Improving Bug Localization with an Enhanced Convolutional Neural Network

Yan Xiao, Jacky Keung, Qing Mi, Kwabena E. Bennin
Department of Computer Science, City University of Hong Kong, Kowloon, Hong Kong

CityU 香港城市大學 City University of Hong Kong
Department of Computer Science

Outline

- Background
- Key limitation of existing techniques
- DeepLocator
- Experimental results
- Conclusions

Background

- Bug
  - coding mistakes that may induce unexpected or abnormal behaviors in the software project.
- Bug report
  - provide information about the scenarios in which the software does not behave as expected and how to reproduce this abnormal behavior.
- Bug localization
  - extract various information from the bug report, and then search through the source files to locate the potential buggy files that may be relevant to the bug.

Outline

- Background
- **Key limitation of existing techniques**
- DeepLocator
- Experimental results
- Conclusions

Key limitation of existing techniques

- Traditional features related with program analysis information
  - ➢ extract static features from the source code or execution information.
- Information Retrieval (IR) based approaches
  - ➢ measure the textural similarity between bug reports and source code (names of classes or methods).
  - ➢ this kind of information is more about the texts than semantic information.

Key limitation of existing techniques

- Machine Learning (ML) based approaches
  - match the topics of bug reports with those of source code or classify the source files to each bug report by using former fixed files.
- Deep learning based approaches
  - the recently proposed deep learning based model for bug localization is the combination of deep neural networks and IR techniques. The results are influenced by the performance of IR.
  - many DNN models are used together, which is essentially a very complex model that is hard to adjust the weights accurately.

Outline

- Background
- Key limitation of existing techniques
- **DeepLocator**
- Experimental results
- Conclusions

DeepLocator

DeepLocator is composed of an enhanced CNN considering bug-fixing experience, together with proposed rTF-IDuF and pre-trained word2vec technique.

- employ CNN, a well-known deep learning model, to understand the underlying semantical information of the bug reports and source code.
- correlate the bug reports to the corresponding buggy files instead of textural similarity used in IR-based approaches.

DeepLocator

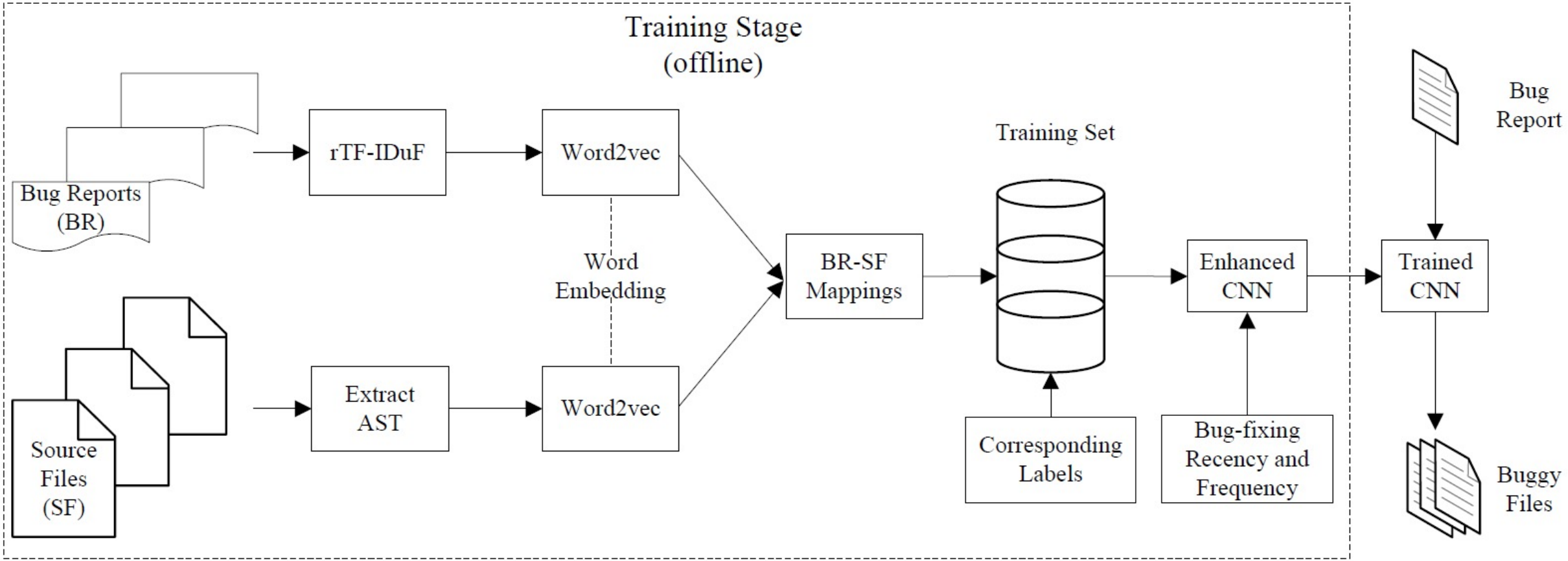

Fig. 3. The Overall Workflow of DeepLocator.

