Bank Check Verification Using Semi-Lexical Analysis

Briti Gangopadhyay, Somnath Hazra and Pallab Dasgupta

Automated bank cheque verification using image processing is used to complement the current cheque verification systems and reduce human intervention. However, the use of deep learning techniques like Convolutional Neural Networks (CNN's) for image processing can provide wrong interpretation of semi lexical tokens without any explanation resulting to rejection of a valid check. In this document we analyse the verification problem in a sem-lexical setting. The experiments are performed on a cheque dataset obtained from IDRBT [1]. The dataset contains 112 high-resolution images of completely filled (with amount, date, amount in words and signature) Indian bank cheques. All the cheques are handwritten. The following sections describe the problem and the processing steps to extract the amount contained in the handwritten cheques. The fact that these documents are handwritten makes the process of validation difficult; since OCR processes such as Tesseract [6] fail to perform extraction of handwritten tokens with many variations. Studies related to handwritten cheque verification have been conducted previously, e.g., in [4, 5], but without addressing the detection of semi-lexical tokens that can occur in handwritten characters.

1. Problem description

Our aim is to verify the validity of a bank cheque through match between the amount in words and numbers, signature and other components of the cheque that can be learnt. Since the verification is performed on a document containing human handwriting, there is a considerable amount of noise in the input. A human handwritten character also has inherent semi-lexical connotations i.e they can show membership in multiple classes. Following the definition of semi-lexical languages bank check verification problem is defined in the following way:

- The alphabet set, $\Sigma_1 = \mathcal{W}$ where $\mathcal{W}$ is a set of words from the English corpora and $\Sigma_2 = \{0, \ldots, 9\}$
- We consider words of the form: $\omega_1 = w_1 |||| \ldots |||| w_9$, where w_i represents the words in the cheque and $\omega_2 = d_1 |||| \ldots |||| d_9$ where d_i represents the digits in the cheque. Detected words ω_1 and ω_2 belongs to $\mathcal{L}$ only if it satisfies the following set $\mathcal{R}$ of constraints:

    1. $w_i \in \mathcal{V}$ where $\mathcal{V}$ is a valid set of words that can occur in a cheque like $\{zero \ldots ninety\ nine\} \cup \{hundred, thousand, crore \ldots\}$
    2. $w_i \in \omega_1$ is in correct order forming a valid amount
    3. $\omega_1 = words(\omega_2)$ the amount in words matches the amount in numbers.

- The set $\mathcal{T}$ of semi-lexical tokens consists of various handwritten images of the digits and handwritten words. Each image is tagged with a member of $\Sigma_1 \cup \Sigma_2$, that is, a digit from $1, \ldots, 9$ or a word from $\mathcal{W}$.
- A set, C, of semi-lexical integrity constraints, as follows:

    Reasoning assisted similarity constraint : The interpretation of each word $w_i \in \omega_1$ should match the handwritten digits $d_i \in \omega_2$ in correct order i.e, $w_i = word(d_i)$. Also we check the validity of the handwritten date and suggest a feasible interpretation. For example, if the year is read as 2027, it may be the case that the writing was 2021, and the writing of 1 was read as 7 by the system.

    Reasoning assisted dissimilarity constraint : The signature on the bank cheque should be close to the authorized signature. This is formalized by measuring the SIFT distance between the authorized signature and the signature on the cheque.

2. Methodology and Experiment details

The following image Fig 1 shows a sample cheque from the IDRBT dataset. All the images are cropped to size with no rotations or flips. Thus a deterministic pre-processing extracts the required text and number from the image.

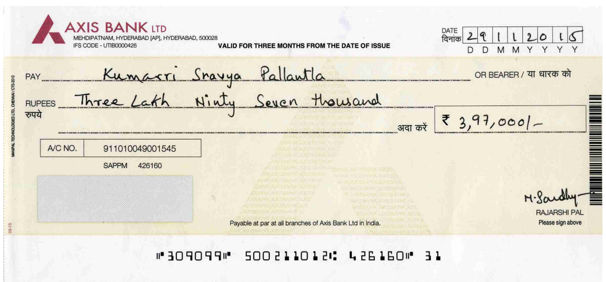


Figure 1: Sample image of a cheque from the IDRBT dataset [1]. The amount (in words and numbers) is written by hand. OCR algorithms thus fail to work in this case.

For recognizing the words in the text, a pre-trained model is used [2], which is trained on IAM dataset [7]. The model structure is described in the repository. Although there is a pre-trained sentence level model, it was not used since the sentences in the cheques do not form an actual sentence that make sense in natural language. The IAM dataset does not contain Indian words like "lakhs" or "crores"; which are very common in the cheque dataset images, as can be seen in figure 1. These can be corrected by distance measures such as edit distance.

For recognizing the digits in the number a CNN was trained with the MNIST dataset as in this repository [3]. These results were more accurate compared to the word model. The identified number is converted to words, followed by an edit distance word to word match with the identified words from the word model. The algorithm is described next.

