Opening Sentences in Academic Writing

How Security Researchers Defeat the Blinking Cursor

Didem Demirag
Concordia University
Montreal, QC, Canada
d_demira@encs.concordia.ca

Jeremy Clark
Concordia University
Montreal, QC, Canada
j.clark@concordia.ca

ABSTRACT

Traditionally, education in computer science focuses on stakeholders like teachers, undergraduate students, and employers. However researchers also educate themselves about recent results and new subject matters. An important vehicle in this informal, self-education process is reading peer-reviewed academic papers—papers that are also used in the curriculum of graduate-level research courses. Technical writing skills are important in this domain, as well as engaging the reader with interesting text. This paper is a study of academic writing. We study in depth the first sentence used by researchers in opening their academic papers and how this sentence operates to draw the reader in. We use a corpus of 379 papers from a top-tier cybersecurity conference and use qualitative analysis (coding from grounded theory) to create a taxonomy of 5 general types and 14 sub-types of opening sentences. In this paper, we define and illustrate each type through examples, and reflect on what we learned about writing after examining all of these sentences.

CCS CONCEPTS

• **Social and professional topics → Informal education**; **Computational thinking**.

KEYWORDS

Scientific Writing; Education; Cybersecurity

ACM Reference Format:
Didem Demirag and Jeremy Clark. 2022. Opening Sentences in Academic Writing: How Security Researchers Defeat the Blinking Cursor. In *Proceedings of the 53rd ACM Technical Symposium on Computer Science Education V. 1 (SIGCSE 2022), March 3–5, 2022, Providence, RI, USA.* ACM, New York, NY, USA, 7 pages. https://doi.org/10.1145/3478431.3499378

1 INTRODUCTORY REMARKS

What makes the writing style of an academic paper stand out? Strong technical writing is partially founded on *SIGCSE* research dating back to the 1990s on how to move writing from the English department into computer science [27, 36, 55, 62]. Technical writing is now considered a necessary communication skill in our curriculum (*cf. ACM/IEEE Computing Curricula 2020* [15, 16]).

In a departure from many *SIGCSE* papers, educators in our work are not teachers in a classroom but are authors of academic papers who transfer knowledge to their readers. A well-written paper can reach a wide audience beyond conference attendees and others who benefit from direct communication with the authors. Further, academic papers are used in the classroom, particularly in courses offered to graduate students. For a paper to be effective, it must first catch the attention of the course instructor and then engage the interest of the students. While most researchers are, or were, students, and writing is included in modern curricula, writing research papers is an advanced form of writing considered too ambitious for teaching to undergraduates [30]. Therefore, crafting academic papers is often self-taught and/or passed from supervisors to students through mentorship [48].

Writing tools and systems help produce competent text, but tend not to enhance elegance and style. If writing style is difficult to teach and analyze, what can be done? In this paper, we build a 'zoo' of writing samples (conceptualized by Miro [48] at *SIGCSE'11*), placed in a taxonomy we develop, to exhibit different writing styles for writers and readers to study and learn from. The specific lessons drawn from viewing our zoo are meant to be subjective, and depend on the viewer's personal context. A secondary contribution of our paper is our methodology for building a zoo. We hope to see this applied to other domains of writing.

Our methodology is empirical and positive. To illustrate what we mean, consider a research paper that designs and tests a pedological tool in an educational setting. By contrast, an *empirical* paper might survey a set of courses from around the world. Often before a *normative* approach (*i.e.,* what ought to be done) can be formulated, it is instructive to first consider a *positive* approach (what is being done). Our zoo is not a curation of 'good' writing samples (beyond being acceptable for publication in a top-tier conference) but offers a large set of samples that are carefully organized by our interpretation of what the writers intended to convey.

The Opening Sentence. We believe it is fruitful to study the writing style of academic computer science articles at different levels of granularity. As a general trade-off, short writing samples enable the study of a large number of samples, while longer writing samples enable a broader representation of the writing style. In this work, we choose to look at a large number of very short samples—each only a single sentence. In choosing which sentence to study from each paper, the first sentence is an intuitive candidate. The opening sentence of a paper needs to be bold, convey the importance of the subject of the paper, and hook the reader. The novelist Stephen

SIGCSE 2022, March 3–5, 2022, Providence, RI, USA.

ACM ISBN 978-1-4503-9070-5/22/03…$15.00
https://doi.org/10.1145/3478431.3499378

King is said to spend "months or even years" writing an opening sentence [25]. The academic Steven Pinker notes that,

> "Good writing starts strong. Not with a cliché ('Since the dawn of time'), not with a banality ('Recently, scholars have been increasingly concerned with the question of . . . '), but with a contentful observation that provokes curiosity." [56]

In this paper, we study the opening sentences of 379 papers from the field of information systems security. We use a variant of grounded theory [31] to classify the opening sentences of each of these papers according to what the sentence is doing to advance an argument and engage with the reader. For example, a paper might start with a historical fact, argue the importance of the subject, tell a story, or open with a question (as this paper itself does). We develop 5 main categories, with 14 sub-types in total. We provide many examples of each and a guide to distinguishing them—an opening sentence 'zoo.' Such a set of examples, called a zoo by Miro in *SIGCSE'11*, is suggested for students to develop style and taste (beyond competence) in writing [48].