DeepLocator

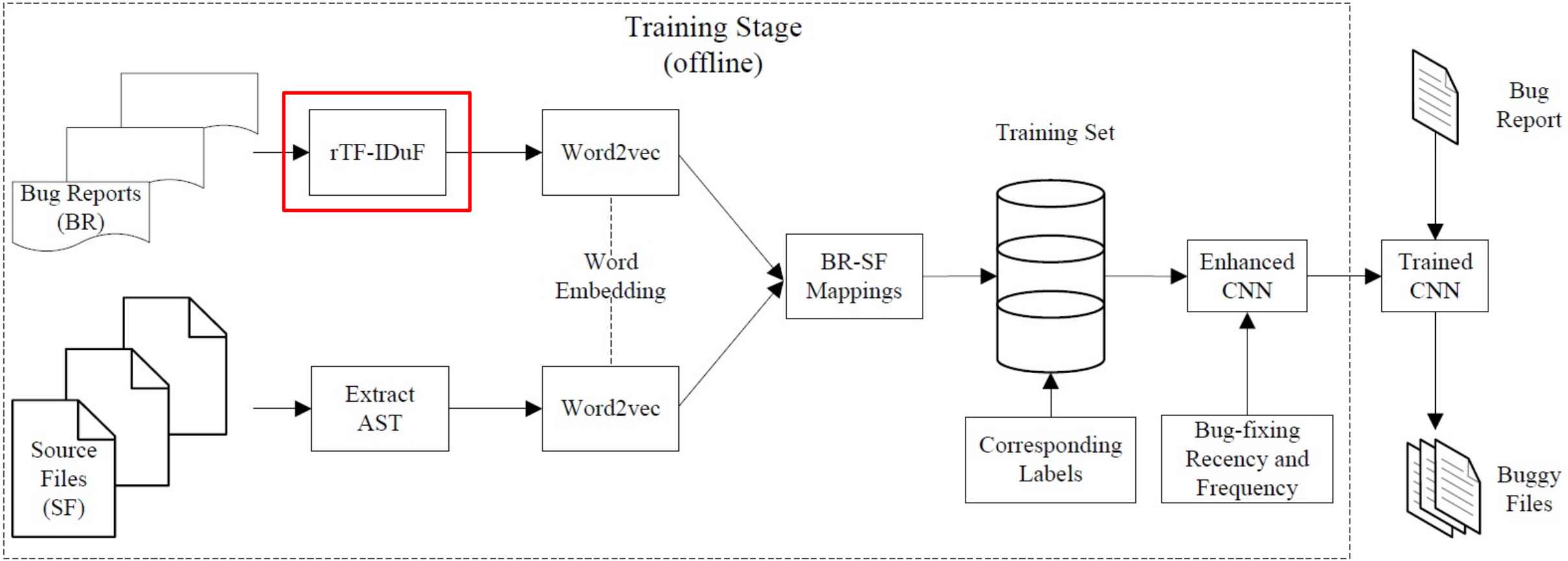

Fig. 3. The Overall Workflow of DeepLocator.

revised TF-IDuF (rTF-IDuF)

TF-IDF

$$tf_{t,d} = f_{t,d} \qquad idf_{t,d} = \log\left(\frac{N}{df_t}\right)$$

$$tf_{t,d} = \log(f_{t,d}) + 1$$

$$\omega_{tf-idf} = tf_{t,d} \times idf_{t,d}$$

TF-IDuF

$$tf_{t,d} = f_{t,d} \qquad idf_{t,d} = \log\left(\frac{N_u}{n_{t,u}}\right)$$

$$\omega_{tf-idf} = tf_{t,d} \times idf_{t,d}$$

rTF-IDuF $\qquad \omega_{tf-idf} = (\log(f_{t,d}) + 1) \times \log\left(\frac{N_u}{n_{t,u}}\right)$

