Recommending Functions in Spreadsheets from the Fuse Corpus

Shaown Sarker, Emerson Murphy-Hill
Department of Computer Science
NC State University
Raleigh, North Carolina
Email: {ssarker, emurph3}@ncsu.edu

***Abstract*—The most common form of end-user programming software is spreadsheets. Despite spreadsheets having a large array of functions built-in, most of these functions are underused. To address this, we investigate recommender system technology and present a collaborative filtering based function recommender system for spreadsheets. Our evaluation shows that the recommender system performs significantly better in comparison with a most popular algorithm based system. In this paper, we discuss the implementation of the system and the evaluation process. In addition, we outline a road-map to integrate the system as an effective tool in Microsoft Excel.**

I. Introduction

End-user programmers range from children to professionals including teachers, accountants, administrators, managers, and research scientists [1]. While many of these users are not formally trained software developers, the tasks they perform on a frequent basis are quite similar to that of a professional software engineer's including detecting errors, debugging, and understanding formulas and macros written by others.

Spreadsheets are the most common form of end-user programming software. According to a study in 2005 [2], nearly 23 million Americans use spreadsheets, constituting 30% of the workforce. The use of spreadsheets is also very common in industry for analytical purposes. Winston [3] estimates that around 90% of all analysts in industry perform calculations in spreadsheets. From a spreadsheet user's perspective, formulas can be viewed as program fragments as formulas can contain programming constructs such as variables, conditional statements, and references to other parts of the spreadsheet. It is important that these end-user programmers using spreadsheet makes efficient use of the functionality available to them `[Citation needed]`.

In Microsoft Excel, there are almost 350 unique functions, (472 in Microsoft Excel 2013 including the compatibility functions[1]). But, most of these are rarely used. To facilitate these very large number of spreadsheet analysts to use these functions, it is viable that we explore effective function recommendation in spreadsheets.

`Not sure how to tie in Fuse here.`

Researchers have been assembling and preserving spreadsheet corpora for the purpose of better understanding of end-user activities and designing tools to assist them [4], [5]. In this paper, we look into Fuse [6], the largest reproducible spreadsheet corpus known to date. We applied a slightly modified version of the collaborative filtering algorithm on the spreadsheets in Fuse to get function recommendations for an input spreadsheet. We evaluated the performance of our system against another commonly used recommender algorithm and found out that our collaborative filtering-based recommendation system recommends functions in spreadsheet by `insert future calculated percentage here`.

We first review previous work on recommendation systems in softwares and related work on making spreadsheet formula usage efficient (Section II). We then provide details of the spreadsheet corpus used for our system and the details of the algorithm used (Section III) before describing our evaluation measures (Section IV) and their outcome (Section V). We discuss the various aspects of our system (Section VI) along with the limitations and future works (Section VII) based on our contribution in this paper afterwards.

II. Related Work

Recommender systems have been applied with success in both academia and industry to help users cope with information overload and provide personalized suggestions [7]–[9]. Recommender systems assist users by distinguishing products and services of interest to the user in a scenario when the number and diversity of options exceeds the user's decision making capability.

Collaborative filtering is one of the most prominent recommending technologies available. The core concept of collaborative filtering is to apply a nearest neighbor method between a user's ratings or preferences and the preference of a large user community, and provide the user with recommendations by extrapolating based on how her selection or preference relate to that of the community. Collaborative filtering has been applied to recommend movies [10], news [8], books [11], [12], research papers [13], and school courses [14], [15].

However, there has been little effort to use recommendation systems to help users of a large and complicated software system to learn and explore the functionalities and commands of the system. One of the notable attempts to use recommender systems to recommend command is the OWL [16], which makes command recommendations to Microsoft Word users.

[1] https://support.office.com/en-us/article/Excel-functions-by-category-5f91f4e9-7b42-46d2-9bd1-63f26a86c0eb

OWL recommends commands to an individual if she is not using certain commands at all but the community is using them on a frequent basis.

A similar approach to recommend commands in AutoCAD, a computer aided drafting software, called CommunityCommands has been developed by Matejka and colleagues based on collaborative filtering [17]. The researchers of CommunityCommand showed that the collaborative filtering derived command recommendations were preferred by the AutoCAD users over the history based recommendation approach used in OWL.

```
What should be the domain of the
recommendations - the user or the
spreadsheet
```

In this paper, we build on these prior works by applying the concept of recommender systems in spreadsheet to recommend functions. We use the Fuse spreadsheet corpus as a source of function usage preference of the spreadsheet users' community and recommend functions contextualized to an individual spreadsheet by applying a slightly modified collaborative filtering algorithm.

