Smart Industrial Scanner for Implementation of Relevant Data Parsing from Prescriptions Using SSWF Algorithm

Jephin V. Jose, Sherin Eliyas, Sathish Kumar, and Angeline Benitta

Abstract Scanners have a wide range of functions. When it comes to an embedded system that converts scanned documents into meaningful data, there is seemingly a void in the industry. Data that can be mined from prescriptions are invaluable and any approach to make meaningful sense of such data is always a beneficial. Such approaches not only push the bounds of how much creative programming can achieve, but also how much it means to the people who can benefit from such innovations. We implement one such innovative approach to convert medicine data within a prescription to make meaningful sense of the data that resides within the prescriptions. We propose a device for converting a medical prescription into a standardized format. To achieve this, we implemented a HP Ink Tank 410 scanner connected to a Raspberry Pi 4 running Ubuntu, we propose a revised algorithm for detecting and parsing related medicinal information from a prescription. This algorithm guarantees a reduction in processing time and improve improves performance.

Keywords Medicinal data · Improved processing · Improved performance · Prescription digitization

---

The original version of this chapter was revised: The second author's name has been changed to "Sherin Eliyas". The correction to this chapter is available at https://doi.org/10.1007/978-981-19-7982-8_57

J. V. Jose · Sherin Eliyas · Sathish Kumar (✉) · A. Benitta
Hindustan Institute of Technology and Science, Chennai, India
e-mail: sathishkm@hindustanuniv.ac.in

J. V. Jose
e-mail: 20248001@student.hindustanuniv.ac.in

Sherin Eliyas
e-mail: sherine@hindustanuniv.ac.in

A. Benitta
e-mail: dangeline@hinduatnuniv.ac.in

N. Marriwala et al. (eds.), *Mobile Radio Communications and 5G Networks*,
Lecture Notes in Networks and Systems 588,
https://doi.org/10.1007/978-981-19-7982-8_52

1 Introduction

With the advent of advanced techniques in the field of deep learning and innovative strategies to convert any handwriting into text has provided us with an advantage of making innovations to such techniques that will improve the efficiency of these algorithms as well as come up with solutions to issues that plague the healthcare industry. In Fig. 1, the growth of digital medical industry demands the need for digitized health data and there is a need for efficient methods for generating such data.

The medical industry generates an enormous amount of data, but unfortunately they require highly specialized set of tools to parse them and convert them into meaningful information. Developing such specialized tools require immense research. Although there are a lot of studies being conducted to find novel approaches to equipping the medical industry with advanced technologies that has the potential to improve lives of many people, which include the use of robotics to improved healthcare service providers, they tend to focus on standardizing the formats and services the industry offers. The need for technology to parse information from previously unstandardized forms of data can be very valuable in studying a patient's previous medical history and much more. These kinds of data combined with the modern infrastructure of advanced healthcare ecosystems can lead to many use cases where such technologies can be implemented. The techniques required for such innovations can be formulated by simply studying and researching the current drawbacks in the industry and can be implemented to provide useful services. We propose a similar strategy for parsing relative information into a standardized format for digitizing data contained in prescriptions.

A HP Ink Tank 410 scanner is capable of scanning images at 1200dpi. This scanner is connected to a Raspberry Pi Ubuntu and is connected via USB and CUPS. HP Ink Tank 410 scanner is set to scan the images at 400dpi as this is seemingly a good

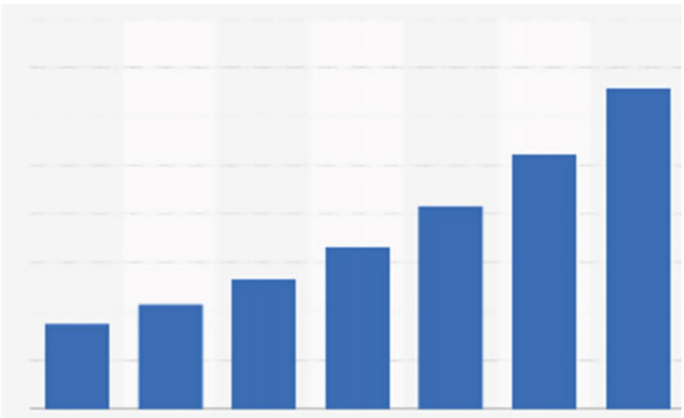


Fig. 1 Growth of digital health market

tradeoff between the image qualities and file size. The Raspberry Pi uses PyCharm as the preferred IDE for development.

