HAIF: A Hierarchical Attention-Based Model of Filtering Invalid Webpage

Chaoan ZHOU†, Member, Jianping ZHAO†, Tai MA†, and Xin ZHOU†, Nonmembers

SUMMARY In Internet applications, when users search for information, the search engines invariably return some invalid webpages that do not contain valid information. These invalid webpages interfere with the users’ access to useful information, affect the efficiency of users’ information query and occupy Internet resources. Accurate and fast filtering of invalid webpages can purify the Internet environment and provide convenience for netizens. This paper proposes an invalid webpage filtering model (HAIF) based on deep learning and hierarchical attention mechanism. HAIF improves the semantic and sequence information representation of webpage text by concatenating lexical-level embeddings and paragraph-level embeddings. HAIF introduces hierarchical attention mechanism to optimize the extraction of text sequence features and webpage tag features. Among them, the local-level attention layer optimizes the local information in the plain text. By concatenating the input embeddings and the feature matrix after local-level attention calculation, it enriches the representation of information. The tag-level attention layer introduces webpage structural feature information on the attention calculation of different HTML tags, so that HAIF is better applicable to the Internet resource field. In order to evaluate the effectiveness of HAIF in filtering invalid pages, we conducted various experiments. Experimental results demonstrate that, compared with other baseline models, HAIF has improved to various degrees on various evaluation criteria.

key words: invalid webpage filtering, attention mechanism, hierarchical-level attention calculation, webpage features, deep learning

1. Introduction

When users use search engines to retrieve information, for the influence of business, user clicks, and poor regulation, a host of returned result webpages that do not meet the users’ search goals will be included in the returned list, such as recruitment advertisement webpages, service failure webpages, etc. Such the webpages that do not contain useful information are called invalid webpages in this paper. The webpages that contain valuable information are called valid webpages. Users have different invalid webpage definition standards for diverse demand information. For example, when searching for geographic information, users will regard webpages that do not have geographic information (such as news information, service failures, advertisements, etc.) as invalid webpages, but when searching for surrounding news, they will regard news information webpages as valid webpages. Automatically filtering invalid webpages can enable netizens to obtain beneficial information more intuitively, save time and cost in information retrieval, and reduce waste of Internet resources. For invalid web filtering tasks, the following problems urgently need to be solved: 1) choosing which technology to better build the model can ensure the accuracy and applicability of the filtered webpages; 2) The webpages have complex structural information and semantic information, how to construct reasonable webpage features to describe webpage information.

Invalid Webpage Filtering (IWF) model is a learner that belongs to text classification task. When the computing ability of computers is not well developed, many text classification models based on kernel functions have been proposed [1], [2]. Such the models have limited feature learning capabilities, so it is arduous to capture complex and obscure information in webpages. With the improvement of hardware and the deepening of algorithm research, the models based on deep learning were proposed and constructed. Kim applied Convolutional Neural Networks (CNN) to the text classification task of Natural Language Processing (NLP). The experimental results demonstrate that the classification model based on deep learning shows excellent performance compared to the kernel functions [3]. With the proposal of CNN classification model, more and more deep neural network models [4]–[8] have achieved gratifying results in the task of text classification.

HyperText Mark-up Language (HTML) is currently the most widely used language on the Internet, and it is also the main language that constitutes web documents. HTML documents are the text data that browsers parse to display information on webpages. They are the basis of webpages. HTML documents contain HTML tags and plain text. These data are used to describe webpage information. Different HTML tags and their corresponding plain text data have different description directions for webpage information [9]. Using HTML data for web mining [10] has been a research hotspot. In many research works, HTML tags and plain text data are constructed with different features [11], [12]. If the IWF models take into account the semantics and sequence information of HTML plain text, and constructs and analyzes the HTML tag features reasonably, the webpage features can be better extracted and suitable for the Internet application environment. Therefore, this paper proposes an IWF model combining webpage features and hierarchical attention mechanism called HAIF. The novel works of HAIF are as follows:

(1) Model structure based on deep learning: Com-
pared with the kernel functions, deep neural networks have stronger feature learning ability and high accuracy of discrimination. HAIF has strong robustness, fault-tolerance to noisy data and positive generalization.

(2) Text embeddings combining semantic and sequence features: In terms of webpage feature construction, lexical-level embeddings are used to represent lexical semantics of text, and paragraph-level embeddings are used to represent text sequence features. Unlike feature construction based on manual construction or external knowledge, the pre-trained word embedding models are used to achieve text sequence feature construction, which is suitable for increasingly updated Internet data resources.

(3) Local-level attention calculation: Perform local-level attention calculation on input embeddings. Increase the attention to local feature information with large influence on the prediction result, and reduce the attention to local feature information with little influence on the prediction result, so that the features representation of plain text data are more reasonable and sufficient.

(4) Tag-level attention calculation: According to different webpage tag categories, the feature matrix of hidden information is divided to implement tag-level attention calculation. Tag-level attention calculation enables HAIF to introduce webpage structural features on the basis of local-level attention calculation and deep learning feature extraction, thereby making HAIF’s analysis and discrimination more applicable to Internet application environment.