Source Data. Before initiating our work, we were unsure if it would be possible to understand the opening sentences of academic papers without some domain knowledge of the field (we discuss our conclusions on this in Section 4). Thus we made the initial decision to look at papers from our own research area: security.

The field of security has hundreds of conferences and workshops but is recognized as having four top conferences (as ordered within a calendar year): *The Network and Distributed System Security Symposium (NDSS)*, *IEEE Symposium on Security and Privacy*, *USENIX Security Symposium*, and *ACM Conference on Computer and Communications Security (CCS)*. These conferences have considerable overlapping authors[1] and program committee members, and we hypothesize (but did not test) that the analysis would be invariant to which conference the papers came from.

We selected *USENIX Security*, which has always offered its proceedings as open access, allowing us to side-stepping any potential copyright issues with opening our dataset and work. It also offers its proceedings in formats that were useful for our project: *i.e.,* all papers in a single file; and `epub` format in addition to `pdf`. We used three consecutive years (2014–2016) for a total of 379 papers. We analysed every opening sentence from the main body of the paper (as opposed to the abstract, which we view as serving an archival role for the paper).

Open Access. Our raw data, the full set of original papers, and the full set of our codes are available as an NVIVO database on a public GitHub repository.[2]

2 PRELIMINARIES AND RELATED WORK

Opening Sentences. Our idea to examine opening sentences came from Pinker's style guide [56], which is devoted in part to improving technical, scientific, and academic writing. Pinker also devotes a section of the book to the role of the opening sentences of a work (illustrating with popular non-fiction books). When we reflected on our own difficulty with 'getting a paper going,' we conceived of this project. To our knowledge, opening sentences have not been systematically categorized before. In an older paper, King looks at opening sentences in medical research [38]. The article provides stylistic advice—draw the readers' attention, be concise and clear about stating the main theme of the paper—and he gives examples from medical writing and explains different ways to simplify overcomplicated sentences by shortening them. By contrast, our paper is not normative at all (it is not on how to improve an opening sentence) but is instead a neutral classification of how others have decided to open their papers, looking for trends and the variations in approach.

Academic Writing. Miro at *SIGCSE'11* presents a set of falsifiable writing principles to make good writing similar to engineering [48]. For example, principles at the granularity of a single sentence include: 'put known information at the beginning of a sentence and new, important information at the end;' 'sentences begin and end well;' and 'each sentence exposes only one idea' (note these are for any sentence, and not restricted to an opening sentence). However Miro considers style—personality and enjoyment—as beyond these principles. Instead, he suggests building a 'zoo' of writing exemplars for students to study, process, and reflect on. Our paper strives to build such a zoo for opening sentences.

Côté and Custeau [18] describe in the *SIGCSE Bulletin* an evaluation method for grading a student's article (part of a pedagogical exercise they are testing) along 23 criteria: *e.g.,* 'the topic is pertinent, original, and interesting;' 'redundancy adds to clarity;' and 'references are made in the text to illustrations.' In our work, we choose one of their criteria for in-depth study: '**the introduction draws attention and is interesting to read**.' Future work might adapt our methodology for studying other criteria.

Grounded Theory. Grounded theory is an analysis method for qualitative data [31]. In grounded theory, one or more practitioners will examine the data to divisions between different concepts. The data is then partitioned at these points and concepts are labeled with a code. By performing coding, the aim is to come up with new high-level theories and concepts at the end of the process. Coding is an iterative process and several rounds of coding can be performed to refine the categories. At the end of the process, a new theory that is based on the data is presented.

3 CATEGORIZATION

Figure 1 shows the taxonomy of our codes where the area of the box is proportional to how many times they were used. We provide a description of each category, as well as several examples.

Methodology. To arrive at this figure, we followed the coding techniques of grounded theory assisted by the qualitative analysis software tool NVIVO. To start, one of us partitioned out the opening sentence of each paper, which also anonymized the writing samples. Together, we read dozens of sentences without coding and establish a guiding principle: to code what the authors are trying accomplish; not just the content, but the intension of the sentence.

For the first year of proceedings, both of us 'open coded' the papers together to establish an initial codebook. For the following two years of proceedings, one of us coded and the other reviewed, discussing and resolving disagreements together. If more than one

[1] System Security Circus 2020: http://s3.eurecom.fr/~balzarot/notes/top4
[2] GitHub: https://github.com/MadibaGroup/2020-Opening