1.1 Algorithm Description

Selective sliding window filtering (SSWF) algorithm is an improvement on C-Cube algorithm and 3-step filtering algorithm. The algorithm comprises of 4 steps that can be used to efficiently provide with the required data that can be parsed from the prescription. This technique is novel in the sense that it utilizes the vertices data that in Fig. 2 Google Vision API returns along with the text OCR conversion and uses it to the advantage of filtering out the irrelevant data from the text that is parsed from the API.

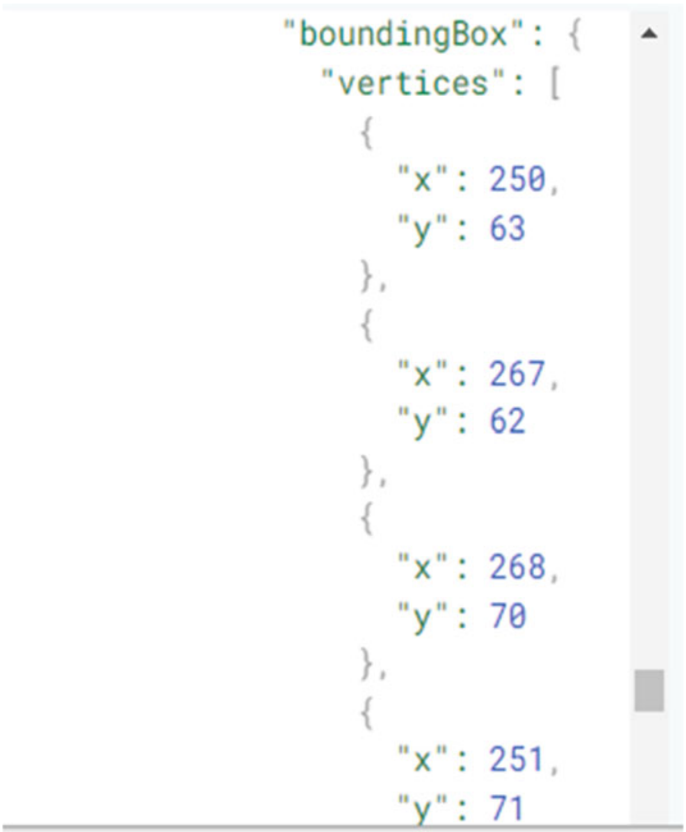


Fig. 2 Response from Google Vision API

The algorithm selectively searches for the relevant words in the data and then searches for information relevant to the newly found piece of data. This approach, however, is scalable and can be implemented in a wide array of use cases that struggle from parsing niche categories of data.

The algorithm then proceeds to rank the similarities of the relevant data with an internal dataset. The data can be used to implement the keyword search as search for a large dataset consisting of the data that can be parsed. The parsed information is then stored in a structured format of key, value pairs and can then be used to make meaningful sense of the data and perform analysis on the data acquired.

1.2 Datasets

The algorithm is tuned to efficiently use as little data as required to perform the searches. This algorithm uses two different datasets. The first one for the initial search and the second one for finally searching the relative information relevant with respect to the data gathered from the first dataset. An efficient approach to data collection and processing was of high priority as the use case demands highly efficient performance figures in terms of usability and scalability. This prompted the use of innovative strategies to minimize the amount of times the algorithm interacts with the datasets. These are described in detail later on. Data was collected in house and from external sources like WHO and so on. Data that was collected was done after immense research implementing state-of-the-art techniques for processing, cleaning and collecting only the data that will be required for the most general use case.

1.3 Environment

Choosing the most efficient and user friendly platforms and frameworks was key in choosing the runtime environment for implementing the algorithm. The system is implemented using Python, Flask, Numpy, Pandas and Google Cloud Vision API Library. This was implemented in a REST architecture utilizing all the advantages of a REST API. This consists of, but not limited to faster performance, quick integration and flexibility. Python was chosen as the standard language due to the immense community support and the vast amount of libraries and frameworks that can be utilized. Flask was chosen as a quick and efficient way of implement a REST API for dealing with the request from the client system that consists of the Base64 encoded prescription. Pandas stands as the unparalleled data storage and processing framework for such data heavy tasks as it is highly efficient in dealing with big data and can be utilized for easy manipulation, cleaning and pre-processing of the dataset. Numpy was chosen to provide the algorithm with the arbitrary math operations to make the functionalities possible.