DeepLocator

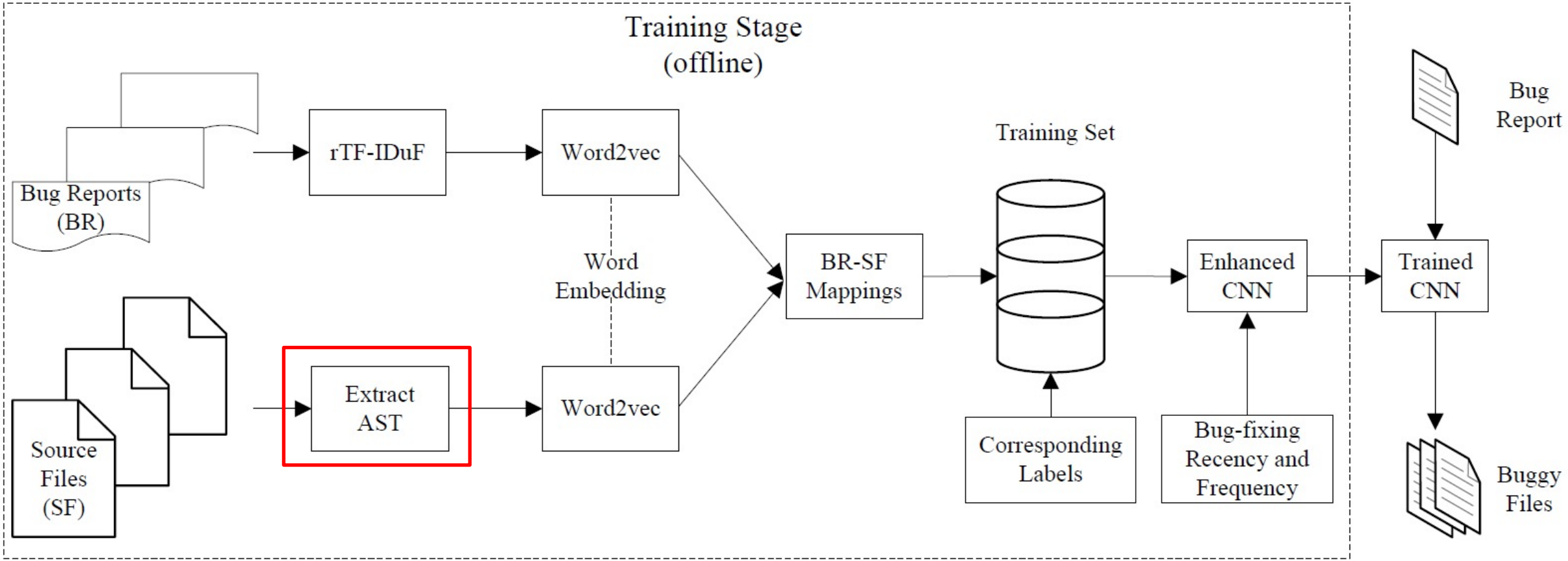

Fig. 3. The Overall Workflow of DeepLocator.

