al. (Eds): FCST, CMIT, SE, SIPM, SAIM, SNLP - 2022
pp. 101-109, 2022. CS & IT - CSCP 2022 DOI: 10.5121/csit.2022.120808

Some existing methods on this topic include setting a time, and turning off the light when the time is detected, this allows the user to turn off the light after they are asleep if they approximately know when they are going to fall asleep. However, firstly, this is only available in a certain smart home software, and not usable if you use hardware from another company [5]. Secondly, they will need to set a different time on the app everyday if they do not sleep at the same time everyday, which is not very convenient because you only know that you are going to sleep when you feel really sleepy, and an extra thing to do when feeling sleepy is just inconvenient. They will also need to turn off the set time every morning if they are not sleeping at the same time everyday to prevent their lights from suddenly turning off when they are awake the next day, which adds more to the inconvenience. And people could also just forget to do it. Thirdly, it will be hard for people to know when they are asleep, especially for the ones with sleeping problems. I personally know relatives with sleeping problems, and they can go to bed at 10pm but are unable to fall asleep until 1am. It will be bad if they set the time too early, which can make them awake from the almost sleeping state if they find themselves awake in total darkness, and it can also stress them to sleep faster before the light is turned off, which will make them even less likely to fall asleep due to the stress [6]. It will also not be too good if they set the time too late because it means they will sleep with the light on for a while, and therefore enjoying less good sleeping quality.

Our goal is to detect the user’s sleep. The method I use is detecting the heart rate from the user’s and evaluating if the user is sleeping or not. I train the program to learn the sleeping heart rate and awake heart rate. Then I have the program compare how similar the heart rate sound of the user is to the trained model. Then I will have the AI determine if the user is asleep or not. If the user is asleep, I turn the light off. The first strength of my method is that I use the wearOS system in the method, and it can be integrated with switching lights off with, which is something that regular fitness bands cannot really do because they do not have an outside interface and cannot be connected to the internet [7]. The second strength is that my program builds it’s AI model on top of the specific user’s heart rate instead of heart rate from a big database [8]. Which is more personalized and can sometimes mean higher detection rate for people who have a more irregular heart rate pattern.

First, I tested the app on myself, and the app gives an 63% accuracy rate immediately after detecting sleep. It gives a 73% accuracy in 5 minutes of sleep, which shows the method is an effective method. I tested it in the scenario where I wear the wearable, and the smart plug I used is wyze, which is connected to IFTTT [9]. The smart plug is connected to the lamp in my room, and my room has minor background noises. In other scenarios where I trained the machine learning model of the app on myself for a week. In 73% of the trials, where the light immediately closes after sleep is detected, the user is asleep. And in 80% of the trails, where the light closes 5 minutes after sleep is detected, the user is asleep. The watch also trains data on the same person over time, therefore increasing the accuracy over time. Data will be fitted more and more throughout time, and accuracy will increase. The part where the signal sent to the smart plug currently always works, as long as there is internet, showing this is a very compatible process, and it also works with most of the plugs out there.

The paper is organized into six sections. The first section is this section, which is the introduction. The second section is challenges, it’s about the challenge I face during the process of experimenting and creating the program. The third section is about my solution to the problems that I face in section two. The fourth section is about the details of my experiment and program.

The fifth section is about related works to my paper, and comparison between our work. The sixth section is about the conclusions, and it also points to what I will be adding on to this project and experiment in the future.

2. CHALLENGES

In order to build the project, a few challenges have been identified as follows.

2.1. Finding how to detect people's sleep

The first way is using an accelerometer, but it is to irregular, people can have really different habits [10]. The second way is using a wrist band or smart watch to detect user's heartbeat. And when the heartbeat drops to a certain level, determine if the user is asleep. It could also be not so accurate because different people can have different heart rate. For example, awake older people can have lower heart rate than sleeping young man. Then I tried to use dynamic heart rate determination based on the user's normal heart rate, but it still does not work quite well because people's heart rate can flow up and down even during sleep and it is hard to actually determine if one is actually asleep or the heart rate just dropped by accident.

2.2. Finding ways to turn off light switch

My first thought of myself for turning off the light switch is trying to find an apis of smart home companies to have my code able to connect with the light switch. But that does not really work because they don't really have only open apis. I later learned it is because it would make the smart plugs too easy to hack with an open api. And it would cause security issues. Because most smart plugs are connected with google home. But that does not really work either as it requires code to run off of a computer or raspberry pi. But I am sending code off of a watch, and google home doesn't really support that. I also tried to look at android documentation of talking to google home on android device, but that documentation is outdated and cannot be used now.

2.3. Finding ways to send a signal from a watch to the server

I originally wanted to use a Firebase to connect with the watch. But the server I choose cannot detect Firebase changes. It however can detect changes in a set google sheet or an email sent to the server. I first tried with google sheet but it does not really work because works that alter google sheet cannot really be publicly used without a license. So I decided to send the server an email as a signal. The problem is that most of the demonstrations online and APIs are being demonstrated off of a phone. And when I try to use them on the watch, it shows no error but it does not work. There is really no demonstration of sending email off of a watch online because it is a less exposed area.

3. SOLUTION

The whole system works by first detecting the heart rate of the user by the use of a smart watch. Then we input the heart rate into a machine learning system that is written in python. I do that by using an implementation called chaquopy. The program will not be usable on the first day, as on the first day, the heart rate data will be written into two files, one for resting heart rate and one for sleeping heart rate. On the second day, the AI will be able to determine if the user is asleep or not based on the model created on the first day. Once the AI determines that the user is asleep, it will send an email to an outside app called IFTTT. To send an email, the first thing that you need to do is get host names, smtp services and more. Then I need the program to detect how the thing works. Then I will need to pass the username and password of the test email in to send it. After the email is sent, it triggers a trigger in the IFTTT, which can connect to any service or products that is already connected with IFTTT, including many smart home apps. Then just choose a light to close in the smart home product menu to stop the light in your home.

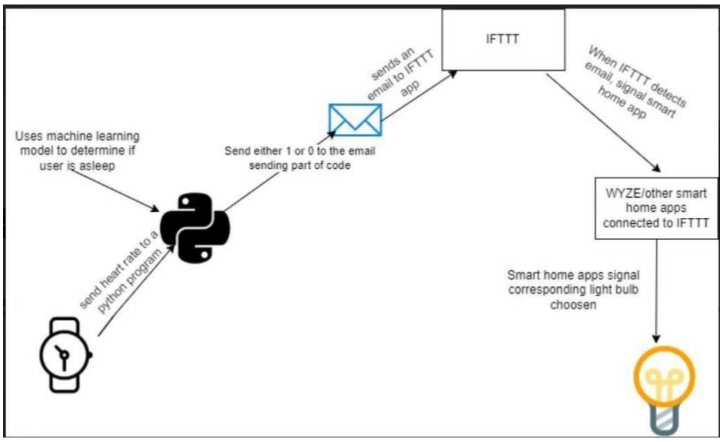
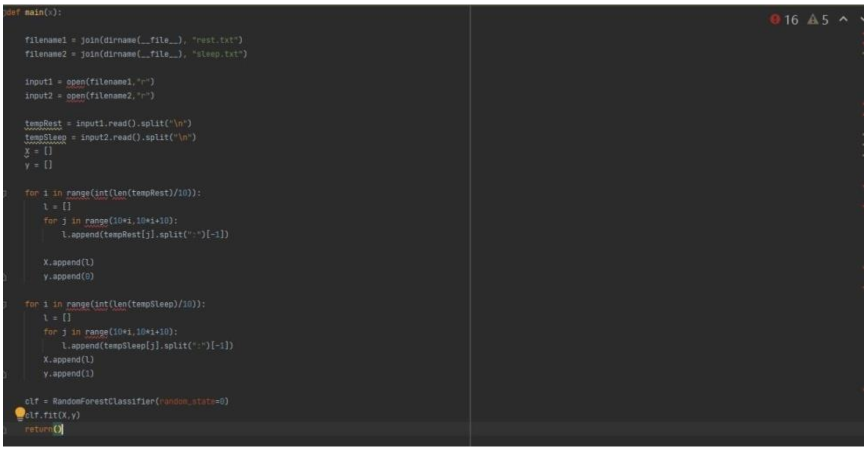

IFTTT
sends an email to IFTTT app
When IFTTT detects email, signal smart home app
Uses machine learning model to determine if user is asleep
Send either 1 or 0 to the email sending part of code
WYZE/other smart home apps connected to IFTTT
send heart rate to a python program
Smart home apps signal corresponding light bulb choosen

