Responses to reviewers

Please see our replies to reviewer comments, below. Note that anything in *italics* is from the review team.

A. Comment from Editor:

Sorry for the delay in the response to your revision. Both reviewers acknowledge good progress, but have some outstanding issue you need to look into:

Not a problem, thanks for organizing such great reviews. Based on that feedback we have made some significant additions to this draft:

- As suggested by reviewers, we now provide the comparison of DE with random search on topic stability and GA for the supervised task in section 5.8.
- As requested by both reviewers, we also show results applying LDADE to understand developer dialogs in StackOverflow (see section 5.2) which reduced the topic instability (which was previosuly observed with untuned LDA).

As to the specifics of your last round of comments:

- *Why categorizing StackExchange data into relevant and non relevant categories useful (Motivation of the SENG related example)*

We answered the reviewers concern in our reply to A.2

- *Use LDA-GA for the classification task in Section 5.3 and demonstrate whether or not LDADE outperforms LDA-GA*

As mentioned above, we have now down that (please see Section 5.8). Thank you for that suggestion.

- *Provide a deeper discussion on the very low number of evaluations of the proposed algorithm.*

As per the suggestion from our Reviewer 2, we provided our reasoning of such low number of evaluations in our reply to B.2. We also performed another experiment where we compared LDADE with the random search shown in Section 5.8.

A. Reviewer 1

Thank you for the addressing my comments. I'm happy with most of the changes made.

Thank you for those kind words.

A.1

Let me clear one misunderstanding. You wrote:

> *> The experiments need to be improved in the following ways/ First, LDA has been used to help many realistic software engineering tasks (for example, tasks considered by papers included in Table 2. There is a need to expand the experiments to compare LDA and LDADE on those realistic software engineering tasks. It is unclear if the task considered in the experiments (Section 5.3) is realistic (why categorizing StackExchange data into relevant and non relevant categories useful?). More than one tasks should have been considered (similar like Panichella et al.s work).*
>
> *> Our goal in this revision was to be as responsive as possible to your suggestions (as as you can see below, in A.4, A.5 and A.7, we were able to implement much of your advice.) But as to applying this to more realistic SE task, we might have a different perspective on what is a valid SE task. We say this since it sounds like you are saying the unsupervised tasks conducted by 23 of 28 recent highly cited LDA papers are not realistic? And that we should evaluate this paper only via the supervised tasks seen in 4 of the 28 papers? That is not our view, please see the notes above in A.1 on the need for stability in unsupervised SE tasks.*

From my original comment "LDA has been used to help many realistic software engineering tasks (for example, tasks considered by papers included in Table 2)", I'm not saying that the unsupervised tasks considered prior work listed in Table 2 are not realistic. The point that I would like to convey is: the paper does not evaluate LDADE and LDA directly for any of the supervised or unsupervised tasks listed in Table 2. Considering several of the unsupervised or supervised tasks in Table 2 will be good. I leave it to the authors to either perform this analysis or not, but at least the limitation should be acknowledged in the paper and maybe considered as a future work.

Thank you for clarifying your concern. Based on your suggestion, we added a analysis where LDADE provides a set of stable topics which were unstable before with the use of LDA. Please remember from Table 3, we showed how highly unstable LDA was if used with default configurations. We added another Table 8 in section 5.2, where same analysis was performed as before but this time with more stable configurations that was found by LDADE. LDADE found 27 stable topics. We also have acknowledged the limitation of this study for other SE tasks in our threats to validity section.

A.2

Also, there is a need to better motivate the supervised task considered in Section 5.3 (to answer the question: why categorizing StackExchange data into relevant and non relevant categories useful?).

While working with an industrial partner, their main product is about reading legal documents which they can mark as relevant or irrelevant in a legal case. They want to reduce the cost and improve the efficiency of labeling the involved party's legal case documents. We generated a similar mock-up with our StackExchange documents. This is the reason why categorizing this data into relevant and non relevant became useful. Since most of our SE tasks can be divided into the same case scenario where SE community want to reduce the cost and improve the efficiency, such as categorizing the test reports into buggy and not buggy.

