Hybrid ASCII Art Extraction Algorithm Based on String Distance

2024 IEEE/ACIS 22nd International Conference on Software Engineering Research, Management and Applications (SERA) | 979-8-3503-9134-3/24/$31.00 ©2024 IEEE | DOI: 10.1109/SERA61261.2024.10685590

Xiujuan Wang
Faculty of Information Technology
Beijing University of Technology
Beijing, China
xjwang@bjut.edu.cn

Xiaotong Wu
Faculty of Information Technology
Beijing University of Technology
Beijing, China
contrail123@sina.com

Haochen Shi
Faculty of Information Technology
Beijing University of Technology
Beijing, China
1572505830@qq.com

Shuaibing Lu
Faculty of Information Technology
Beijing University of Technology
Beijing, China
lushuaibing@bjut.edu.cn

***Abstract*—ASCII art detection and recognition is an important branch of current network information processing. However, due to ASCII art's text-based organization and image-based semantic expression, traditional natural language processing (NLP) and image recognition fail to yield ideal results. This paper designs an ASCII art localization and extraction algorithm based on string distance for highly mixed text and ASCII art, aiming to segment clean ASCII art for subsequent recognition. Additionally, an evaluation standard for ASCII art extraction effectiveness is defined. Experimental results show that the proposed algorithm performs well in locating and extracting ASCII art.**

Keywords—ASCII art; extraction; dictionary; string distance

I. INTRODUCTION

With the advancement of technology, network technology has become widely used, significantly altering entertainment and social life. People can anonymously post any content online or contact strangers via email, leading to attention on network information management issues like sensitive remarks, personal attacks, and cyber violence.

Text-based images are called ASCII art, commonly used in web pages, emails, and text-based interfaces. ASCII art has a strong impact on first-time users[1].ASCII art can be broadly categorized into two types: structure-based ASCII art and tone-based ASCII art[2]. Structure-based ASCII art represents images created by drawing the outlines of objects using characters. Figure 1 shows an example of structure-based ASCII art depicting a snail. Even a simple smiling face like ":-)" is classified as structure-based ASCII art. On the other hand, tone-based ASCII art represents grayscale images composed of characters. Figure 2 shows an example of a tone-based ASCII art depicting a flower. Recent ASCII art not only uses ASCII code characters but also Unicode characters, making it more expressive[3].

ASCII art is popular on social networking platforms due to its ability to convey image information through the arrangement of characters, as well as its capacity to express rich emotional content and information.

However, ASCII art poses significant challenges to natural language processing. Since ASCII art uses the form of characters to convey information without following language rules, existing natural language processing methods are unable to handle it effectively. Additionally, ASCII art exists in text form within the text stream, making it difficult for image-based approaches to be applied. Both domestic and international researchers focusing on ASCII art extraction and recognition are few in number, leading to slow progress in this area.

Fig.1 Structured art

Fig.2 Color tone art

This paper addresses the highly mixed situation of text and ASCII art and proposes a ASCII art localization and extraction algorithm based on string distance. This algorithm aims to segment clean ASCII art, laying the foundation for subsequent recognition. Furthermore, we define an evaluation criterion for ASCII art extraction effectiveness. Finally, we test the algorithm using actual network comment data containing ASCII art, demonstrating the effectiveness of the proposed method.

II. RESEARCH STATUS AT HOME AND ABROAD

The research on ASCII art extraction is mainly divided into two categories: one is the extraction method proposed by Japanese scholar Suzuki, which is independent of textual language, and its improvements. The other type of research is related to textual language and a series of network software.

Tanioka et al. proposed a ASCII art recognition method based on Support Vector Machine (SVM) as a natural language processing technique, which can determine whether a given text is a ASCII art object[4]. The training data for SVM is a set of 262-dimensional vectors. Each vector consists of two parts. The first 256 elements of the vector represent byte patterns, where the i-th element ($0 \leqslant i \leqslant 255$) of the vector counts the occurrences of byte value i in the byte stream of the UTF-8 encoded text. The authors classified Japanese word classes into 6 groups. The remaining 6 elements of the

vector represent the occurrence of groups in the text. Since this method is specific to Japanese, it is not suitable for other languages.

