# Life in Software Engineering Conferences: A Text Mining Study

George Mathew
Computer Science, NCSU
Raleigh, North Carolina
george.meg91@gmail.com

Tim Menzies
Computer Science, NCSU
Raleigh, North Carolina
tim.menzies@gmail.com

Amritanshu Agrawal
Computer Science, NCSU
Raleigh, North Carolina
aagrawa8@ncsu.edu

## ABSTRACT

This paper aims to examine the trends and patterns in Software Engineering(SE) conferences. From 1993 to 2013 there have been more than 9000 papers published in top 11 Software Engineering(SE) conferences in numerous topics and with varying trends which naturally leads to their study. We model SE papers from the last two decades using topic modelling to identify prominent topics in software engineering over the last 20 years and study (a) Author publishing trends (b) Diversity between conferences (c) Evolution of conferences (d) Existence of PC bias & (e) Evolution of SE. This study will aid future researchers in selecting the SE sections that call for new research and the approach to be taken to successfully publish in SE.

## Keywords

Software Engineering; Bibliometrics; Topic Modelling; Text Mining

## 1. INTRODUCTION

Software Engineering Conferences are a very integral part of the research ecosystem. Apart from being a platform that help the inception of new topics in SE, they also help numerous researchers showcase their works and build their research career. That said, not many researchers have analyzed how SE conferences has changed over time and what does it take for researchers to have a successful life publishing in SE conferences. Considering that close to 10000 papers from over 2300 different authors have been published in top SE conferences, there must exist some pattern and trends which can be used to study the characteristics of SE conferences.

The naive approach to address this study would be via bibliometrics. Bibliometrics studies in SE have focused in the following areas; (a) generating ranking lists of top performing institutions and scholars [10], (b) citation analysis to identify the most popular articles [27], and (c) content analysis of SE research [3, 4, 11]. Papers in area (a) can mainly be used internally within the SE research community. Papers on areas (b) and (c) can be used to explain our science to outsiders, e.g. to funding authorities or to scientists representing other disciplines. Additionally, such works can be helpful in teaching students about software engineering research or to highlight the top areas under study to industry, and help outsider to get acquainted with the latest research trends. Thus, bibliometrics papers can be important aid in distributing knowledge beyond the software engineering community. But, there is a need for identifying the patterns in the papers published in SE and how it has changed over the years. This can directly aid future researchers in selecting the SE sections that call for new research and the approach to be taken to successfully publish in SE.

Thus to address the deficits of the bibliometric based studies we take a topic modelling based approach. Topic Modelling was used for a similar study in Computer Science publications [13] and more recently using just the titles of SE papers for modelling [8]. The promising results and the novelty of the method in modelling text based data prompts us to take this approach.

To study the characteristics of SE conferences we propose the following research questions:

- **RQ1: What research career is more rewarded?**
  Do authors who publish in multiple conferences of diverse topics or do they publish in select conferences with specific conferences?

**Result 1**
*We find that most authors publish in around half of all the topics and only very few authors venture into all the topics in software engineering.*

- **RQ2: Are conferences diverse?**
  Do most conferences in SE cover similar sets of topics? or Are different conferences specific to different topics?

**Result 2**
*80% papers from all conferences are from 60% of the conferences. i.e Papers from a limited set of topics are mostly published in all the conferences.*

- **RQ3: Do conferences evolve?**
  Does the focus of a conference change over time? or Does a conference tend to focus on similar topics?

*WOODSTOCK '97 El Paso, Texas USA*
© 2016 ACM. ISBN 123-4567-24-567/08/06...$15.00
DOI: 10.475/123_4

**Result 3**
*Most conferences change over time but there are conferences that prefer concentrating on a niche set of topics*

- **RQ4: Is there a program committee bias?**
  What percentage of papers published in a conference have authors in the program committee(PC) of the conference?

**Result 4**
*Most conferences have around 30% average PC bias; Few conferences have around 50% bias; A rare conference has less than 15% bias.*

- **RQ5: Do topics evolve?**
  What percentage of papers published in a conference have authors in the program committee(PC) of the conference?

**Result 5**
*From papers in top 11 conferences of SE, there is a change in the topic of focus almost every 3-4 years.*

The rest of the paper is organized as follows; §2 summarizes similar work. §3 describes the approach taken to conduct our study and how we harnessed data for it. §4 shows the results to our proposed research questions. §5 discusses the takeaways from our research questions and finally §6 presents any threats to the validity of our experiments.

## 2. RELATED WORK

Numerous bibliometric studies analyzing SE conferences have been published in SE, most of which have been in the last 2 decades. This section discusses few of the prominent works with their notable findings.