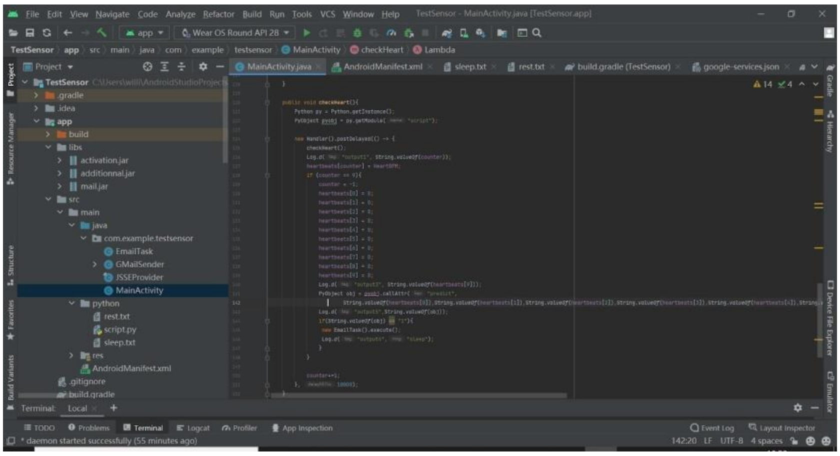
Figure 1. Overview of the system

The implementation of the heart rate detecting uses android permission and sensor system. It uses the sensorOnChange detection system. Which returns data every time that the sensor changes value, which is really fast. The implementation of machine learning is done by python with the chaquopy. There are two choices on the Java UI, one is detect rest heart rate and another is detect sleep heart rate. The python code takes in the heart rate detected from rest and sleep, and input them into two different clusters. Then train the model based on the clusters so that when a new heart rate comes in, the machine is able to determine if the user is asleep or not. The implementation of sending email uses three Java Jars: Mail.jar, activation.jar, and additional.jar. It creates smtp services to send email. The implementation of all 3 packs ensures that email sending is doable on all of the WearOS devices because some wearOS devices are different from others. The implementation of triggering light turning off is through the use of an outside program IFTTT. It is a program that can trigger actions in other registered apps. The user can set the triggered app to any smart home app registered on IFTTT. For testing purposes, I use a wyze application in the triggered application. Then you can set the trigger to anything you want. I set the trigger to email sent with a certain tag. The tag is an uuid given to the user username so the IFTTT can distinguish which user sends the email. The heart rate detection gives heart rate input to the chaquopy that’s connected to a machine learning model in python every 10 seconds. Every 100 seconds, there will be 10 heart rates in the python program, and the program will put the 10 heart rates together into a machine learning model because sometimes people’s heart rate can fall very low during awake state, so one heart rate cannot determine if one’s asleep. The machine learning model will return either a 0 or a 1 after analysis. A 0 means the user is not asleep and 1 means the user is asleep. Once the Java side of the code gets back a 1 value. Then, the email sending code will be activated. It sends an email through the internet to the IFTTT trigger. Then as the IFTTT trigger detects the email with a tag in it’s application, it will signal the triggered element the email tag trigger connects to. It uses the tag on the email I send out to specify which specific triggered element is wanted to trigger, then it sends the signal to the right smart home plug. The plug will turn it’s light off as it receives this signal.



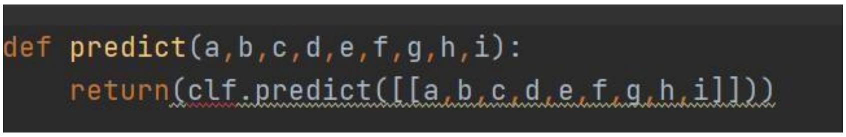
```python
def main(x):

    filename1 = join(dirname(__file__), "rest.txt")
    filename2 = join(dirname(__file__), "sleep.txt")

    input1 = open(filename1,"r")
    input2 = open(filename2,"r")

    tempRest = input1.read().split("\n")
    tempSleep = input2.read().split("\n")
    X = []
    y = []

    for i in range(int(len(tempRest)/10)):
        l = []
        for j in range(10*i,10*i+10):
            l.append(tempRest[j].split(":")[-1])

        X.append(l)
        y.append(0)

    for i in range(int(len(tempSleep)/10)):
        l = []
        for j in range(10*i,10*i+10):
            l.append(tempSleep[j].split(":")[-1])
        X.append(l)
        y.append(1)

    clf = RandomForestClassifier(random_state=0)
    clf.fit(X,y)
    return()
```

Figure 2. Python code that inputs data into the model



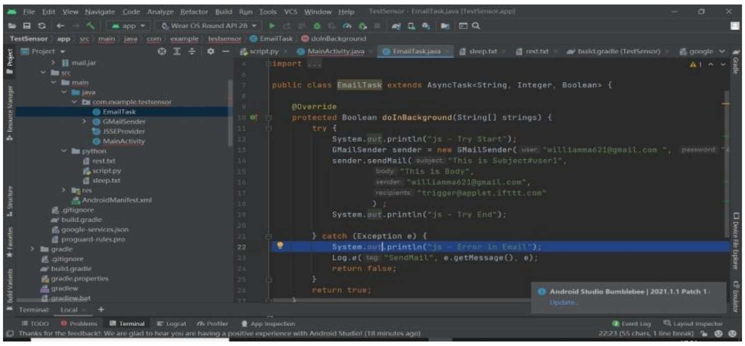
```java
    }

    public void checkHeart(){
        Python py = Python.getInstance();
        PyObject pyobj = py.getModule( name: "script");

        new Handler().postDelayed(() -> {
            checkHeart();
            Log.d( tag: "output1", String.valueOf(counter));
            heartbeats[counter] = heartBPM;
            if (counter == 9){
                counter = -1;
                heartbeats[0] = 0;
                heartbeats[1] = 0;
                heartbeats[2] = 0;
                heartbeats[3] = 0;
                heartbeats[4] = 0;
                heartbeats[5] = 0;
                heartbeats[6] = 0;
                heartbeats[7] = 0;
                heartbeats[8] = 0;
                heartbeats[9] = 0;
                Log.d( tag: "output2", String.valueOf(heartbeats[9]));
                PyObject obj = pyobj.callAttr( key: "predict",
                        String.valueOf(heartbeats[0]),String.valueOf(heartbeats[1]),String.valueOf(heartbeats[2]),String.valueOf(heartbeats[3]),String.valueOf(heartbeats[4]),String
                Log.d( tag: "output5",String.valueOf(obj));
                if(String.valueOf(obj) == "1"){
                 new EmailTask().execute();
                 Log.d( tag: "output6", msg: "sleep");
                }
            }

            counter+=1;
        }, delayMillis: 10000);
    }
```

Figure 3. Java code that inputs prediction into python side



```python
def predict(a,b,c,d,e,f,g,h,i):
    return(clf.predict([[a,b,c,d,e,f,g,h,i]]))
```

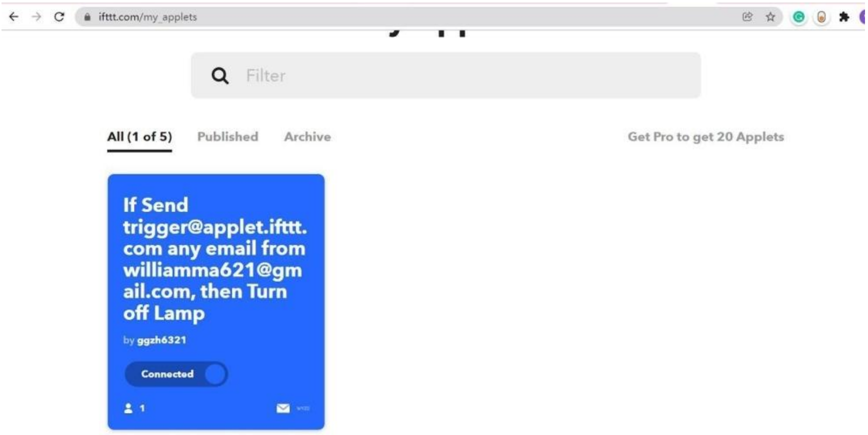
Figure 4. Python that predicts if user is asleep



File Edit View Navigate Code Analyze Refactor Build Run Tools VCS Window Help
TestSensor - EmailTask.java [TestSensor.app]
TestSensor app src main java com example testsensor EmailTask doInBackground
script.py
MainActivity.java
EmailTask.java
sleep.txt
rest.txt
build.gradle (TestSensor)
google
Android Studio Bumblebee | 2021.1.1 Patch 1
Update...
Event Log Layout Inspector
Thanks for the feedback! We are glad to hear you are having a positive experience with Android Studio! (18 minutes ago)
22:23 (55 chars, 1 line break)

```java
import ...

public class EmailTask extends AsyncTask<String, Integer, Boolean> {

    @Override
    protected Boolean doInBackground(String[] strings) {
        try {
            System.out.println("js - Try Start");
            GMailSender sender = new GMailSender( user: "williamma621@gmail.com ",  password: "
            sender.sendMail( subject: "This is Subject#user1",
                    body: "This is Body",
                    sender: "williamma621@gmail.com",
                    recipients: "trigger@applet.ifttt.com"
                    ) ;
            System.out.println("js - Try End");

        } catch (Exception e) {
            System.out.println("js - Error in Email");
            Log.e( tag: "SendMail", e.getMessage(), e);
            return false;
        }
        return true;
```

Figure 5. Email Task Code



ifttt.com/my_applets

Filter

All (1 of 5) Published Archive Get Pro to get 20 Applets

If Send trigger@applet.ifttt.com any email from williamma621@gmail.com, then Turn off Lamp

by ggzh6321

Connected

1

Figure 6. IFTTT

4. EXPERIMENT

4.1. Experiment 1

My solution solves the problems by having a good accuracy rate for detecting sleep and having a very good rate of success in connecting to the smart plug to turn off the light. The experiment is scientific because it’s repeatable and there are statistically enough trials taken to show a good accuracy. The first day I detected the user’s resting heart rate using the program when he’s awake. Then I detected the user’s sleeping heart rate by detecting the user’s heart rate throughout the night, and taking the heart rate collected at intervals of time in which the user is sure he is asleep. Then on the second day, the program will analyze the data collected on the first day to decide whether the user is asleep, and send a signal immediately when the program model decides that the user is asleep to the light bulb. Due to experimental purposes, the program will also produce a sound to wake the user up. After he is woken up, he will state if he was asleep or not when the sound is played. In an attempt to increase the rate in which the user is asleep when the light is turned off, we also tested sending a signal to the light bulb and playing the sound 5 minutes after the model declared the user is asleep.

