A Quarter of Million Insights about End-User Programmers

Kevin Lubick*, Titus Barik*†, Emerson Murphy-Hill*
*North Carolina State University, Raleigh, North Carolina, USA
†ABB Corporate Research, Raleigh, North Carolina, USA
kjlubick@ncsu.edu, titus.barik@us.abb.com, emerson@csc.ncsu.edu

***Abstract*—We are proposing the FUSE corpus to be used as the dataset for the MSR 2016 data challenge. The FUSE corpus was presented at the MSR 2015 data showcase, containing 249,376 unique spreadsheets — an order of magnitude larger than previous corpora. In this paper we make a case for how this diverse corpus of spreadsheets will offer large benefits to the software engineering and research communities. We conclude with some promising lines of research FUSE makes possible and give an overview of the tools we provide to aid researchers participating in the challenge.**

I. INTRODUCTION

End-user programmers today constitute a broad class of users, including teachers, accountants, administrators, managers, research scientists, and even children [1]. Although these users are typically not professional software developers, their roles routinely involve computational tasks that, in many ways, are similar to those of developers — not just in activity, but also in their underlying cognitive demands on users [2].

Spreadsheets are perhaps the most ubiquitous form of end-user programming software [3]. *Cells* within the tables of these spreadsheets are augmented with computation, such as expressions, functions, and macros [4]. This interplay between presentation and computation within the spreadsheet environment has garnered significant interest from the software engineering research community [5]. Researchers have adopted techniques and approaches to studying errors [6], code smells [7], and refactoring in spreadsheets [8], similar to traditional programming environments.

We have previously presented a corpus of 249,376 spreadsheets, called FUSE [9], extracted from the over 26.83 billion web pages in the Common Crawl index. This corpus gives researchers the potential to extract unprecedented insight into how end-user use such an ubiquitous form of programming.

II. PROPOSED MSR CHALLENGE

Previous MSR challenges have been predominantly focused on the types of activities performed by professional programmers (see Table I). For example, years 2006 through 2011 of MSR challenges focused on source code repositories, bug data, and mailing lists for particular software projects. In 2012, the MSR challenge was extended to a different domain — mobile devices — but the sources of data considered were still essentially change logs and bug reports.

In more recent years, the MSR challenges have been expanding their datasets to more diverse information sources, such as Stack Overflow. Well-received mining papers, like the one presented by Chowdhury and Hindle [10], tend go further and incorporate additional diverse datasets like YouTube comments.

TABLE I
PREVIOUS MSR CHALLENGE DATA SOURCES

YEAR	Data Source
2006	PostgreSQL and ArgoUML
2007	Eclipse and Firefox
2008	Eclipse
2009	GNOME Desktop
2010	FreeBSD, Debian, and GNOME
2011	Eclipse and NetBeans, Firefox and Chrome
2012	Android
2013	Stack Overflow
2014	GitHub
2015	Stack Overflow

End-user programmers outnumber professional programmers by an order of magnitude [3]. Given their role and importance will only increase with time, our proposed MSR challenge asks participants to obtain insights about end-user programmers and the artifacts that they produce using spreadsheets. First, we think that such a challenge would increase the diversity of the MSR community, by garnering interest from researchers in the end-user communities. Second, spreadsheets may serve as a data source to allow software engineering researchers to try their tools and techniques, such as static analysis, on a very different domain than what they typically study. Finally, we hope that this MSR challenge will inspire new tools and generate new insights about end-user programmers, and that such findings will drive the directions of a broad range of software engineering research.

III. THE CONTENTS OF FUSE

In this section, we summarize the contents of the FUSE corpus. For readers curious on the details of the extraction process that took about 60,000 instance hours, see our MSR data paper [9].

The primary contribution of FUSE is the 249,376 unique spreadsheets extracted from the Common Crawl web index[1]

[1]The Common Crawl non-profit organization provides this index to companies and individuals at no cost for the purpose of research and analysis. For more information, see http://www.commoncrawl.org.