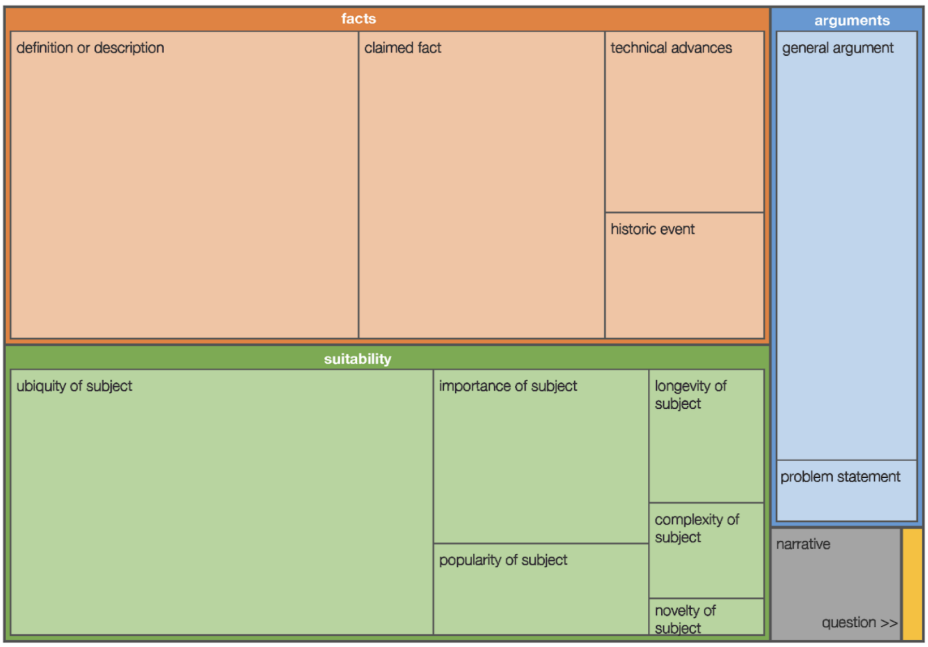

Figure 1: A treemap of our codes for the opening sentences of 379 research papers. The size (in area) of each visualized code is proportional to the number of sentences with that code.

code applied to a sentence, we sought the best fit and avoided coding a sentence with more than one code.

Periodically, we reviewed the codebook itself together. First, we would look at the codes themselves (as opposed to the sentences) for relationships ('axial coding'). Second, we looked at all the samples within each specific code to see if they were coherent and to re-code or further sub-divided as necessary ('code upon').

We halted after coding the third year of proceedings, as we determined that the newest samples did not substantially modify our codebook (*i.e.,* in terms of sub-categorizing, reclassifying or creating new codes) or the proportion of samples in each code. We did not measure this explicitly but relied on our judgement. The final step of grounded theory is to pull general theories out of the codes and their categorization, which we did not pursue since our end goal was classification. Thus we use the coding techniques of grounded theory rather than exactly grounded theory itself.

Replication, Validity & Extension. A replication of our study (starting with the same writing samples) has some potential of producing different results. Starting with the codebook itself, we expect the codes would be named differently, the spectrum between objective fact and subjective argument can be sub-divided differently, and the fidelity of sub-categorization of codes might also differ. However we expect the highest-level intentions (facts, arguments, narratives, *etc.*) should be conveyed in any open coding of the samples, and should a study start with our codebook, we are confident the classification of the samples themselves would not deviate significantly.

We encourage researchers to use our methodology to build zoo's for other writing domains. For close domains (*e.g.,* other CS disciplines), we would advise skipping the open coding step, and use our codebook directly. It is quite general and appears applicable to different forms of persuasive writing. Of course, during coding, researchers should revisit the codebook and use axial coding and code upon techniques to customize the codebook as necessary. It would be interesting to compare zoos across disciplines (within and outside of CS), across languages, and author demographics.

Note on Quotations. Quoted sentences are presented verbatim. When the quoted sentence includes a citation, we leave the citation intact. Thus that citation will not be found in the bibliography of our paper, but can be found by looking at the paper being quoted.

3.1 Facts

3.1.1 Facts: Definition or Description. Many papers start with a straightforward definition of the subject of the paper. These tend to be neutral and like something you would read in a glossary.

- "Malware sandboxes are automated dynamic analysis tools that execute samples in an isolated and instrumented environment." [32]
- "Secure two-party computation allows two parties to process their sensitive data in such a way that its privacy is protected." [20]
- "HMAC is a cryptographic authentication algorithm, the 'Keyed-Hash Message Authentication Code,' widely used in conjunction with the SHA-256 cryptographic hashing primitive." [6]

Similarly, papers might provide a description of what the subject of a sentence does or how it works. These are also neutral statements and like something you'd read in a textbook.

- "Traditionally, digital investigations have aimed to recover evidence of a cyber-crime or perform incident response via analysis of non-volatile storage." [59]
- "Mobile social applications discover nearby users and provide services based on user activity (what the user is doing) and context (who and what is nearby)." [43]
- "To reduce the memory footprint of a system, the system software shares identical memory pages between processes running on the system." [68]

3.1.2 Facts: Claimed Fact. Another neutral approach to an opening sentence is to provide a fact that is relevant to the subject of the paper. Later we will discuss arguments which are often expressed as if they are facts but are only debatably true. A claimed fact's correctness should either be apparent or at least provable (i.e., falsifiable).

- "Users are often advised or required to choose passwords that comply with certain policies." [39]
- "Mobile apps frequently demand access to private information." [64]
- "For several decades, car keys have been used to physically secure vehicles." [29]

Some sentences use stronger and more vivid language but are still factually based.

- "In spite of extensive industrial and academic efforts (e.g., [3, 41, 42]), distributed denial-of-service (DDoS) attacks continue to plague the Internet." [26]

3.1.3 Facts: Technical Advances. Many opening sentences lay out a technical advance in the subject of the sentence. This creates a window of opportunity for the researcher to later identify a novel research problem caused by the changing technology. It is common to see words like: evolve, become, transition, and improve.

