VI. ANNEX

Our tool has been successfully applied in ZTE's requirement review system. Due to the company policy, it is inconvenient to open the source code of MRDQA and the dataset. We describe MRDQA's building process in detail so that readers can adapt it to other software projects.

A. Dataset Labeling

We invited seven ZTE requirement review experts with extensive professional experience in this field. Each requirement document is randomly assigned to three experts for labeling. In order to ensure the quality of the labels, the experts abide by the following criteria.

1) Correctness Whether the description of the requirement is consistent with the goal of the product.
2) Unambiguity Whether the description is clear and unambiguous.
3) Scenarios Whether the application scenarios are described completely.
4) Interfaces Whether the user interface and interfaces between modules are defined.
5) Acceptance criteria Whether the specifications of function, performance, safety, and reliability are included.

A requirement document is accepted as a valid sample if the corresponding three experts give the same labels. Otherwise, it will undergo a new round of labeling. If the results are still inconsistent, the three experts will discuss it together to give the final label.

B. Content Parser

In order to extract the visual and textual content of requirement document page, we developed a content parser. The parser takes the URL of a requirement web page as input and outputs its textual features and a 1000x2000 pixel screenshot with the irrelevant information (e.g., the ZTE logo) removed.

The textual features include completeness, element statistics and metrics of the requirement web pages. The completeness is measured by a pre-defined template, which contains all the necessary elements, including Scenario, Function, Performance, etc. The element statistics include number of words in each element, number of figures, number of tables, etc. The metrics about the requirement web page include number of view times, number of viewers, number of editors, etc. All the textual features are shown in Table II. The "No." is the abbreviation of "Number of".

C. MRDQA Training

1) Data Preparation: We enlarge the dataset for training and validation. Specifically, we adopt the nearest filling method to augment the image data with a width shift range of 0.05, a height shift range of 0.05 and a zoom range of 0.05. Image augmentation is only for the screenshot, while the corresponding textual features are copied directly. Then we apply 90% of the augmentated dataset as training set and 10% as validation set. Detailed statistics are shown in Table III.

TABLE II
TEXTUAL FEATURES.

Completeness	Element Statistics	Web Page Metrics
Scenario	No. Words in each element	No. View times
Function	No. Figures	No. Viewers
Performance	No. Tables	No. Editors
Safety	No. Hyperlinks	No. Edit times
Reliability		No. Comments
User interface		No. Downvote
Module interface		No. Upvote
Acceptance criteria		
Release version		
Revision record		

TABLE III
DATASET STATISTICS.

Dataset	High-quality	Medium-quality	Low-quality	Total
Training	1071	855	774	2700
Validation	119	95	86	300
Test	50	42	36	128

2) Model Training: The pre-trained Efficientnet-B2 and Efficientnet-B3 models of the visual cognition subnetwork are fetched from the internet[2]. They are used to obtain the visual rendering representation, which is a 2,688 dimensional vector. Then the SVD layer is applied to extract the most significant 256 dimensions. In the textual characteristics subnetwork, the embedding layer has size of 64 and then we use a multi-head self-attention layer with 2 heads and hidden size of 32 for each head. We adopt Adam optimizer with a learning rate of 0.0001 and a batch size of 16. Besides, we adopt the early stop training strategy to prevent overfitting, where we stop training if the accuracy does not increase for 10 epochs.

D. Definition of Metrics

In this section, we define the metrics used in the previous experiments. Table IV shows the confusion matrix of MRDQA on the 128 test samples. Label_h represents that a requirement document's label is high and Prediction_h represents that MRQDA predict that a requirement document's quality is high.

TABLE IV
CONFUSION MATRIX OF MRDQA.

	Label_h	Label_m	Label_l
Prediction_h	**40**	7	0
Prediction_m	9	**31**	3
Prediction_l	1	4	**33**

[2]https://www.kaggle.com/chopinforest1986/efficientnetb0b7-keras-weights

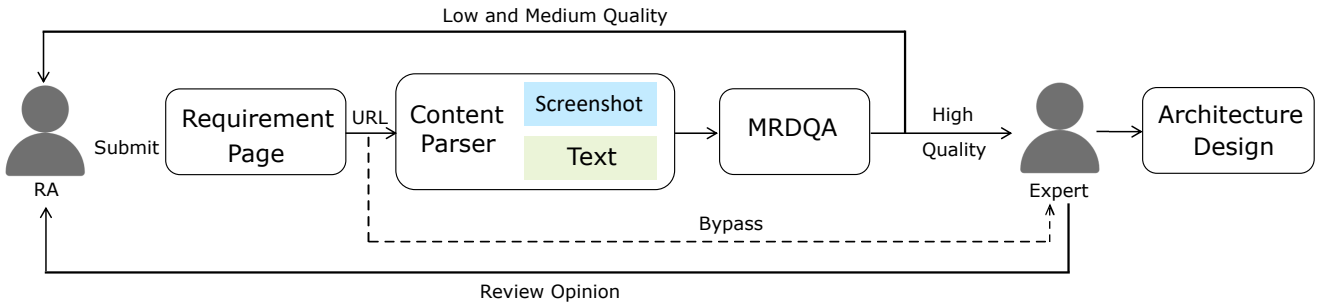


Fig. 2. The Application Solution of MRDQA.

