Automated assertion generation via information retrieval and its integration with deep learning

Hao Yu, Yiling Lou, Ke Sun, Dezhi Ran, Tao Xie, Dan Hao, Ying Li, Ge Li, Qianxiang Wang

Problem Statement

Unit testing plays a crucial role in software development by validating the correctness of basic units within a software system. Manual creation of test assertions for unit tests can be time-consuming and error-prone. While deep learning approaches like ATLAS have been introduced to automate assertion generation, their effectiveness is limited. To improve the efficiency and accuracy of assertion generation, there is a need to explore the integration of Information Retrieval (IR) techniques with deep learning.

Background

Assertion
- Validate the correctness of the method

Traditional Assertion Generation
- Existing test generation tools can automatically generate assertions with two main categories:
  - Capture and assert
  - Differential testing
- Limitations
  - Detect only crashing/regression faults with generated assertions being unmeaningful
  - Require the code under test to be compiliable and executable.

ATLAS
- ATLAS takes a test method without any assertions along with a focal method (method under test) to output an assertion.
- Limitations
  - Assertions generated by ATLAS are not explainable due to unexplainable nature of DL.
  - Sequence-to-sequence models suffer exposure to bias and disappearance of gradient.

Proposed Solution

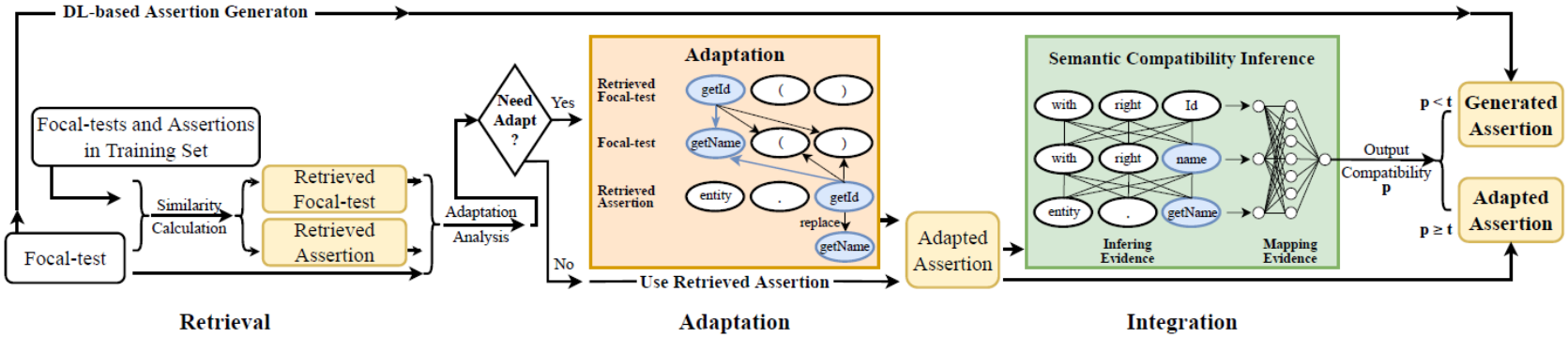


Figure 1: Workflow of the integration approach

IR-based approach

- Retrieval (IR-based Assertion Retrieval, short as IR_{ar})
- Adaptation (Retrieved-Assertion Adaptation, short as RA_{adapt})

Integration (IR-based approach + Deep Learning)

IR-based approach

1) **IR-based Assertion Retrieval (IRar)**

- The first component of the IR-based approach is the IR-based assertion retrieval (IRar) technique.
- IRar aims to retrieve the most relevant assertion for a given focal-test based on similarity measures.
- It involves tokenizing focal-tests in the training and test sets using tools like javalang.
- By calculating the Jaccard similarity coefficient between focal-tests, IRar identifies the most similar focal-test in the training set.
- The assertion corresponding to the retrieved focal-test is then selected as the output assertion for the given focal-test.

Continued...