1.4 Test Hardware

We implemented the scanner using a HP Ink Tank 410 scanner at 400dpi, coupled with a very minimally powerful Raspberry Pi 4. This combination of devices can efficiently convert scanned data into a well-structured format.

2 Literature Review

While only the words that contain the names of the medicine are of interest, as they are mixed with other irrelevant words extracted from the prescription, it is non-trivial to isolate them [1]. Artificial intelligence and ingenious solutions have enabled us to automate different elements of the screening process and present metrics for evaluating the performance of the automated process [2]. The usage of extended MNIST has been explored and the results support the efficiency of proposed model to identify the poor legibility of handwriting and transform it into readable correct text recognition [3]. In order to recognize doctors' handwriting with higher accuracy, more medical words and associate recognized terms with prescriptions will be dealt with by using a larger medical term corpus. The proposed doctors' handwriting recognition system will make it possible to reduce medical errors and save medical cost and ensure healthy living [4]. People face problems in collecting all the documents including debts and receipts. In the current COVID like pandemic situations, it is even worse if one needs to visit a hospital to collect documents [5]. The results indicate that major hospitals are, at present, using AI-enabled systems to augment medical staff in patient diagnosis and treatment activities for a wide range of diseases. In addition, AI systems are making an impact on improving the efficiency of nursing and managerial activities of hospitals [6]. Scientists seek various possibilities using computer technologies, especially artificial intelligence (AI) enhanced methods, for healthcare services and medical diagnoses [7]. The image data contains a lot of redundant information; how to use the effective information in the image to transform the image style becomes very important [8]. The upsurge and flourishing of text mining paved the way for a new beginning in the area of information extraction (IE) and information retrieval (IR). As the term suggests, it is to mine relevant information from the text document. The text document could be un-structured or semi-structured. There exist different approaches and methods for text mining and most of these techniques are computational linguistics with Python library related functions [9]. Since the handwriting of physicians is hard to peruse out, the proposed system simplifies the process of comprehending such handwriting by the methods for handwriting recognition utilizing the OCR techniques [10]. Raspberry Pi (RPi) is a credit-sized mini-computer with great capabilities similar to a PC. In this study, it is used as a remote enrollment node. The application of Raspberry Pi and cloud computing has given a new direction of research into the field of Internet-of-Things (IOT) [11]. In the pre-processing step, the system digitizes a paper-made document into a grayscale image using an optical scanner and converts the grayscale image into a binary image. Furthermore, the regions containing text are located and each image is segmented from the word. Then, a smoothing and normalization processing is applied for eliminating noise and variation of size, slant and rotation before performing the recognition. The recognition methods consist of feature extraction and classification [12].

3 Modules

The system consists of six modules. Modules are categorized based on the amount of tasks each module performs and by the frequency of their operations.

3.1 Module 1

Module 1 consists of input data management. This module takes in the input and converts the data into an image file that is temporarily stored in the local storage. The image is stored according to the content type that is mentioned in the Base64 code. Python Base64 library is used for this purpose.

3.2 Module 2

Module 2 consists of the Google Vision API environment setup and sending the data to the Google Vision API. The environment setup consists of authentication, data conversion, response management and response parsing.

The response from the Google Vision API consists of a JSON-like syntax where each word detected consists of its description, vertices for the bounding box in Fig. 3 that encapsulates the word and so on.

