Effectiveness of Vector Space Model Feature Location Technique

Ramesh Neupane
Computer Science Departmenr
Boise State University
Boise, ID, US
rameshneupane@u.boisestate.edu

***Abstract*—In this paper, we evaluated the efficacy of the Vector Space Model (VSM) for the Feature Location Technique (FLT). The result shows that on average the rank of the best method is very low that it is not feasible to use this model for feature location technique.**

I. Introduction

It is estimated that there are 111 billion lines of code written every year by software developers [1]. Even an average sized software company has a minimum of a hundred thousand lines of code [2]. Therefore, it is a challenge for every software company in software maintenance and evolution. The software maintenance and evolution includes removing bugs, adding new features, and improving the existing functionalities [3]. The process of identifying the right places in the code that corresponds to some specific functionality is also called concept location or feature location technique (FLT) [4]. It is generally a manual and tedious task. We can, however, increase the productivity of FLT by generating a ranked list of the most likely methods in the code base. One of the approaches to generate such a ranked list is Vector Space Model (VSM). VSM has been previously applied to applications like automatic indexing [5], information retrieval [6], etc.

In this report, we performed an analysis on the effectiveness of the VSM feature location technique. In our case study, we found that with a given query, the best position will rank around 300 position which is not feasible in practice.

II. Vector Space Model

A. Implementation

The implementation of this VSM feature location technique is done in Python3. The external python libraries used are pandas, numpy and sklearn. Until the calculation of the similarity matrix, the example presented on the homework question paper was tested and then it is updated to handle the large and practical corpus.

The implementation step is as follows:

- The weighted term frequency – inverse document frequency (tf-idf) matrix is computed for each term present in the file CorpusMethods-jEdit4.3-AfterSplitStopStem.txt.
- The queries present in the file CorpusQueries-jEdit4.3-AfterSplitStopStem.txt are then loaded and converted into query vectors.
- The similarity matrix between the corpus of methods and the query is then calculated.
- The ranked list of every method corresonding to the featureID is then iterated to compute the best rank and to generate the required CSV.

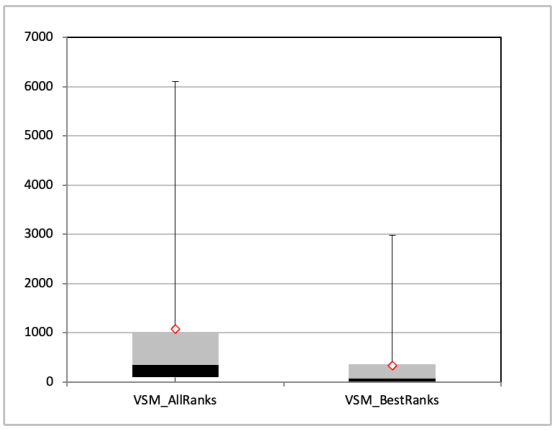

Fig. 1. A comparison of best and all ranks

TABLE I
Descriptive statistics of results

	Min.	25%	Med.	75%	Max.	Avg.	St. Dev.
All	1	92	341	1008	6102	1079.3	1586.7
Best	1	11.5	70	355	2984	338.8	695.1

B. Issues Encountered

While working with this implementation, there are various issues I have to face. The major problem was with generating the CSV. The formatting of the CSV makes it a lot harder to generate the CSV as required. Specially, to include the best rank of the first line of each methodID. Another issue I faced is that the run time while generating the similarity matrix is too high. It might be because of my programming logic. The computational time was more than an hour.

III. EVALUATION

A. Results and Discussion

TABLE I shows the statistics to quantify the effectiveness of the VSM model for feature location technique. We can see

that the minimum value for both method types is 1, the 25^{th} percentile of all methods is 92 and of the best method is 11.5. Likewise, the average mean of the rank is 338.8 for the best methods and 1079.3 for all methods. The standard deviation is 695.1 for the best method and it is more than double for all ranks.

Similarly, Fig. 1 shows the box plot of the data presented in TABLE I. We can see that the average position of the best rank is 338, that means the best method according to the query will rank at around this position. It means that if we implement this model, we need to look for around 300 results to get the best methods, which is not feasible. However, on the positive side, we narrowed down the options by 95%.

IV. CONCLUSION

Vector Space Model is one of the methods that can be used for feature location technique. We implemented the technique to determine the efficacy of this method. We found out that even though looking at the average rank of the best and all the methods based on the query, it feels infeasible to use this technique. However, it reduces the search space by 95% which is really effective.

REFERENCES

[1] "Application security report," Cybercrime Magazine, 03 2018. [Online]. Available: https://cybersecurityventures.com/application-security-report-2017/

[2] V. Capitalist, "Here's how many millions of lines of code it takes to run different software - business insider," Business Insider, 02 2017. [Online]. Available: https://www.businessinsider.com/how-many-lines-of-code-it-takes-to-run-different-software-2017-2

[3] B. Dit, M. Revelle, M. Gethers, and D. Poshyvanyk, "Feature location in source code: a taxonomy and survey," *Journal of software: Evolution and Process*, vol. 25, no. 1, pp. 53–95, 2013.

[4] V. Rajlich and N. Wilde, "The role of concepts in program comprehension," in *Proceedings 10th International Workshop on Program Comprehension*. IEEE, 2002, pp. 271–278.

[5] G. Salton, A. Wong, and C.-S. Yang, "A vector space model for automatic indexing," *Communications of the ACM*, vol. 18, no. 11, pp. 613–620, 1975.

[6] B. Abu-Salih, "Applying vector space model (vsm) techniques in information retrieval for arabic language," *arXiv preprint arXiv:1801.03627*, 2018.