Nakazawa et al. proposed a byte pattern-based method[5]. It scans a line of UTF8 encoded text and detects areas in the text that contain ASCII art. Before extracting the ASCII art, an SVM model is constructed through a learning algorithm. Each training data consists of a class and an extended byte pattern. The class represents whether it is ASCII art. The extended byte pattern for each line is the concatenation of byte patterns of this line, the preceding N lines, and the following N lines. In this ASCII art extraction approach, the constructed SVM model is used to calculate the likelihood of this line being ASCII art from the extended byte patterns of a single line. The byte pattern-based method uses smoothed ASCII art likelihood to avoid splitting ASCII art into multiple parts. If the smoothed ASCII art likelihood of a line is greater than or equal to 50%, then that line is considered a part of the ASCII art. The likelihood of a line of smoothed ASCII art is calculated based on the ASCII art likelihood of 2M+1 lines (i.e., this line, the preceding M lines, and the next M lines).

On a Japanese website, there is a free software called "AA scan" [6], which aims to help users collect ASCII art from BBS 2channel. Although the author did not disclose specific recognition details, the documentation mentions that this is a heuristic method based on Japanese characters. For example, it uses the frequency of characters in the text, including not only English letters but also hiragana, katakana, kanji, and other Japanese characters.

Japanese scholar Suzuki proposed a text-language-independent extraction method in [7], which is different from the aforementioned methods. This method consists of three steps: window scanning, ASCII art detection, and reduction of non-ASCII art lines before and after. Firstly, k lines are selected as a window, and the entire text area is scanned using this window. For the scanned window sections, a trained ASCII art binary classifier is applied to roughly locate the ASCII art positions. Finally, the extracted ASCII art is obtained by removing empty lines before and after the ASCII art, as well as non-ASCII art lines. For training the binary classifier, this method proposes several indicators to determine whether the text contains ASCII art: H (number of horizontally continuous identical characters), R (compression ratio after RLE compression), G (compression ratio after LZ77 compression), L (number of text lines), S (number of text characters), W (number of words outside the vocabulary), and B (byte patterns). Among these attributes, W and B are language-related attributes.

In Suzuki's subsequent work [8-11], the method introduced in [7] was fine-tuned. This involved adding new parameters to accommodate different ASCII art texts, incorporating new methods to handle empty characters, and comparing the effects of different compression modes, thereby enhancing the effectiveness of the initial method.

III. ASCII ART EXTRACTION ALGORITHM BASED ON STRING DISTANCE

In existing research, the sliding window ASCII art extraction algorithm proposed by Japanese scholar Suzuki and his research team is widely recognized as the most effective algorithm. However, due to its use of metrics such as text length, text byte count, and text compression ratio as inputs to its decision tree, it operates at the line level, making decisions based on line-level data. While this approach effectively addresses the problem of locating and extracting ASCII art in typical ASCII art texts, as shown in Figure 3, which consist of plain characters, it falls short when dealing with texts that have a high degree of mixture between ASCII art and semantic content, as depicted in Figure 4. Existing algorithms are unable to handle the finer boundary determination at the character level and can only achieve coarse extraction, causing significant interference in subsequent ASCII art recognition tasks.