2) Retrieved-Assertion Adaptation (RAadapt)

- Decide whether the assertion should be modified.
- Decide which token (i.e, invoked method, variable or constant) should be modified.
- Decide what value a candidate token should be replaced with.
  - Heuristics-based strategy ($RA^{H}adapt$)
    - The same sub-token after hump-split
    - The position of the token in the assertion
  - Neural-network-based-strategy ($RA^{NN}adapt$)
    - Embedding first-order semantic information
    - Embedding first-order information with high-order semantic information
    - Combining semantic information with lexical similarity

Integration

An IR-based approach retrieves assertions from the training set.

- Could be highly effective especially when there are similar cases in the training set.

A DL-based approach learns to generate assertions based on the training set.

- Is capable of generating “new” assertions absent from the training set.

An integration approach based on a compatibility inference model.

- Calculates the “compatibility” between the retrieved assertion and the current focal-test method to determine whether to directly return the retrieved assertion or apply a DL-based approach to generate a new assertion.
- The task of neural language inference is usually solved with a binary neural inference model.
- Since the assertion is retrieved from its training focal-test, but also consider the compatibility between the retrieved focal-test and the input focal-test.
- We model the local evidence of compatibility among the retrieved assertion, the retrieved focal-test, and the input focal-test with an attention-based RNN model.

Datasets

$Dataset_{old}$

- It excludes the assertions that contain okens absent from the focal-test and the vocabulary. Such tokens are called unknown tokens.

$Dataset_{new}$

- Added those excluded cases with unknown tokens back to Dataset.

Table 1: Detailed statistics of each type in $Data_{old}$ and $Data_{new}$

AssertType	Total	Equals	True	That	NotNull	False	Null	ArrayEquals	Same	other
$Data_{old}$	15,676	7,866(50%)	2,783(18%)	1,441(9%)	1,162(7%)	1,006(6%)	798(5%)	307(2%)	311(2%)	2(0%)
$Data_{new}$	26,542	12,557(47%)	3,652(14%)	3,532(13%)	1,284(5%)	1071(4%)	735(3%)	362(1%)	319(1%)	3,030(11%)

Experimental Results

RQ1: How does the proposed IR-based assertion retrieval technique IRar perform compared to the latest DL-based assertion generation approach (i.e, ATLAS)?

RQ2: How do the proposed retrieved-assertion adaptation techniques RA^{H}adapt and RA^{NN}adapt improve the effectiveness?

RQ3: How does the proposed integration approach boost DL-based and IR-based approaches?

RQ1: Effectiveness of IRar

Overall effectiveness of IRar

IRar technique significantly outperforms the DL-based approach ATLAS in terms of accuracy and BLEU scores on both datasets. The difference is more pronounced on the challenging dataset with unknown tokens, suggesting that DL-based methods may struggle with generating assertions for such cases, leading to a notable drop in accuracy.

Table 2: Accuracy of IR_{ar}, RA^{H}_{adapt}, RA^{NN}_{adapt}, and integration

Dataset \ Approach	ATLAS	IR_{ar}	RA^{H}_{adapt}	RA^{NN}_{adapt}	integration
$Data_{old}$	31.42	36.26	40.97	43.63	46.54
$Data_{new}$	21.66	37.90	39.65	40.53	42.20

Table 3: Muti-BLEU of IR_{ar}, RA^{H}_{adapt}, RA^{NN}_{adapt}, and integration

Dataset \ Approach	ATLAS	IR_{ar}	RA^{H}_{adapt}	RA^{NN}_{adapt}	integration
$Data_{old}$	68.51	71.48	73.28	73.95	78.86
$Data_{new}$	37.91	57.98	59.81	59.81	60.92

Continued...

a) Effectiveness on different assertion types

The image below describes a comparison between the effectiveness of IRar and ATLAS on various types of assertions. It is noted that IRar consistently outperforms ATLAS across all assertion types, demonstrating the versatility and effectiveness of IRar in generating a wide range of assertions.