The rest of this paper consists of four sections. Section 2 introduces the related work and selection of core methods. Section 3 introduces HAIF model, technical details, and proposes a baseline model. The baseline model is based on ensemble learning and local-level attention calculation, which we call EACNN. Section 4 states the experimental verification, in which HAIF was evaluated using the baseline models. Section 5 summarizes the research achievement and suggests the future works.

2. Related Work

To purify Internet resources and filter invalid webpages, many scholars and organizations have carried out related research work on webpage text analysis techniques. Webpage analysis technology based on HTML text data mainly involves two key links: webpage feature representation and analysis model establishment.

Traditional web feature design methods are mostly based on people’s professional experience. Sheu JJ et al. distinguished medical webpages from erotic webpages by manual designing webpage features combined with decision tree algorithm, and filtered erotic webpages [13]. Kan MY et al. used URL as features to achieve fast webpage classification, which is an experience-based method [14]. Since webpages contain a large amount of unstructured information that is frequently updated, it is very difficult to analyze webpages in a way that completely relies on manual feature design. Word embedding technology can implement language model and representation learning technology in NLP, and can be introduced in webpage feature representation. Mikolov T et al. proposed the word2vec [15] tool and PV-DM [16] method for constructing word embeddings and paragraph vector. These methods have achieved significant advantages in the performance of all NLP downstream tasks.

On the basis of manual feature construction, the introduction of text feature representation based on word embedding technology can better describe webpage information, so that the representation of webpage has better generalization and is more suitable for complex, diverse and fast-updated Internet data resources.

In terms of establishing web analysis models, early work was mainly based on traditional machine learning models. Chau M et al. took Web content and Web structure as features, and filter invalid webpages based on Support Vector Machine (SVM) [17]. Kamarudin A et al. used Naive Bayes (NB) to build a classifier to analyze webpage text to effectively filter the invalid content of Internet [18]. Kikuchi H et al. proposed an algorithm of fuzzy clustering specialized for Webpage directory [19]. Models established solely by statistical learning methods are week in learning hidden features, and it is difficult to generate highly accurate and robust webpage analysis models in the face of complex and diverse Internet data resources. In recent years, the models based on deep learning manifest better performance on this task. Afzal MZ et al. classified documents based on CNN [20]. Buber E et al. built a webpage classification model based on the deep learning architecture of Recurrent Neural Network (RNN) and took the meta tag information of title, description and keywords as features [21]. Sumathi S et al. proposed a model based on random forest and deep neural networks to predict spam emails in Internet data resources [22]. Dong L et al. designed an adaptive RNN called AdaRNN for target-dependent sentiment classification in Twitter [23]. These works utilize neural networks to capture the latent features of webpage text, and the recognition effect in documents or specific webpages is superior. Attention-based CNN (ABCNN) [7] can be applied to NLP text classification and shows better than the CNN-based model. The CNN model based on the hierarchical attention mechanism studied by Luong M et al. which considered the extraction of two-level features and have a superior performance [24]. These research shows that attention mechanism can be utilized to further enhance the model’s feature representative effect and play the role of optimizing the “understanding” of models.

In view of the above research, we use word embedding technology to represent the feature information of webpage text, and propose a webpage analysis model based on multi-level attention calculation and deep learning.

3. HAIF Model

In this section, we describe the proposed HAIF. First, it gives an overview of HAIF, and introduces the task definition and notation. Next, the details of the embedding layer,
3.1 Overview of HAIF

The IWF model is proposed in this paper, called HAIF. On the basis of deep learning, HAIF utilizes different levels of embeddings to represent the semantic and sequence information of plain text. In addition, HAIF utilizes hierarchical attention calculation to enhance its ability to extract plain text and HTML tag features. As shown in Fig. 1, HAIF mainly includes the following contents:

**Embedding Layer**: applies pre-trained embedding models to transform the original input text sequence into embeddings. Embeddings are used as model input. Embeddings are composed of lexical-level embeddings and paragraph-level embeddings.

**Local-level Attention Layer**: the local-level attention is calculated on input embeddings, the local attention and embeddings information are concatenated, a new feature matrix is generated, and the feature description ability of text semantics is improved.

**Convolution Layer**: performs convolution operation on the feature matrix to extract hidden information.

**Tag-level Attention Layer**: assigns different attention weights to convolutional matrices at different webpage tag levels, realizes tag-level attention calculation, and improves HAIF’s ability to represent webpage HTML tag features.

**Maxpooling Layer**: is utilized to re-extract features and improve features robustness.

**Fully Connected Layer & Output Layer**: applies sigmoid function to obtain the judgment results of whether the webpage needs to be filtered.

**Task definition and notation**: According to HTML tag categories, the webpage $wp = [t_1, t_2, \ldots, t_m]$ is divided into $m$ sections of tags’ plain text, each text sequence $t_j = \{w_{j1}, w_{j2}, \ldots, w_{jn}\}$ contains $n$ words, and $w_k$ represents the $k$-th word in text sequence $t$. The goal of task is to obtain the prediction result of whether the input webpage $wp$ is an invalid webpage. The prediction results are divided into two cases: valid page and invalid page.