Table 1. User sleep rate when triggered immediately after the program detects sleep

Trials	Success	Rate
30	19	63.3%

Table 2. User sleep rate when triggered 5 minutes after the program detects sleep

Trials	Success	Rate
30	22	73.3%

Summary: when the program sends a signal to the light bulb 5 minutes after sleep, there is a high rate that the user is already asleep and the user would not be awake.

4.2. Experiment 2

The second experiment uses the same setting as the first experiment, but this time, the program gets the data from the whole week of data on the user's resting heart rate, sleeping heart rate and user's feedback on the machine. This time, the number of trials is lesser because the amount of time required to take enough data for each trial. And below is the result of the experiment on the 8th day.

Table 3. User sleep rate when triggered immediately after the program detects sleep

Trials	Success	Rate
15	11	73%

Table 4. User sleep rate when triggered 5 minutes after the program detects sleep

Trials	Success	Rate
15	12	80%

Summary: when the program sends a signal to the light bulb 5 minutes after sleep, there is a high rate that the user is already asleep and the user would not be awake.

The experiment result shows that there is a higher rate of success when the signal is sent 5 minutes after the user's sleep. It also shows that accuracy rates have an increasing trend. After 1 week, the rate of success in sending signals five minutes after detecting sleep is good as it reaches 80%. This also means that it is probable for the program to continue growing in success rate as more and more data is trained. The success rate of turning lights off is also very good. There are only two cases throughout the entire experiment in which the light bulb did not turn off. After some investigation, both cases are caused by some sort of accidental internet connection failure. Because both the rate of sleep detection and sending light signals is good, it proves the program is useful.

5. RELATED WORK

Sleep detection using an accelerator [11]. This work is great, it uses an accelerator and studies an individual over a long time. It is very accurate. But it requires the user to have a certain LSAD wearable on. The data cannot really be further passed on to other devices. Although mine is less accurate, it can be put on any wearable devices installed with wearOS, and it can be used on many different devices. The main difference is different methods of detection. This work uses an accelerator to study the person over a long period of time, which is hard to integrate with other things. While what I did is use a program that runs on a wearable, and possibly even on a phone in the future, that detects heart rate, and can be integrated to close light or do other things related to sleep.

This work uses sleep detection using EEG, which is certainly very accurate as it directly detects brain waves [12]. This method is very strengthful in it's accuracy and it can also gather data of many other parts about sleep besides falling asleep such as how good the sleep is, and how fast the different stages of sleep cycle in the user. But it is hard to pass data from this headset to do other things on the internet. The headset is also difficult to get. While my method certainly does not have as good of accuracy and detection of many other important things, it has enough accuracy and is compatible with turning lights off when sleep is detected.

This work uses many sensors on both phone and wearable, and passes data between phone and wearable to provide very accurate detection [13]. This is great in it's accuracy and multitude of detection, but this requires users to have too many devices to close the light when sleep is detected. It requires setup on the phone, smart watch, and plug to inter-connect all 3 to work together. This is an overkill on both the phone and the smart watch, as both devices need to stay awake and close to the user, which uses a lot of battery power. My method is easier in doing this because I only require connection between a wearable and any plug, which is much easier although with less accuracy.

6. Conclusions

In this research, the thing I did is using an android application that detects heart rate, and then, when the machine learning model finds that the heart rate is low enough, it indicates that the user is asleep [14]. As the user is asleep I send an email using 3 implementations, Mail, activation, and additional. They first connect to an smtp online, then send email with the username and password I provided. As I applied the application to experiment, my result contains an 80% accuracy after it is trained on the user for a whole week, and the program sends the signal 5

minutes after detection. Although it does not have 100% accuracy, it proves that this method is a workable approach. The Experiment result indicates that this is an effective way to solve the problem. Although it's accuracy is not entirely accurate, it is sufficient to close the light, and the thing is it is compatible with many different smart home plugs. You do not need to buy a specific one if you already have it. It's a cheap and compatible way to solve the problem without causing too much money and stuff.

The accuracy really needs to be increased as it is the most important part of this project. It's low when first trained, and not so high even when trained after a week. Although I can now still handle this problem by just delaying the time when the program closes the light to increase accuracy, this does not give the user optimal experience. It's also not as practical because it requires both a wrist band and a smart plug, in which many people probably do not have both and would need to buy them to use the program.

In the future, I plan to use machine learning algorithms on a computer to study the user's breathing pattern on a computer [15]. This way, the user will not need to wear a wrist band. And I probably would not need to use IFTTT as an interconnecter between my application and the smart home application because accessing permissions on the computer is much easier.

References

[1] Rezk, Mario. "Auto-control: nociones básicas e investigación fundamental." Revista latinoamericana de psicología 8.3 (1976): 389-397.

[2] Malvy, D., and François Chappuis. "Sleeping sickness." Clinical Microbiology and Infection 17.7 (2011): 986-995.

[3] De Santo, Rosa Maria, et al. "Sleeping disorders in early chronic kidney disease." Seminars in nephrology. Vol. 26. No. 1. WB Saunders, 2006.

[4] Léger, Damien, et al. "An international survey of sleeping problems in the general population." Current medical research and opinion 24.1 (2008): 307-317.

[5] Xu, Ke, et al. "Toward software defined smart home." IEEE Communications Magazine 54.5 (2016): 116- 122.

[6] Baum, Andrew, Jerome E. Singer, and Carlene S. Baum. "Stress and the environment." Journal of social issues 37.1 (1981): 4-35.

[7] Liu, Renju, and Felix Xiaozhu Lin. "Understanding the characteristics of android wear os." Proceedings of the 14th annual international conference on mobile systems, applications, and services. 2016.

[8] van Ravenswaaij-Arts, Conny MA, et al. "Heart rate variability." Annals of internal medicine 118.6 (1993): 436-447.

[9] Ovadia, Steven. "Automate the internet with “if this then that”(IFTTT)." Behavioral & social sciences librarian 33.4 (2014): 208-211.

[10] Ravi, Nishkam, et al. "Activity recognition from accelerometer data." Aaai. Vol. 5. No. 2005. 2005.

[11] Girardin Jean-Louis, Daniel F Kripke, Roger J Cole, Joseph D Assmus, Robert D Langer. Sleep detection with an accelerometer actigraph: comparisons with polysomnography, Science Direct, Jan. 2001, www.sciencedirect.com/science/article/abs/pii/S0031938400003553.

[12] Hal, Bryan V., et al. Low-cost EEG-based sleep detection, IEEE, 2014, ieeexplore.ieee.org/abstract/document/6944641.

[13] Martinez, Gonzalo J., et al. Improved Sleep Detection Through the Fusion of Phone Agent and Wearable Data Streams, IEEE, 2020, ieeexplore.ieee.org/abstract/document/9156211/authors#authors.

[14] Enck, William, et al. "A study of android application security." USENIX security symposium. Vol. 2. No. 2. 2011.

[15] Mahesh, Batta. "Machine learning algorithms-a review." International Journal of Science and Research (IJSR).[Internet] 9 (2020): 381-386.

TRAC: AN APPROACH TO TEACHING SECURITY-AWARE PROGRAMMING IN UNDERGRADUATE COMPUTER SCIENCE COURSES

Rochelle Elva

Department of Mathematics and Computer Science,
Rollins College, Florida, USA

ABSTRACT

The unfortunate list of software failures, attacks, and other software disasters has made it apparent that software engineers need to produce reliable code. The Department of Homeland Security reports that 90% of software exploits are due to vulnerabilities resulting from defects in code. These defects are easy to exploit. They are potentially dangerous as they create software vulnerabilities that allow hackers to attack software, preventing it from working or compromising sensitive data. Thus, these defects need to be addressed as part of any effort to secure software. An effective strategy for addressing security-related code defects is to use defensive programming methods like security-aware programming. This paper presents TRAC, an approach to teaching security-aware programming. The acronym stands for Teach, Revisit, Apply and Challenge. It also describes the implementation of the approach and the results of a small case study (n = 21), in a senior-level elective course.

KEYWORDS

Security-Aware Programming, Secure Coding, Software Security, Teaching Secure Coding.

1. INTRODUCTION

In our modern world, every facet of our lives from the most mundane activities in our homes to the more complex activities like medical care, all rely to some extent on software. These systems utilize software to manage a wide range of applications and services. They handle sensitive data including personal and financial information and control processes that are at times life-threatening. As a result of this strong dependency on software, even a minor security breach can have a ripple effect resulting in tremendous damage that cannot be contained or localized. The unfortunate list of software failures, attacks, and other software disasters has made it very apparent that as software engineers, we need to produce reliable code- that is secure code.

The Department of Homeland Security reports that 90% of software exploits are due to vulnerabilities that result from defects in code [1]. These defects are easy to locate and exploit. They are potentially dangerous as they create software vulnerabilities allowing hackers to attack software, rendering it non-operational and/or compromising sensitive data [2]. Thus, any effort to secure software must include the management of these code defects.