Table 4: Detailed statistics of ATLAS and IR_{ar} for each assert type

AssertType / Approach	Total	Equals	True	That	NotNull	False	Null	ArrayEquals	Same	Other
ATLAS-$Data_{old}$	4,925(31%)	2,501(32%)	966(35%)	248(17%)	598(51%)	229(23%)	236(30%)	100(33%)	47(15%)	0(0%)
IR_{ar}-$Data_{old}$	5,684(36%)	2,957(38%)	1,039(37%)	449(31%)	439(38%)	314(31%)	285(36%)	111(36%)	89(29%)	1(50%)
ATLAS-$Data_{new}$	5,749(22%)	2,900(23%)	619(17%)	537(15%)	388(30%)	126(12%)	85(12%)	47(13%)	37(12%)	1,010(33%)
IR_{ar}-$Data_{new}$	10,059(38%)	4,664(37%)	1,436(39%)	1,070(30%)	600(47%)	394(37%)	286(39%)	147(41%)	113(35%)	1,349(45%)

Continued...

b) Effectiveness with different similarity coefficients

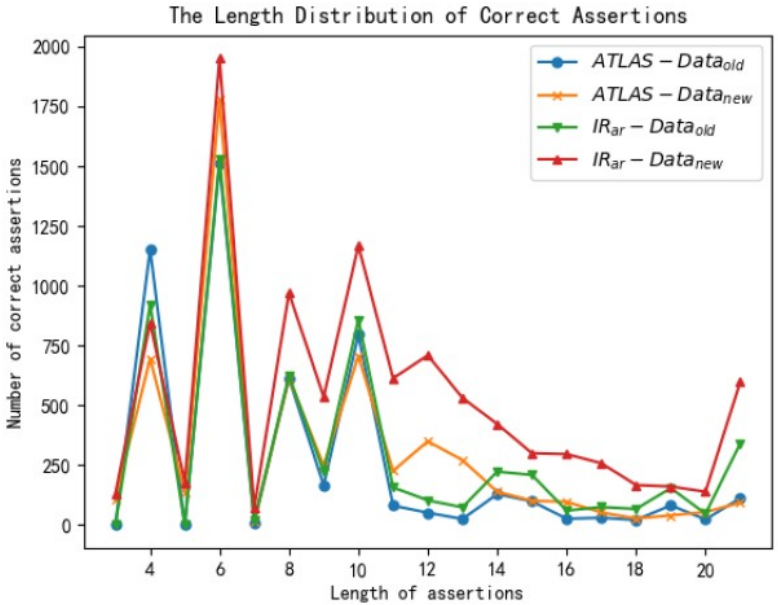
The results, presented in Table 5, indicate that varying the similarity coefficients has minimal influence on the performance of IRar, suggesting the robustness and versatility of the IRar approach in generating assertions regardless of the specific similarity metric used.

Table 5: Accuracy of different similarity coefficients

Approach / Dataset	Jaccard	DICE	Overlap
$Data_{old}$	36.26	36.26	36.12
$Data_{new}$	37.90	37.90	37.74

Continued...

Correct generated assertions



Correct assertions generated by ATLAS

- In $Dataset_{old}$, among 4,925 correct assertions that can be generated by ATLAS, 4,560 (92.59%) assertions exist in the training set if the abstraction strategy is applied.
- In $Dataset_{new}$, we also observe that the majority of correct generated assertions (i.e, 98.76%) are the assertions existing in the training set.

Continued...

Incorrect generated assertions

- The edit distance between the incorrect generated assertions and the labeled assertions.

Table 6: Edit distance between correct assertions and incorrect assertions generated by ATLAS and IR_{ar}

Dataset \ Edit	1	2	3
ATLAS-$Data_{old}$	2,840(26%)	763(7%)	914(9%)
IR_{ar}-$Data_{old}$	2,966(30%)	1,198(12%)	572(6%)
ATLAS-$Data_{new}$	2,614(13%)	1,595(8%)	1,522(7%)
IR_{ar}-$Data_{new}$	4,532(31%)	1,912(13%)	1,091(8%)

Table 7: Token types to be modified within one edit distance

Dataset \ Token	total	api	variable	constant	assertType
ATLAS-$Data_{old}$	2,840	27	865	1,171	738
IR_{ar}-$Data_{old}$	2,966	145	998	1,042	677
ATLAS-$Data_{new}$	2,614	225	482	1,051	274
IR_{ar}-$Data_{new}$	4,532	448	1,014	2,100	395

- Three categorizations of incorrect generated assertions whose edit distance is only one token away from the correct assertions.