DeepLocator

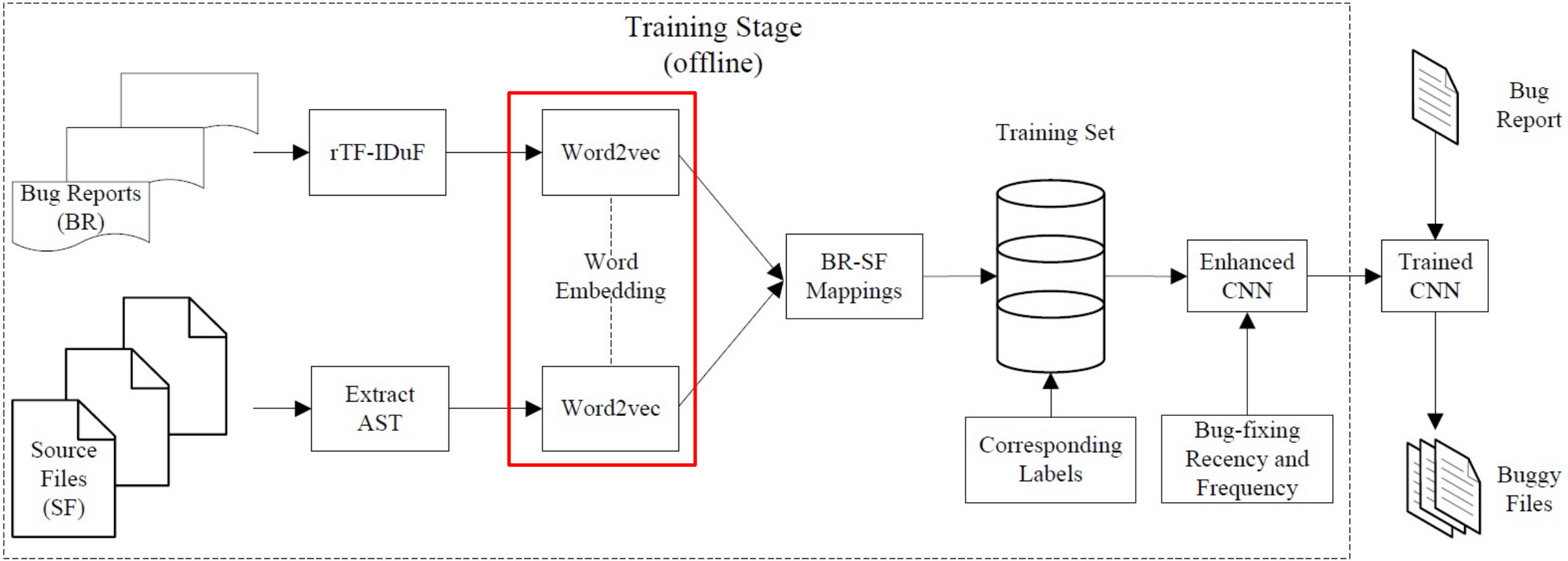

Fig. 3. The Overall Workflow of DeepLocator.

DeepLocator

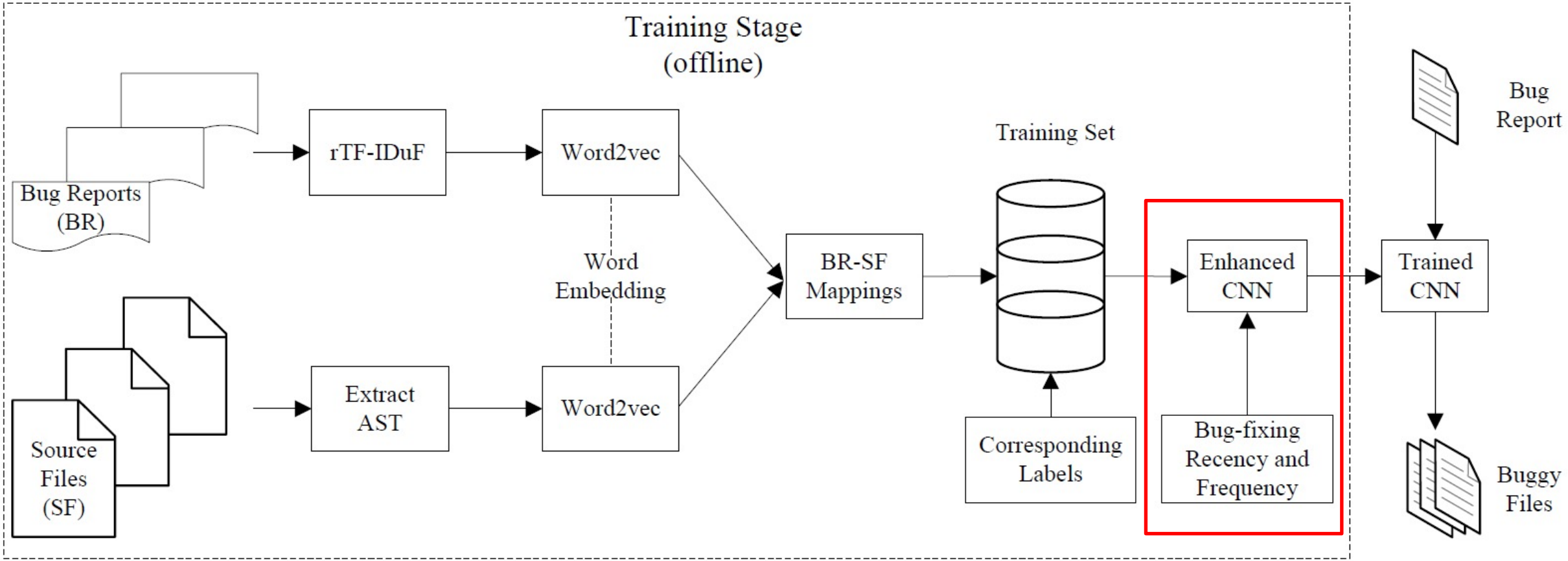

Fig. 3. The Overall Workflow of DeepLocator.