2.1. Algorithm

In this part, we describe the procedure of matching the amount in words, to that in numbers. For ease of description, we use certain notations for this text, that are disjoint from those in the main text. The final string recovered from the amount in words is considered to be ω_1; that from the amount in numbers, ω_2, converted to words is $\mathcal{S}_2$. The set of possible words in a bank cheque is considered to be $\mathcal{V}$. The algorithm can be outlined through the following steps:

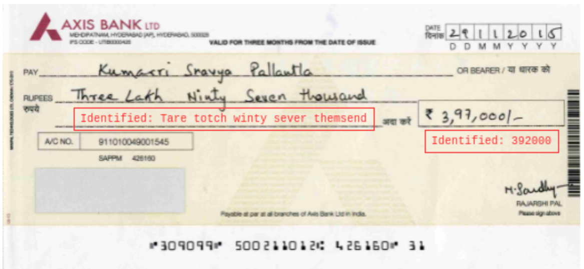
1. Run the word identification model to find the maximum score of w_i among $\mathcal{V}$. If identified append it to the final string, ω_1.
2. If no such word is identified, run the word identification model again without limiting the output corpus and choosing from $\mathcal{W}$. Append the word to ω_1 from $\mathcal{V}$ with minimum edit distance to the identified word. For example if ode is identified as w_i then one will be appended to ω_1.
3. If no word is identified, mark it with a default marker '—'.
4. Run the MNIST model to obtain ω_2 and frame the sentence $\mathcal{S}_2$ from the amount in words.
5. Compare ω_1 and $\mathcal{S}_2$. In case of any word mismatch, choose the word from ω_1. Mark it for human verification. Replace default markers in ω_1 with the words from $\mathcal{S}_2$. Output the final string (with flag for human verification if needed).
6. Additional constraints to be checked include information such as "thousand" can't follow "hundred", and validity of date such that date cannot be from future and must represent a valid date.
7. Checking SIFT distance between cheque signature and authorized signature.

2.2. Results

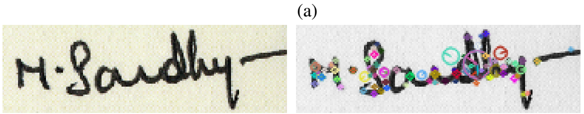
This procedure is interpretable since one can trace the steps of deducing a certain output. A CNN trained on the dataset cannot account for semi-lexical tokens and cannot provide interpretation for the decisions. In this section, a result is explained on the image 1. The example shows verification of amount in words and numbers.

The identified number ω_2 3,92,000 is converted to words to form $\mathcal{S}_2$ = "Three lakh ninty two thousand". As can be seen from the image 2a, both the identifications have errors that need correction. The amount in words ω_1 is recognized as "Tare totch winty sever themsend".

While correction, sometimes multiple words can have the same minimum edit distance. For example, the first identified word in ω_1, 'Tare', has the same edit distance with ['one', 'two', 'three']. Since this is the first



(a)



(b)

Figure 2: a) The identified words and number identified by the word model and the MNIST model respectively are shown in red inside the image. Other comparisons can be similarly done. The identified strings shown here do not represent ω_1 or ω_2. b) SIFT distance between the authorized signature and the signature on cheque.

Table 1: Comparison of accuracy for the Bank Cheque Verification Problem

	Case Type		
Methodology	Valid	Ambiguous	Invalid
End to End CNN Model	70.0	12.5	80.0
Semi-Lexical Analysis	100	78.04	100

word, words such as ‘and’, ‘lakh’, etc. can be automatically ruled out if they appear. Among [‘one’, ‘two’, ‘three’], ‘three’ has least edit distance with ‘Tare’, so ‘three’ is confirmed as the first word. For the second word ’totch’, there is larger edit distance error with the list of words generated [‘one’, ‘two’, ‘three’, ‘four’, ‘ten’, ‘forty’, ‘lakh’], since the word model is not trained on Indian corpus. Among the words, ‘lakh’ seems most probable because it is the only word that can appear between two numbers. Similar processing follows for the other words except for ‘sever’. The edit distance similarity list is [‘one’, ‘three’, ‘four’, ‘five’, ‘seven’]. The corresponding word in $\mathcal{S}_2$ which is obtained from amount in numbers is ’two’. According to MNIST error maps, similar to those in the Sudoku example in the main text, the digit 2 is similar to the digit 7 and sever has least edit distance with seven. Thus ‘seven’ is selected as the last word. However, it is flagged for human validation.

Since there are many corrections involved here, it requires human verification before further progress; but the cheque can be verified to be correct. Processing of signature 2b and date can be similarly done. A comparison with end to end CNN model is shown in Table 1.

References

[1] IDRBT Cheque Image Dataset `https://www.idrbt.ac.in//icid.html` Accessed: 2021-07-30

[2] Scheidl, Harald, Handwritten Text Recognition with TensorFlow, 2021, GitHub, GitHub repository, `https://https://github.com/githubharald/SimpleHTR`

[3] Chintala, Soumith and Osherovich, Eli, Basic MNIST Example, 2020, GitHub, GitHub repository, `https://github.com/pytorch/examples/tree/master/mnist`

[4] Agrawal, Prateek and Chaudhary, Deepak and Madaan, Vishu and Zabrovskiy, Anatoliy and Prodan, Radu and Kimovski, Dragi and Timmerer, Christian, Automated bank cheque verification using image processing and deep learning methods, Multimedia Tools and Applications, 2021, Springer

[5] Raghavendra S.P and A. Danti, “A novel recognition of Indian bank cheques based on invariant geometrical features,” 2015 International Conference on Trends in Automation, Communications and Computing Technology (I-TACT-15), 2015, pp. 1-5, doi: 10.1109/ITACT.2015.7492682.

[6] R. Ramanathan, S. Ponmathavan, N. Valliappan, L. Thaneshwaran, A. S. Nair and K. P. Soman, “Optical Character Recognition for English and Tamil Using Support Vector Machines,” 2009 International Conference on Advances in Computing, Control, and Telecommunication Technologies, 2009, pp. 610-612, doi: 10.1109/ACT.2009.155.

[7] Marti, UV., Bunke, H. The IAM-database: an English sentence database for offline handwriting recognition. IJDAR 5, 39–46 (2002). https://doi.org/10.1007/s100320200071