III. Methodology

A. Fuse Spreadsheet Corpus

B. Modified Collaborative Filtering

IV. Evaluation

V. Results

VI. Discussion

VII. Limitations & Future Work

Acknowledgment

References

[1] A. J. Ko, R. Abraham, L. Beckwith, A. Blackwell, M. Burnett, M. Erwig, C. Scaffidi, J. Lawrance, H. Lieberman, B. Myers *et al.*, "The state of the art in end-user software engineering," *ACM Computing Surveys (CSUR)*, vol. 43, no. 3, p. 21, 2011.

[2] C. Scaffidi, M. Shaw, and B. Myers, "Estimating the numbers of end users and end user programmers," in *Visual Languages and Human-Centric Computing, 2005 IEEE Symposium on*. IEEE, 2005, pp. 207–214.

[3] W. Winston, "Executive education opportunities millions of analysts need training in spreadsheet modeling, optimization, monte carlo simulation and data analysis," *OR MS TODAY*, vol. 28, no. 4, pp. 36–39, 2001.

[4] M. Fisher and G. Rothermel, "The euses spreadsheet corpus: a shared resource for supporting experimentation with spreadsheet dependability mechanisms," in *ACM SIGSOFT Software Engineering Notes*, vol. 30, no. 4. ACM, 2005, pp. 1–5.

[5] F. Hermans and E. Murphy-Hill, "Enrons spreadsheets and related emails: A dataset and analysis," Delft University of Technology, Software Engineering Research Group, Tech. Rep., 2014.

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

[7] W. Hill, L. Stead, M. Rosenstein, and G. Furnas, "Recommending and evaluating choices in a virtual community of use," in *Proceedings of the SIGCHI conference on Human factors in computing systems*. ACM Press/Addison-Wesley Publishing Co., 1995, pp. 194–201.

[8] P. Resnick, N. Iacovou, M. Suchak, P. Bergstrom, and J. Riedl, "Grouplens: an open architecture for collaborative filtering of netnews," in *Proceedings of the 1994 ACM conference on Computer supported cooperative work*. ACM, 1994, pp. 175–186.

[9] U. Shardanand and P. Maes, "Social information filtering: algorithms for automating word of mouth," in *Proceedings of the SIGCHI conference on Human factors in computing systems*. ACM Press/Addison-Wesley Publishing Co., 1995, pp. 210–217.

[10] B. N. Miller, I. Albert, S. K. Lam, J. A. Konstan, and J. Riedl, "Movielens unplugged: experiences with an occasionally connected recommender system," in *Proceedings of the 8th international conference on Intelligent user interfaces*. ACM, 2003, pp. 263–266.

[11] G. Linden, B. Smith, and J. York, "Amazon. com recommendations: Item-to-item collaborative filtering," *Internet Computing, IEEE*, vol. 7, no. 1, pp. 76–80, 2003.

[12] J. Liu, C. K. Wong, and K. K. Hui, "An adaptive user interface based on personalized learning," *IEEE Intelligent Systems*, no. 2, pp. 52–57, 2003.

[13] S. M. McNee, N. Kapoor, and J. A. Konstan, "Don't look stupid: avoiding pitfalls when recommending research papers," in *Proceedings of the 2006 20th anniversary conference on Computer supported cooperative work*. ACM, 2006, pp. 171–180.

[14] R. Farzan and P. Brusilovsky, "Social navigation support in a course recommendation system," in *Adaptive hypermedia and adaptive web-based systems*. Springer, 2006, pp. 91–100.

[15] M.-H. Hsu, "A personalized english learning recommender system for esl students," *Expert Systems with Applications*, vol. 34, no. 1, pp. 683–688, 2008.

[16] F. Linton, D. Joy, H.-P. Schaefer, and A. Charron, "Owl: A recommender system for organization-wide learning," *Educational Technology & Society*, vol. 3, no. 1, pp. 62–76, 2000.

[17] J. Matejka, W. Li, T. Grossman, and G. Fitzmaurice, "Community-commands: command recommendations for software applications," in *Proceedings of the 22nd annual ACM symposium on User interface software and technology*. ACM, 2009, pp. 193–202.