Enhanced CNN

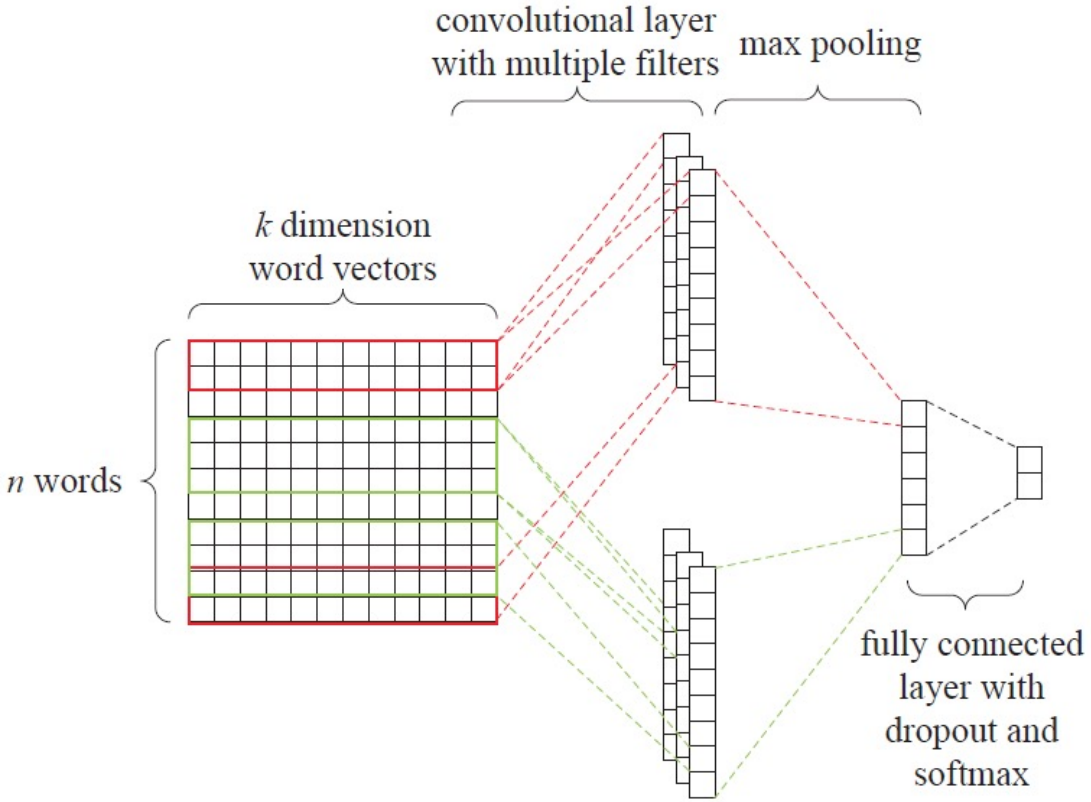

Fig. 4. The Architecture of CNN for a Sentence. n is the number of words in the sentence and k is the vector dimension of each word that is tuned experimentally (details are shown in Figure 5).

Enhanced CNN

Original

$$R_i = \frac{1}{r.month - s.month + 1}$$

$$F_i$$

Scaled

$$r_i = \begin{cases} 0 & if\ R_i < R_{min} \\ \frac{R_i - R_{min}}{R_{max} - R_{min}} & if\ R_{min} < R_{max} \\ 1 & if\ R_i > R_{max} \end{cases}$$

$$f_i = \begin{cases} 0 & if\ F_i < F_{min} \\ \frac{F_i - F_{min}}{F_{max} - F_{min}} & if\ F_{min} < F_{max} \\ 1 & if\ F_i > F_{max} \end{cases}$$