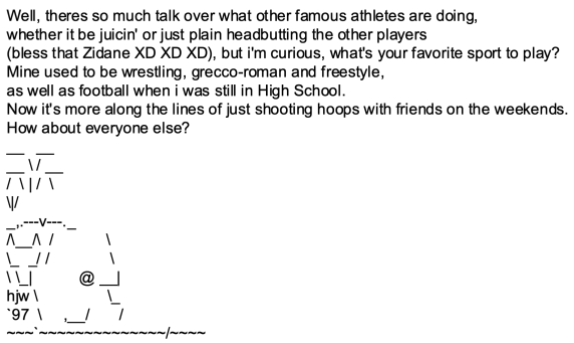

Fig.3 Texts with Typical ASCII art

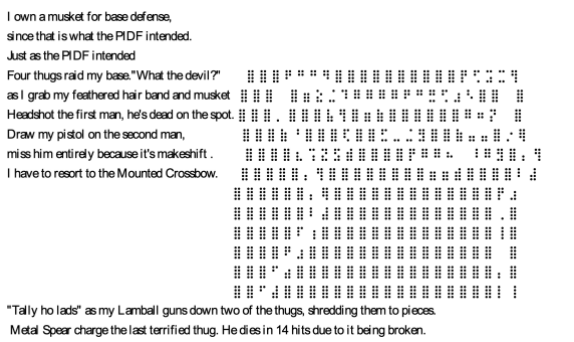

Fig.4 Mixed text

Upon observing the mixed text, it is evident that extracting ASCII art from this type of text requires determining the boundaries between ASCII art and regular semantic text. The composition of ASCII art mostly involves non-semantic ordinary characters such as '.', '\', '?', etc. Therefore, this paper proposes a semantic-level judgment at the word-level granularity from mixed text to distinguish between ASCII art and regular semantic text.

The overall structure of this algorithm is shown in Figure 5. In this study, a dataset containing ASCII art from real online comments was used. The data text was scanned and segmented using a sliding window to obtain several word sequences. Subsequently, the standard string distance for each word sequence was computed, and the resulting vectors were fed into a multilayer perceptron for discrimination. Finally, all strings judged as ASCII art were extracted to obtain clean ASCII art text. English text was used for discussion in this chapter's algorithm.

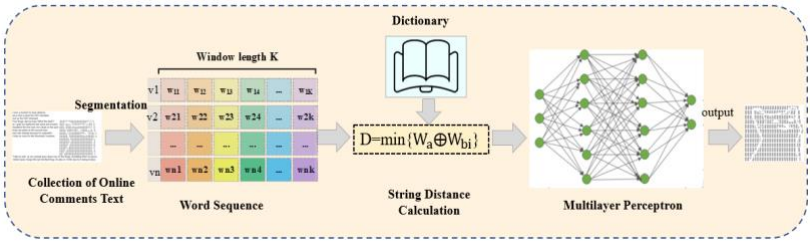

Fig.5 Overall structure

A. Character Segmentation

The function of character segmentation is to convert text into words. Word segmentation is the first step in natural language processing tasks. However, in the case of ASCII art texts, they often contain symbols and letters, making it impossible to use punctuation marks as markers for word segmentation. Therefore, for word segmentation in the mixed text of English text and ASCII art used in this study, we adopt a simple approach using space characters as markers for segmentation.

B. Text Scanning

After character segmentation, we obtain a word sequence T, where T={w1 w2 ... wn} represents the mixed text containing ASCII art and regular semantic text, with w1, w2...wn denoting words naturally segmented by spaces. To facilitate discrimination, this paper uses a sliding window approach with a length of K and a step size of N to traverse the above word sequence and generate a series of fixed-length word sequences vi. As shown in Figure 6, the algorithm sets N=K, taking N=K=3 as an example. Thus, we have v1={w1, w2, w3}, v2={w4, w5, w6}... vj={wn-2, wn-1, wn}.

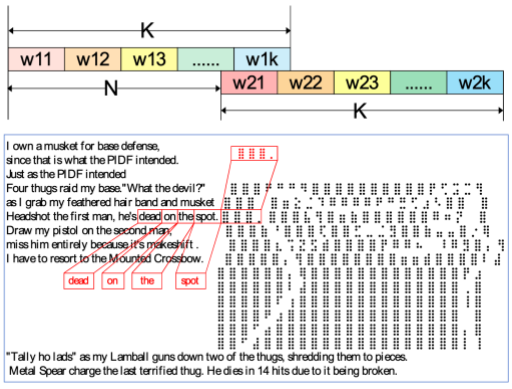

Fig.6 Diagram of text scanning

C. String Distance Calculation

The mixed text of ASCII art and regular text cannot use punctuation marks as the basis for word segmentation, thus requiring the use of space characters as word delimiters. This segmentation approach leads to two main issues. Firstly, punctuation marks that do not belong to ASCII art are included as part of words. For example, in the sentence "I like basketball.", the last word will be segmented as "basketball.", which is unacceptable for traditional natural language processing algorithms. Secondly, parts of ASCII art may be included within words.