David C. Wyld et al. (Eds): FCST, CMIT, SE, SIPM, SAIM, SNLP - 2022
pp. 111-129, 2022. CS & IT - CSCP 2022 DOI: 10.5121/csit.2022.120809

An effective strategy for addressing security-related code defects is to use defensive programming methods [3], such as security-aware programming. These programming methods are designed to build reliable systems. They achieve their objective by incorporating security considerations into the code design process so that the software produced is free from flaws that will make it vulnerable to attack. Such systems can be trusted to perform reliably, even under unexpected conditions. From a coding standpoint, software developed defensively will be free of security-related defects such as buffer overflow errors, null pointer deference, and improper input and output validation errors.

Incorporating secure coding instruction in the undergraduate curriculum would provide our students with the ability to code securely. This is a necessary skill to prepare them for their careers [4]. Currently, there is often a knowledge gap between the coding demands of the industry and the ability of graduating students to write robust code. To bridge this gap, many high-tech companies must provide security training for new hires [5]. Security awareness training discussed by Banerjee and Panday in [6], is just one approach that some companies use. The use of security-aware coding instruction - like TRAC, in our university programs, would fill this knowledge gap and provide graduates with skills that will make them immediately marketable.

This paper presents TRAC, an approach to teaching security-aware programming. The acronym stands for Teach, Revisit, Apply and Challenge. It is a four-step approach devised to facilitate the development of mastery in writing secure code. Our approach is implemented as a module across multiple existing, core and elective courses in the computer science curriculum. This paper describes the implementation of our approach to security-aware programming and presents the results of a small case study, used as a pilot test.

2. BACKGROUND AND REVIEW OF LITERATURE

In this section, we will define the terms that will be used in this paper, explain our rationale for teaching security-aware programming, and provide a review of related work in software security - particularly software security education. We will also discuss some of the obstacles that contribute to the lack of security-aware programming instruction in the undergraduate computer science curriculum.

2.1. What is Security-Aware Programming?

To define security-aware programming, we first must define two fundamental concepts: code defects and security-related software vulnerabilities. Code defects refer to errors in code. We will focus on logical errors, not syntax errors. We assume that the target student group is capable of writing basic code that compiles. Security-related software vulnerabilities are weaknesses in software, that stem from code defects that can be exploited. Their presence in code, therefore, makes it less secure. For the purposes of this paper, security-aware programming is defined simply as coding securely. This is the skilled practice of designing and writing code so that the final product is free of defects, that could lead to security-related vulnerabilities. As a result, the code (and software produced) is robust and reliable. Like any other skill, security-aware programming is developed and refined through repeated practice. We use the contextual approach to learning presented in [7]. This involves learning to identify the code defects to be avoided and engaging in the application of the relevant strategies to prevent them in a variety of situations. Thus security-aware programming involves content from the cross-cutting bodies of knowledge including software engineering and the fundamentals of software and program development [3, 8]. The actual implementation of security-aware programming would incorporate all the skills required to build robust code. These skills would include identifying test cases that provide full coverage of the code and testing code throughout the development process including code review. In related work, the concept of security-aware programming is referred to as defensive programming [3, 4, 9], robust programming, secure programming [3, 4], having a security mindset [10], and coding using risk management [11].

According to the report of the 2008 Secure Coding Workshop, while coding security features in code is the job of only a few security specialists, security-aware programming is the responsibility of every programmer. They continue with the claim that security-aware programming is a requisite for meeting the security requirements of code [4].

2.2. The Rationale for Teaching Security-Aware Programming

Gary McGraw states that external approaches to securing software are nowhere as effective as designing software that is secure in the first place [12]. The Department of Homeland Security cites The Software Engineering Institute as reporting that 90% of software exploits are due to vulnerabilities that result from defects in code [1]. The presence of these defects needs to be addressed since they are easy to identify and exploit during attacks such as DOS (denial of service) [2]. The failure to practise defensive or secure coding has been identified as the cause of many of the defects in software [3, 13]. As a result, there has been some discussion and research on the value of teaching security-aware programming and how this skill can be incorporated into the undergraduate computer science curriculum. At the 2008 Secure Coding Workshop, industry representatives lamented the time and other resources needed to train new employees in the skills required to write secure code. They also advised that students should enter the job market already skilled in secure software development [4]. In 2010, the Summit on Education in Secure Software was convened to identify the specifics of the security content that students need to learn and to suggest effective teaching strategies [14]. Then the 2013 Computer Science Curricula added Security as part of the Computer Science Body of Knowledge in undergraduate Computer Science programs. Nine core hours were allocated for the security knowledge areas. This included fundamental concepts in security, design principles, and defensive programming [9]. It is evident from all of these efforts, that at all levels, stakeholders agree that security should be an integral part of every Computer Science program. However, for many of the reasons stated in Section 2.3 security-aware programming is totally absent, left to chance, or taught in a very limited way in many of our undergraduate Computer Science programs [15].

Our review of literature strongly supports the idea that security-aware programming should be an essential component of computer science education [2– 4, 6, 8, 9, 12, 16–18]. This is because security is a functional requirement for all software in our modern social environment. Most times students discover how to make their code robust through a process of trial and error, but the topic is hardly ever discussed in undergraduate courses, particularly at the introductory level. The TRAC approach to security-aware programming being proposed in this paper is designed specifically to provide multiple opportunities for students to develop the skill of writing robust code. A primary difference between the proposed approach and current practice is that the learning of secure coding skills is facilitated by actual curriculum design, instead of just being left to chance, as is often the case currently.

Another benefit of teaching security-aware programming is the positive impact on students' careers. Learning good secure coding habits includes understanding the value of test coverage to evaluate the efficacy of code. This is important since a large proportion of the coding aspects of the technical interview evaluate just that. Unfortunately, even students who are good programmers, often fail this aspect of the interview because they lack the skill of writing code that is fully robust. The security-aware code development paradigm will provide opportunities to develop the requisite skills, thus making students more marketable [4].

The vast amount of software failures and disasters has made it very apparent that the production of reliable code is a fundamental requirement and ethical responsibility of every software engineer. By its very definition, secure code is reliable code. Therefore, it is our responsibility as educators to teach security as part of software development. In addition, students who learn to code securely from the onset are more likely to continue this practice in their careers as this would have become second nature to them after years of repeated practice. It is also easier to teach individuals to master a skill by teaching the correct technique the first time, rather than attempting to correct deficiencies from years of bad practice [4].

The teaching of security-aware programming has taken one of three forms [8]:

- single concentrated course [19, 20]
- threading or integration in courses already existing in the curriculum [10, 21–25]
- concentration/track in a degree program [26]

Deciding on which approach to use is important. There are arguments for each approach [23]. For example, having a separate security class facilitates focus and depth of learning, and tends to be very effective since it is taught by faculty who are invested in the topic [3]. The integrated approach also has its advantages since there are multiple opportunities for concept formation through repetition. This approach also has high impact value since more students would have the opportunity to be exposed to the security content with little disruption of the curriculum. However, it would involve ’buy-in’ from all faculty teaching the classes with software security content. The third option, teaching security awareness as a concentration/track, has the disadvantage of needing specially trained faculty and the possibility of low impact, since students may not select the concentration/track. However, this approach would provide the benefit of depth of learning for the students who do select the concentration/track.

The TRAC approach to teaching security programming is a threaded approach that provides several advantages and addresses some of the issues just discussed. This approach allows faculty who appreciate the value of security education to include the approach across their classes. The use of TRAC does not alter the course schedules and provides learning opportunities for students across multiple courses. This will therefore provide impact across the curriculum even if only one faculty member ‘buys in’.

2.3. Obstacles to Teaching Security-Aware Programming

Although the Computer Science Curricula 2013 has recommended that security be infused into the computer science curriculum at all levels [9], almost 10 years later, this recommendation has not been implemented in several programs. This can be attributed to two primary reasons: perceived lack of resources and failure to believe in the merit of teaching security-aware coding. We will now discuss some barriers to the teaching of security-aware programming that we have identified through our research.

The issue of lack of resources has two main components: faculty, and curriculum bandwidth. Several institutions state that they do not have faculty with software security training [27]. Some faculty also complain about the absence of teaching resources [5]. Yet, several resources have been developed with materials that they can use. These include course modules and e-learning materials such as the Seed Project, OWASP WebGoat, and SWEEP project [8]. faculty are either unaware of their existence or they are not convinced of the value of the required time investment. Another problem is that many of the resources are more advanced and complex than what would be needed by faculty who are not security specialists. These resources tend to focus on Web-based projects and cybersecurity frameworks and are therefore inappropriate for introductory-level courses. The second resource issue is a lack of both time and space in an already packed curriculum. Creating new security courses or adding content to already existing courses is considered an unnecessary burden, causing an increased workload for both faculty and students [4, 5]. In addition, some faculty believe that introductory classes should focus solely on code algorithms and syntax. The insertion of new content is considered disruptive [3].

The other issue is the perceived merit of teaching secure programming. Many faculty do not value security-aware programming as a necessary addition to the curriculum. Some believe that they are already teaching these concepts - although student feedback suggests the contrary (see Section 4.2.3). Also, while many companies would like new hires to be skilled in secure programming, they do not explicitly include this as a requirement in posted job descriptions. Consequently, some faculty and even students do not prioritize secure programming skills in the undergraduate computer science programs [4]. According to Bishop in [27], some faculty also believe that it is a myth that the security of software will be improved by teaching students to code securely. Their rationale is that this ignores the impact of other contributors to security. It is true that one cannot overlook the value of the other facets of software security, such as security infrastructure. However, as more companies begin to accept the value of security, investment in secure infrastructures will become standard and the need for secure code will remain a standard. Teaching students to code securely, will not solve every software security problem, however, it will contribute to the solution.