Cai & Card [3] analyzed 691 papers from 7 leading journals SE and 7 leading conferences SE in the year 2006. Among their findings was that 89% of conference papers focus on 20% of subjects in SE, including software/program verification, testing and debugging, and design tools and techniques. Moreover, the average number of 7 top international conferences in SE references cited by a conference paper is 24. A large scale study using 70,000 SE articles from DBLP[1] was performed by Fernandes [5] in 2014. He provides evidence that the number of authors of articles in software engineering is increasing on average around ?0.40 authors/decade. The results also indicate that until 1980, the majority of the articles have a sole author, while nowadays articles with 3 or 4 authors represent almost half of the total. Garousi & Vahid in 2014 presents a bibliometric analysis of the Turkish SE community (researchers and institutions) based on the Scopus academic search engine. Based on their study they find that there was a lack of diversity in the general SE spectrum(for eg. limited focus on requirements engineering, software maintenance and evolution, and architecture). They also identify a low involvement from the industry in SE. Although this research was conducted in only in Turkish SE community, this study can be extended to identify similar patterns in the entire SE community. Similar study was conducted early in 2010 on the Canadian SE community [9]. More recently, a citation based study was conducted by Garousi and Fernandes to identify the top cited paper in SE community [7]. The authors use two metrics: total number of citations and average annual number of citations to identify the top paper. The authors also try characterizing the overall citation landscape in SE motive and hope that this method encourages further discussions in the SE community towards further analysis and formal characterization of the highly-cited SE papers. Although the authors encourage an important need in SE of characterization, the approach using citations has other alternatives.

In Computer Science(CS), a 2013 paper by Hoonlor et al.highlighted the prominent trends in CS [13].This paper identified trends, bursty topics, and interesting inter-relationships between the American National Science Foundation (NSF) awards and CS publications, finding, for example, that if an uncommonly high frequency of a specific topic is observed in publications, the funding for this topic is usually increased. The authors adopted a Term Frequency Inverse Document Frequency(TFIDF) based approach to identify trends and topics. A similar approach can be performed in SE considering how closely CS is related to SE.

Garousi and Mantyla recently have adopted a Topic Modeling and Word Clustering based approach to identify topics and trends in SE [8] similar to the research by Hoonloor et al. [13]. Although their method is very novel and in line with the current state of the art, they use only the titles of the papers for modelling topics. This might lead to inaccurate topics as titles are generally not very descriptive of the field the paper is trying to explore. This issue is addressed in the current work where we use the abstracts of the paper to build topic models.

Ren & Taylor in 2007 assessed both academic and industrial research institutions, along with their scholars to identify the top ranking organizations and individuals [21]. They provide an automatic and versatile framework using electronic bibliographic data to support such rankings which produces comparable results as those from manual processes. This method although saves labor for evaluators and allow for more flexible policy choices, the method does not provide a spectrum of the topics and the publication trends in SE. More recently in 2013, Hummel et al.analyzes the expressiveness of modern citation analysis approaches like h-index and g-index in SE by analyzing the work of almost 700 researchers in SE [14]. They conclude that on an average h-index for a top author is around 60 and g-index is about 130. The authors however do not shed light on the publication pattern of these top authors and in §4.1 we analyze this using a topic modelling based bibliometric study.

Since, SE conferences are the focus of research in this paper, a natural question would be if there is a bias from the Program Committee(PC). The PC of any conference is the authority comprising of the best researchers in the program who are responsible for selecting among submissions a subset of papers that are believed to be most adequate for presentation in that conference. Systä et al.in their 2012 publication “Inbreeding in Software Engineering” [23] analyzes acceptance of papers (co)authored by PC members in 6 leading SE conferences. They conclude that there is a lot of variance in the acceptance rate(0-70%) of papers (co)authored by PC members. Although their research methodology is very insightful, the data used in their study is limited to 6

[1] http://dblp.de

conferences over a period of 5 years. Thus, in this study, in §4.4 we expand on their research using data from 11 conferences over 20 years to draw a clearer conclusion if there is a bias from the PC and if this bias is constant or changes over time.

# 3. DATASET & RESEARCH METHOD

## 3.1 Gathering Data

For studying and analyzing SE conferences we use a database of 11 conferences and its program committee members from 1993-2013. This data was obtained in the form of a sql dump from the work of Vasilescu et al.“A historical dataset of software engineering conferences” in MSR 2013 [26] and is shown in Table 1. The first column in the table represents the acronym for the conference which will be used throughout the paper; the second column shows the name of the paper and the third column shows the first edition of the conference in the available dataset.

| Short | Name | Start |
|---|---|---|
| ASE | IEEE/ACM International Conference on Automated Software Engineering | 1994 |
| CSMR | European Conference on Software Maintenance and Reengineering | 1997 |
| FASE | International Conference on Fundamental Approaches to Software Engineering | 1998 |
| FSE | ACM SIGSOFT Symposium on the Foundations of Software Engineering | 1993 |
| GPCE | Generative Programming and Component Engineering | 2000 |
| ICPC | IEEE International Conference on Program Comprehension | 1997 |
| ICSE | International Conference on Software Engineering | 1994 |
| ICSM | IEEE International Conference on Software Maintenance | 1994 |
| MSR | Working Conference on Mining Software Repositories | 2004 |
| SCAM | International Working Conference on Source Code Analysis & Manipulation | 2001 |
| WCRE | Working Conference on Reverse Engineering | 1995 |

Table 1: Conference Names with their Acronyms

The data published by Vasilescu et al.does not contain the abstract or other sections of the papers which can be used to perform text based analytics. Thus, we join the data from the SQL tables with the abstracts of all papers from the Association of Computing Machinery(ACM) harvested by Tang et al.on “aminer” [24][2].