- "Recent advances in cloud computing enable customers to outsource their computing tasks to the cloud service providers (CSPs)." [54]
- "Browsers have evolved over recent years to mediate a wealth of user interactions with sensitive data." [34]
- "Since its beginning in the early nineties, the Web evolved from a mechanism to publish and link static documents into a sophisticated platform for distributed Web applications." [42]

3.1.4 Facts: Historic Events. A final type of neutral opening sentence will refer to some historic event.

- "In 1996, Wagner and Schneier performed an analysis of the SSL 3.0 protocol [67]." [65]
- "In February 2011, a new Tor hidden service [16], called "Silk Road," opened its doors." [61]
- "The Network Time Protocol (NTP) is one of the Internet's oldest protocols, dating back to RFC 958 [15] published in 1985." [23]

In some cases, a paper opens with a "compound" sentence that makes reference to a historic event in one clause of the sentence, while having additional clauses of a different category. For example, the following sentence refers to a historic event as well as a technical advance.

- "Starting from Denning's seminal work in 1986 [9], intrusion detection has evolved into a number of different approaches." [12]

In our analysis, some sentences are coded with more than one code but we do so sparingly.

3.2 Arguments

3.2.1 Arguments: General Argument. Many opening sentences issue a subjective argument that represents the authors' opinion. Unlike a fact, it isn't straightforward that the reader will accept it as true. While arguments are less neutral than facts, they can be more interesting and provocative, which can help draw the reader into the paper.

The arguments we categorize under "general arguments" do not fit elsewhere in our categorization system. As we go through more categories, we will see other more specific kinds of arguments.

- "It is a truth universally acknowledged, that password-based authentication on the web is insecure." [44]
- "The dismissal of human memory by the security community reached the point of parody long ago." [7]
- "In recent years, unwanted software has risen to the forefront of threats facing users." [63]
- "The phenomenal growth of Android devices brings in a vibrant application ecosystem." [13]

3.2.2 Arguments: Problem Statement . A special type of argument is a "problem statement" which uses the opening sentence to establish a problem or challenge to be solved.

- "A key challenge when running untrusted virtual machines is providing them with efficient and secure I/O." [60]
- "Determining the semantic similarity between two pieces of binary code is a central problem in a number of security settings." [24]
- "It is difficult to keep secrets during program execution." [57]

For some sentences, the problem is not stated explicitly but can be inferred from what is said. For example, the "pressure to respond" in the following sentence implies a problem.

- "As popular applications rely on personal, privacy-sensitive information about users, factors such as legal regulations, industry self-regulation, and a growing body of privacy-conscious users all pressure developers to respond to demands for privacy." [28]

3.3 Suitability

3.3.1 Suitability: Importance of subject. A large set of sentences make a special kind of argument: that the subject of the opening sentence is suitable or worthy of research. The exact reasons they are suitable fall into a few sub-categories: the subject is important, ubiquitous, complex, novel, popular with other researchers, or has been around a long time.

Many opening sentences state that their subject is important, with the implication that it is thus suitable for research.

- "Security has now become an important and real concern to connected and/or automated vehicles." [14]
- "Error handling is an important aspect of software development." [35]

- "SSL/TLS is, due to its enormous importance, a major target for attacks." [46]

Some sentences do not explicitly use the word "important" but find other ways to convey the same notion. For example, a concern or component might be described as essential or crucial or serious.

- "The threat of data theft in public and private clouds from insiders (e.g. curious administrators) is a serious concern." [22]
- "The same-origin policy (SOP) is a cornerstone of web security, guarding the web content of one domain from the access from another domain." [70]

3.3.2 Suitability: Ubiquity of subject. The most popular kind of opening sentence argues that a subject is suitable for research because it is ubiquitous and widely used.

- "Billions of users now depend on online services for sensitive communication." [45]
- "Embedded systems are omnipresent in our everyday life." [17]
- "Android is the major platform for mobile users and mobile app developers." [52]

3.3.3 Suitability: Popularity of subject. While the ubiquity of a subject corresponds to how widely it is used, a closely related variant points out that the subject has received a lot of attention. Often, this means attention from other researchers which lends credibility to the subject for further research.

- "Protecting the privacy of user data within mobile applications (apps for short) has always been at the spotlight of mobile security research." [50]
- "The black-market economy for purchasing Facebook likes, Twitter followers, and Yelp and Amazon reviews has been widely acknowledged in both industry and academia [6, 27, 37, 58, 59]." [66]
- "Since the first widely-exploited buffer overflow used by the 1998 Morris worm [27], the prevention, exploitation, and mitigation of memory corruption vulnerabilities have occupied the time of security researchers and cybercriminals alike." [21]

3.3.4 Suitability: Longevity of subject . In this category, how long a subject has been around is the key component to why it is a suitable subject for study. In some cases, a specific duration is provided and in others, it is implied that the amount of time is significant.

- "Redaction of sensitive information from documents has been used since ancient times as a means of concealing and removing secrets from texts intended for public release." [4]
- "Since its beginning in the early nineties, the Web evolved from a mechanism to publish and link static documents into a sophisticated platform for distributed Web applications." [42]

3.3.5 Suitability: Complexity of subject. In this category, the complexity of the subject is highlighted, implying that the complexity creates new issues or requires further research. The complexity might be inherent to the subject itself. Or there might be a complex set of external factors to consider.