3.2 Embeddings

Embedding layer builds the input embeddings by combining word2vec embeddings [15] and distributed memory model of paragraph vectors (PV-DM) [16] (Fig. 2). Word2vec model maps each word to a vector, which can be used to express the relationship between word-to-word. PV-DM applies an unsupervised method to generate text paragraph vectors of arbitrary length, and takes into account word order when extracting the semantics between words and words of word vectors. This enables the extraction of sequential features of text data of different lengths from webpages.

Text sequence $t$ consists of $n$ words, denoted as $t = \{w_1, w_2, \ldots, w_n\}$, each word $w_k$ can be transformed into a vector by pre-trained word2vec embedding model, $V$ is the fixed-sized vocabulary, $d$ is the dimension of word embedding. The word embedding of $w_k$ is represented as:
\[ \text{ew}_k = Ev^k \quad (1) \]

\( E \) represents the word embedding model pre-trained by Skip-gram [15], which is used to represent the features of the words. \( v^k \) represents the a one-dimensional vector used to represent the index of words in \( E \). \(|V|\) is the number of words in the word embedding model and is also the dimension of \( v^k \). For example, the position vector \( v^1 \) of the first word \( w_1 \) is \([1,0,\ldots,0] \). The PV-DM embeddings of text sequence \( t \) is represented as \( ep \). The embeddings of \( j \)-th tag’s plain text is finally represented as \( \text{emb}_j = \{\text{ew}_{j1}, \text{ew}_{j2}, \ldots, \text{ew}_{jn}, \text{ep}\} \). \( ep \) and \( \text{ew}_{jk} \)'s vector dimension is \( d \). The input embeddings of the webpage \( wp \) is represented as \( \text{wp}_{\text{emb}} = \{\text{emb}_{1}, \text{emb}_{2}, \ldots, \text{emb}_{m}\} \), and the number of tag categories is \( m \).

### 3.3 Local-Level Attention Layer

Local-level attention layer introduces attention mechanism to calculate the importance of input embeddings’s local-level information. Different local-level features are given different weights according to the influence of information on the model. The calculation equations of local-level attention are shown in Eq. (2)-(4).

\[
m_{jk} = \text{tanh}(\omega_{jk}^T + b_j) \quad (2)
\]

\[
a_{jk} = \text{softmax}(m_{jk}^T A_{\text{local}}) \quad (3)
\]

\[
x_j = \sum_k a_{jk} e_j \quad (4)
\]

First, the local-level attention layer linearly transforms the embedding based on tanh function. Next, the local-level attention layer calculates the importance (\( a_{jk} \)) of each local-level feature based on the softmax function. Finally, the local-level attention calculation is completed by weighted average of the embeddings. \( e_j \) represents the embedding of local-level feature, where \( n \) is the number of words tag \( j \), \( \omega \) and \( b_j \) represent weight matrix and bias matrix respectively. \( A_{\text{local}} \) is the parameter matrix of local-level attention to be trained. \( a_{jk} \) represents the local-level attention weight of \( e_{jk} \). \( x_j \) represents the output matrix after performing local-level attention weighting on \( j \)-th tag of webpage, which shape is \((n + 1) \times d\).

### 3.4 Convolutional Layer

Convolutional layer extracts text features by performing a convolution operation on the input matrix \( x_j \). Convolutional layer learns data features by slide windows. The set of vectors passing through the \( o \)-th \((0 \leq n + 1)\) slide window is represented as \( X_{jo} = \{x_{jo}, x_{j(o+1)}, \ldots, x_{j(o+p-1)}\} \), where the slide window size is \( p \). The convolution operation of HAIF is as follows:

\[
x_{jo}^{\text{con}} = f(X_{jo} \odot \omega + b) \quad (5)
\]

Each window vector \( X_{jo} \) obtains the weight through the slide windows. In Eq. (5), \( \omega \) is the weight matrix, \( b \) is the bias value, \( f \) represents the non-linear activation function ‘RELU’. The equation of ‘RELU’ is as follows:

\[
f(x) = \max(0, 0.01x, x) \quad (6)
\]

\( x_{jo}^{\text{con}} \) represents the hidden information of the \( j \)-th HTML tag of the webpage after convolution, which is constructed by concatenating the output result vector of each slide window. The feature matrix of webpage \( wp \) is represented as \( X_{\text{con}} = \{x_{1}^{\text{con}}, x_{2}^{\text{con}}, \ldots, x_{m}^{\text{con}}\} \), and \( m \) is the number of tag categories. The convolutional layer of HAIF uses convolution slide windows with sizes of 3, 4, and 5, respectively. Each window has 100 filters to reduce the possibility of losing important signals.