## 3.2 Latent Dirichlet Allocation

Latent Dirichlet Allocation (LDA) is a generative statistical model that allows sets of observations to be explained by unobserved groups that explain why some parts of the data are similar. The basic LDA model [2,17] is shown in Figure 1. A collection of D documents is assumed to contain $T$ topics expressed with $W$ different words. Each document $d \in D$ of length $N_d$ is modelled as a discrete distribution $\theta(d)$ over the set of topics. Each topic corresponds to a multinomial distribution over the words. Discrte priors $\alpha$ are assigned to the distribution of topics vectors $\theta$ and $\beta$ for the distibutions of words in topics. In Figure 1, the outer plate spans documents and the inner plate spans word instances in each document (so the $w$ node denotes the observed word at the instance and the $z$ node denotes its topic). The inference problem in LDA is to find hidden topic variables $z$, a vector spanning all instances of all words in the dataset. LDA is a problem of Bayesian inference. The original method used is a variational Bayes approximation of the posterior distribution [2], alternative inference techniques use Gibbs sampling [12] and expectation propagation [16].

LDA learns the various distributions (the set of topics, their associated word probabilities, the topic of each word, and the particular topic mixture of each document).To find these various distributions, LDA takes particular set of parameters namely, number of clusters ($k$), number of iterations ($n$), document topic prior ($\alpha$), and word topic prior ($\beta$). Literature suggests that different sets of these parameters can give rise to stable and unstable topics. To overcome this problem, we tune the above mentioned parameters of LDA using Differential Evolution (DE) [22]. Tuning the parameters and different configurations were used by [15, 18, 19]. They [6, 25] achieved higher stability by just increasing the number of cluster size.

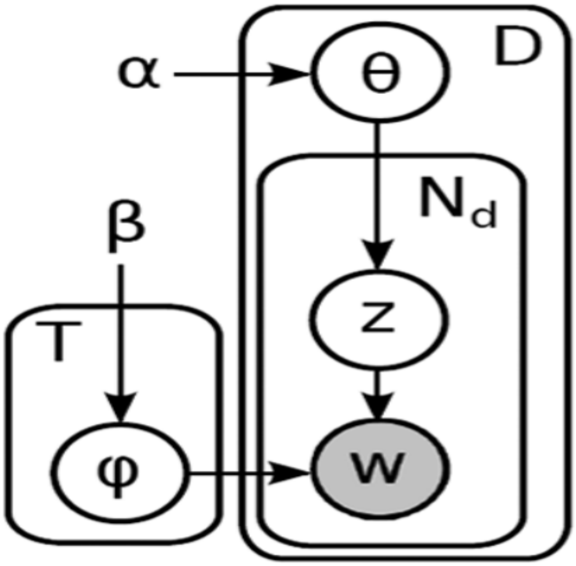

Figure 1: LDA Model

From the flowchart given in Figure 2, we can see how DE works. A small pseudocode is given in Algorithm 1. Each box is shown with the Line numbers which correspond to pseudocode in ALgorithm 1. It starts by generating an initial population of candidate solutions (here LDA parameters). These solutions are moved around in the search space by using simple mathematical mutation, crossover and selection operators to combine the positions of existing solutions from the population. If the new position of a solution is an improvement (some fitness value) it is accepted and forms part of the population, otherwise the new position is simply discarded. The process is repeated till we reach a stopping criteria and by doing so a satisfactory solution (parameters)

[2] http://aminer.org

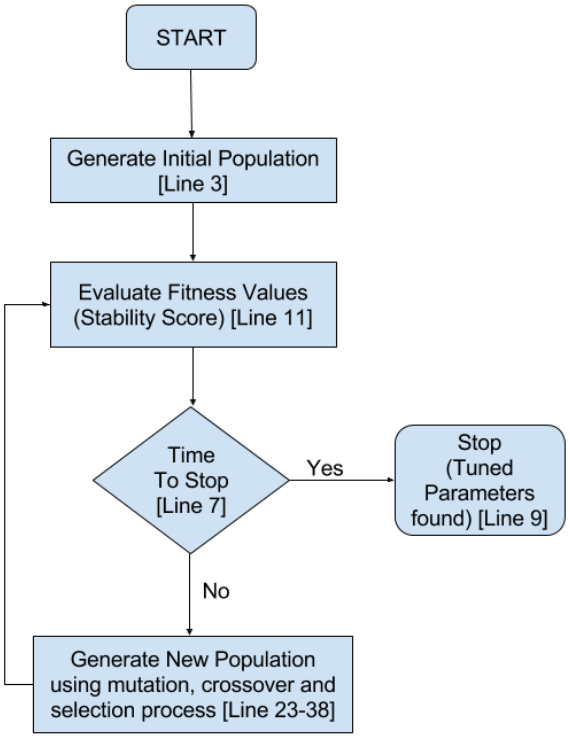
START
Generate Initial Population [Line 3]
Evaluate Fitness Values (Stability Score) [Line 11]
Time To Stop [Line 7]
Yes
Stop (Tuned Parameters found) [Line 9]
No
Generate New Population using mutation, crossover and selection process [Line 23-38]

Figure 2: Flowchart of DE

is discovered.

The fitness value is measure similar to the Jaccard Similarity. We say topics are stable, when there are x times n% of terms overlap in a topic across m runs of LDA with the same parameters. We take median of all these scores.

## 4. STUDIES

In order to answer the research questions we adopt a topic modeling based approach coupled with a bibliometric based study(§3.2) on the dataset described in §3.1. Topic modeling using LDA needs three parameters to be set primarily; 1) #: Number of topics 2) $\alpha$ : Dirichlet prior on the per-document topic distributions 3) $\beta$: Dirichlet prior on the per-topic word distribution.