3.3 Module 3

Module 3 consists of a text parsing component that scans the document containing texts against the dataset containing keywords. These keywords consists of carefully designed words that corresponds to the different combinations of the dosages of medicines. As they are key to finding relevant medicine information. These keywords also consist of different words/combination of characters and numbers that correspond to the dosage, i.e., the digit 5 may also be returned by the Google Vision API as the letter S in the English alphabet. This is done mainly to account for the errors that

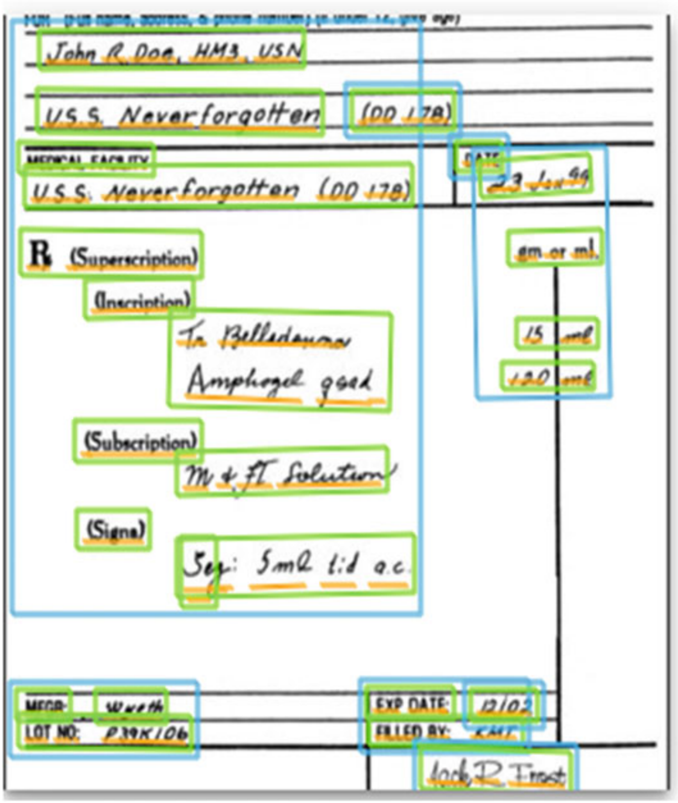


Fig. 3 Bounding boxes created by OCR

Google Vision API may return. These are mostly due to the handwriting of doctors being very complex.

This module hence scans the document for words returning a fuzzy match percentage of above 98 for every word in the keyword dataset. This then returns a candidate dosage that is accurate enough for the algorithm to do the rest of the processing.

3.4 Module 4

Module 4 consists of the second layer of the data parsing architecture. This is, however, the most complex and crucial component of the algorithm. This component is key to finding the relevant information from the medical prescription. As the candidate word that was returned in module 3 also contains relative position of the word in the document by means of the vertices of the bounding box which encapsulates the word, this can be used to our advantage in parsing the words in the immediate vicinity of the currently found word. This section of the algorithm searches for words in the immediate vicinity of the candidate word and returns a list of potential medicines.

3.5 Module 5

Module 5 consists of checking the list of words returned by module 4 and checks them against the second dataset in the architecture. This datasets consist of names of medicines currently in circulation at the time of performing this task. Every potential word in the list returned by the module 4 is checked against the dataset and are then ranked according to their fuzzy match score which is tuned to be above 85. The word that has the maximum score is selected as the name of the medicine.

3.6 Module 6

Module 6 consists of the data matching component of the architecture. The words found to be the name of the dosage and the name of the medicine are paired together to form a key value pair of data. This data can be utilized for efficiently processing the data and can be useful for a wide number of operations.

4 Our Approach

Our approach for creating a very efficient and scalable data parsing utilizes our novel selective sliding window filtering (SSWF) algorithm for parsing relative data contained within a medical prescription. This approach consists of two different datasets that can be used to find the accurate words that are found in the datasets.

4.1 Data Collection

Data collection for the datasets have been carried out keeping the possible biases in mind, this includes methods of determining the most common keywords that can be present within a prescription. It has also been taken to account that certain words can be mislabeled in the output of the OCR and a Rosetta Stone like dataset consisting of possible errors that can be captured by the OCR has been included in the dataset that consists of the different variations to a keyword and its actual correct representation added as key value pairs. This dataset is very crucial in finding the correct word in the document [13]. We are currently only focusing on the dosage information of every medicine to avoid the biases that can be caused by other common words in the document like tablets, syrup, etc., and are currently not considering any such words. We are only detecting words that correspond to the dosage information of a particular medicine. Dosage words include words like 10 mg, 25 ml and so on.