### 3.5 Tag-Level Attention Layer

To reduce the weight of irrelevant tags’ information and increase the weight of influential tags’ information, HAIF adds the tag-level attention layer. Tag-level attention layer is used to calculate the attention degree of webpage tag information. The convolutional feature matrix set of a webpage is represented as \( X_{\text{con}} = \{x_{1}^{\text{con}}, x_{2}^{\text{con}}, \ldots, x_{m}^{\text{con}}\} \). \( x_{jo}^{\text{con}} \) (\( j < m \)) represents the feature matrix of the text data in the webpage’s \( j \)-th tag after the feature extraction of the convolutional layer, which is used to describe the hidden information of text of HTML tags for webpages. The tag-level attention layer introduces tag feature information by calculating a weight matrix to the feature matrix of HTML tags of webpages. Adjust the influence of each tag on webpage information through tag-level attention calculation. The calculation equations are as follows:

\[
m_{j\text{tag}} = \text{tanh}(\omega_{j\text{tag}}x_{j}^{\text{con}} + b_j) \quad (7)
\]

\[
a_{j\text{tag}} = \text{softmax}(m_{j\text{tag}}^T A_{\text{tags}}) \quad (8)
\]

\[
H = \sum_k a_{j\text{tag}} x_k^{\text{con}} \quad (9)
\]

First, the tag-level attention layer linearly transforms the embedding based on tanh function. Next, calculate the importance (\( a_{j\text{tag}} \)) of each tag-level feature matrix based on the softmax function. Finally, the tag-level attention calculation is completed by weighted average of the convolutional feature matrices. \( x_{j}^{\text{con}} \) represents the output matrix after convolution on \( j \)-th tag of the input webpage. \( \omega \) and \( b_j \) represent tag-level attention layer’s weight matrix and bias matrix respectively. \( A_{\text{tags}} \) is the parameter matrix of tag-level attention to be trained. \( a_{j\text{tag}} \) represents the tag-level attention weight of \( x_{j}^{\text{con}} \). \( H \) represents the output after performing tag-level attention weighting on the input webpage.

Then perform the maxpooling operation on the webpage \( wp \)’s presentation matrix \( H \), which is defined as:

\[
\mathcal{H} = \text{Maxpooling}(H) \quad (10)
\]

Sigmoid function is used to perform probability output on \( \mathcal{H} \) after maxpooling operation. The output \( Y \) of sigmoid function is the final judgment result of whether the webpage
needs to be filtered.

\[ Y = \text{sigmoid}(\omega \hat{Y} + b_f) \]  

(11)

Sigmoid function is used to convert the internal function into a probability function to achieve a binary output. \(Y\) is the output result. If \(Y \geq 0.5\), \(w_p\) is predicted as a valid webpage, otherwise \(w_p\) is predicted as an invalid webpage. \(\omega_s\) represents the weight matrix, and \(b_f\) is the bias value.

### 3.6 Model Training

We need to optimize all the training parameters in HAIF, local-level attention layer: \(\omega_i; b_i; A_{\text{local}}\), convolution layer: \([\omega, b]\), tag-level attention layer: \(\omega_i; b_i; A_{\text{tags}}\), sigmoid layer: \([\omega, b]\). HAIF uses an end-to-end backpropagation for training. HAIF is optimized by minimizing the cost function shown below.

\[ H_{Y'}(Y_i) = - \sum_i Y'_i \log Y_i \]  

(12)

Where \(Y'_i\) is the label value of whether webpage \(i\) actually needs to be filtered, and \(Y_i\) is the prediction result calculated by using sigmoid activation function. In the model training phase, AdaDelta [25] is used to accelerate convergence and reduce the amount of calculation. The initial learning rate is 0.001 and the mini-batch size is 32. To avoid overfitting HAIF exploits dropout [26] strategy and \(K\)-fold cross-validation. The parameters of dropout and \(K\) are 0.5 and 3, respectively.

### 3.7 EACNN Model

For HTML documents, the plain text in distinct tags is of different importance to the user to understand webpage information. Compared to subtags, the plain text in meta tags can clearly achieve a better overview of the webpage. In response to this concept and related research work, based on deep neural networks, HAIF utilizes lexical-level embeddings and paragraph-level embeddings to express plain text sequence information, and performs different levels of attention calculations on the plain text and HTML tags to improve the model’s performance. Zhou Z et al. found that randomly assigning weights to multiple networks and using genetic algorithms to evolve the weights can improve the performance of networks [27]. Based on this study and to evaluate the performance of HAIF, we propose a comparative baseline model called EACNN, which divides HTML documents into different data sets according to tag categories to train base learners. Each base learner is based on local-level attention mechanism and CNN, which is the same as part of the content of HAIF (Sects. 3.2-3.4).

EACNN and HAIF introduce webpage structure information in different ways. EACNN utilizes ensemble learning to replace HAIF’s tag-level attention calculation. Figure 3 is the schematic diagram of EACNN. EACNN exploits the subsets based on tag categories to train base learners.

Different from the output method of the probability average of Bagging ensemble learning algorithm [28], EACNN assigns an optimization weight to the output of each base learner. The output of each base learner is the product of the original output and the optimized weight, and the final output of EACNN is the average of the cumulative sum of base learners’ outputs. In training strategy, the output of each base learner is randomly weighted first, and then the Particle Swarm Optimization (PSO [29]) algorithm is used to optimize the weights. The specific calculation process is as follows:

HTML documents are sampled into \(m\) data subsets according to tag categories and represented as \(w_p = \{t_1, t_2, \ldots, t_m\}\). Each data subset is used to train a base learner. \(O(w_p)\) represents the \(i\)-th webpage's EACNN output, the calculation equation is:

\[ O(w_p) = \frac{\sum_{j=1}^{m} W_j O_j(w_p)}{m} \]  

(13)