Therefore, to determine whether the segmented strings have regular semantic meaning, this algorithm incorporates an open-source English dictionary provided by Princeton University[12]. It matches the strings with words in the dictionary using fuzzy matching. For example, the string "basketball." can be matched semantically with "basketball". For each word w_a extracted through the sliding window, the standard string distance d is defined as shown in Equation 1.

$$d(w_a)=min\{w_a \oplus w_{b1}, w_a \oplus w \ldots w_a \oplus w_{bm}\} \quad (1)$$

Where w_a represents the word to be computed, w_{b1}, $w_{b2} \ldots w_{bm}$ are words in the dictionary, and $\oplus$ denotes the shortest character distance between words. The calculation steps corresponding to A⊕B are as follows:

- The initial string distance is set to 0, and i is initialized to 1.
- Compare the lengths of words A and B, let the difference in word lengths be denoted as l, and let Len represent the length of the longer word.
- Align the i-th letter of the shorter word with the i-th letter of the longer word, and compare the corresponding letters one by one. If they are not the same, increment the string distance by 1, and at the end of the comparison, obtain the string distance for this round.
- Set i = i + 1, repeat step three until i = Len - l + 1, and record the shortest string distance h.
- Then A⊕B = h + l, denoted as H for A⊕B.
- Calculate the standard string distance d(A) for word A by performing the above calculations for word A with all words in the dictionary, and take the minimum value of H as H_{min}.

Calculate the standard string distance for the string "footbell.((" as shown in Figure 7.

Fig.7 Example of calculating standard string distance

D. Multilayer Perceptron

During the training process, the standard string distance sequence *x(v1) x(v2)… x(vM)* calculated from the training set is used as input, and this standard string distance component includes ASCII art as labels for training the multilayer perceptron.

IV. EXPERIMENT

A. Evaluation of ASCII Art Extraction Performance

For line-level ASCII art extraction, when the extracted ASCII art's starting and ending line numbers in the original text are correct, it is considered a successful extraction; otherwise, it is considered a failed extraction.

For word-level ASCII art extraction, as there is no precedent in the experimental field, existing evaluation criteria for ASCII art detection algorithms cannot measure the effectiveness of this experiment. Considering that there is no scaling issue after ASCII art extraction, this paper proposes the ASCII art accuracy to evaluate the effectiveness of ASCII art extraction, as shown in Equation (2).

$$\text{Accuracy}_{\text{ascii}} = \frac{|\text{Num}_{\text{original}} - \text{Num}_{\text{Extracted}}|}{\text{Num}_{\text{original}}} \quad (2)$$

In this context, $\text{Accuracy}_{\text{ascii}}$ represents the accuracy of ASCII art extraction, $\text{Num}_{\text{original}}$ is the number of characters in the original ASCII art, and $\text{Num}_{\text{Extracted}}$ is the number of characters extracted.

B. Experimental Setup

Baseline algorithm adopts the ASCII art detection algorithm proposed by Japanese scholar Suzuki and his research team [6]. The algorithm parameters include text length, number of characters in the text, text compression ratio, and window length set to 1. The machine learning algorithm chosen is the decision tree C4.5 provided by Weka[13].Component heads identify the different components of your paper and are not topically subordinate to each other.

The dataset used in this experiment is the social network comments and ASCII art dataset provided by Japanese scholar Suzuki and his research team. The social network comments and ASCII art data are randomly recombined to generate 991 mixed texts containing ASCII art and normal semantic text. Among them, 700 texts are used as training data, and 291 texts are used as testing data.

The learning rate of the multilayer perceptron is set to 0.0001, using cross-entropy loss function, Elu activation function, and Adam optimizer, with a batch size of 32, and trained for 100 epochs.

C. Experimental Results and Analysis

a) Experiment 1: Impact of Window Size K

The size of the sliding window directly affects the word sequence and indirectly influences the final accuracy of ASCII art extraction. To ensure better experimental results, this experiment sets the window size K to 3, 4, 5, 6, and 7 respectively for training and testing. Figure 8 shows the accuracy of line-level ASCII art extraction, while Figure 9 illustrates the final extraction results and corresponding accuracy of character-level ASCII art extraction. It can be observed that our method significantly outperforms the baseline method for line-level ASCII art extraction. As for character-level ASCII art extraction, the table indicates that the extraction accuracy is highest when K is set to 5. This is because with a smaller window, there is a higher likelihood of the entire window being covered by ASCII art characters, and the length of ASCII art strings does not exhibit clear regularity. Consequently, the standard string distance in such cases lacks clear patterns, making it difficult to determine whether the text is ASCII art or not. Conversely, when the window is too large, there is an increase in cases where short ASCII art is mistakenly classified as normal punctuation. Considering the overall experimental results, selecting a window size of K=5 yields optimal performance.