- "Today, large and complex software is built with many components integrated using APIs." [69]
- "The capabilities and limitations of disassembly are not always clearly defined or understood, making it difficult for researchers and reviewers to judge the practical feasibility of techniques based on it." [3]
- "As popular applications rely on personal, privacy-sensitive information about users, factors such as legal regulations, industry self-regulation, and a growing body of privacy-conscious users all pressure developers to respond to demands for privacy." [28]

3.3.6 Suitability: Novelty of subject. Finally, a degree of novelty is an important component in any research question so it is unsurprising that papers begin arguing novelty from their opening sentence. In this category, sentences focus on something that is new or emerging.

- "In the last few years, a new class of cyber attacks has emerged that is more targeted at individuals and organizations." [40]
- "Although the operating system (OS) kernel has always been an appealing target, until recently attackers focused mostly on the exploitation of vulnerabilities in server and client applications— which often run with administrative privileges—as they are (for the most part) less complex to analyze and easier to compromise." [37]

This sentence manages to appeal to both longevity and novelty by relating two subjects.

- "While cryptocurrency has been studied since the 1980s [22, 25, 28], bitcoin is the first to see widespread adoption." [33]

3.4 Narrative

A potentially interesting way to draw a reader into a paper is by establishing a narrative: a scenario that gets the reader thinking about themselves or other people and what they might do.

- "Consider that you are a domain owner, holding a few domain names that you do not have a better use of." [2]
- "Consider the setting where a client owns a public input x, a server owns a private input w, and the client wishes to learn z := F(x,w) for a program F known to both parties." [5]

Narratives might also set a scene, like the academic version of an establishing shot from films and TV.

- "Our phones are always within reach and their location is mostly the same as our location." [47]
- "We live in a "big data" world." [41]
- "The battle for the living room is in full swing." [53]

3.5 Question

Making the reader curious is another good way to begin a paper, and this can be accomplished using a question. In our sample, this was underused with only one example.

- "Do programmers leave fingerprints in their source code?" [10]

4 DISCUSSION

Domain Expertise. We pondered whether we could apply our analysis to a domain in which we were not experts. Obviously a non-expert in security would not understand the technical content of many of our sentences, but would they be able to tell the difference between, say, a historic fact and a suitability argument?

We are unsure but skeptical. Consider the following sentence: "The widespread adoption of DEP, which ensures that all writable pages in memory are nonexecutable, has largely killed classic code injection attacks" [11]. A non-expert who is unfamiliar with DEP or code injection could understand the gist of the sentence: code injection is an *attack*, therefore it is *bad*; DEP is *good* because it reduces something that is bad. But what if the authors of the sentence had dropped the word 'attack' and instead simply ended: "DEP… has largely killed classic code injection." Now it is ambiguous to a non-expert whether DEP is good or bad, in fact the strength of the word 'kills' might lead the non-expert to conclude DEP is bad. While most sentences are not like this, we see this as evidence that our analysis should be undertaken by domain experts and not general readers.

Sentence Length. We examined the shortest and longest sentences in our dataset. Short sentences are striking and easy to remember. The shortest ones from our data set were four words long: "Video is ineffably compelling" [9] and "Software bugs are expensive" [58]. Long sentences can obviously convey more information or make a complex argument, but they might have to be read a few times to fully parse what they are saying. King has a similar argument in [38], where he gives an example of a long opening sentence in medical writing and he simplifies the sentence by shortening it and as a result making it more concise and easier to parse.

Security-specific Approach. There was one set of sentences that usually fell under the *suitability: importance of subject* code that was very specific to security. In security, it is important to researchers to tackle the biggest threats. Many use their opening sentence to position themselves as doing so. Many of them flatly state that their research area is the most prevalent threat ("In recent years, unwanted software has risen to the forefront of threats facing users" [63]; "Today, runtime attacks remain one of the most prevalent attack vectors against software programs" [19]; "Remote malware downloads currently represent the most common infection vector' [51]). Some prefer to make their point in a stronger way with a colourful choice of words ("In spite of extensive industrial and academic efforts (e.g., [3, 41, 42]), distributed denial-of-service (DDoS) attacks continue to plague the Internet" [26]). We look at colourful sentences next, with another example using the word 'plague.'

Colour. We also analyzed to what extend the authors add 'colour commentary' in the opening sentence. The decision to use this type of commentary is related to how researchers perceive and define the seriousness of their topic. Is avoiding all colour in the text a way to prove that the problem is significant and should be taken seriously? Or should writers add some colour to make the writing more engaging and easier to hook the reader? Examples of sentences that use a more vivid choice of words include: "The defacement and vandalism of websites is an attack that disrupts the operation of companies and organizations, tarnishes their brand, and plagues websites of all sizes, from those of large corporations to the websites of single individuals" [8]; "The dismissal of human memory by the security community reached the point of parody long ago" [7]; "Video is ineffably compelling" [9].

Using colourful language is probably best in moderation. Text that is very ornate and elaborate is deridingly called 'purple prose.' We did not find any examples in our data set, but to what extent is this a reflection of writing in information technology? An interesting area of future work is to examine the usefulness of our taxonomy in both other STEM domains, but also in academic writing in arts and the humanities.