If \(O(w_p) > 0.5\), is labeled as an invalid webpage, otherwise \(w_p\) is labeled as a valid webpage. \(O_j(w_p)\) represents the prediction output of the \(j\)-th base learner on whether the \(i\)-th webpage needs to be filtered. \(W_j\) represents the output weight of the \(j\)-th base learner. To ensure the excellent effect of EACNN, it is necessary to calculate the optimal solution of the weights of different learners. PSO algorithm has the ability to optimize the objective function of continuous parameters, and has the advantages of fast search speed, high efficiency, and is suitable for real-valued processing. Therefore, EACNN applies PSO algorithm to optimize the weights \((W = \{W_1, W_2, \ldots, W_m\})\).

\[ \text{score}(W) = \sum_{i=1}^{m} \text{rank}(w_p_i) \]  

(14)

\[ \text{rank}(w_p_i) = \begin{cases} 1, Y_i = Y'_i \\ 0, Y_i \neq Y'_i \end{cases} \]  

(15)

\(\text{score}(W)\) represents the current effect evaluation.
of EACNN setting weights ($W$). $rank(wpi)$ represents EACNN’s prediction result of $i$-th webpage.

$Y_i$ represents whether $wpi$ needs the filtered prediction label. $Y'_i$ represents the actual label of $wpi$.

4. Experiments

4.1 Experimental Setting

4.1.1 Dataset

As shown in Fig. 4, users can obtain title data and description data of webpage from the search engine according to the search content. HTML text data of the webpage is obtained according to URL data. By analyzing the structural and functional characteristics of webpages, we divide the HTML text data of webpages into the following five subsets according to their HTML tag types, as shown in Table 1: 1) title text data returned by search engines; 2) description text data returned by the search engine; 3) meta tag; 4) sub-titles and other Special font; 5) page body text and table tag.

![Fig. 4](image) An example of search engine returning webpage.

Table 1 Subsets of HTML tags for webpage.

| Categories | Tags |
|------------|------|
| 1          | Title text data returned by the search engine |
| 2          | Description text data returned by the search engine |
| 3          | `<meta>` metadata |
| 4          | `<h1>...<hn>` sub-titles, `<strong>`, `<em>`, `<b>`, `<i>` and other Special font |
| 5          | `<p>` webpage body, `<table>` table tag |

![Fig. 5](image) Examples of experimental data.

We use the frequently-used geographic information search task as experimental scenario. Therefore, the webpages containing geographic information are labeled as valid pages, and the webpages not related to geographic information (such as commercial shopping, advertising, and community Q&A) are labeled as invalid pages. We utilize a real Internet application (Baidu) as the source of experimental data, and set “City-District-Street-Location Type (Attribute)” as the search information format. The format is a commonly used information description method in Chinese, which gradually narrows the coverage of geographic information description from cities to streets and roads (for example, Changchun City-Chaoyang District-Weixing Road-Bank), and searches according to the location type of the current retrieval target.

In the experimental data collection, we collected 4,390 Chinese actual webpage data from Baidu search engine (https://www.baidu.com) based on crawler technology and search information format as the experimental data set, and manually labeled each sample. The dataset includes 2,790 valid webpages and 1,600 invalid webpages. 70% of the samples in the dataset are used to train models, and the remaining samples are used to test the models’ performance.

As shown in Fig. 5, in addition to the data of each attribute in Table 1, each sample also contains data ID, URL text and Boolean data labeling the types of webpages.

4.1.2 Data Preprocessing

Before the model training, the NLP tool jieba (https://github.com/fxsjy/jieba) is used to perform text segmentation, and remove stopwords and punctuation. In terms of text pre-training embedding matrix generation, we use the Skip-gram method in [15] to implement the construction of 300-dimensional embedded Chinese word2vec vectors. And based on the method of [16] to complete the generation
of PV-DM embeddings, the dimension of PV-DM is also 300.

4.1.3 Evaluation Index

We use Accuracy, Precision, Recall, F1 score, Receiver Operating Characteristic (ROC) and Area Under Curve (AUC) value as the evaluation criteria for model performance. The X-axis of ROC curve is False invalid page Rate (FNR), and the Y-axis is True invalid page Rate (TPR). The calculation equations of FPR and TPR as follows:

\[
FPR = \frac{FP}{FP + TN} \tag{16}
\]

\[
TPR = \frac{TP}{TP + FN} \tag{17}
\]

True invalid page (TP) refers to the result of being correctly identified as an invalid webpage; true valid page (TN) refers to the result of being correctly identified as a valid webpage; false invalid page (FP) refers to being incorrectly identified as the result of an invalid webpage; false valid page (FN) refers to the result of being erroneously identified as a valid webpage. The closer the ROC curve is to the upper left corner, the better performance of the model will be. AUC is defined as the area under the ROC curve, and the value of AUC range is \([0, 1]\). The larger the AUC value, the better the model performance.

4.2 Model Performance Optimization

4.2.1 The Positive Effect of Paragraph Embeddings

HAIF is based on deep learning and attention calculation. In order to improve feature extraction in text sequence, on the basis of lexical-level word2vec embeddings, concatenating PV-DM embeddings can improve the Accuracy of discrimination from 96.12% to 96.58%. And HAIF’s F1 score increased from 96.95% to 97.49%. This manifests that the concatenating of vocabulary-level embedding and paragraph-level embedding of the input text can more fully represent the segment information, so that the performance of HAIF is more excellent. It means the introduction of paragraph embedding vectors plays a positive role in training effect of HAIF.