$$L(\theta) = \frac{1}{N}\sum_{i=1}^{N} cost_i$$

$$cost_i = -\sum_{j=1}^{T} t_{ij}\log(y_{ij})$$

$$cost_i = -\sum_{j=1}^{T} t_{ij}\log(y_{ij}) - \omega_1 r_i - \omega_2 f_i$$

DeepLocator

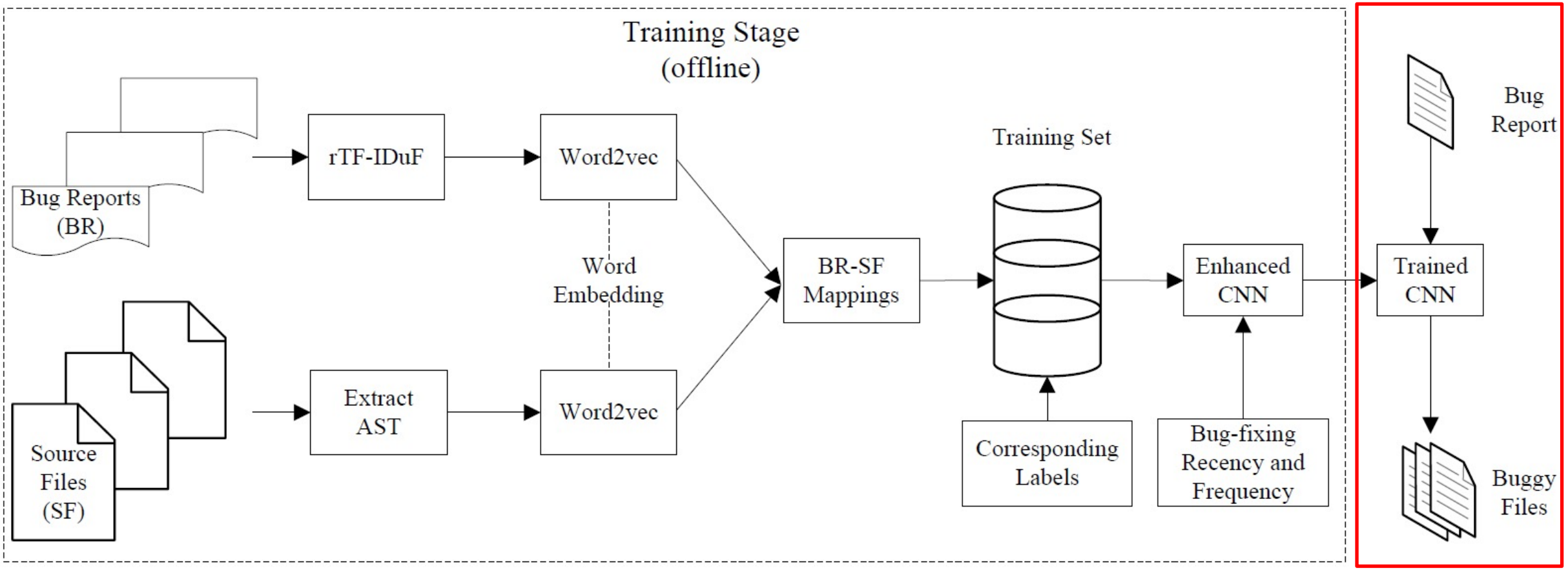

Fig. 3. The Overall Workflow of DeepLocator.

Outline

- Background
- Key limitation of existing techniques
- DeepLocator
- **Experimental results**
- Conclusions

Experimental dataset and evaluation metrics

Table I Subject Projects

Project	Time Range	# of Bug Reports	# of Fixed Buggy Files	Avg. # of Buggy Files per Bug
AspectJ	03/02-01/14	593	1,151	4.0
Eclipse UI	10/01-01/14	6,495	6,228	2.7
JDT	10/01-01/14	6,274	5,002	2.6
SWT	02/02-01/14	4,151	1,415	2.1
Tomcat	07/02-01/14	1,056	1,038	2.4

Metrics

$$P = \frac{true\ positive}{true\ positive + false\ positive}$$

$$R = \frac{true\ positive}{true\ positive + false\ negative}$$

$$F = \frac{2 * Precision * Recall}{Precision + Recall}$$

$$MAP = \frac{1}{Q}\sum_{i=1}^{Q}\frac{AvgP(i)}{Q}$$

Questions

- What is the effect of rTF-IDuF and pre-trained word2vec on DeepLocator?
- What is the effect of model settings on DeepLocator?
  - the size of filters
  - the number of convolutional layers
- What is the performance of DeepLocator?
  - Enhanced CNN vs Conventional CNN
  - DeepLocator vs HyLoc

Property of rTF-IDuF.

AspectJ
Bug ID: 415266
Summary: Bug 415266 LTW not working when JMX is enabled.
Description: When I enable JMX remote management on a JVM along with AspectJ load-time weaving (LTW), our Aspect doesn't appear to get woven in. This are the JVM arguments ...

Eclipse UI
Bug ID: 201616
Summary: Bug 201616 [Min.Max] Need to be able to change the orientation of a minimized stack.
Description: We currently use the aspect ratio of the existing view stack to set the orientation to be used when showing the view as fast views. While this works in most cases there are enough ...

Fig. 6. Bug Reports that Contain 'aspect' in Project AspectJ and Project Eclipse UI.