The Arc of Time. Appealing to the notion that a domain is important because of its longevity is also a common way to start a paper in our data set. The idea of appealing to years or decades of work is extremely common ("Over the last few years there have been numerous reports…" [49]; "The last decade in cryptography…" [1]; "Over the past few years, face authentication systems…" [67]; "For several decades, car keys…" [29]). The longest timeframe in our dataset is: "Redaction of sensitive information from documents has been used since ancient times as a means of concealing and removing secrets from texts intended for public release" [4].

5 CONCLUSION

We hope that our results are interesting, but that they also raise consciousness of thoughtful writing in research papers. As researchers ourselves, we have often found ourselves staring at a blinking cursor, trying to come up with a compelling way to start our paper. After completing our analysis, we find ourselves proactively choosing how to start our writing. The zoo also allows us to try on different approaches (*e.g.,* 'what if we started with a narrative instead of the importance of the domain?') to find one that works. Finally, we may not have fully answered the question we started with ('what makes writing style stand out?') but we have equipped readers with a set of samples to examine while asking themselves this question.

ACKNOWLEDGMENTS

The *SIGCSE* reviews were thoughtful, insightful, and enthusiastic—thank you to the reviewers! J. Clark acknowledges funding of this project from the National Sciences and Engineering Research Council of Canada (NSERC).

REFERENCES

[1] Erdem Alkim, Léo Ducas, Thomas Pöppelmann, and Peter Schwabe. 2016. Post-quantum key exchange—a new hope. In *USENIX Security*. 327–343.
[2] Sumayah Alrwais, Kan Yuan, Eihal Alowaisheq, Zhou Li, and XiaoFeng Wang. 2014. Understanding the dark side of domain parking. In *USENIX Security*. 207–222.
[3] Dennis Andriesse, Xi Chen, Victor Van Der Veen, Asia Slowinska, and Herbert Bos. 2016. An in-depth analysis of disassembly on full-scale x86/x64 binaries. In *USENIX Security*. 583–600.
[4] Frederico Araujo, W Kevin, et al. 2015. Compiler-instrumented, dynamic secret-redaction of legacy processes for attacker deception. In *USENIX Security*. 145–159.
[5] Eli Ben-Sasson, Alessandro Chiesa, Eran Tromer, and Madars Virza. 2014. Succinct non-interactive zero knowledge for a von Neumann architecture. In *USENIX Security*. 781–796.
[6] Lennart Beringer, Adam Petcher, Q Ye Katherine, and Andrew W Appel. 2015. Verified Correctness and Security of OpenSSL HMAC. In *USENIX Security*. 207–221.
[7] Joseph Bonneau and Stuart Schechter. 2014. Towards reliable storage of 56-bit secrets in human memory. In *USENIX Security*. 607–623.
[8] Kevin Borgolte, Christopher Kruegel, and Giovanni Vigna. 2015. Meerkat: Detecting website defacements through image-based object recognition. In *USENIX Security*. 595–610.
[9] Matthew Brocker and Stephen Checkoway. 2014. iSeeYou: Disabling the MacBook Webcam Indicator LED. In *USENIX Security*. 337–352.
[10] Aylin Caliskan-Islam, Richard Harang, Andrew Liu, Arvind Narayanan, Clare Voss, Fabian Yamaguchi, and Rachel Greenstadt. 2015. De-anonymizing programmers via code stylometry. In *USENIX Security*. 255–270.

[11] Nicholas Carlini and David Wagner. 2014. ROP is still dangerous: Breaking modern defenses. In *USENIX Security*. 385–399.

[12] Marco Caselli, Emmanuele Zambon, Johanna Amann, Robin Sommer, and Frank Kargl. 2016. Specification mining for intrusion detection in networked control systems. In *USENIX Security*. 791–806.

[13] Kai Chen, Peng Wang, Yeonjoon Lee, XiaoFeng Wang, Nan Zhang, Heqing Huang, Wei Zou, and Peng Liu. 2015. Finding unknown malice in 10 seconds: Mass vetting for new threats at the google-play scale. In *USENIX Security*. 659–674.

[14] Kyong-Tak Cho and Kang G Shin. 2016. Fingerprinting electronic control units for vehicle intrusion detection. In *USENIX Security*. 911–927.

[15] Alison Clear and Allen Parrish. December 31, 2020. *A Computing Curricula Series Report, CC2020.* Technical Report. Association for Computing Machinery (ACM), IEEE Computer Society (IEEE-CS).

[16] Alison Clear, Allen Parrish, Ming Zhang, and Gerritt C van der Veer. 2017. CC2020: A Vision on Computing Curricula. In *SIGCSE*. 647–648.

[17] Andrei Costin, Jonas Zaddach, Aurélien Francillon, and Davide Balzarotti. 2014. A large-scale analysis of the security of embedded firmwares. In *USENIX Security*. 95–110.

[18] Vianney Cote and Guy Custeau. 1992. An integrating pedagogical tool based on writing articles. *ACM SIGCSE Bulletin* 24, 1 (1992), 38–41.

[19] Lucas Davi, Ahmad-Reza Sadeghi, Daniel Lehmann, and Fabian Monrose. 2014. Stitching the gadgets: On the ineffectiveness of coarse-grained control-flow integrity protection. In *USENIX Security*. 401–416.

[20] Daniel Demmler, Thomas Schneider, and Michael Zohner. 2014. Ad-hoc secure two-party computation on mobile devices using hardware tokens. In *USENIX Security*. 893–908.

