Test Migration for Android Applications

Farideh Khalili

Test Reuse for Android applications

- Problem?
  a. Automatic migration of functionality tests among applications with similar functionalities.

- Ultimate goal:
  a. Test store

- Subject:
  a. GUI based applications

- Why?
  a. Interactive: Test cases are a chain of events (either actions or oracles).
  b. Many applications with similar set of functionalities
  c. GUI Interfaces for the same functionality tend to be semantically similar

e1

e2

e3

e4
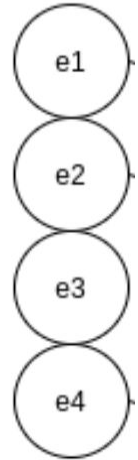
Functionality: Searching in a mailbox

```
[
 {
   "content-desc": "Search",
   "Resource-id": "ru.mail.mailapp:id/toolbar_action_search",
   "text": "",
   "action": [
     "click"
   ]
 },
 {
   "content-desc": null,
   "resource-id": "ru.mail.mailapp:id/search_text",
   "text": "Search Text",
   "action": [
     "send_keys_and_enter",
     "Automated"
   ]
 }
]
```

```
[
 {
   "content-desc": "Search",
   "resource-id": "com.my.mail:id/toolbar_action_search",
   "text": "",
   "action": [
     "click"
   ]
 },
 {
   "content-desc": null,
   "resource-id": "com.my.mail:id/search_text",
   "text": "Search Text",
   "action": [
     "send_keys_and_enter",
     "Automated"
   ]
 }
]
```

Semantic Matching in Test Reuse

1. Tools:
   a. AppTestMigrator (ATM) - 2019
   b. CraftDroid - 2019

2. Identify Reusable Components.
   a. Corpus of Documents
   b. Word Embedding
   c. Event Descriptor Extractor
   d. Semantic Matching Algorithm

3. Introduce a new semantic Matching Algorithm.
   a. SemFinder - 2021

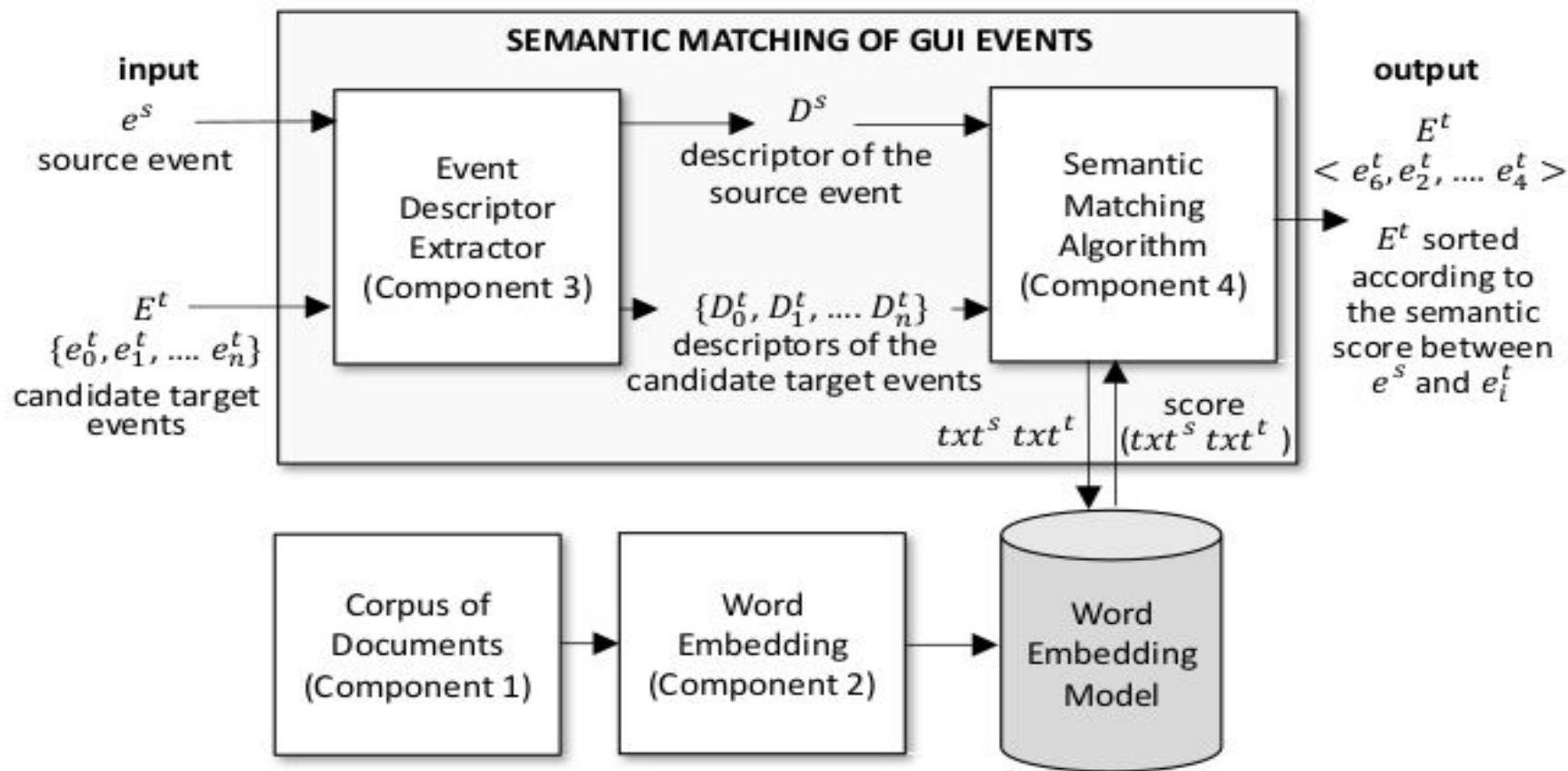4. Investigate the impact of Semantic Matching Components and their instances in isolation