4.2 Selective Sliding Window Filtering (SSWF) Algorithm

The selective sliding window filtering (SSWF) algorithm stands for the approach taken for finding words in the immediate vicinity of the detected dosage words in the prescription document. This approach is motivated by the need for a highly efficient architecture for parsing information from the document [14]. This works by first using the detected dosage words and extracting its bounding box vertices returned by the Google Vision API.

4.3 Priorities

The priorities of this algorithm have been decided after immense research and are tuned to efficiently return the word in the immediate vicinity of the dosage word. This is done in three different directions, left, top and bottom. These directions are chosen as they are the most common places a medicine name is seen with respect to the dosage information [15]. However, the most common direction in a prescription is toward the left of the dosage information, e.g. Ranitidine 150 mg in Fig. 4. So this is the default location for search.

If a medicine name is not found in this direction, the algorithm then checks in the direction corresponding to the top of the dosage word. Same followed by the direction corresponding to the relative bottom of the dosage information.

4.4 Proposed Method

The search for a word in the immediate vicinity of the dosage word consists of extracting the vertices and setting a new bounding box through which all words falling within them is detected. The new bounding box is decided with respect to the vertices of the dosage word. If $[x, y]$ vertices of the dosage word that correspond to the bottom left, bottom right, top right and top left are assigned as A, B, C and D. And $[x, y]$ vertices of the new box that correspond to the bottom left, bottom right, top right and top left are assigned as W, X, Y and Z. In the case of searching toward left of the dosage word, the vertices are assigned to the new box are set as

$$W = A[x] - \text{decrement value of } x,\ A[y] - \text{decrement value of } v$$

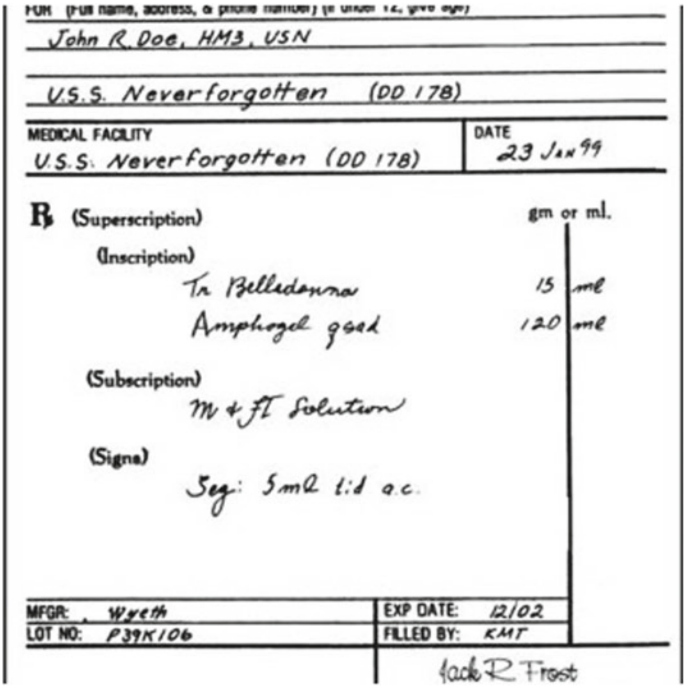


Fig. 4 Medicine name to the left of dosage words

$$X = A[x, y]$$

$$Y = D[x]-\text{decrement value of } x,\ D[y]-\text{decrement value of } v$$

$$Z = D[x, y]$$

Here, decrement value of x, decrement value of y are constants that allow the system to move the search window every time it does not return a value.

Here, the solid blue line refers to the bounding box created by the OCR and the dotted green line refers to the box created by the algorithm in Fig. 5 to look for potential medicine names.

The values are set globally independently for each direction. These values have been tuned to fit the widest use cases and to reduce errors while searching for words.

The resulting list of words is checked with an external dataset consisting of medicine names in Fig. 6. This is then fuzzy matched with each of the resulting words. If an accurate match is found that satisfies the threshold set, it will return the name of the medicine. This word is then paired with its respective dosage word and then returned as a key, value pair.