Table II Term Weights of 'aspect'

	TF-IDF		rTF-IDF	
DeepLocator	Bug 415266 in AspectJ	Bug 201616 in Eclipse UI	Bug 415266 in AspectJ	Bug 201616 in Eclipse UI
'aspect'	7.36	3.09	1.48	6.99

Impacts of Pre-trained Word2vec.

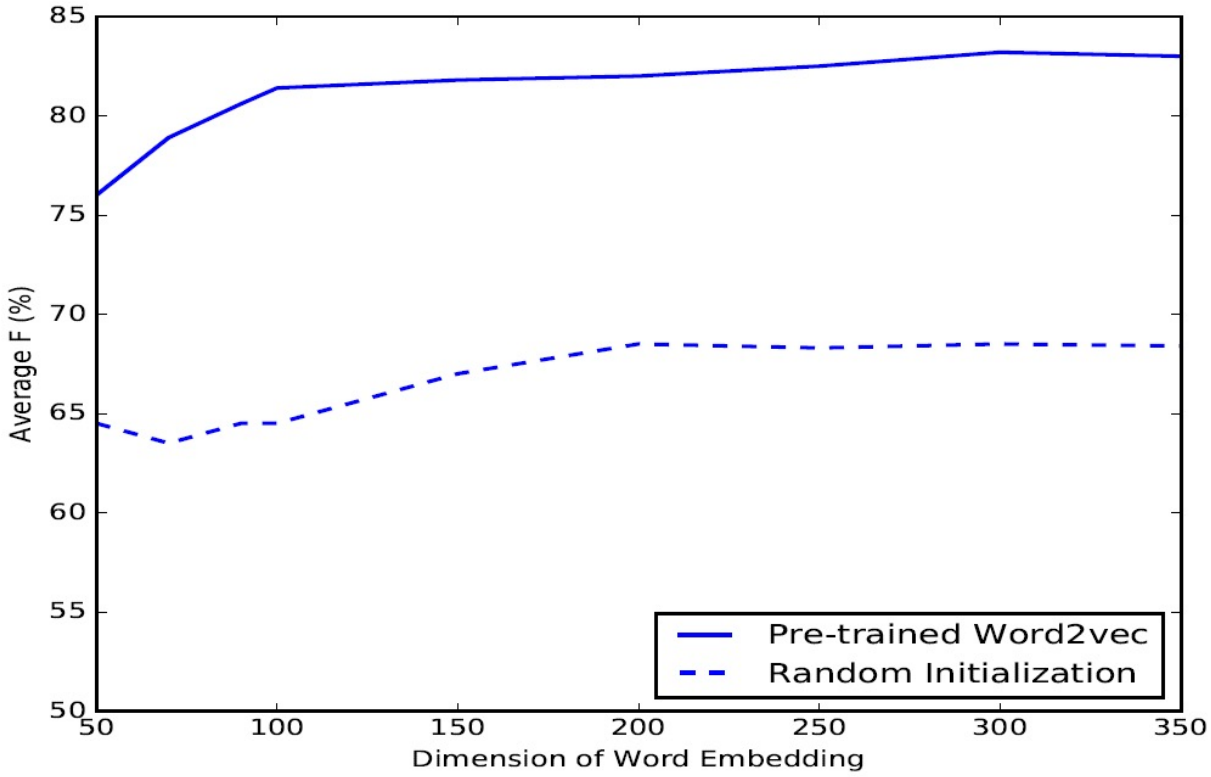

Fig. 7. Performance of Word2vec and Random Initialization along Dimension of Word Embedding.

Accuracy Under Different Model Settings.