Figure 2: Logical workflow of the semantic matching

Evaluation

- Evaluation
  a. 337 text queries (source events) to find the best match in a list of candidate events.
  b. Based on the rank of the true event in the list of candidate events.
  c. Top1
  d. MRR (Mean Reciprocal Rank)

$$\text{TOP1} = \frac{1}{|Q|} \sum_{i=1}^{|Q|} \left\{ \begin{array}{ll} 1 & \text{if } rank_i = 1 \\ 0 & \text{otherwise} \end{array} \right\} \in [0;1]$$

$$\text{MRR} = \frac{1}{|Q|} \sum_{i=1}^{|Q|} \frac{1}{rank_i} \in (0;1]$$
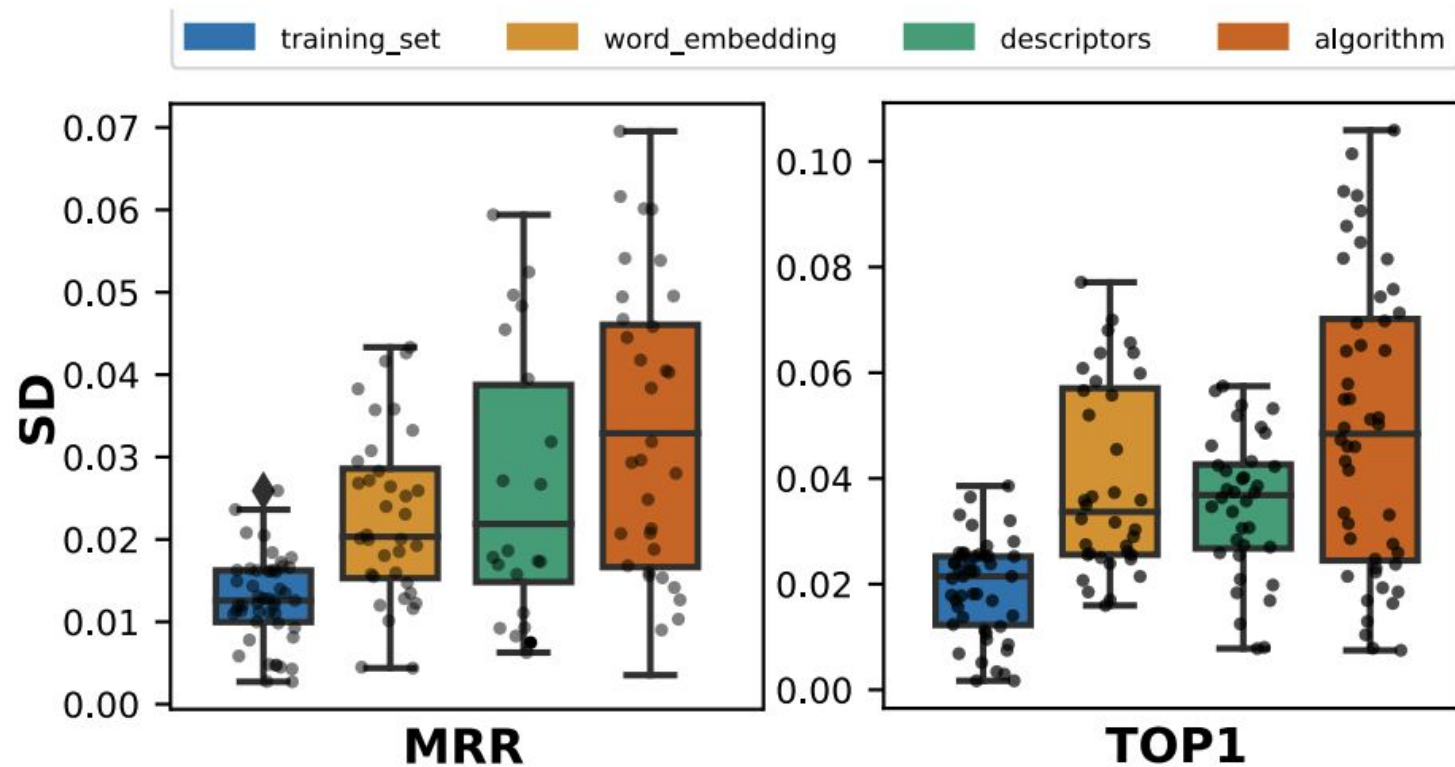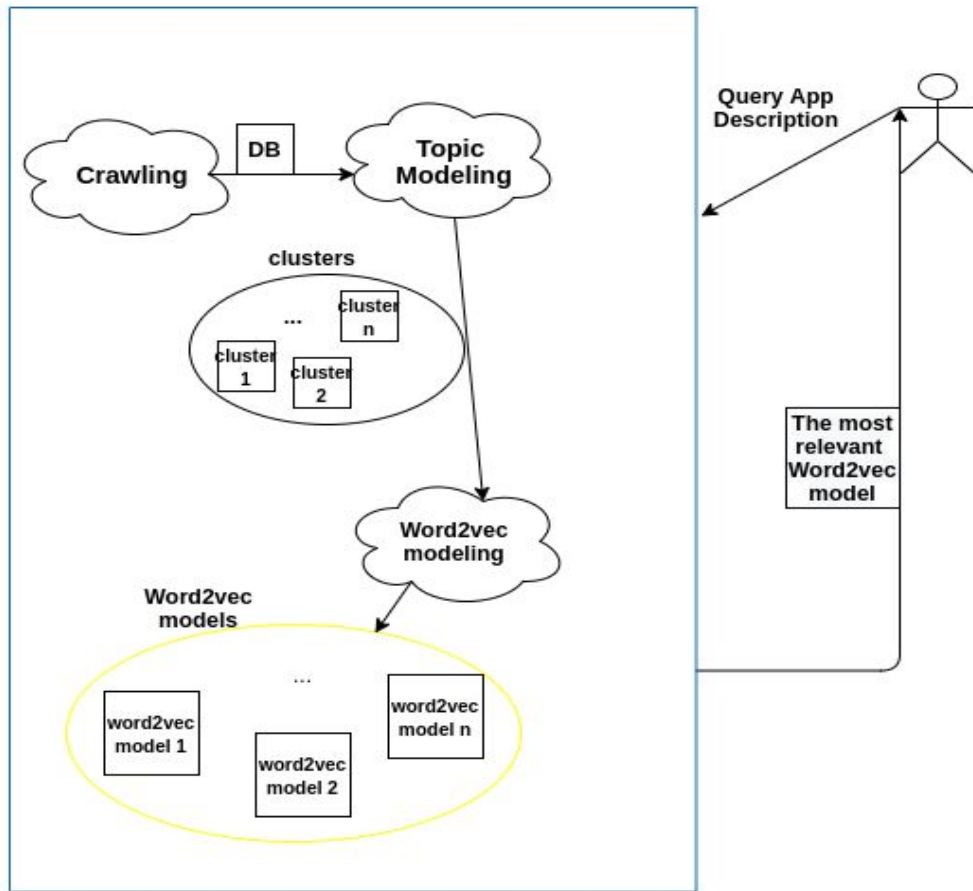
Figure 5: impact analysis