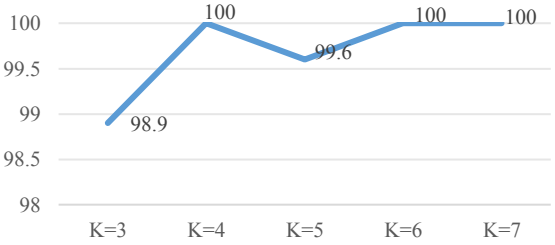

Fig.8 The accuracy of line-level ASCII art extraction under different Ks

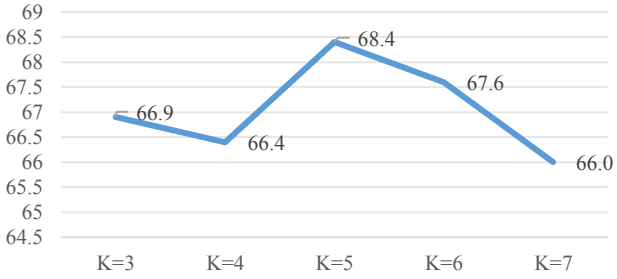

Fig.9 The accuracy of character-level ASCII art extraction under different Ks

b) Experiment 2: Impact of Perceptron Layer Numbers

Under the condition of a sliding window size of K=5, this study further investigates the influence of perceptron layer numbers on the accuracy of character-level ASCII art extraction. The structures of the multilayer perceptron are as follows: Structure 1: {5,20,20,15,5}, Structure 2: {5,20,20,18,15,5}, Structure 3: {5,20,20,18,16,15,5}, and Structure 4: {5,20,20,18,16,15,15,5}. The experimental results are shown in Table 3, where it can be observed that the number of layers in the multilayer perceptron does not significantly affect the accuracy of character-level ASCII art extraction, as indicated in Table I.

TABLE I THE IMPACT OF MULTILAYER PERCEPTRON STRUCTURE ON ASCII ART EXTRACTION ACCURACY

Option	$\text{Accuracy}_{\text{ascii}}$
Structure1	68.35
Structure2	68.21
Structure3	68.32
Structure4	68.30

c) Experiment 3: The Comparison of ASCII Art Extraction Performance with the Baseline Algorithm

The difference in accuracy between this algorithm and the baseline algorithm for line-level ASCII art extraction and

character-level ASCII art extraction is shown in Figure 10. It can be observed that our algorithm outperforms the baseline algorithm in terms of ASCII art extraction accuracy at different levels.

The results of ASCII art extraction are shown in Figure 11, where it can be seen that our algorithm successfully filters out the normal semantic text parts unrelated to ASCII art, resulting in clean ASCII art without semantic text.

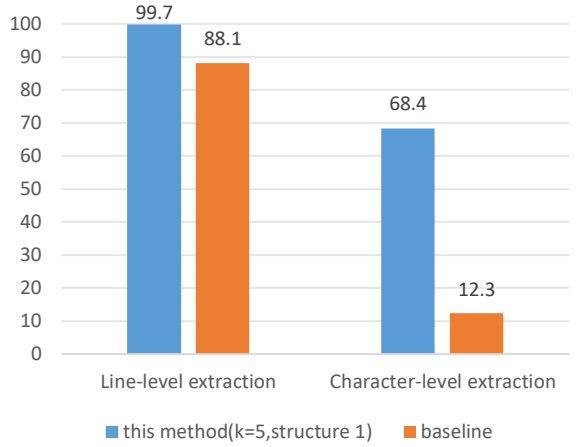

Fig.10 The comparison of ASCII art extraction accuracy with the baseline algorithm