RQ2: Effectiveness of RAadapt

Overall accuracy of RA^{H}adapt and RA^{NN}adapt

Table 8: Detailed statistics of RA^{H}_{adapt} and RA^{NN}_{adapt} for each assert type

Approach \ AssertType	Total	Equals	True	That	NotNull	False	Null	ArrayEquals	Same	Other
RA^{H}_{adapt}-$Data_{old}$	6,423(41%)	3,300(42%)	1,151(41%)	536(37%)	553(48%)	335(33%)	316(40%)	120(39%)	111(36%)	1(50%)
RA^{NN}_{adapt}-$Data_{old}$	6,839(44%)	3,509(45%)	1,225(44%)	551(38%)	610(52%)	342(34%)	341(43%)	134(44%)	126(41%)	1(50%)
RA^{H}_{adapt}-$Data_{new}$	10,525(40%)	4,882(39%)	1,487(41%)	1,142(32%)	651(51%)	403(38%)	297(40%)	154(43%)	121(38%)	1,388(46%)
RA^{NN}_{adapt}-$Data_{new}$	10,758(41%)	4,988(40%)	1,526(42%)	1,161(33%)	691(54%)	401(37%)	308(42%)	162(45%)	126(39%)	1,395(46%)

Unsuccessful cases of the adaptation technique:

- On Dataset$_{old}$, 419 and 737 incorrect retrieved assertions are modified into correct by RA^{H}adapt and RA^{NN}adapt
- On Dataset$_{new}$, 563 and 679 incorrect retrieved assertions are modified into correct by RA^{H}adapt and RA^{NN}adapt

Successful cases of the adaptation technique:

- On Dataset$_{old}$, 25 and 42 correct retrieved assertions are modified into incorrect by RA^{H}adapt and RA^{NN}adapt
- On Dataset$_{new}$, 68 and 84 correct retrieved assertions are modified into incorrect by RA^{H}adapt and RA^{NN}adapt

RQ3: Effectiveness of Integration

The complementarity between DL-based (ATLAS) and IR-based (IRar and RAadapt) approaches in generating correct assertions, showing unique strengths of each approach is discussed. The integration of both approaches is highlighted as enhancing assertion generation effectiveness, indicating the promising potential of combining DL-based and IR-based methods for automated assertion generation.

Novelty of proposed solution

- The paper introduces an innovative approach that leverages Information Retrieval (IR) techniques for assertion generation in unit testing.
- This marks a significant departure from traditional deep learning-based methods by incorporating IR principles to enhance the efficiency and accuracy of assertion generation processes.
- The construction of a comprehensive dataset that includes practical and challenging cases in assertion generation is a novel aspect of the research.
- The thorough evaluation of the IR-based approach on both existing and newly constructed datasets, showcasing significant improvements in accuracy over state-of-the-art DL-based methods, demonstrates the novelty and effectiveness of the proposed techniques.

Limitations of proposed solution

1) The proposed methods primarily focus on unit testing and may not directly apply to other types of testing or broader software engineering challenges without further adaptation and evaluation.
2) The IR-based approach relies on the availability of a comprehensive dataset of assertions for retrieval and adaptation, which might limit its applicability in environments with insufficient existing tests.
3) The adaptation technique, RA_adapt, currently focuses mainly on the replacement operation. The paper suggests that it could be extended with more edition operations, such as addition or deletion, in future work. This indicates a limitation in the current scope of the adaptation technique, which may affect the accuracy and applicability of the generated assertions.

Discussion Points

1. The study primarily uses accuracy and BLEU scores to evaluate the effectiveness of assertion generation techniques. However, these metrics may not fully capture the quality and usefulness of generated assertions in real-world testing scenarios.
2. The adaptation technique, RA_adapt, currently focuses mainly on the replacement operation. This indicates a limitation in the current scope of the adaptation technique, which may affect the accuracy and applicability of the generated assertions.
3. While the paper discusses the effectiveness of the integration approach in generating correct assertions, it does not address scalability concerns. As software projects grow in size and complexity, the computational resources and time required for assertion generation using this integrated approach may become significant.

Thank You