[21] Alessandro Di Federico, Amat Cama, Yan Shoshitaishvili, Christopher Kruegel, and Giovanni Vigna. 2015. How the ELF Ruined Christmas. In *USENIX Security*. 643–658.

[22] Tien Tuan Anh Dinh, Prateek Saxena, Ee-Chien Chang, Beng Chin Ooi, and Chunwang Zhang. 2015. M2r: Enabling stronger privacy in mapreduce computation. In *USENIX Security*. 447–462.

[23] Benjamin Dowling, Douglas Stebila, and Greg Zaverucha. 2016. Authenticated network time synchronization. In *USENIX Security*. 823–840.

[24] Manuel Egele, Maverick Woo, Peter Chapman, and David Brumley. 2014. Blanket execution: Dynamic similarity testing for program binaries and components. In *USENIX Security*. 303–317.

[25] Joe Fasser. 2013. Why Stephen King Spends 'Months and Even Years' Writing Opening Sentences. *The Atlantic* (July 2013).

[26] Seyed K Fayaz, Yoshiaki Tobioka, Vyas Sekar, and Michael Bailey. 2015. Bohatei: Flexible and elastic ddos defense. In *USENIX Security*. 817–832.

[27] Harriet J Fell, Viera K Proulx, and John Casey. 1996. Writing across the computer science curriculum. *ACM SIGCSE Bulletin* 28, 1 (1996), 204–209.

[28] Matthew Fredrikson and Benjamin Livshits. 2014. ZØ: An optimizing distributing zero-knowledge compiler. In *USENIX Security*. 909–924.

[29] Flavio D Garcia, David Oswald, Timo Kasper, and Pierre Pavlidès. 2016. Lock it and still lose it—on the (in) security of automotive remote keyless entry systems. In *USENIX Security*.

[30] Alan Garvey. 2010. Writing in an Upper-Level CS Course. In *SIGCSE*.

[31] Barney G Glaser and Anselm L Strauss. 1967. The discovery of grounded theory; strategies for qualitative research. (1967).

[32] Mariano Graziano, Davide Canali, Leyla Bilge, Andrea Lanzi, Elaine Shi, Davide Balzarotti, Marten van Dijk, Michael Bailey, Srinivas Devadas, Mingyan Liu, et al. 2015. Needles in a haystack: Mining information from public dynamic analysis sandboxes for malware intelligence. In *USENIX Security*. 1057–1072.

[33] Ethan Heilman, Alison Kendler, Aviv Zohar, and Sharon Goldberg. 2015. Eclipse attacks on bitcoin's peer-to-peer network. In *USENIX Security*. 129–144.

[34] Nav Jagpal, Eric Dingle, Jean-Philippe Gravel, Panayiotis Mavrommatis, Niels Provos, Moheeb Abu Rajab, and Kurt Thomas. 2015. Trends and lessons from three years fighting malicious extensions. In *USENIX Security*. 579–593.

[35] Suman Jana, Yuan Jochen Kang, Samuel Roth, and Baishakhi Ray. 2016. Automatically detecting error handling bugs using error specifications. In *USENIX Security*. 345–362.

[36] David G Kay. 1998. Computer scientists can teach writing: an upper division course for computer science majors. In *SIGCSE*. 117–120.

[37] Vasileios P Kemerlis, Michalis Polychronakis, and Angelos D Keromytis. 2014. ret2dir: Rethinking kernel isolation. In *USENIX Security*. 957–972.

[38] Lester S King. 1967. The Opening Sentence. *JAMA* 202, 6 (1967), 535–536.

[39] Saranga Komanduri, Richard Shay, Lorrie Faith Cranor, Cormac Herley, and Stuart Schechter. 2014. Telepathwords: Preventing weak passwords by reading users' minds. In *USENIX Security*. 591–606.

[40] Stevens Le Blond, Adina Uritesc, Cédric Gilbert, Zheng Leong Chua, Prateek Saxena, and Engin Kirda. 2014. A look at targeted attacks through the lense of an NGO. In *USENIX Security*. 543–558.

[41] Mathias Lécuyer, Guillaume Ducoffe, Francis Lan, Andrei Papancea, Theofilos Petsios, Riley Spahn, Augustin Chaintreau, and Roxana Geambasu. 2014. Xray: Enhancing the web's transparency with differential correlation. In *USENIX Security*. 49–64.

[42] Sebastian Lekies, Ben Stock, Martin Wentzel, and Martin Johns. 2015. The unexpected dangers of dynamic javascript. In *USENIX Security*. 723–735.

[43] Matthew Lentz, Viktor Erdélyi, Paarijaat Aditya, Elaine Shi, Peter Druschel, and Bobby Bhattacharjee. 2014. SDDR: Light-Weight, Secure Mobile Encounters. In *USENIX Security*. 925–940.

[44] Zhiwei Li, Warren He, Devdatta Akhawe, and Dawn Song. 2014. The emperor's new password manager: Security analysis of web-based password managers. In *USENIX Security*. 465–479.

[45] Marcela S Melara, Aaron Blankstein, Joseph Bonneau, Edward W Felten, and Michael J Freedman. 2015. CONIKS: Bringing Key Transparency to End Users. In *USENIX Security*. 383–398.