. The second component is a myriad of metadata about these spreadsheets, which is detailed below. Finally, we released the 2,127,284 web URLs that returned spreadsheet-related MIME types, of which the URLs that returned actual spreadsheets are a subset.

For our corpus, we augmented the JSON web record returned from Common Crawl with additional metadata from four *plugins*: `Apache Tika`[2], which contbutes file metadata including the MIME type, a best-guess file extension, a SHA-1 signature, and the length in bytes of the spreadsheet; `InternetDomain`, which uses the Google Guava[3] library to extract domain-related information from the `WARC-Target-URI`; `LingPipe`[4], which extracts language-related information from the spreadsheet. and `Apache POI`[5], which obtains metrics on the content of the spreadsheets, such as the use of functions;

The last two elements are provided as a way for researchers to query for spreadsheets of the most interest, and we detail them a bit more here. The `LingPipe` plugin extracts the token stream (keywords) from spreadsheets, lowercases these tokens, removes English stop words (such as 'a' or 'the'), and filters out non-words (such as numbers). As an example, these tokens could be used to locate financial spreadsheets by looking for words such as "accounts receivable", "depreciation" and so on.

The `Apache POI` plugin gathers preadsheet metrics to get a high-level overview of the content of the spreadsheets. There are over 450 such metrics, which include the number of times a given Excel function (such as `SUM` or `VLOOKUP`) is used, the total number of input cells (i.e., cells that are not formulas), the number of numeric input cells, and the most common formula used.

IV. Potential Spreadsheet Research Questions

There are many ways spreadsheets have been studied in the past. In this section, we present a few of them as well as a few of our own with the hope of sparking the minds of the creative MSR community.

One similarity found among several spreadsheet corpora (Table II) is that not every spreadsheet uses formulas. In fact, the proportion of formula-containing spreadsheets is typically less than half. This has been noticed by even the developers of Excel:

> Everybody thought of Excel as a financial modeling application, [but] we visited dozens of Excel customers, and did not see anyone using Excel to actually perform what you would call 'calculations.' Almost all of them were using Excel because it was a convenient way to create a table.
> — Joel Spolsky [11]

The open question we have is **Can you detect spreadsheets that could use formulas but don't using techniques like machine learning?** As far as we know, no one has directly answered this.

Several end-user researchers have investigated techniques of determining the quality of spreadsheets through static analysis, using similar approaches to their software engineering counterparts. For example, Cunha and colleagues have created a quality model of spreadsheets [12] and the work by Pinzger and others introduced the idea of code smells to spreadsheets [7]. Most recently, Jannach and colleagues came up with a technique for automatically finding and fixing spreadsheet errors [13]. One downside to these recent approaches is that they all relied on the well-known EUSES corpus [14], which contains about 5,000 spreadsheets circa 2005. It would be interesting to **replicate findings of these static analysis tools on FUSE**, and **to expand these techniques to include modern spreadsheet tools and functions**.

We found some spreadsheets that changed at least once over the course of the Common Crawl. The fact that FUSE has this temporal component opens the door to literally a new dimension of analysis, such as **How do spreadsheets and their formulas change over time?** We anticipate that researchers will be able to use a spreadsheet-diffing tool like SheetDiff created by Chambers and colleagues [15] to assist with answering this family of research questions.

The final set of research questions digs into the structure of spreadsheets and clustering or otherwise drawing conclusions about the set. The work by Abraham and colleagues [16] looked at **what can you extrapolate about a spreadsheet based on its structure**, such as "This looks like a gradebook and this looks like an inventory". Along this vein, **How likely are formulas to co-exist across sheets? Do some functions cluster together and can we make recommendations based on this? Are there metrics that can be used to identify "interesting" spreadsheets, for some definition of interesting?** The space of questions is enormous and we end the list here, having inspired potential challenge participants.