In similar studies [8], the authors suggest using 67 topics and default values of $\alpha$ and $\beta$ as provided by the LDA package in R statistical toolkit [20]. The choice for the number of topics was taken by minimizing the number Likelihood error. We optimize these tunable parameters as described in §3.2. After tuning, the optimal values for #, $\alpha$ & $\beta$ are 12, 0.847 & 0.764 respectively. Using these parameters, LDA was performed on the abstracts(titles if abstract was not available) of the dataset. Table 2 show the 12 topics and the top 10 terms associated with each topic. This clearly gives an understanding of the broad categories of SE. For example from Table 2, Topic 0 highlights testing based papers, Topic 3 describes source code analysis based papers, Topic 10 shows bugs and defects based papers etc.

### 4.1 RQ1: What research career is more rewarding?

In this section we study the number of authors who publish papers in different topics and identifying the optimal

Table 2: Topics with top 10 terms

| # | Top 10 Terms |
|---|---|
| 0 | test, testing, cases, fault, tests, coverage, techniques, suite, generation, regression |
| 1 | applications, web, application, systems, service, security, services, aspect, business, oriented |
| 2 | patterns, pattern, anti, ibm, traits, micro, sub, 1800, assembler, catalog |
| 3 | code, source, tools, comprehension, program, tool, refactoring, programming, developers, information |
| 4 | optimal, search, spreadsheet, spreadsheets, methods, guarantee, sbse, factor, advantage, solutions |
| 5 | special, members, formal, methods, welcome, easier, conference, experiment, reasons, good |
| 6 | division, leads, consolidation, company, information, corporation, procurement, corporate, robert, vice |
| 7 | software, engineering, development, research, process, project, tools, paper, projects, work |
| 8 | program, analysis, programs, dynamic, data, technique, execution, static, approach, paper |
| 9 | model, design, software, based, systems, approach, paper, models, language, requirements |
| 10 | bug, bugs, reports, fix, report, fixing, fixed, fixes, crash, patches |
| 11 | software, code, approach, changes, paper, study, results, systems, source, analysis |

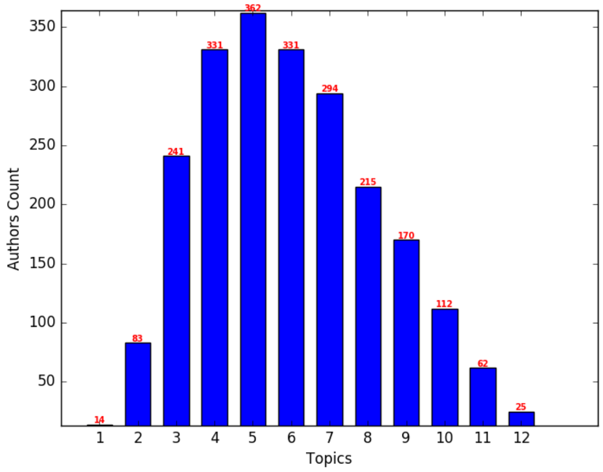

Figure 3: Number of authors vs Number of Topics Used.

**Algorithm 1** Pseudocode for DE with a constant number of evaluations

```
Input: $np = 10, f = 0.7, cr = 0.3, iter = 3, Goal \in \{Data, term, ...\}$
Output: $S_{best}, final\_generation$
function DE($np, f, cr, iter, Goal$)
    $Cur\_Gen \leftarrow \emptyset$
    $Population \leftarrow$InitializePopulation(np)
    for $i = 0$ to $np - 1$ do
        $Cur\_Gen$.append($Population$[i],score($Population$[i], $term$)
    end for
    for $i = 0$ to $iter$ do
        $NewGeneration \leftarrow \emptyset$
        for $j = 0$ to $np - 1$ do
            $S_i \leftarrow$Extrapolate($Population$[j],Population,cr,f,np)
            if score($S_i$) $\geq$ $Cur\_Gen$[j][1] then
                $NewGeneration$.append($S_i$,score($S_i$, $term$))
            else
                $NewGeneration$.append($Cur\_Gen$[j])
            end if
            $Cur\_Gen \leftarrow NewGeneration$
        end for
    end for
    $S_{best} \leftarrow$ GetBestSolution($Cur\_Gen$)
    $final\_generation \leftarrow Cur\_Gen$
    return $S_{best}, final\_generation$
end function
function EXTRAPOLATE($old, pop, cr, f, np$)
    $a, b, c \leftarrow threeOthers$(pop, old)
    $newf \leftarrow \emptyset$
    for $i = 0$ to $np - 1$ do
        if $cr \leq$ random() then
            $newf$.append($old[i]$)
        else
            if typeof($old$[i])== bool then then
                $newf$.append(not $old[i]$)
            else
                $newf$.append(trim(i,($a$[i]+$f*$($b$[i] $-$ $c$[i]))))
            end if
        end if
    end for
    return $newf$
end function
```

success strategy for publishing in software engineering conferences. Figure 3 shows the number of authors publishing in “x” number of distinct topics where “x” varies from 1 to 12 as shown on the x-axis of the graph.It is clear that from over 2240 thousand authors publishing in various software engineering conferences in the past 20 years close to 46% of the authors publish in 4-6 topics and close to 80% of the authors publish in 3-8 topics. But the question arises what percentage of the top authors fall in the 80% of authors publishing between 3-8 topics.