We define T_{hh}: the prediction value (the first subscript letter) is of high quality and the label (the second subscript letter) is high, the prediction result is correct. Similarly, we can get T_{mm} and T_{ll}. We define F_{hm}: the prediction value is of high quality but the label is medium, the prediction result is wrong. By this way, we can get F_{hl}, F_{mh}, F_{ml}, F_{lh}, and F_{lm}. The N denotes the total number of samples in the test dataset. The Accuracy, Precision, and Recall are defined as follows. $Precision_h$ represents the precision of high quality category.

$$Accuracy = \frac{(T_{hh} + T_{mm} + T_{ll})}{N} \tag{1}$$

$$Precision_h = \frac{T_{hh}}{T_{hh} + F_{hm} + F_{hl}} \tag{2}$$

$$Precision_m = \frac{T_{mm}}{T_{mm} + F_{mh} + F_{ml}} \tag{3}$$

$$Precision_l = \frac{T_{ll}}{T_{ll} + F_{lh} + F_{lm}} \tag{4}$$

$$Recall_h = \frac{T_{hh}}{T_{hh} + F_{mh} + F_{lh}} \tag{5}$$

$$Recall_m = \frac{T_{mm}}{T_{mm} + F_{hm} + F_{lm}} \tag{6}$$

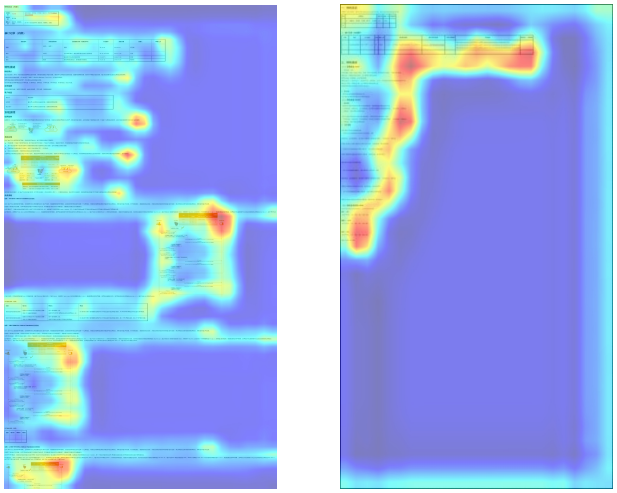
$$Recall_l = \frac{T_{ll}}{T_{ll} + F_{hl} + F_{ml}} \tag{7}$$

E. The Application Solution of MRDQA

Our purpose of MRDQA is to accelerate the requirement review process and save labor costs, so we embedded MRDQA into the process as a pre-filter. For the low and medium quality requirements, MRDQA automatically send the prediction result to the Requirement Analyst (RA), who can make modification and submit again if accept the result, otherwise push the original requirement to the review expert directly. For the high quality requirements, it passes them to the experts who will give RA review opinions if the requirements should be improved. Finally, The qualified requirement documents are used for the architecture design. In this way, all the requirement documents review by the experts are of high quality, which reduces a lot of workload and improves review efficiency. Fig. 2 shows the application solution of MRDQA.

F. How The Visual Information Works?

To better understand how the visual information works, we generated the class activation map for visualizing the discriminative regions for classification of MRDQA-V. Two examples of class activation map are illustrated in Fig. 3.



(a) High quality requirement (b) Low quality requirement

Fig. 3. Class activation maps of two example requirement documents with different quality labels.

The brighter the color is in the figure, the more attention the model pays to. In Fig. 3(a), there are quite a few discriminative regions corresponding to various presentational forms, e.g., the multiple lines of the tables on the left top, the figures and diagrams on the middle and bottom areas. So MRDQA-V can successfully learn the layout of high quality requirement document. On the contrary, In Fig. 3(b), the range of attention is relatively narrow. Obviously, its content is very abridged, lacking of concrete description, figures, or diagrams. MRDQA-V mainly focused on the right and bottom of the requirement content, capturing the short vertical and horizontal lengths of its content, which are the usually representatives of low quality requirement documents.