A.3

There is also a need to use LDA-GA for the classification task in Section 5.3 and demonstrate whether or not LDADE outperforms LDA-GA. Even if LDADE does not outperform LDA-GA, it is still okay. The paper can inform researchers to use LDADE for unsupervised tasks or tasks requiring short parameter tuning time or tasks requiring better stability, and LDA-GA for supervised tasks or tasks where parameter tuning can be done overnight. However, without the comparison, researchers may not be able to make such a decision in a future work.

Thank you for suggesting us to look further into the comparison of LDA-GA and LDADE. We added the details in the text (see Section 5.8). From Figure 20, we did not see big difference in F1 results among the 2 methods. The reason being the value of k matters for the classification analysis and both the methods find the same range of k. But LDA-GA is still 24-70 times slower than LDADE which gives advantage of using LDADE (see Figure 19). We would still suggest researchers to perform LDADE for any task, since LDADE is orders of magnitude faster and gets better or same improvement.

B. Reviewer 2

First of all, I would like to thank the authors for the effort they put to address my comments. The paper is improved a lot.

Thank you for those kind words.

B.1

But, I have still a couple of comments. I was really surprised that with just 30 evaluations the algorithm is able to achieve a stable configuration. This is - in my opinion - a result that in some way belittles the issue raised in the paper. If by exploring only 30 different configurations Im able to obtain a stable solution, probably LDA is in general stable and I dont need a sophisticated approach to find a much stable solution.

Good point, but in reply we note that DE is more than just "try 30 different configurations". Consider how it works:

- It samples from a space that is continuously improving. In every generation, after item i the frontier contains items better than at least one thing seen in the last generation (as do items $1..i$). During the next generation, mutation happens between 3 better candidates and, even after being 50% through one generation, odds are that the 3 candidates have all passed the "better' test. So DE builds better solutions from a space of candidates that it is continually refining and improving– a process we cannot characterize as "just try 30 things at random".
- GAs and SA mutate all their attributes independently. But DE supports vector-level mutation that retain the association between variables in the space [86].

As to LDA being generally stable, we can't quite agree with you there. Our evidence is very much to the contrary (see Table 3).

What you might be saying, and this is where we do agree, is that there might be better stabilities achievable after just a little more CPU. One research direction we have here in the lab is to try and change the inner workings of LDA, perhaps add an inertia constant to the Gibbs sampling to reduce the instabilities within that algorithm. Turns out, this is a potentially very promising idea but we have no definitive results to report at this time.

B.2

I just need to perform some trials and pick the configuration that provides the best results. Indeed, which are the benefits of DE as compared with a Random Search? I would like to see in the paper a deeper discussion on the very low number of evaluations of the proposed algorithm and if possible a comparison with Random Search.

After considering your suggestions we did perform the comparison of DE with random search and observed that the improvement achieved with LDADE outperforms the improvement achieved with random search. (Please see Figure 21).

B.3

In addition, I concur with Reviewer 1 that the proposed

algorithm should be experimented in a scenario more specific for the software engineering community.

This is a good point and one that is readily resolved. The last draft of this paper did not include results on re-stabilizing the Table 3 results:

- Please consider Table 3 which was based on the second most cited paper (in the last five years) from the Empirical Software Engineering journal[5]. That paper was called "What are developers talking about? An analysis of topics and trends in Stack Overflow". It used standard LDA to make conclusions about developer discussions within StackOverflow [7].
- Now please consider Table 8. This is the same analysis, this time using LDADE. Note how the first table's topics are mostly unstable while the second table's topics are mostly stable.

The clear conclusion here is that the research community needs to revisit certain prominent recent results in the SE research community which we take to be a specific scenario from the SE community (studying developer exchanges in StackOverflow).

[5] As stated by Google metrics, https://goo.gl/UM8Bxd