4.2.2 Sample Length Threshold Calibration

Sample length is the number of words contained in the text sequence. The statistical curves of the sample length of each HTML tag sub-data set are near-log normal distribution. The text lengths of different samples in the dataset are different. The statistical curves of the text length of each HTML tag sub-data set of the webpage are near-log normal distribution. The text lengths of different samples in the data set are different. Simply take the sample with the longest length and fill the other samples with the same length, which will waste computing resources and trigger the information between samples to deviate excessively. Therefore, we design a threshold equation to optimize the sample length. The calculation content of the \(L_{\text{Max}}\) is the sum of the average value of the sample length and the standard deviation of twice the sample length. The equation is shown as Eq. (18).

\[
L_{\text{Max}} = \text{mean(words)} + 2 \times \text{std(words)} \tag{18}
\]

In two cases, the sample threshold and the original sample length, we used the base learners (HAIF unlabeled attention layer) trained with different tag datasets to conduct a comparative experiment of F1 scores. Figure 6 shows the performance of the models using different sample length setting strategies.

The results indicate that setting the sample length threshold can improve the overall performance of the models. It proves the rationality of the sample length threshold setting.

4.3 Experimental Results

4.3.1 Effect of Tag-Level Attention Layer on HAIF Performance

We use different tag subsets to construct various IWF models (without tag-level attention layer) which are the base learners of HAIF. Through comparative experiments to analyze the impact of the introduction of tag-level attention layer on model performance. The comparison results are presented in Table 3.

Compared with model trained with full text (only local-level attention, with out tag-level attention), the HAIF’s Precision decreased from 97.95% to 97.14%, but Accuracy, Recall and F1 increased from 91.80%, 90.19%, 93.77% to 96.58%, 97.49%, 97.31%. It demonstrates that the introduction of webpage structural features is better than only use of text features to identify invalid webpages. In addition, we found from the experimental results that different HTML tags have different effects on model performance. In
Table 2 Evaluation of embedding method.

| Model (embedding method) | Accuracy | Precision | Recall | F1     |
|--------------------------|----------|-----------|--------|--------|
| HAIF (word2vec)          | 96.12%   | 96.78%    | 97.13% | 96.95% |
| HAIF (word2vec+PV-DM)    | 96.58%   | 97.14%    | 97.49% | 97.31% |

Table 3 Comparison HAIF and base models.

| Model   | Accuracy | Precision | Recall | F1     |
|---------|----------|-----------|--------|--------|
| All Text| 91.80%   | 97.95%    | 90.19% | 93.77% |
| Tags1   | 92.94%   | 94.75%    | 94.63% | 94.57% |
| Tags2   | 93.74%   | 93.88%    | 96.39% | 94.99% |
| Tags3   | 90.89%   | 88.78%    | 97.87% | 92.97% |
| Tags4   | 89.64%   | 88.04%    | 96.73% | 92.02% |
| Tags5   | 90.66%   | 88.89%    | 97.61% | 92.92% |
| HAIF    | 96.58%   | 97.14%    | 97.49% | 97.31% |

general, the overall performance of the models established by using the title (Tags1) and description (Tags2) data of the webpages returned by the search engine is better than the models established by using the other HTML tag data (Tags3, Tags4, Tags5). This shows that the HTML tag data introduces more noise information while the text length is longer, which leads to the poorer effect of the base learner based on it. The experimental results show that the performance of HAIF that introduces the webpage tag features is better than the other IWF models. This proves that introducing the tag-level attention mechanism into deep-learning-based model can improve its performance on webpage analysis tasks.

4.3.2 Influence of Various Tag Data in Webpage on Feature Description

To further observes the representation effect of the data in different tags on the characteristic information of the webpage. We use different HTML tag subsets to train local-level attention CNN models (Web_tags_model) and compare these models with HAIF. The comparison results of ROC curves and AUC scores of different models are shown in Fig. 7.

The experimental results demonstrate that AUC (0.9586) and ROC curve of HAIF are better than other Web_tags_models. We found that the performance of Web_tags_models based on Tags1 and Tags2 data are better than other Web_tags_models. This shows that title data and description data contain less noise information and can better summarize the webpage information. However, the information contained in font tag, body tag and other subtags are relatively complicated. Webpage filtering model’s performance is affected by the quality of tag data. This demonstrates that on the basis of traditional text features, the introduction of webpage structural features has played a positive role in webpage filtering models’ construction. HAIF can better extract HTML text features of webpages by adding local-level and tag-level hierarchical attention calculations. This indicates that the structure of HAIF is reasonable.

4.3.3 Comparative Experiment

We have selected some baseline models to verify the advance of HAIF, including machine learning models based on kernel function, deep learning models, attention-based deep learning model and ensemble learning-based model (EACNN). The base learners’ embedding layer, local-level attention layer and convolution layer of EACNN are the same as HAIF’s.