Figure 4 shows the results of the study which was repeated considering only the top 1% cited authors in SE. From the figure we can see that all of the top 1% authors publish between 10-12 topics. Thus, none of the top 1% authors fall into the category of authors who publish in the majority crop of authors who publish in 3-8 topics.

**Result 6**
*Successful authors publish in different topics.*

## 4.2 RQ2: Are conferences diverse?

In this section we check if most conferences are same or different. A heatmap as shown in Figure 5 shows which conferences are similar to each other. The y axis of the heatmap shows the index of the topic ranging between 0-11 and the x axis shows the conference names used in the study.The color on the heatmap indicates how strongly or weakly a topic on the y-axis is covered in a conference on x-axis.

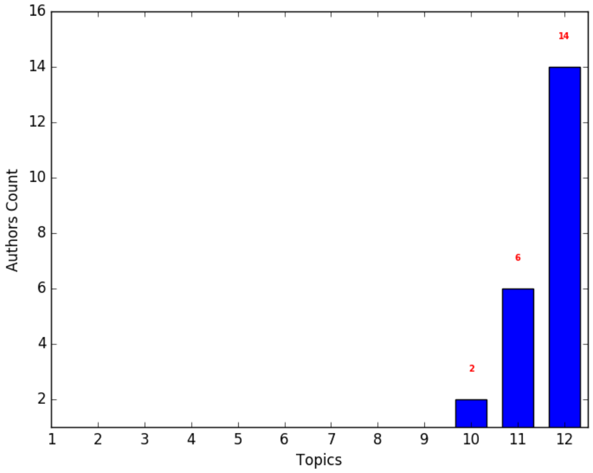

Figure 4: Number of authors vs Number of Topics Used for top 1% of the authors

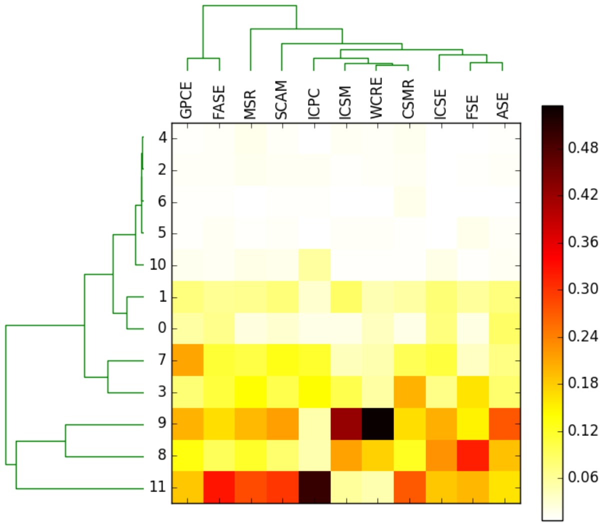

Figure 5: Heatmap showing correlation between topics and conferences.

From the hierarchical tree on the x-axis we can see that there is a strong correlation between conferences with respect to the topics covered in them. For example, ICPC, ICSM, WCRE and CSMR cover similar topics and top conferences like ICSE, FSE and ASE cover similar topics. On the other hand, conferences like GPCE and FASE are completely different from other conferences. Thus we can conclude that, there are conferences staggeringly similar to each other but there are also conferences which significantly differ in the topics published in them.

**Result 7**
*Top conferences publish papers on similar topics but there exists smaller conferences that focus on different topics.*

### 4.3 RQ3: Do conferences evolve?

In this section we examine if the topics concentrated in a conference changes over time. In Figure 6 9 top conferences ICSE, ICSM, WCRE, CMSR, MSR, ICPC, FSE, SCAM and ASE are used to study topic coverage % of papers published in them from 1993-2013. The y-axis in each figure represents a stacked bar chart in descending order indicating the topic coverage percentage in a conference(title) for a certain year(x-axis). The x-axis represents the year of the conference from 1993-2013.

Consider ICSE, in years 1998-2004 topic 7 is the most published topic and the percentage of papers published in topic 7 decreases almost consistently. Similarly topic 11 slowly gains popularity from 2007-2010 and eventually becomes the most popular topic from 2011-2013. Similar trends can be observed in ASE and FSE.But when it comes to MSR, we can see that the most popular topic has always been topic 11(expect for 2010). In WCRE on the other hand, Topic 9 has slowly moved from the most popular topic to the verge of becoming obsolete.

**Result 8**
*Most conferences evolve over time but there are conferences that prefer concentrating on a niche set of topics.*

### 4.4 RQ4: Is there a program committee bias?

In this section we check what percentage of papers published in a conference are (co)authored by members of the program committee(PC) and examine if a bias exist in paper acceptance. Table 3 shows the % of papers (co)authored my PC members per year from 1993 - 2013(columns) in eleven prominent conferences(rows). We can observe that in ICSE, FSE and ASE less than 30% of papers published have atleast one of the authors in the PC(average of 15%). Whereas in MSR, WCRE, ICPC over 40% of papers in average each year have atleast one of the authors in the PC. FASE is an exceptional case where on an average less than 15% of the papers are (co)authored by PC members.

**Result 9**
*PC Bias Ranges from 0% to 70% with mostly around 30%. Most conferences have around 30% average PC bias; Few conferences have around 50% bias; A rare conference has less than 15% bias.*

### 4.5 RQ5: Do topics evolve?

In this section we study the trend in topics of papers published from 1993-2013 and check if there is a change in affinity towards a topic every year. Figure 7 shows the stacked coverage % of top 5 topics in descending order(y-axis) in a year(x-axis) for 9281 papers from top 11 conferences over 20 years.