- Threats to validity and challenges:
  - Subject selection
    - the effectiveness of such approach depends heavily on the subjects
    - The effectiveness of the approach is negatively correlated with the length of the test case
  - Assumption of one to one mappings between the source and the target application
  - Depends heavily on the availability and variety of source tests.
  - The effectiveness of semantic matching in isolation is not necessarily correlated with the effectiveness of the test reuse approach that adopts it

- Future work:
  - Identify categories that work well with this approach
  - Consider one-to-many and many-to-one mappings of events
  - Evaluate semantic matching approaches with respect to its effect on test migration to see if there is a correlation with its performance in isolation
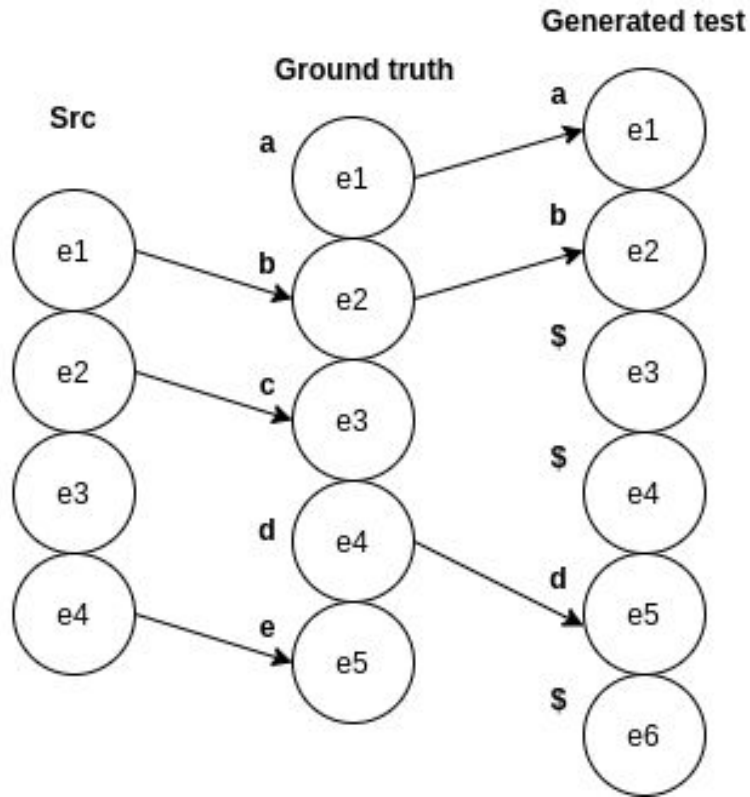
- Ground truth: *"abcde"*
- Generated test: *"ab$$d$"*
- Levenshtein(ground truth, generated) = 3
- Effort = 3
- #gtEvents = 5
- Reduction = 0.4

References

- [1]: Behrang, Farnaz, and Alessandro Orso. "Test migration between mobile apps with similar functionality." *2019 34th IEEE/ACM International Conference on Automated Software Engineering (ASE)*. IEEE, 2019.

- [2]: Lin, Jun-Wei, Reyhaneh Jabbarvand, and Sam Malek. "Test transfer across mobile apps through semantic mapping." *2019 34th IEEE/ACM International Conference on Automated Software Engineering (ASE)*. IEEE, 2019.

- [3]: Mariani, Leonardo, et al. "Semantic matching of gui events for test reuse: are we there yet?." *Proceedings of the 30th ACM SIGSOFT International Symposium on Software Testing and Analysis*. 2021.

- [4]: Khalili, Farideh, et al. "The ineffectiveness of domain-specific word embedding models for GUI test reuse." *Proceedings of the 30th IEEE/ACM International Conference on Program Comprehension*. 2022.

- [5]: Zhao, Yixue, et al. "Fruiter: a framework for evaluating ui test reuse." *Proceedings of the 28th ACM Joint Meeting on European Software Engineering Conference and Symposium on the Foundations of Software Engineering*. 2020.