2.4. Related Work

Table 1 summarizes the approaches to teaching software security that we identified in our review of the literature. In the 12 articles identified, the majority used the approach of integrating security modules into already existing courses. However, only two of these spoke specifically about threading these modules across multiple courses at different levels in the undergraduate program [18, 23]. In two of the articles, a single specialized course was used to introduce an in-depth coverage of software security [19, 20]. While all the articles suggested ways to incorporate the learning of software security principles into the undergraduate curriculum, their approach and focus were different. Some approached the teaching of software security as secure software design, while others used defensive coding. Numbers 1-4 and 9 in Table 1, used the secure software design approach, adding a level of security to system development life cycle (SSDLC). Numbers 5-7 and 12 in Table 1, used the defensive coding approach. In four of the articles, the focus was on teaching security as soon as possible, so the target group was the introductory computer science classes. These are represented by numbers 5-8 in Table 1. Two articles focused on specific software security issues: secure mobile computing [28] and digital forensics [29]. In two of the articles, the focus was to educate faculty. This served both to train the faculty and to provide resources that they could reuse in their courses [5, 23].

Table 1. Summary of the approaches to teaching software security that we identified in our review of literature

Teaching Format	#	Approach	Focus	Source
Single Course	1	Software Design	Secure Software Design	[19]
	2	Secure Software Design SSDLC	Senior-level security course	[20]
Integration in existing course/s	3	Software Design	Data Structures Software Design	[21]
	4	Software Design (SS-DLC)	Intro to Java Software Engineering	[22]
	5	Defensive coding	Introductory Classes	[10]
	6	Defensive coding	Introductory Computer Science classes	[16]
	7	Defensive coding	Introductory Computer Science classes	[24]
	8	General Security Topics - digital forensics	Introductory course	[29]
	9	Secure Software Design for Mobile Apps	Courses in Mobile app development	[28]
	10	Resources & Tools	Faculty training workshop	[5]
Integration in existing course/s and Threaded throughout the curriculum	11	Secure Software Design SSDLC using software case studies	Faculty training workshop	[23]
	12	Secure Software Design and defensive programming	6 courses (including introductory core courses) throughout the curriculum	[18]

3. TRAC APPROACH TO TEACHING SECURITY-AWARE PROGRAMMING

In this section, we present a detailed description of the TRAC approach to developing the skill of security-aware programming. We also identify the set of software defects that will be the focus of our instruction.

TRAC is an acronym for Teach, Revisit, Apply and Challenge.

Our approach is intended for use as a module in any code-based computer science course. To overcome some of the obstacles to teaching secure coding (discussed in Section 2.3), our approach works with existing courses. The techniques used can be implemented by faculty without specific training in software security. It can also be used across multiple courses, to facilitate incremental skill development through repeated practice in a variety of contexts. This is supported by Ambrose et al. in their book on how to learn. They claim that it takes at least 21 repetitions of the correct way to perform a skill before it becomes a habit.

3.1. Implementation of TRAC

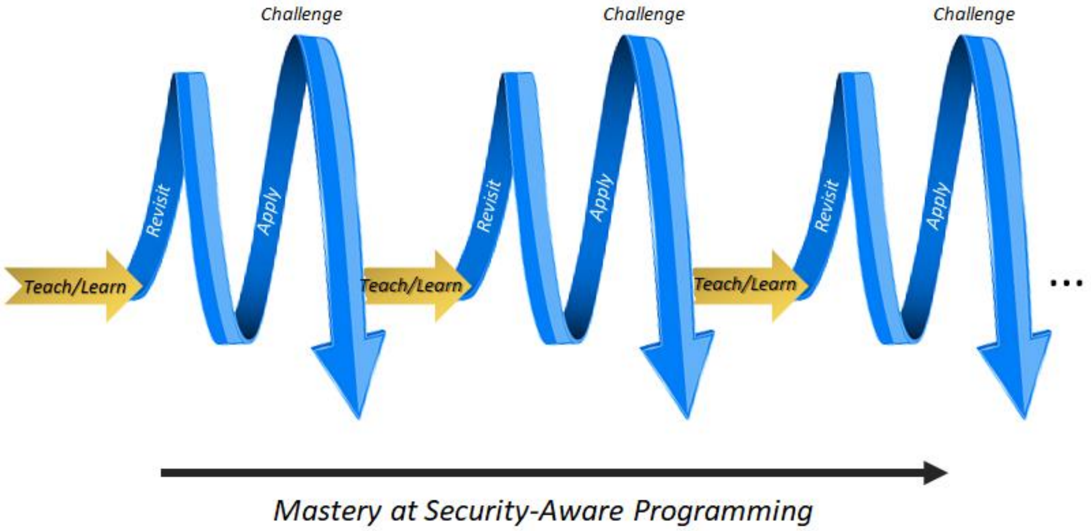


Figure 1. The TRAC Approach to Teaching Security-Aware Programming

Figure 1 illustrates our thinking on how mastery of security-aware programming skills can be developed using the TRAC approach. Learning will occur in repeated phases of teaching/learning, revisiting written code to evaluate its level of security, applying the knowledge of security-related code defects to adjust code to improve security, and challenging each other during code review. The ultimate challenge is the self-challenge - to be able to write code needing little or no security improvements.

We selected a spiral and not a cycle to represent our learning process because while the phases are repeated, the starting point for repetitions is not the same. Each experience improves the level of skill development, thus moving the learner further along the learning continuum, and closer to mastery.

3.1.1. Teaching

During the teaching stage, students will be instructed using the content described in Section 3.2.1. As a result of these experiences, students should be able to name the defects described in Table 2. They should also be able to identify these code defects in new code examples. Students will also use traditional code design tools like activity diagrams and class diagrams to identify code interface points and the data traveling across those interface points. They will learn to use these to determine potential data security issues.

3.1.2. Revisiting Stage

Once students have developed proficiency in the identification of security flaws, they will be invited to examine their previous coding assignments to find the unchecked security-related code defects that made their programs vulnerable. They will then select examples that they feel comfortable sharing with the class. The class will then discuss the presentations, identifying the most common defects found and any others that might have been missed by the presenters.

3.1.3. Application Stage

At the beginning of the module, the application stage will involve two types of activities. The first will be to apply the knowledge of security-aware programming to correct the security flaws identified in a previous assignment submission. The second will be to use the skills learned, in a completely new assignment to write code that is even more secure than their previous work. As students master the skill of security-aware programming, they will automatically use the strategies for the avoidance of code defects, to produce secure code that is error free. Through practice, writing secure code will become more natural and second nature.

3.1.4. Challenge Stage

During the challenge stage, students are given coding problems that they will solve in groups. As they work on their solutions, they will make a list of the security checks that they have considered. All team members will contribute to the final deliverable. Teams will then challenge each other to break the code created. As students progress through this process, the challenge stage will evolve into formal code reviews. This will prepare students for the code review process that is a common practice in the industry. Through the activities of this stage, students will learn how to prepare their code for review and how to critically review code prepared by their peers.

3.2. The Learning Goals of TRAC

The TRAC approach is designed to create opportunities for students to acquire the skill of writing robust code through awareness of security concerns associated with software. This goal of the approach is expressed in the following two learning outcomes: as a result of using the TRAC approach, students should:

- acquire code security knowledge
- develop mastery in the skill of security-aware programming

3.2.1. Acquisition of Knowledge

We believe that code security knowledge involves both the learning of software security content and an understanding of the contextual relevance of coding securely. Thus, facilitating the acquisition of code security knowledge instruction in TRAC begins with building a rationale for secure programming. Teaching security-aware programming using TRAC, fits into the Information Assurance and Security knowledge areas, added to the computer science curriculum in 2013 [9]. This knowledge area has the following five learning outcomes:

1. Analysis of the trade-offs of balancing security properties
2. A description of risks, threats, and vulnerabilities and how these relate to security attacks
3. Understanding the concepts of trust and trustworthiness in terms of software
4. OS SEcurity and Network Security
5. HCI

The TRAC approach addresses the first three of these learning outcomes. This is expressed in our teaching/learning objectives that students should be able to:

- explain the rationale for security-aware programming
- list and identify common code defects that are security risks
- apply design and coding principles of defensive programming to mitigate security-related code defects.