From the Figure 7, we can see that topic 7 was highly popular from 1998-2004 and since then the percentage of accepted papers have gradually reduced. On the contrary, topic 13 has gained popularity gradually from 2008-2013. Topic 8 is slowly gaining popularity from 2005 and based on the trend, it might become the most popular topic in which papers are published in the next 5 years.

**Result 10**
*From papers in top 11 conferences of SE, there is a change in the topic of focus almost every 3-4 years.*

## 5. CONCLUSIONS & FUTURE WORK

This study presented an explanatory analysis of publishing patterns of successful authors, diversity and evolution of conferences, program committee bias and rise & fall of topics in SE conferences.

As the trend in this paper(§4.1) all successful authors publish in large number of different topics of SE rather than few niche sectors. This naturally raises the questions, which field an author should concentrate on enhance his/her reputation? If an author is already successful, which fields should he/she venture into to continue his success? We plan to explore these questions in our subsequent works.

This paper further explores the diversity in conferences and the evolution of these conferences over the years(§4.2 & §4.3 respectively). Firstly, we see that most conferences are similar in nature. How does this similarity compare to other areas of science? Is it because there are too many lesser known venues which publish papers not seen or read by others? Is it related to the quality of papers' or venues'? Aksnes [1] discusses quality metrics and visibility metrics increase the number of citation to a given paper. Perhaps these metrics should be explored in further depth in SE literature. Secondly, we notice that focus areas of conferences change overtime and some topics are discarded as time passes and some topics become more prominent over time. A closer study can help researchers in selecting conferences more appropriate to their field of study.

We also explore if there is a bias from the PC in a conference. The bias ranges from 0-70% with an average of around 30%. Our conclusions are similar to that observed by Systa et al. [23]; Why is it easier for members of the PC to get their papers published in most conferences? The prime reason could be that PC members are probably more experienced researchers, and thus it can be assumed that they are mature scientists and thus also write better papers than average authors.

Finally, the paper studies the change in focus areas in SE over the last two decades. Some topics have gradually gained popularity over the years, while some of them are on the verge of becoming obselete. This raises the question, can we identify topics that need conferences to focus on?

## 6. THREATS TO VALIDITY

Like any study based on empirical analysis, biases during the study can affect the final results. Therefore, any conclusions made from this work must be considered with the following issues in mind:

1. **Sampling Bias:** Sampling bias threatens any classification experiment; i.e., what matters there may not be true here. For example, the data sets used here