TABLE II
COMPARISON OF FUSE AND OTHER SPREADSHEET CORPORA

	FUSE	**EUSES**	**Enron**
Size (n)	249,376	6,000	15,570
Space (GB)	87	0.64	23.3
Access	All	Researchers	All
Unique formulas	894,361	84,004	784,380
Extendable	Yes	Yes	No
Framework	Hadoop	Excel/VBA	Scantool
Scalable	Yes	No	No
Collection period	2013-2014	2006	2006
Primary origin	CC	Google	Enron
Binaries	Yes	Yes	Yes
Metadata	Yes	No	No
Domains	4,318	???	-
Distinct functions	219	209	139

V. The use of other spreadsheet corpora

Keeping with the spirit of diverse submissions, we would also like to point to two other spreadsheet corpora that partici-

[2]https://tika.apache.org/
[3]https://github.com/google/guava
[4]http://alias-i.com/lingpipe/
[5]http://poi.apache.org/

pants may consider including in their analysis. We summarize these corpora in Table II. The EUSES corpus [14] was created by scraping Google search queries for spreadsheets and keywords like "finance". As a part of the Enron trial, over 15,000 spreadsheets were included in the released emails, which were extracted to make the Enron corpus [17].

These three corpora offer three different viewpoints into the user of spreadsheets — for example, the Enron corpus offers insight into professional, financial spreadsheets. Our challenge would not require the use of EUSES or Enron corpora, but we would like participants to be aware of them and be free to use them to compare or contrast to FUSE.

VI. Useful Analysis notes for Participants

In addition to releasing FUSE, we also released the source code of the tools we used to create the corpus. Of note, we used Apache POI as the main library for analyzing the spreadsheets, as it is fast, written in Java, open source and works well on most spreadsheet formats. One thing to keep in mind is that Apache POI uses up a fair amount of memory - to analyze spreadsheets that are 3MB or bigger requires at least 2 GB of heap space to avoid out-of-memory errors.

One potential concern participants may have with our dataset is it requiring an unreasonable amount of processing power for such a challenge. We would like to allay those concerns by sharing the fact that creating the metadata for all of the spreadsheets took around 400 instance hours on Amazon EMR, a far more reasonable number than the 60,000 hours creating the corpus required.

Additionally, analysis time can be reduced further by using the metadata we provide to narrow the search space. For example, participants can load our metadata into MongoDB, query for spreadsheets that contain at least one IF statement, and then analyze only those several thousand spreadsheets.

VII. Past and Present Research involving Fuse

We made the roughly 8 GB dataset available in March 2015, upon submission of the MSR data paper. Since then, we have had tens of downloads and are currently aware of at least three researchers or research groups using Fuse[6].

VIII. Final notes for challenge committee

We intend this paper to be the basis for the MSR 2016 data challenge. However, we request that our MSR 2015 paper [9] be cited as a part of the challenge.

IX. Short bio of the authors

Kevin Lubick is struggling to be a PhD student at North Carolina State University. He has a part time job as a soccer referee, where the 50% approval rate is a drastic improvement over his abysmal paper acceptance rates. He clings to Titus Barik for publications and a resemblence of competency.

Titus Barik is a PhD student at North Carolina State University, a researcher at ABB Corporate Research and a diligent father - that is to say, tired and cranky. Due to a glitch in the Human Resources department, he was accidentally hired to be a 2015 intern at Microsoft Research. He started researching spreadsheet corpora after accidentally replying to an email that mentioned free donuts. He believes he is advised by Emerson Murphy-Hill.

Emerson Murphy-Hill is a professor at North Carolina State University. After several years of hard work, he received tenure, but was dismayed to find out no one cares. He is disillusioned to believe he advises both Titus and Kevin. His favorite food is bananas, an interest he shares with his daughter Zuri.