Table 2. Common Code Weaknesses Adapted From The Common Code Weakness Enumeration

CWE ID	**Weak-ness**	**Highest Position**	**Description**	**Likelihood of Exploit:**	**Negative effects**	**Mitigation Strategies**
787	Out of Bounds Write	1	Code writing data to a position before or after the memory location of a given buffer	high	Code crash, DOS, modifying memory	input validation of write parameters
125	Out of Bounds Read	3	Code reading data from a position before or after the memory location of a given buffer	high	Code crash, DOS, modifying memory	input validation including calculations producing length parameters
20	Improper Input Validation	3	Code receives and uses data without setting in place checks and balances that the values received are legitimate	high	Code crash, DOS, entire system hijacked by ransomware	Adopt a non-trust policy treat all input as untrustworthy analyze code and design for possible areas of insecure input and validate input
190	Integer Overflow	8	the results of a calculation that produces a value larger than an integer; code attempts to store that value as an integer	high	Buffer overflow, Code crash or infinite loop	input validation and validation of the result of integer calculations; using unsigned integers
129	Improper Validation of Array Index	14	Code either fails to validate array index values leading to code errors including out-of-bounds reads and writes	high	Code crash, DOS, unexpected code behavior, memory corruption, out-of-bounds read, out-of-bounds write	Adopt a no-trust policy, data validation including input validation for all data used as array index
476	Null Pointer Deference	14	Code accesses or tries to use null value as if it were an actual object reference	medium	Code crash, unexpected code behavior	Validation of all object data including input validation for all data
754	Improper check for unusual orexcep-tional conditions	15	Code fails to check for edge cases and exceptional conditions in the code	medium	Code crash, DOS, unexpected code behavior	Develop test cases that provide full coverage of code, handle exceptions locally instead of throwing them to other parts of code, anticipate error conditions

						and program code to exit elegantly
835	Infinite Loop	26	Code gets into a loop and does not have a condition to get out	no known attack pattern	Code crash due to consumption of memory, DOS	Check that all loop terminating conditions can be reached; input validation for loops managed by input data
532	Insertion of sensitive data in log file	33	As part of error handling, code unwittingly writes security-sensitive data such as code structure, file names and format to log file.	high	Attackers gaining access of log file have access to sensitive data and an unprotected path to security data	Careful selection of messages sent to log files; Sensitive error log messages used during code development and testing should be erased when no longer needed

We subscribe to the opinion cited in [3, 30] that acquiring the relevant knowledge will affect what the students observe and how they use these observations to solve new problems. For this reason, in the teaching component of our approach, we provide content that will help students to understand why they should care about code security. To establish this context, we review notorious major software failures and discuss and analyze reports from multiple sources including the Department of Homeland Security (DHS), Software Engineering Institute (SEI), and reputable new reports on current events explaining the impact of code defects.

To study code defects, we selected from an established list of verified software code weaknesses. Our source was the Top 25 Common Errors Enumeration from CWE/SANS [2]. This source ranks software defects based on their prevalence, and impact on code security. The most common and harmful defects are found at the top of the list. We examined lists from 2010 to 2021.

Several of the 25 top code weaknesses listed were not relevant to our target audience. For example, many of those listed focused on web-based software applications. We, therefore, filtered the list, keeping application-independent flaws that would be contextually relevant to most students in the computer science undergraduate program. Our final selection was the set of nine code defects shown in Table 2.

For each defect in the Table, we provide a description of the defect, an explanation of the negative impact that it can have on code, and the likelihood that this flaw would be exploited. We also mapped each defect to a list of strategies that can be used to mitigate its occurrence in code. To create a discussion-point on the relative significance of the defects selected, the highest position, held in the top 25, is presented in the Table. This highest position refers to the highest-ranking that each specific defect, ever occupied in the list of top 25 common errors, during the time period that we examined (2010 - 2021). The first eight defects in Table 2 were listed among the top 25 weaknesses at some time during our research time period. The ninth defect in our list (insertion of sensitive data in log files), was never in the top 25, it was listed in the top 35 in 2019. However, we decided to include this error handling defect for the following three reasons: there was a high likelihood that the defect would be exploited; it was a good teaching tool, and it would be relatable to students.

3.2.2. Skill Development

Our primary objective is to take learners from the novice level of security-aware programming to the level of proficiency - as experts. Borrowing from the developmental learning approach in [30], we evaluate mastery by focusing on two dimensions of learning - consciousness, and competence. Consciousness is the achievement of a goal through deliberate choice and focused action. Competence is the ability to perform a task with a high level of mastery or expertise.

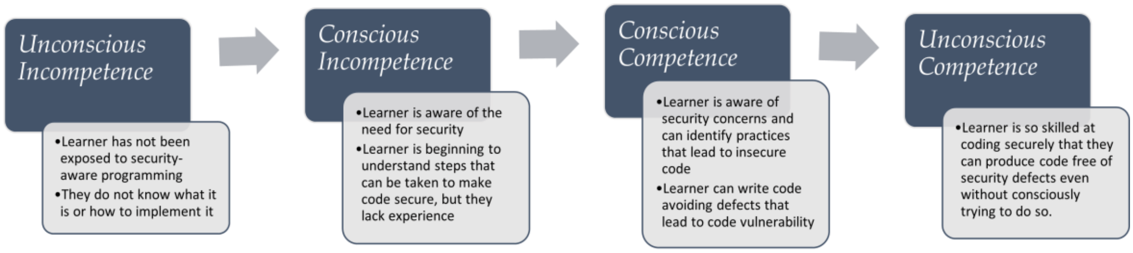

Unconscious Incompetence
•Learner has not been exposed to security-aware programming
•They do not know what it is or how to implement it
Conscious Incompetence
•Learner is aware of the need for security
•Learner is beginning to understand steps that can be taken to make code secure, but they lack experience
Conscious Competence
•Learner is aware of security concerns and can identify practices that lead to insecure code
•Learner can write code avoiding defects that lead to code vulnerability
Unconscious Competence
•Learner is so skilled at coding securely that they can produce code free of security defects even without consciously trying to do so.

Figure 2. Progression from Novice to Expert Secure Programmer using TRAC

It is our belief that the use of TRAC will facilitate learners' maturity from being unconsciously incompetent security-aware programmers to becoming unconsciously competent security-aware programmers. The stages of transitional skill development are depicted in Figure 2.

In the first stage, the learner is unconscious of security concerns and is also incompetent at coding securely (thus unconsciously incompetent). This corresponds with the beginning of the Teaching stage of TRAC. The learner transitions to the second phase, after being schooled in the identification of security-related code defects, and their impact on the vulnerability of software. This second phase is called the consciously incompetent stage because while the learner is aware of the security concerns that need to be addressed, they have limited knowledge and ability to correct them. At this point, the learner is at the stage corresponding with the Revisiting stage of TRAC. Through practice and more learning, the learner will become both more conscious of the security concerns, and competent in the strategies used to reduce and/or avoid code defects. The learner consciously and skilfully applies their learning to produce more secure code. At this point, the learner transitions to the next phase called consciously competent. This will occur during the Application stage of TRAC. With much practice and experience, the learner will effortlessly transition to the next stage. This constitutes mastery. At this point, the practice of secure coding will be second nature. The programmer codes securely on autopilot as it were. This mastery stage is described as the unconsciously competent stage. This stage corresponds with the Challenge stage of TRAC, but it is not a static stage. The learner continues through the stages of TRAC but each time gets further along the learning continuum, and closer to mastery.

4. CASE STUDY

The TRAC approach was tested in an upper-level, elective course, on Secure Software Engineering. Security-aware programming was taught as a course module, over a period of three weeks. The Security-Aware Programming module was taught as a component of the Secure System Development Life Cycle. There were 21 students enrolled in the course: seven graduating seniors, ten juniors, and four sophomores. All students had already completed at least three computer science courses.

The pilot test was evaluated using observation of students' interactions during class, evaluating written assignments, and reviewing student feedback. There were 3 written assignments. The first 2 were identical assignments but were given 2 weeks apart. Students were asked to find a code sample that they had written and submitted for one of their previous computer science classes. They were required to analyze the code to see if there were any security-related code defects. Students were then asked to modify the code sample so that it was more secure. The third assignment was to code the backend of an automated teller machine (ATM), paying special attention to security issues. Students were asked to comment their code to indicate security concerns that they had addressed. These submissions were then presented to the class for an informal code review in the form of a class discussion. A simple assignment was selected because the group of students ranged in experience from first semester sophomores to graduating seniors. The more senior students were given the option to select their own problem and prepare a secure code solution - using the absence of security-related defects as the measure of code security.

At the end of the course, students were asked to volunteer anonymous feedback on their experiences. Data was collected from all students through an anonymous, informal survey, course evaluations, anecdotal records, informal interviews, and unsolicited conversations. No extra credit was assigned for student responses. Data collection was conducted surrounding five feedback questions (FQ). Data was also collected at the end of the semester following the course (almost four months later) to determine if the course in software security had impacted their coding habits. The latter is analyzed as FQ 6.

The following 6 feedback questions were used to obtain student feedback.

1. How would you define secure software?
2. Has your perception of software security changed during this semester? If yes how?
3. What would you say was the greatest takeaway from this course?
4. How did the course match the expectations that you had during registration?
5. Is there any area/topic covered in this course that you will use going forward? If yes, please explain
6. **four-month Check-In:** Are there any security strategies/checks that we studied last semester that you find yourself paying more attention to as you write code now?

4.1. Results from Observing Students in Class and Evaluating Assignment Submissions

From students' interactions in class, it was clear that they were engaged and enjoying the content on software security. All students participated in class discussions. One sign that students were really engaged with the content (in and out of class), is that on three occasions, different students sent an email sharing a software security story, that they had heard from current events on the news. Also, as the semester progressed and students became more confident in their ability to identify and correct security flaws, they became more willing to critique each other's work and to present their own code for critique.

In the submissions for Assignment 1, student examples were almost 100% cases of failure to validate input. By the time students submitted the second assignment, their growth in knowledge was evident. They presented more complex code with a variety of different security flaws and were able to suggest code alternatives that would improve the degree of security of the code. There was one group that struggled with finding a more complex coding example. However, after observing code presented by their peers, they were able to resubmit more complex and accurate code for the assignment. Assignment 3 was very well done. Most students solved the problem assigned. A few groups took the challenge to develop a more complex system. They implemented a database query system, an account login validator, and a batch processor for financial transactions.

From these exercises, it was evident that students were able to identify and correct the set of security-related code flaws listed in Table 2.