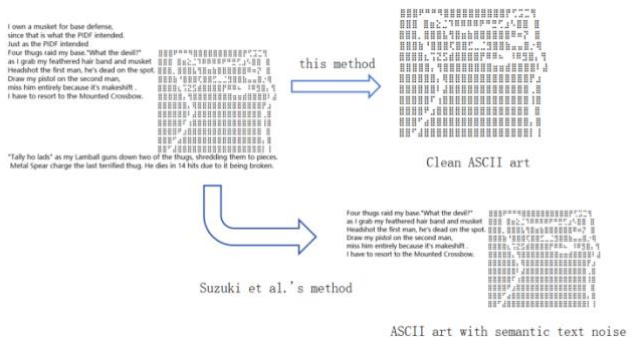

Fig.11 Examples of ASCII art extraction results from different methods

V. CONCLUSION

This paper proposes a hybrid ASCII art extraction method based on standard string distance and machine learning. The method utilizes a sliding window to extract words and calculates the standard string distance of each word as input to the machine learning algorithm for precise word-level ASCII art extraction. Additionally, the paper investigates the impact of word sequence window size and multilayer perceptron depth on the accuracy of ASCII art extraction.

Experimental results demonstrate a significant improvement in line-level ASCII art extraction effectiveness using this method. Furthermore, it successfully addresses the issue of word-level ASCII art extraction.

REFERENCES

[1] Fujisawa, A., Matsumoto, K., Ohta, K., Yoshida, M., & Kita, K. (2020). ASCII Art Classification Model by Transfer Learning and Data Augmentation. In Fuzzy Systems and Data Mining VI (pp. 608-618). IOS Press.

[2] Hiroki, T., Minoru, M.: Ascii Art Pattern Recognition using SVM based on Morphological Analysis. Technical report of IEICE. PRMU 104(670), 25–30(20050218).

[3] Suzuki, Tetsuya. Introduction of N-gram into a Run-Length Encoding Based ASCII Art Extraction Method.15th International Conference, ICWE 2015 Workshops NLPIT, PEWET, SoWEMine Rotterdam, The Netherlands, June 23–26, 2015

[4] Tanioka, Hiroki., Maruyam, Minoru. 2005). Ascii Art Pattern Recognition using SVM based on Morphological Analysis.Technical report of IEICE. PRMU, 104 (670) 25–30, 2005. 0218.

[5] Nakazawa, M., Matsumoto, K., Yanagihara, T., Ikeda, K., Takishima, Y., Hoashi,K.: Proposal and its evaluation of ASCII-art extraction. In: Proceedings of the 2nd Forum on Data Engineering and Information Management (DEIM2010),pp. C9–C4 (2010)

[6] EGG:AAscan(inJapanese),http://www11.plala.or.jp/egoo/download/download_index.html(retrieved on June 13, 2011)

[7] Suzuki, Tetsuya (2010). A Decision Tree-based Text Art Extraction Method without any Language-Dependent TextAttribute.International Journal of Computational Linguistics Research, 1 (1) 12–22, 2010.

[8] Suzuki, Tetsuya (2011). A Comparison of Whitespace Normalization Methods in a Text Art Extraction Method with Run Length Encoding. In: Hepu Deng, Duoqian Miao, Jingsheng Lei, and FuLee Wang, editors, Artificial Intelligence and Computational Intelligence, volume 7004 of Lecture Notes in Computer Science, p. 135–142. Springer Berlin Heidelberg, 2011.

[9] Suzuki, Tetsuya.(2015). Comparison of Two ASCII Art Extraction Methods: A Run-Length Encoding based Method and a Byte Pattern based Method. In: Proceedings of the 6th IASTED International Conference on Computational Intelligence. ACTA Press, 2015. 827-026.

[10] Suzuki, Tetsuya., Hayashi, Kazuyuki (2010). Text Data Compression Ratio as a Text Attribute for a Language-Independent Text Art Extraction Method. In: Proceedings of the 3rd International Conference on the Applications of Digital Information and Web Technologies, p. 513–518.

[11] Suzuki, Tetsuya.Introduction of N-gram into a Run-Length Encoding Based ASCII Art Extraction Method.15th International Conference, ICWE 2015 Workshops NLPIT, PEWET, SoWEMine Rotterdam, The Netherlands, June 23–26, 2015 Revised Selected Papers,p.28-39.

[12] http://wordnet.princeton.edu/

[13] Weka:https://www.cs.waikato.ac.nz/ml/weka/