Table 4 shows the calculation results of the optimal weight of each base learner of EACNN. It was found that the weight coefficients (0.2134 and 0.2415, respectively) based on Tags1 base learner and Tags2 base learner were higher than the weights of other learners. This demonstrates that the title and summary text returned by search engine provides a better overview of the meta-information of the webpage than other HTML tags. The weight of the base learner based body tag (Tags5) is the lowest, at 0.1690. This is because the length of body tag’s plain text is longest, and body tag’s information is more diverse and complex.

The experimental results are shown in Table 5. The deep learning-based models are superior to the traditional machine learning models of SVM and KNN. This proves that the deep learning models have previously traditional machine learning in feature extraction and learning. The discriminant effect experiment using this dataset manifests
The experimental results are shown in Table 7.

Ten experiments to evaluate the computational cost of HAIF. We utilize the average calculation time of and settings are shown in Table 6. We choose a personal computer (PC) to analyze webpages and use time cost to analyze the feasibility of this method in actual Internet applications. The experimental environment and settings are shown in Table 6.

We use the trained HAIF model to analyze random 1000 webpages. We utilize the average calculation time of ten experiments to evaluate the computational cost of HAIF. The experimental results are shown in Table 7.

The experimental results show that the efficiency of using HAIF to identify invalid webpages on ordinary PCs can reach 59.28-63.00s per 1000 webpages. It takes about 0.06s to judge whether a webpage is a valid webpage by HAIF. This shows that the computational efficiency of HAIF can fully meet the actual needs of netizens. The method of filtering invalid webpages through HAIF can reduce Internet noise information, and purify Internet resources and environment, while ensuring the applicability of computational costs.

5. Conclusions

This paper proposes a hierarchical attention-based IWF (HAIF) model. HAIF is used to filter invalid webpage in Internet information retrieval. HAIF is based on CNN, which combines word2vec and PV-DM embeddings to improve the feature representation of HTML text sequence, and utilizes multi-level attention calculations. Tag-level attention calculation can introduce webpage structure information and improve the ability of HAIF. To evaluate the performance of HAIF, we collected experimental dataset from actual Internet resources. Through experimental analysis and results, it can be demonstrated that HAIF is superior to other baseline models. This verifies that HAIF is useful for solving the netizens need of filtering invalid webpages, reducing the results of manually filtering Internet resources, and enabling netizens to find valuable information faster when searching with search engines.

Table 5 Comparison HAIF and baseline models.

| Model     | Accuracy | Precision | Recall  | F1     |
|-----------|----------|-----------|---------|--------|
| SVM [1]   | 76.77%   | 80.00%    | 38.89%  | 52.34% |
| KNN [2]   | 74.87%   | 71.58%    | 45.73%  | 55.81% |
| CNN [3]   | 88.15%   | 84.33%    | 78.47%  | 81.29% |
| BLSTM [4] | 87.93%   | 86.40%    | 75.00%  | 80.30% |
| GRU [6]   | 89.29%   | 87.60%    | 78.47%  | 82.78% |
| ABCNN [7] | 91.80%   | 97.95%    | 90.19%  | 93.77% |
| EACNN     | 96.90%   | 96.23%    | 94.96%  | 95.58% |
| HAIF      | 96.58%   | 97.14%    | 97.49%  | 97.31% |

Table 6 Experimental environment and settings.

| Categories   | Tags                      |
|--------------|---------------------------|
| IDE          | PyCharm CE                |
| Programming language | Python 3.6              |
| Deep learning framework | Tensorflow 1.13.2       |
| CPU          | 2.6 GHz Intel Core i5     |
| RAM          | 8GB 1600MHz DDR3          |

that the three deep learning models (CNN, BLSTM, GRU) have similar performance. CNN and RNN-like networks (BLSTM, GRU) have advantages in extracting local-level features and sequence features, respectively. The text length of the title data and summary data in the webpage tags are short. RNN-likes networks’ structure is used to construct the filtering model, which would waste computing resources and fall into a disadvantage in local feature extraction. Assigning the same weight to the content of each tag in the webpage text will introduce more noise interference features and reduce the focus on valuable information. Compared with ABCNN, HAIF introduces tag-level attention calculations that can be better introduced webpage features to achieve better model performance. ABCNN’s Precision (97.95%) is higher than HAIF’s (97.14%). But on other evaluation criteria, HAIF performs better than ABCNN. This is due to the small sample size of the dataset or uneven data distribution. EACNN using linear tag-level features has obvious advantages over other models, but the simple use of a real weight to represent the features of webpage have certain limitations. The comparison experiment results indicate that HAIF constructed by introducing hierarchical attention calculation achieves the best performance, with F1 scores of 97.31%. HAIF’s Accuracy Precision and Recall are also higher than most of the other baseline models.

4.3.4 Feasibility Analysis for Calculating Cost

We choose a personal computer (PC) to analyze webpages and use time cost to analyze the feasibility of this method in actual Internet applications. The experimental environment and settings are shown in Table 6.

We use the trained HAIF model to analyze random 1000 webpages. We utilize the average calculation time of ten experiments to evaluate the computational cost of HAIF. The experimental results are shown in Table 7.

The experimental results show that the efficiency of using HAIF to identify invalid webpages on ordinary PCs can reach 59.28-63.00s per 1000 webpages. It takes about 0.06s to judge whether a webpage is a valid webpage by HAIF. This shows that the computational efficiency of HAIF can fully meet the actual needs of netizens. The method of filtering invalid webpages through HAIF can reduce Internet noise information, and purify Internet resources and environment, while ensuring the applicability of computational costs.