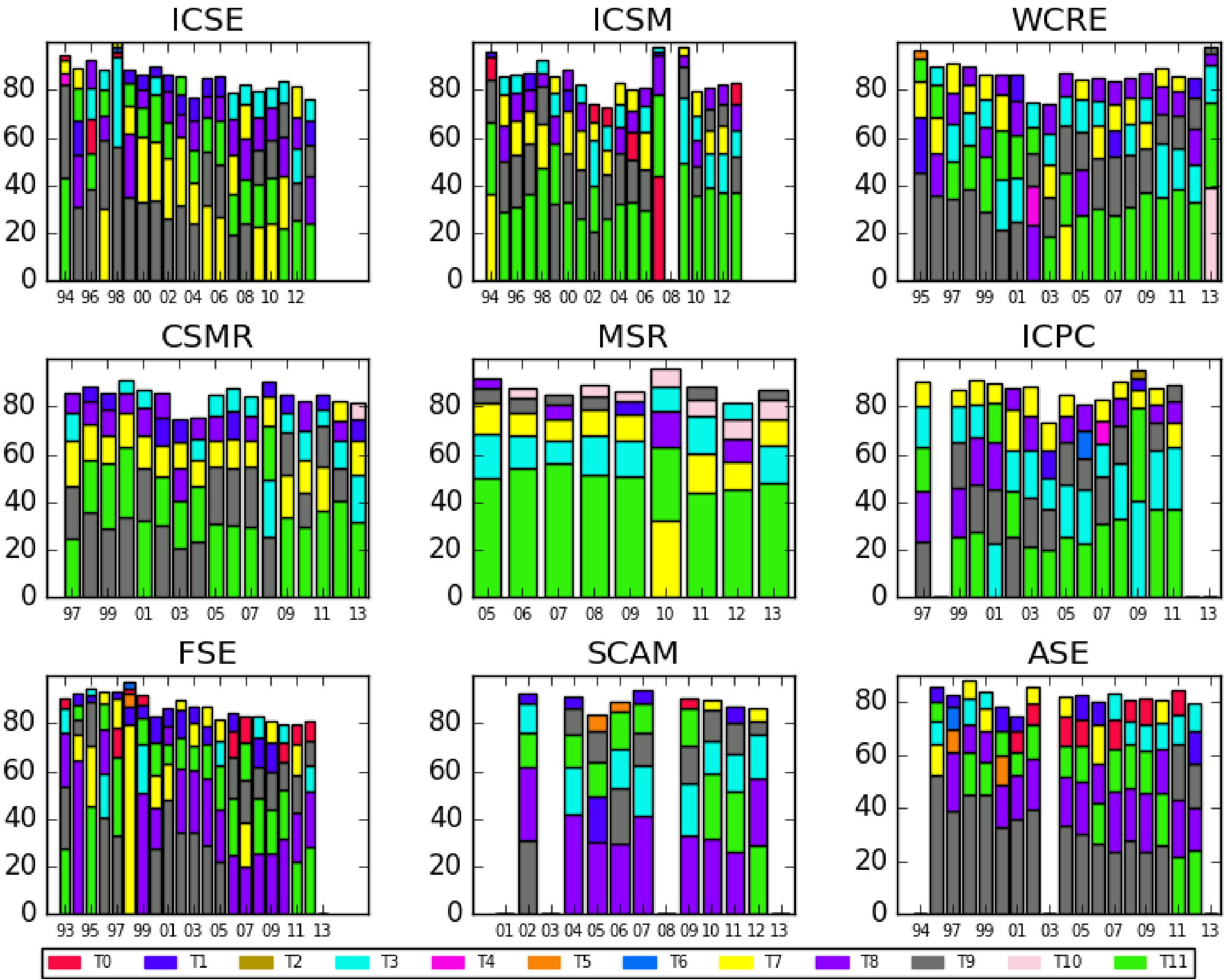

Figure 6: Topic Coverage % vs Year for top conferences from 1993-2013.

| conf | 93 | 94 | 95 | 96 | 97 | 98 | 99 | 00 | 01 | 02 | 03 | 04 | 05 | 06 | 07 | 08 | 09 | 10 | 11 | 12 | 13 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| WCRE | | | 56 | 40 | 63 | 44 | 57 | 40 | 50 | 45 | 41 | 26 | 45 | 53 | 29 | 42 | 32 | 31 | 35 | 38 | 34 |
| FSE | 11 | 18 | 6 | 23 | 15 | 4 | 10 | 24 | 15 | 28 | 9 | 21 | 12 | 28 | 13 | 23 | 21 | 11 | 12 | 20 | 25 |
| GPCE | | | | | | | | 0 | 8 | 5 | 13 | 7 | 10 | 19 | 0 | 21 | 29 | 15 | 23 | 13 | 45 |
| MSR | | | | | | | | | | | | | 32 | 54 | 39 | 43 | 37 | 45 | 44 | 50 | 44 |
| ICPC | | | | | 68 | 46 | 58 | 33 | 44 | 40 | 32 | 42 | 30 | 40 | 61 | 56 | 39 | 50 | 38 | 24 | 47 |
| ICSE | | 16 | 16 | 5 | 13 | 13 | 22 | 16 | 15 | 17 | 21 | 21 | 13 | 9 | 7 | 12 | 16 | 17 | 21 | 12 | 27 |
| ASE | | 25 | 30 | | 19 | 25 | 23 | 20 | 22 | 29 | 22 | 22 | 19 | 14 | 18 | 26 | 15 | 15 | 14 | 29 | 28 |
| ICSM | | 29 | 43 | 48 | 31 | 50 | 39 | 28 | 34 | 35 | 22 | 33 | 31 | 34 | 34 | 49 | 42 | 31 | 38 | 37 | 36 |
| SCAM | | | | | | | | | 39 | 56 | 33 | 41 | 30 | 36 | 26 | 41 | 30 | 19 | 29 | 27 | 14 |
| FASE | | | | | | 0 | 0 | 11 | 8 | 8 | 4 | 14 | 4 | 10 | 9 | 16 | 18 | 19 | 13 | 15 | 12 |
| CSMR | | | | | 14 | 16 | 21 | 30 | 33 | 25 | 19 | 14 | 25 | 46 | 36 | 31 | 39 | 45 | 37 | 29 | 41 |

Table 3: % of papers (co)authored by members of PC from 1993 - 2013

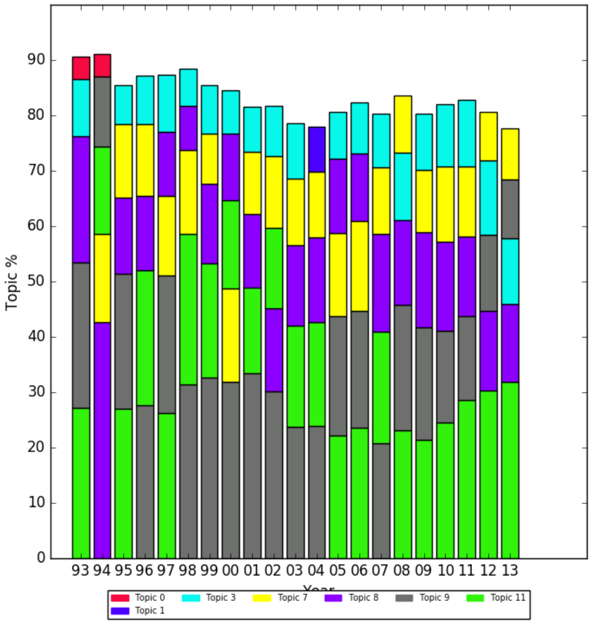

Figure 7: Topic Coverage % vs Year

spans only between 1994-2013 and is limited to top 11 software engineering conferences.

2. **Learner Bias:** For answering our research questions, we chose Latent Dirichlet Allocation since it could classify words into topics based on their association and its promising results [8, 13].

3. **Evaluation Bias:** This paper uses one measure of similarity(Jaccard Similarity) for tuning the parameters. Studying other similarity measures will be a part of future work.

4. **Order Bias:** LDA uses a random seed for building the topic model. For our experiments we fix this seed to 0 for reproducibility of the study.