4.2. Results from Students' Responses to Feedback Questions

In this section, we will discuss students' responses to each of the six feedback questions from Section 4

4.2.1. Responses to FQ 1: Student Definition of Secure Software

Figure 3 illustrates the terms used by students to describe their understanding of secure software. Most students, (between 80% and 100%), defined secure software using the ACID (availability, confidentiality, isolation, and durability) properties, associated with reliable or robust software. All students defined secure software, as the product of secure coding. A little over 70% of the students included strategies used to achieve software security in their definition. They included both reactive and proactive measures.

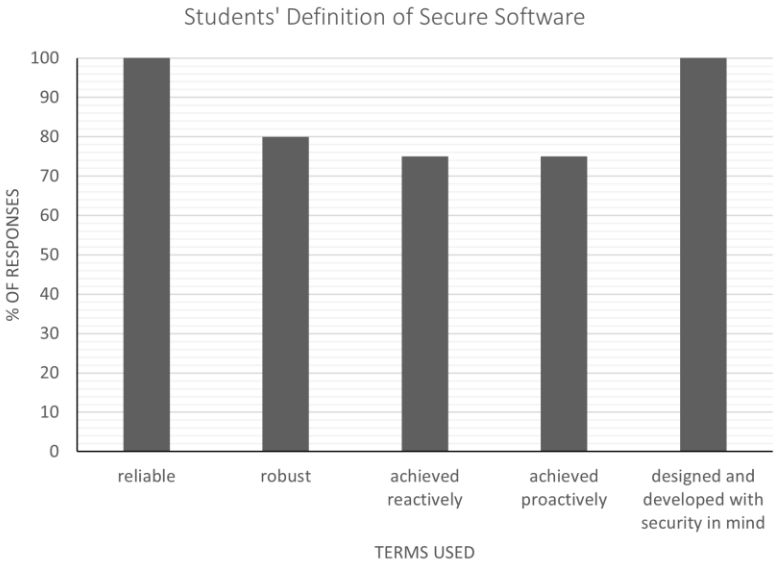


Figure 3. Graph of students' responses to Feedback Question 1

4.2.2. Responses to FQ 2: How Students' Perception of Software Security Has Evolved During the Semester

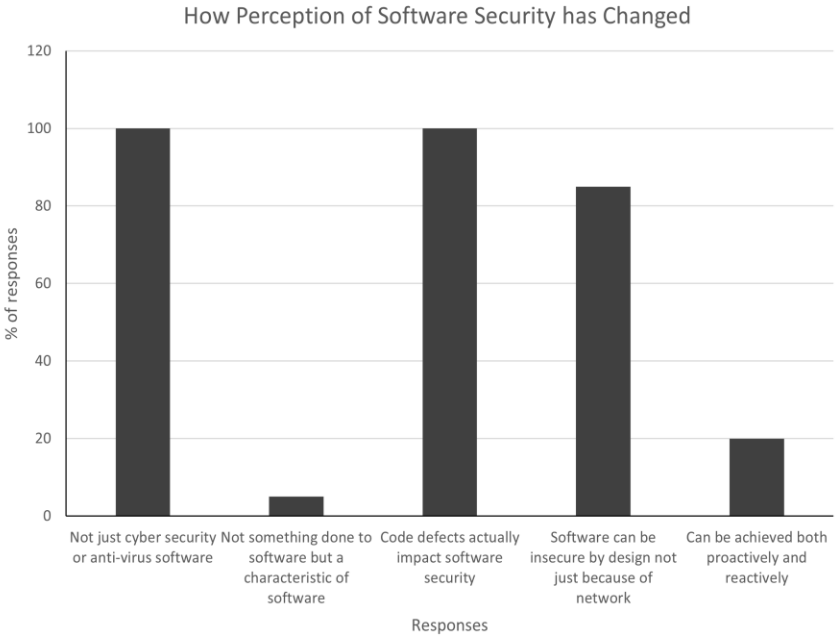


Figure 4. Graph of students' responses to Feedback Question 2

When describing how the course changed their perception of software security, students' responses generally surrounded three main themes:

- Definition of Security
- Value of Security-Aware Programming in achieving Software Security
- Their own ability to implement strategies to make their software more secure

Students explained that they had previously viewed software security as cybersecurity. At the end of the course, that perception had changed to viewing software security as a characteristic of the software and more than just cybersecurity. All students commented on a new understanding that software security can be negatively impacted by code defects. Secure coding was therefore something that they were capable of, by proactively avoiding security-related code defects. Figure 4 summarizes students' responses to this feedback question.

4.2.3. Responses to FQ 3: What Students Considered the most Significant Takeaway from the Course

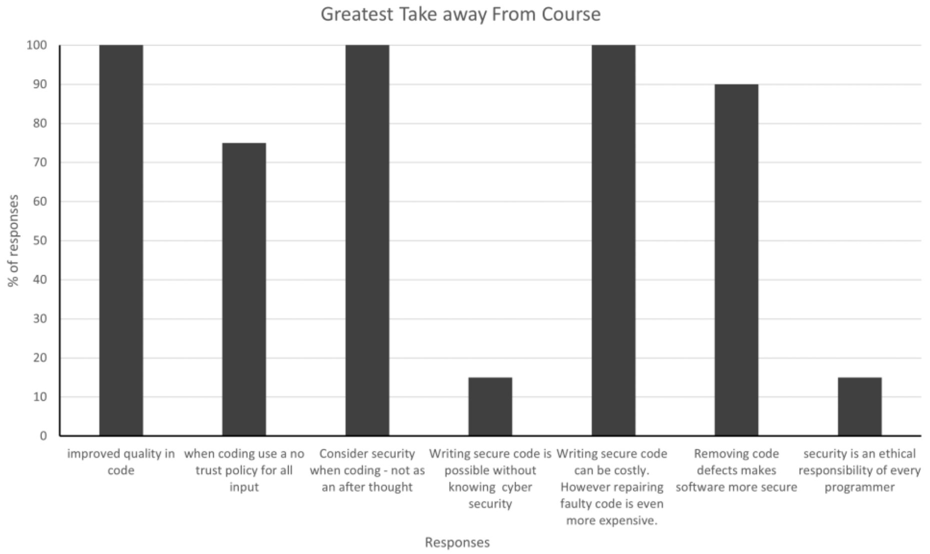


Figure 5. Graph of students' responses to Feedback Question 3

All the students considered the skill developed in the identification of software flaws as one of the greatest takeaways of the course. They all indicated that this was a skill that they were happy to have acquired as it has improved the quality of their code. Most students indicated that their code has improved because they test more extensively to ensure that all edge cases were accounted for.

Three students commented on having adopted a no-trust policy with input data, so they validate all input before using it. Students also commented on the fact that security-aware programming was a surprisingly simple, yet useful tool that they would be using for the rest of their careers. Two students commented that they have been teaching their friends how to code securely when working on collaborative projects. Two students discussed that achieving secure software may be costly in terms of time and human resources. However, the return on this investment was worth it. Another student reported that the primary takeaway was the appreciation that it is the ethical responsibility of all programmers to develop code that is reliable, and that security-aware programming is a good tool for realizing this. Figure 5 summarizes these results.

4.2.4. Responses to FQ 4: Students' Expectations of a Secure Software Engineering Course

All the students, except one, expected a course involving cybersecurity training. The one exception reported to never having thought of software security before and therefore had no expectations for the course. However, 80% of the students commented that developing the skill of security-aware programming was empowering because using security-aware programming made the development of secure code an attainable goal.

4.2.5. Responses to FQ 5: Aspect of the course that will be used going forward

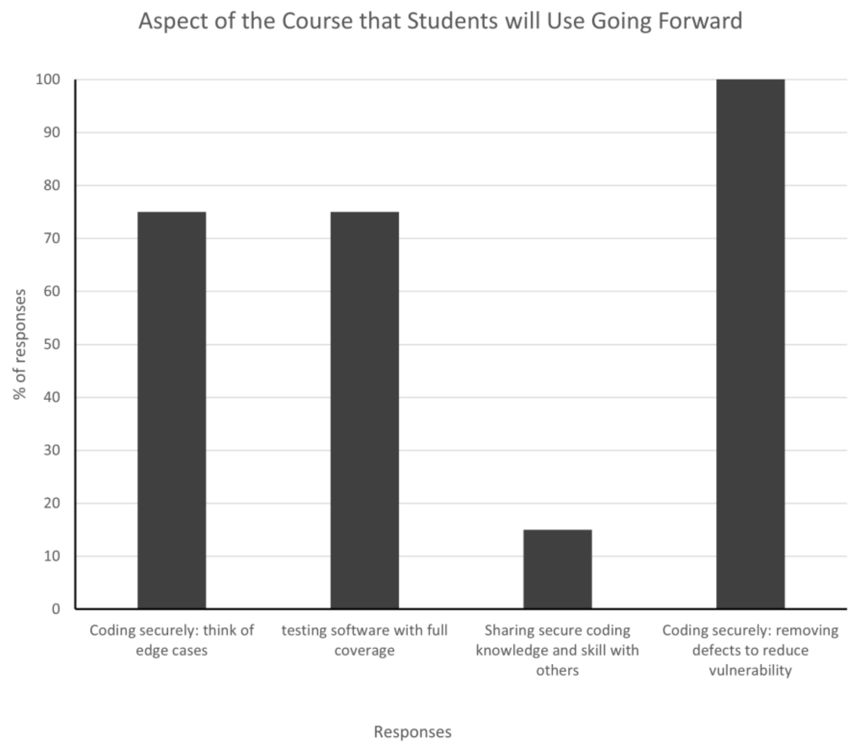


Figure 6. Graph of students' responses to Feedback Question 5

Figure 6 shows students' responses to feedback question 5. All students identified coding securely as a skill that they will continue to use. Some suggested that this would focus on managing code defects while others discussed employing good testing strategies, particularly considering test cases that provide full coverage.