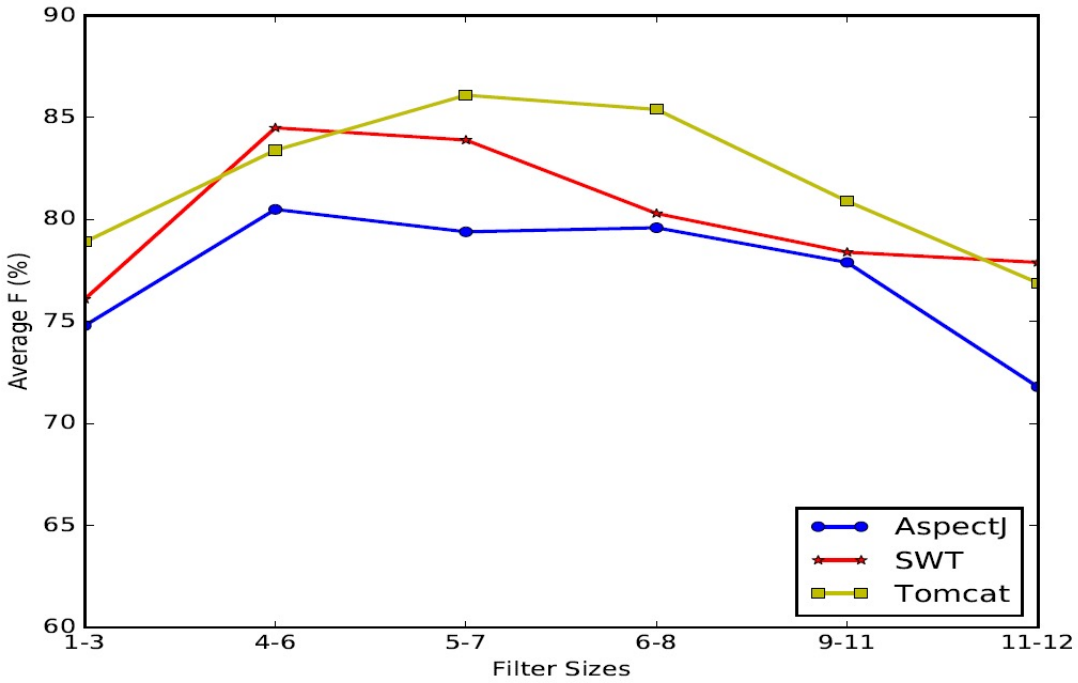

Fig. 8. Performance of Different Filter Sizes.

Table II Accuracy and Time Consumption on Different Number of Layers

Project	One-layer		Two-layer	
	F (%)	Time (s)	F (%)	Time (s)
AspectJ	79.92	8.69	81.16	692.26
Eclipse UI	84.13	8.83	85.58	689.83
JDT	82.28	8.78	83.21	702.18
SWT	83.61	7.93	84.71	682.61
Tomcat	85.04	7.69	85.73	696.67

Performance of Enhanced CNN.

Table III Accuracy of Two Models

Project	Model	P (%)	R (%)	F (%)	F_max (%)	F_min (%)
AspectJ	DeepLocator	77.37	80.23	79.92	81.21	77.94
	CNN	57.83	53.49	55.57	60.42	52.79
Eclipse UI	DeepLocator	82.37	84.58	84.13	85.96	79.19
	CNN	55.35	61.62	59.29	67.17	49.22
JDT	DeepLocator	83.43	79.92	82.28	85.41	78.48
	CNN	58.79	61.46	60.85	65.49	54.23
SWT	DeepLocator	82.33	84.98	83.61	84.92	79.95
	CNN	73.37	75.49	73.84	76.39	69.98
Tomcat	DeepLocator	83.35	87.21	85.04	86.29	83.13
	CNN	58.01	58.62	58.39	63.59	52.81

Performance of Enhanced CNN.

Table IV Prediction Time (seconds) per Bug Report of Three Models

Prediction Time	AspectJ	Eclipse UI	JDT	SWT	Tomcat
DeepLocator	25.7	29.9	45.1	9.1	7.9
CNN	26.4	28.3	43.6	9.2	7.7
HyLoc	144.0	126.0	198.0	108.0	60.0

Table V MAP of DeepLocator and HyLoc on Five Projects

MAP	AspectJ	Eclipse UI	JDT	SWT	Tomcat
DeepLocator	0.34	0.42	0.45	0.40	0.54
HyLoc	0.32	0.41	0.34	0.37	0.52

Why does DeepLocator work?

- Abstract Syntax Tree (AST)
  - represent syntax and extract programming patterns of source code.
- Word embedding
  - map words into semantic vector space where similar words are close to each other.
- Enhanced CNN
  - correlate bug reports to buggy files including both linear and nonlinear relationships.
  - CNN model performs well in the semantic parsing field because of the convolving filters.

Outline

- Background
- Key limitation of existing techniques
- DeepLocator
- Experimental results
- **Conclusions**

Conclusions

- This paper proposed a deep learning based model, DeepLocator, for bug localization, which consisted of enhanced CNN, together with rTF-IDuF and pretrained word2vec.
- Some suggestions about how to build the model:
  - the necessity of using rTF-IDuF and pre-trained word2vec
  - filter size
  - the number of convolutional layers
- A set of experiments are conducted to validate the feasibility and effectiveness of DeepLocator for bug localization.

CityU
香港城市大學
City University of Hong Kong

Department of
Computer Science