## 7. ACKNOWLEDGMENTS

We would like to acknowledge B.Vasilescu et al.for providing a well documented relational database of 11 SE conferences, its program committees and authors. We would also like to thank J.Tang et al.for harnessing the abstracts from the ACM portal.

## 8. REFERENCES

[1] D. W. Aksnes. Characteristics of highly cited papers. *Research Evaluation*, 12(3):159–170, 2003.

[2] D. M. Blei, A. Y. Ng, and M. I. Jordan. Latent dirichlet allocation. *Journal of machine Learning research*, 3(Jan):993–1022, 2003.

[3] K.-Y. Cai and D. Card. An analysis of research topics in software engineering–2006. *Journal of Systems and Software*, 81(6):1051–1058, 2008.

[4] N. Coulter, I. Monarch, and S. Konda. Software engineering as seen through its research literature: A study in co-word analysis. *Journal of the American Society for Information Science*, 49(13):1206–1223, 1998.

[5] J. M. Fernandes. Authorship trends in software engineering. *Scientometrics*, 101(1):257–271, 2014.

[6] L. V. Galvis Carreño and K. Winbladh. Analysis of user comments: an approach for software requirements evolution. In *Proceedings of the 2013 International Conference on Software Engineering*, pages 582–591. IEEE Press, 2013.

[7] V. Garousi and J. M. Fernandes. Highly-cited papers in software engineering: The top-100. *Information and Software Technology*, 71:108–128, 2016.

[8] V. Garousi and M. V. Mäntylä. Citations, research topics and active countries in software engineering: A bibliometrics study. *Computer Science Review*, 19:56–77, 2016.

[9] V. Garousi and T. Varma. A bibliometric assessment of canadian software engineering scholars and institutions (1996-2006). *Computer and Information Science*, 3(2):19, 2010.

[10] R. L. Glass and T. Y. Chen. An assessment of systems and software engineering scholars and institutions (1999–2003). *Journal of Systems and Software*, 76(1):91–97, 2005.

[11] R. L. Glass, I. Vessey, and V. Ramesh. Research in software engineering: an analysis of the literature. *Information and Software technology*, 44(8):491–506, 2002.

[12] T. L. Griffiths and M. Steyvers. Finding scientific topics. *Proceedings of the National academy of Sciences*, 101(suppl 1):5228–5235, 2004.

[13] A. Hoonlor, B. K. Szymanski, and M. J. Zaki. Trends in computer science research. *Communications of the ACM*, 56(10):74–83, 2013.

[14] O. Hummel, A. Gerhart, and B. Schäfer. Analyzing citation frequencies of leading software engineering scholars. *Computer and Information Science*, 6(1):1, 2013.

[15] S. Lohar, S. Amornborvornwong, A. Zisman, and J. Cleland-Huang. Improving trace accuracy through data-driven configuration and composition of tracing features. In *Proceedings of the 2013 9th Joint Meeting on Foundations of Software Engineering*, pages 378–388. ACM, 2013.

[16] T. Minka and J. Lafferty. Expectation-propagation for the generative aspect model. In *Proceedings of the Eighteenth conference on Uncertainty in artificial intelligence*, pages 352–359. Morgan Kaufmann Publishers Inc., 2002.

[17] S. I. Nikolenko, S. Koltcov, and O. Koltsova. Topic modelling for qualitative studies. *Journal of Information Science*, page 0165551515617393, 2015.

[18] R. Oliveto, M. Gethers, D. Poshyvanyk, and A. De Lucia. On the equivalence of information retrieval methods for automated traceability link recovery. In *Program Comprehension (ICPC), 2010 IEEE 18th International Conference on*, pages 68–71. IEEE, 2010.

[19] A. Panichella, B. Dit, R. Oliveto, M. Di Penta, D. Poshyvanyk, and A. De Lucia. How to effectively use topic models for software engineering tasks? an

approach based on genetic algorithms. In *Proceedings of the 2013 International Conference on Software Engineering*, pages 522–531. IEEE Press, 2013.

[20] M. Ponweiser. Latent dirichlet allocation in r. 2012.

[21] J. Ren and R. N. Taylor. Automatic and versatile publications ranking for research institutions and scholars. *Communications of the ACM*, 50(6):81–85, 2007.

[22] R. Storn and K. Price. Differential evolution–a simple and efficient heuristic for global optimization over continuous spaces. *Journal of global optimization*, 11(4):341–359, 1997.

[23] T. Systä, M. Harsu, and K. Koskimies. Inbreeding in software engineering conferences, 2012.

[24] J. Tang, J. Zhang, L. Yao, J. Li, L. Zhang, and Z. Su. Arnetminer: Extraction and mining of academic social networks. In *KDD'08*, pages 990–998, 2008.

[25] K. Tian, M. Revelle, and D. Poshyvanyk. Using latent dirichlet allocation for automatic categorization of software. In *Mining Software Repositories, 2009. MSR'09. 6th IEEE International Working Conference on*, pages 163–166. IEEE, 2009.

[26] B. Vasilescu, A. Serebrenik, and T. Mens. A historical dataset of software engineering conferences. In *Proceedings of the 10th Working Conference on Mining Software Repositories*, pages 373–376. IEEE Press, 2013.

[27] C. Wohlin. An analysis of the most cited articles in software engineering journals-2000. *Information and Software Technology*, 49(1):2–11, 2007.