[46] Christopher Meyer, Juraj Somorovsky, Eugen Weiss, Jörg Schwenk, Sebastian Schinzel, and Erik Tews. 2014. Revisiting SSL/TLS implementations: New bleichenbacher side channels and attacks. In *USENIX Security*. 733–748.

[47] Yan Michalevsky, Aaron Schulman, Gunaa Arumugam Veerapandian, Dan Boneh, and Gabi Nakibly. 2015. Powerspy: Location tracking using mobile device power analysis. In *USENIX Security*. 785–800.

[48] Joe Miró Julià. 2011. An engineering approach to teaching writing.. In *SIGCSE*.

[49] Gabi Nakibly, Jaime Schcolnik, and Yossi Rubin. 2016. Website-targeted false content injection by network operators. In *USENIX Security*. 227–244.

[50] Yuhong Nan, Min Yang, Zhemin Yang, Shunfan Zhou, Guofei Gu, and XiaoFeng Wang. 2015. Uipicker: User-input privacy identification in mobile applications. In *USENIX Security*. 993–1008.

[51] Terry Nelms, Roberto Perdisci, Manos Antonakakis, and Mustaque Ahamad. 2015. Webwitness: Investigating, categorizing, and mitigating malware download paths. In *USENIX Security*. 1025–1040.

[52] Marten Oltrogge, Yasemin Acar, Sergej Dechand, Matthew Smith, and Sascha Fahl. 2015. To Pin or Not to Pin—Helping App Developers Bullet Proof Their TLS Connections. In *USENIX Security*. 239–254.

[53] Yossef Oren and Angelos D Keromytis. 2014. From the aether to the ethernet—attacking the internet using broadcast digital television. In *USENIX Security*. 353–368.

[54] Erman Pattuk, Murat Kantarcioglu, Zhiqiang Lin, and Huseyin Ulusoy. 2014. Preventing cryptographic key leakage in cloud virtual machines. In *USENIX Security*. 703–718.

[55] Linda H Pesante. 1991. Integrating writing into computer science courses. In *SIGCSE*. 205–209.

[56] Steven Pinker. 2015. *The sense of style: The thinking person's guide to writing in the 21st century.* Penguin Books.

[57] Ashay Rane, Calvin Lin, and Mohit Tiwari. 2015. Raccoon: Closing digital side-channels through obfuscated execution. In *USENIX Security*. 431–446.

[58] Alexandre Rebert, Sang Kil Cha, Thanassis Avgerinos, Jonathan Foote, David Warren, Gustavo Grieco, and David Brumley. 2014. Optimizing seed selection for fuzzing. In *USENIX Security*. 861–875.

[59] Brendan Saltaformaggio, Zhongshu Gu, Xiangyu Zhang, and Dongyan Xu. 2014. DSCRETE: Automatic rendering of forensic information from memory images via application logic reuse. In *USENIX Security*. 255–269.

[60] Igor Smolyar, Muli Ben-Yehuda, and Dan Tsafrir. 2015. Securing self-virtualizing ethernet devices. In *USENIX Security*. 335–350.

[61] Kyle Soska and Nicolas Christin. 2015. Measuring the longitudinal evolution of the online anonymous marketplace ecosystem. In *USENIX Security*. 33–48.

[62] Harriet G Taylor and Katharine M Paine. 1993. An interdisciplinary approach to the development of writing skills in computer science students. *ACM SIGCSE Bulletin* 25, 1 (1993), 274–278.

[63] Kurt Thomas, Juan A Elices Crespo, Ryan Rasti, Jean-Michel Picod, Cait Phillips, Marc-André Decoste, Chris Sharp, Fabio Tirelo, Ali Tofigh, Marc-Antoine Courteau, et al. 2016. Investigating commercial pay-per-install and the distribution of unwanted software. In *USENIX Security*. 721–739.

[64] Omer Tripp and Julia Rubin. 2014. A bayesian approach to privacy enforcement in smartphones. In *USENIX Security*. 175–190.

[65] Tom Van Goethem, Mathy Vanhoef, Frank Piessens, and Wouter Joosen. 2016. Request and conquer: Exposing cross-origin resource size. In *USENIX Security*. 447–462.

[66] Bimal Viswanath, M Ahmad Bashir, Mark Crovella, Saikat Guha, Krishna P Gummadi, Balachander Krishnamurthy, and Alan Mislove. 2014. Towards detecting anomalous user behavior in online social networks. In *USENIX Security*. 223–238.

[67] Yi Xu, True Price, Jan-Michael Frahm, and Fabian Monrose. 2016. Virtual u: Defeating face liveness detection by building virtual models from your public photos. In *USENIX Security*. 497–512.

[68] Yuval Yarom and Katrina Falkner. 2014. FLUSH+ RELOAD: A high resolution, low noise, L3 cache side-channel attack. In *USENIX Security*. 719–732.

[69] Insu Yun, Changwoo Min, Xujie Si, Yeongjin Jang, Taesoo Kim, and Mayur Naik. 2016. Apisan: Sanitizing API usages through semantic cross-checking. In *USENIX Security*. 363–378.

[70] Xiaofeng Zheng, Jian Jiang, Jinjin Liang, Haixin Duan, Shuo Chen, Tao Wan, and Nicholas Weaver. 2015. Cookies lack integrity: Real-world implications. In *USENIX Security*. 707–721.