5. Conclusions

This paper proposes a hierarchical attention-based IWF (HAIF) model. HAIF is used to filter invalid webpage in Internet information retrieval. HAIF is based on CNN, which combines word2vec and PV-DM embeddings to improve the feature representation of HTML text sequence, and utilizes multi-level attention calculations. Tag-level attention calculation can introduce webpage structure information and improve the ability of HAIF. To evaluate the performance of HAIF, we collected experimental dataset from actual Internet resources. Through experimental analysis and results, it can be demonstrated that HAIF is superior to other baseline models. This verifies that HAIF is useful for solving the netizens need of filtering invalid webpages, reducing the results of manually filtering Internet resources, and enabling netizens to find valuable information faster when searching with search engines.

References

[1] S. Tong and D. Koller, “Support vector machine active learning with applications to text classification,” Journal of Machine Learning Research, vol. Nov, no. 2, pp. 45–66, 2001.
[2] R. Du, R. Safavi-Naini, and W. Susilo, “Web filtering using text classification,” Proc. 11th IEEE International Conf. Networks, Sydney, Australia, pp. 325–330, Sept. 2003.
[3] Y. Kim, “Convolutional Neural Networks for Sentence Classification,” Proc. 2014 Conf. Empirical Methods in Natural Language Processing, Doha, Qatar, pp. 1746–1751, Oct. 2014.
[4] A. Ray, S. Rajeswar, and S. Chaudhury, “Text recognition using deep BLSTM networks,” Proc. 8th International Conf. Advances in Pattern Recognition, Kolkata, India, pp. 1–6, Jan. 2015.
[5] P. Liu, X. Qiu, and X. Huang, “Recent neural network for text classification with multi-task learning,” Proc. 25th International Joint Conf. Artificial Intelligence, New York, USA, pp. 2873–2879, July 2016.
[6] D. Tang, B. Qin, and T. Liu, “Document Modeling with Gated Recurrent Neural Network for Sentiment Classification,” Proc. 2015
A.N.A. Kamarudin and B. Ranaivo. Malancon, “Simple internet filtering access for kids using naive Bayes and blacklisted URLs,” In-ternets, vol.44, no.2, pp.135–162, 2002.

Y. Yang, S. Slattery, and R. Ghani, “A study of approaches to hyper-text categorization,” Journal of Intelligent Information Systems, vol.18, no.2-3, pp.219–241, 2002.

R.Y.K. Lau, C.L. Lai, and Y. Li, “Leveraging the web context for context-sensitive opinion mining,” Proc. 2nd IEEE International Conf. Computer Science and Information Technology, Beijing, China, pp.467–471, Aug. 2009.

J.J. Sheu, “Distinguishing medical web pages from pornographic ones: An efficient pornography websites filtering method,” IJ Network Security, vol.19, no.5, pp.839–850, Sept. 2017.

M.-Y. Kan and H.O.N. Thi, “Fast webpage classification using URL features,” Proc. 14th ACM international Conf. Information and knowledge management, Bremen, Germany, pp.325–326, Oct. 2005.

T. Mikolov, K. Chen, G. Corrado, and J. Dean, “Efficient estimation of word representations in vector space,” arXiv preprint arXiv:1301.3781, Jan. 2013.

Q. Le and T. Mikolov, “Distributed representations of sentences and documents,” Proc. 31th International Conf. Machine Learning, Beijing, China, pp.1188–1196, Jan. 2014.

M. Chau and H. Chen, “A machine learning approach to web page filtering using content and structure analysis,” Decision Support Systems, vol.44, no.2, pp.482–494, Jan. 2008.

A.N.A. Kamarudin and B. Ranaivo, Malancon, “An Overview of Standards and Related Technology in Web Services,” Distributed and Parallel Databases, vol.12, no.2, pp.135–162, 2002.

T. Mikolov, K. Chen, G. Corrado, and J. Dean, “Efficient estimation of word representations in vector space,” arXiv preprint arXiv:1301.3781, Jan. 2013.

Q. Le and T. Mikolov, “Distributed representations of sentences and documents,” Proc. 31th International Conf. Machine Learning, Beijing, China, pp.1188–1196, Jan. 2014.

M. Chau and H. Chen, “A machine learning approach to web page filtering using content and structure analysis,” Decision Support Systems, vol.44, no.2, pp.482–494, Jan. 2008.

A.N.A. Kamarudin and B. Ranaivo, Malancon, “An Overview of Standards and Related Technology in Web Services,” Distributed and Parallel Databases, vol.12, no.2, pp.135–162, 2002.

T. Mikolov, K. Chen, G. Corrado, and J. Dean, “Efficient estimation of word representations in vector space,” arXiv preprint arXiv:1301.3781, Jan. 2013.

Q. Le and T. Mikolov, “Distributed representations of sentences and documents,” Proc. 31th International Conf. Machine Learning, Beijing, China, pp.1188–1196, Jan. 2014.

M. Chau and H. Chen, “A machine learning approach to web page filtering using content and structure analysis,” Decision Support Systems, vol.44, no.2, pp.482–494, Jan. 2008.