4.2.6. Responses to FQ 6: Impact of taking the Course - 4 Months Later

Only half of the students (10) from the original case study responded to question 6. The 7 seniors had graduated, and 4 other students did not submit a response. Of those who did respond, 100% indicated that they were still using the security-aware programming approach. The strategies being employed the most were robust testing of code to spot errors, input validation, and exception handling. Three of the students reported that they were proudly sharing their knowledge and skill with their peers. One student who is a peer tutor for Computer Science 1, reported that he uses the approach to help underclassmen, during tutoring sessions. However, the greatest endorsement was from the student who wrote "...I cannot help myself now. All my code has to be secure. I cannot code anymore without looking for the security vulnerabilities. I wish that we had been taught this before! Every student should take this course!"

4.3. Other Feedback from Students

In this section, we report on other student feedback from informal face-to-face conversations and emails. Generally, students were excited about learning to code securely. Only three students felt that they had been introduced to the concept of secure coding before. All three of those students had previously been taught by the researcher. Interestingly, faculty claim that they teach security measures in their courses. However, somehow this had not translated into student perception. Two of the more advanced students said that they had learned some version of secure programming over time, by trial and error. Several students also reported that the content covered in the course had changed how they were programming in their other computer science classes being taken in the same semester. One senior who was in the process of interviewing for a full-time software engineering position said that the security-aware approach to programming was helping with the technical component of their interviews. All the students suggested that some of this content should be included in the introductory courses.

5. Discussion and Conclusion

The case study points to the potential for using TRAC to develop security awareness in students. From student feedback and the observation of student interactions in class, students demonstrated knowledge of software code defects. They also demonstrated the ability to effectively identify and purge those defects from code, to make it more secure. From their feedback, it also appears that students have gained confidence in using the approaches covered in class. They also appear to feel more competent and confident in their ability to write code that is security-aware. All the students who responded to FQ 6 have reported that they are continuing to use the security-aware approach to coding.

One surprise from the case study was that even when faculty think that they are teaching students to code securely, students do not see it that way. This experience emphasizes the importance of designing security content in our courses instead of leaving it to chance. It is also important to present the security content in a contextually relevant way (as is done in the first stage of TRAC). This will help students to understand the rationale for the approach and the trade-offs for not using secure coding measures.

It was also evident that teaching security-aware programming even to seniors, did not always require complex skills like cybersecurity strategies. The basic code defect identification and avoidance appear to be very effective. Often one of the main arguments for not teaching this approach is that faculty are not trained in security. This case study shows that any faculty member who has learned to code can provide opportunities for students to practice and develop the skill for secure programming.

The case study suggests that the TRAC approach can be beneficial to students. However, the size of the study group was a limitation. Also, only half of the tested group responded to FQ 6 - the four-month follow-up. A more complete picture of the impact could be obtained from a larger number of responses.

The current research was designed as a pilot study. In the future, we plan on using this approach to security-aware programming in three other courses: Introduction to Java, Data Structures and Algorithms, and Object-Oriented Programming. This will increase the sample size. We will also gather independent feedback on students' progress, by monitoring their activity using a version control platform. This can be done anonymously, by providing students with random account credentials.

The researchers are aware that teaching students to develop code with security awareness, is not the silver bullet to making all software secure. However, we believe that it is an important tool, that students can use to contribute to the inherent security of their software products.

REFERENCES

[1] Software assurance. [Online]. Available: https://www.cisa.gov/uscert/ sites/default/files/publications/infosheet SoftwareAssurance.pdf

[2] B. Martin, M. Brown, and S. M. Christey, “2010 cwe/sans top 25 most dangerous software errors,” 2010.

[3] M. Dark, I. B. Ngambeki, M. Bishop, and S. Belcher, “Teach the hands, train the mind ... a secure programming clinic,” 2015.

[4] K. L. Nance, B. N. Hay, and M. Bishop, “Secure coding education: Are we making progress?” 2012.

[5] S. Chung, L. Hansel, Y. Bai, E. Moore, C. Taylor, M. E. Crosby, R. S. Heller, and B. Endicott-Popovsky, “What approaches work best for teaching secure coding practices,” 2014.

[6] C. Banerjee and S. K. Pandey, “Research on software security awareness: Problems and prospects,” SIGSOFT Softw. Eng. Notes, vol. 35, no. 5, p.1–5, Oct. 2010. [Online]. Available: https://doi-org.ezproxy.rollins.edu/10.1145/1838687.1838701

[7] S.-F. Wen and B. Katt, “Learning software security in context: An evaluation in open source software development environment,” Proceedings of the 14th International Conference on Availability, Reliability and Security, 2019.

[8] X. Yuan, L. Yang, B. Jones, H. Yu, and B. tseng Chu, “Secure software engineering education: Knowledge area, curriculum and resources,” 2016.

[9] A. f. C. M. A. Joint Task Force on Computing Curricula and I. C.Society, Computer Science Curricula 2013: Curriculum Guidelines for Undergraduate Degree Programs in Computer Science. New York, NY, USA: Association for Computing Machinery, 2013.

[10] V. Pournaghshband, “Incorporating the security mindset into introductory programming courses,” International journal of advanced computer science, vol. 5, 2015.

[11] K. Frazer, “Building secure software: how to avoid security problems the right way,” ACM SIGSOFT Softw. Eng. Notes, vol. 27, pp. 71–72, 2002.

[12] G. McGraw, “Building secure software: better than protecting bad software,” IEEE Software, vol. 19, pp. 57–58, 2002.

[13] J. A. Whittaker, “Why secure applications are difficult to write,” IEEE Secur. Priv., vol. 1, pp. 81–83, 2003.

[14] B. Taylor, M. Bishop, D. L. Burley, S. Cooper, R. C. Dodge, and R. C. Seacord, “Teaching secure coding: report from summit on education in secure software,” in SIGCSE ’12, 2012.

[15] M. Bishop, “Learning and experience in computer security education (invited paper),” 2013.

[16] H. Yu, N. Jones, G. Bullock, and X. Yuan, “Teaching secure software engineering: Writing secure code,” 2011 7th Central and Eastern European Software Engineering Conference (CEE-SECR), pp. 1–5, 2011.

[17] S. Chung and B. Endicott-Popovsky, “Software reengineering based security teaching,” 2010.

[18] B. Taylor and S. Azadegan, “Threading secure coding principles and risk analysis into the undergraduate computer science and information systems curriculum,” in InfoSecCD ’06, 2006.

[19] M. A. Talib, A. Khelifi, and L. Jololian, “Secure software engineering: A new teaching perspective based on the swebok,” Interdisciplinary Journal of Information, Knowledge, and Management, vol. 5, pp. 083–099, 2010.

[20] M. L. Stamat and J. W. Humphries, “Training 6= education: putting secure software engineering back in the classroom,” western canadian conference on computing education, 2009.

[21] N. Jones, Q. Yu, K. Schell, and H. Yu, “Teaching secure program design,” 2019.

[22] H. Kim, N. Meghanathan, and L. Moore, “Enhancement of an undergraduate software engineering course by infusing security lecture modules,” pp. 265–269, 01 2013.

[23] S. Chung and B. Endicott-Popovsky, “Software reengineering based security teaching,” in Proceedings of the 7th Annual International Conference on International Conference on Cybernetics and Information Technologies, Systems and Applications (CITSA 2010). Orlando, FL, 2010.

[24] B. Taylor and S. Kaza, “Security injections@towson: Integrating secure coding into introductory computer science courses,” ACM Trans. Comput. Educ., vol. 16, pp. 16:1–16:20, 2016.

[25] K. Williams, X. Yuan, H. Yu, and K. S. Bryant, “Teaching secure coding for beginning programmers,” Journal of Computing Sciences in Colleges, vol. 29, pp. 91–99, 2014.

[26] E. B. Fern´andez, S. Huang, and M. Larrondo-Petrie, “A set of courses for teaching secure software development,” 19th Conference on Software Engineering Education and Training Workshops (CSEETW’06), pp. 23–23, 2006.

[27] B. Taylor, M. Bishop, E. Hawthorne, and K. Nance, “Teaching secure coding: the myths and the realities,” 03 2013, pp. 281–282.

[28] K. Qian, D. C.-T. Lo, R. M. Parizi, F. Wu, E. O. Agu, and B. tseng Chu, “Authentic learning secure software development (ssd) in computing education,” 2018 IEEE Frontiers in Education Conference (FIE), pp. 1–9, 2018.

[29] K. L. Nance, “Teach them when they aren’t looking: Introducing security in cs1,” IEEE Security & Privacy, vol. 7, 2009.

[30] S. A. Ambrose, M. W. Bridges, M. Dipietro, M. C. Lovett, and M. K. Norman, How learning works: Seven research-based principles for smart teaching. Tantor Audio, 2021.

AUTHOR

Rochelle Elva is an Assistant Professor of Computer Science at Rollins College in Florida, USA. Her research interests are Software Quality Assurance, Software Security, Security-Aware Programming, The Personal Software Process, and Computer Science Education.

AUTHOR INDEX