[6]Links to known researchers or research labs using Fuse: http://emeryberger.com/research/, http://research.csc.ncsu.edu/dlf/, and http://www.felienne.com/

References

[1] A. J. Ko, B. Myers, M. B. Rosson, G. Rothermel, M. Shaw, S. Wiedenbeck, R. Abraham, L. Beckwith, A. Blackwell, M. Burnett, M. Erwig, C. Scaffidi, J. Lawrance, and H. Lieberman, "The state of the art in end-user software engineering," *ACM Computing Surveys*, vol. 43, no. 3, pp. 1–44, Apr. 2011.

[2] A. Blackwell, "First steps in programming: A rationale for attention investment models," in *IEEE 2002 Symposia on Human Centric Computing Languages and Environments*, 2002, pp. 2–10.

[3] C. Scaffidi, M. Shaw, and B. Myers, "Estimating the numbers of end users and end user programmers," in *VL/HCC '05*, 2005, pp. 207–214.

[4] B. A. Nardi and J. R. Miller, "The spreadsheet interface: A basis for end user programming," in *Human-Computer Interaction: INTERACT '90*, 1990, pp. 977–983.

[5] M. Burnett, "What Is End-User Software Engineering and Why Does It Matter?" in *End-User Development SE - 2*, ser. Lecture Notes in Computer Science, V. Pipek, M. Rosson, B. de Ruyter, and V. Wulf, Eds. Springer Berlin Heidelberg, 2009, vol. 5435, pp. 15–28. [Online]. Available: http://dx.doi.org/10.1007/978-3-642-00427-8_2

[6] S. G. Powell, K. R. Baker, and B. Lawson, "A critical review of the literature on spreadsheet errors," *Decis. Support Syst.*, vol. 46, no. 1, pp. 128–138, Dec. 2008.

[7] M. Pinzger, F. Hermans, and A. van Deursen, "Detecting code smells in spreadsheet formulas," in *ICSM '12*, 2012, pp. 409–418.

[8] S. Badame and D. Dig, "Refactoring meets spreadsheet formulas," in *ICSM '12*, 2012, pp. 399–409.

[9] T. Barik, K. Lubick, J. Smith, J. Slankas, and E. Murphy-Hill, "Fuse: A reproducible, extendable, internet-scale corpus of spreadsheets," 2015.

[10] S. A. Chowdhury and A. Hindle, "Mining stackoverflow to filter out off-topic irc," 2015.

[11] J. Spolsky. (2012) How Trello is different. [Online]. Available: http://www.joelonsoftware.com/items/2012/01/06.html

[12] J. Cunha, J. P. Fernandes, C. Peixoto, and J. Saraiva, "A quality model for spreadsheets," in *Quality of Information and Communications Technology (QUATIC), 2012 Eighth International Conference on the*. IEEE, 2012, pp. 231–236.

[13] D. Jannach, T. Schmitz, B. Hofer, and F. Wotawa, "Avoiding, finding and fixing spreadsheet errors–a survey of automated approaches for spreadsheet qa," *Journal of Systems and Software*, vol. 94, pp. 129–150, 2014.

[14] M. Fisher and G. Rothermel, "The EUSES spreadsheet corpus," in *ACM SIGSOFT Software Engineering Notes*, vol. 30, no. 4, Jul. 2005, pp. 1–5.

[15] C. Chambers and C. Scaffidi, "Struggling to excel: A field study of challenges faced by spreadsheet users," in *Proceedings of the 2010 IEEE Symposium on Visual Languages and Human-Centric Computing*, ser. VLHCC '10. Washington, DC, USA: IEEE Computer Society, 2010, pp. 187–194. [Online]. Available: http://dx.doi.org.prox.lib.ncsu.edu/10.1109/VLHCC.2010.33

[16] R. Abraham and M. Erwig, "Inferring templates from spreadsheets," in *Proceeding of the 28th international conference on Software engineering - ICSE '06*, May 2006, p. 182.

[17] F. Hermans and E. Murphy-Hill, "Enron's spreadsheets and related emails: A dataset and analysis," in *ICSE SEIP '15*, 2015, to appear.