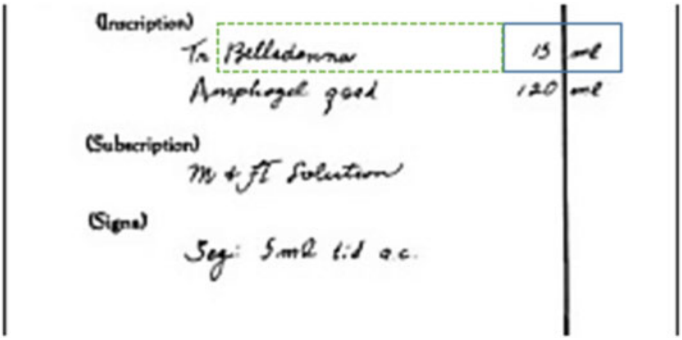
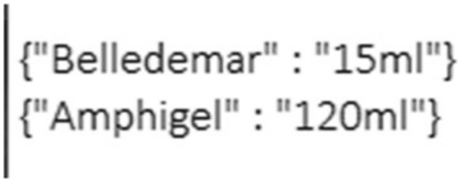


Fig. 5 SSWF algorithm searching for words

Fig. 6 Medicine data in key, value pair



We have also a check for supplicates at this step to account for any duplicates that might be detected by the algorithm. Hence, duplicate key, value pairs are removed from the final result.

5 Conclusion

In this paper, an efficient and scalable algorithm is proposed for innovatively search for medicine names in a prescription and return their details along with their dosage information using a HP Ink Tank 410 scanner and a Raspberry Pi 4. The devices are connected via USB and CUPS and is simple to implement. This algorithm innovatively searches for potential words in three different directions relative to the keyword first detected. This method not only works on printed prescriptions but also works efficiently on handwritten prescriptions. The future revisions of this algorithm can be potentially improved upon with a more accurate OCR system that has trained exclusively on prescriptions. Such a system can alleviate the need for correcting errors in the OCR text and hence reducing the complexity of the algorithm and further improving performance. One more aspect of a better result out of this algorithm would be the reality of a structured prescription format that is standardized. This would be beneficial in resolving some of the major issues that were noted in research.

References

1. Gupta M, Soeny K (2021) Algorithms for rapid digitalization of prescriptions. Vis Inf
2. Soeny K, Pandey G, Gupta U, Trivedi A, Gupta M, Agarwal G (2021) Attended robotic process automation of prescriptions' digitization. Elsevier
3. Shaw U, Tania Mamgai R, Malhotra I (2021) Medical handwritten prescription recognition and information retrieval using neural network. IEEE
4. Tabassum S, Takahashi R, Rahman MM, Imamura Y, Sixian L, Rahman MM, Ahmed A (2021) Recognition of doctors' cursive handwritten medical words by using bidirectional LSTM and SRP data augmentation. IEEE
5. Das A, Anand R, Dash A, Buddala R (2021) A study on shift towards digitalization of medical reimbursement by insurance companies during COVID like pandemic situation. IEEE
6. Lee DH, Yoon SN (2021) Application of artificial intelligence-based technologies in the healthcare industry: opportunities and challenges. 1ORCID
7. Yan K, Ji Z, Jin Q, Wang Q-G (2021) Machine learning for AI-enhanced healthcare and medical services: new development and promising solution. IEEE
8. Li H-A, Zheng Q, Qi X, Yan W, Wen Z, Li N, Tang C (2021) Neural network-based mapping mining of image style transfer in big data systems. IEEE
9. Sakthi Vel S (2021) Pre-processing techniques of text mining using computational linguistics and python libraries. IEEE
10. Butala S, Lad A, Chheda H, Bhat M, Nimkar A (2021) Natural language parser for physician's handwritten prescription. IEEE
11. Shah D, Haradi V (2016) IoT based biometrics implementation on Raspberry Pi. Elsevier
12. Menon DR, Keerthika P, Nancy Madonna G, Nandhini S, Jayanthi AN (2021) Automatic number plate recognition system using Raspberry Pi and Python. IRJET
13. Marriwala N, Rathee P (2012) An approach to increase the wireless sensor network lifetime. In: 2012 World congress on information and communication technologies. IEEE, pp 495–499
14. Bhattacharyya S, Das N, Bhattacharjee D, Mukherjee A (eds) (2016) Handbook of research on recent developments in intelligent communication application
15. Gupta V, Marriwala N, Gupta M (2021) A GUI based application for low intensity object classification & count using SVM Approach. In: 2021 6th International conference on signal processing, computing and control (ISPCC). IEEE, pp 299–302