AN INTELLIGENT SYSTEM TO AUTOMATE THE INQUERY IN LOGISTICS INDUSTRY USING AI AND MACHINE LEARNING

Leo Liao[1] and Ang Li[2]

[1]Crean Lutheran High School, 12500 Sand Canyon Avenue, Irvine, CA 92618, USA
[2]California State University, Long Beach, USA

ABSTRACT

Operator and sales employees in the logistics industry often have to submit the same inquiry repetitively to different vendors and opt in for the quotation that will generate the greatest profit for the company [4]. This process can be very laborious and tedious. Meanwhile, for smaller companies that do not have a well-constructed database for quotation information, monitoring employee's work is simply difficult to achieve [5]. To increase the efficiency of sales' workflow in this particular industry, this application devises a platform that automates the inquiry process, analyzes quotations from different vendors, retrieves the most profitable one, and documents all inquiries an employee has committed [6].

The results, after a series of intensive testing, prove to be promising and satisfying. The machine learning model can successfully fetch the most cost-effective price after analyzing a list of emails containing common languages used in the industry. All histories of an employee's inquiry can be correctly displayed on any front-end device. Overall, the obstacle presented above is largely solved.

KEYWORDS

Automation, Quotation, Analysis.

1. INTRODUCTION

The logistics industry is an industry that provides transportation services from one geological point to another [7]. Logistics companies manage and process service requests from upstream parties (usually importing and exporting corporations) [8]. When clients seek these companies for a service, companies return a quotation. If clients accept the quotation they receive, logistics companies then process these requests by arranging specific transportation services, communicating with customs, and finally carrying the goods to the destination. Although this general workflow is divided into different tasks and is much standardized, specific responsibility assigned to individual workers in the working process sometimes is not structured and generalized [9]. It means that there is still a potential way to optimize some steps in the workflow. More specifically about the problem itself, when sales and operators render a quotation to clients, they have to contact different trucking companies to request a quote and service for clients. Traditionally, sales and operators have to email different parties with the same information several times and receive the quotation they have obtained. By comparing various costs and previous impressions of these companies, logistics industry employees make an independent decision about which trucking companies to cooperate with. This working experience is, first of

David C. Wyld et al. (Eds): NATP, ACSTY, CCCIOT, MLSC, ITCSS - 2022
pp. 105-113, 2022. CS & IT - CSCP 2022 DOI: 10.5121/csit.2022.120109

all, not productive and controllable for a company. Second of all, it is not pleasing for all employees due to its repetitive and redundant nature.

One solution is then proposed to address this awkward situation for both logistics industry companies and employees. Our platform develops an interface and server to allow users to input information only for one time and submit a request individually to different trucking companies. By implementing some pricing algorithms, the server will retrieve the most profitable one and suggest that to the user from the email response sent back. In one word, one key contribution this platform makes is the reduction of time-consuming working experience and increase in productivity of the entire company [10].

Employees in this particular workflow still primarily rely on phone and emails. They will make a call or send an email to different trucking companies to obtain the information about the quotations. Based on their working experience and consultations, employees will make an individual decision on which trucking companies to choose. Although emails reduce some redundancies of the workflow, inputting the same information several times is not convenient for all involved parties, especially for employees. Moreover, there is a lack of standardization of which trucking cooperators. Employees make decisions that might not be the most secure or profitable ones for a company. In the same logic, although sometimes experience can pave the way for the most fit option, newer, inexperienced employees choosing a profitable and stable cooperator is also hard to accomplish. Often a company faces a deficit on a single service due to an too expensive cooperating trucking service that leads to higher costs but brings a low income at the end. They also have to accept the risk when choosing a low-cost cooperator because cargos can get lost due to an unqualified service. Additionally, obtaining the costs through email can be redundant to perceive because employees have to process these prices by adding them together and then comparing. Different trucking companies' quotations require employees to do different calculations everytime, which is extremely inefficient and repetitive. Finally, it is difficult for companies to record the history of quotations because all information is processed by hand and email for which no one is accounting. Due to the lack of recording, unqualified trucking companies cannot be detected and filtered out for future cooperation, leaving another risk of a deficit. In one word, conventional communication methods are not modern and advanced enough to carry out current workflow and make the most profit for logistic companies [11].

Our method is to simplify the information inputting process to a one-time work. The platform allows employees to input the information about the inquiry only once. After clicking the submit button placed at the bottom of the screen, all emails will be distributed to each destination. Compared with traditional methods, employees do not have to compose an email with each service provider, waiting for them to reply. Another is automation of calculation in emails. Conventionally, employees have to do a lot of calculations by adding other surcharges indicated by trucking companies to obtain a final price. On this platform, the server implements algorithms to analyze each email and extract all prices from the email to calculate a final cost. This new method saves employees time to complete other important tasks. In addition to automatic calculation, this platform also helps compare each price and keep them in record. Employees do not have to then rely on paper and pencil to record. Lastly, employees are free from the burden of the management of all inquiries as the platform will save and display their work immediately after they submit their requests. Companies also enjoy this benefit as well because prior to the invention of this platform, they did not have the opportunities to check individual employee's working progress [12]. Constructing a management structure will help businesses to monitor each inquiry and its status of completion.

Proving the outcome from the platform is relatively simple. We employ a number of previous emails, feeding it to the model and checking if it will compute the desired calculation and return results. Using approximately fifty emails sent back to the company, the machine model can detect the price ($+number) in each clause and extract it from the text sample. Although some little errors remain, most of the algorithm is successful and effective. The second stage of proof is to input information by oneself to stimulate a real application of the platform. Throwing in what employees usually compose in an email from the front-end, the retrieving result is satisfying.

As a comparison, employees manually send individual emails to vendors to request a quotation. Email sent back will be artificially analyzed to extract a price. Although it is mostly successful, the speed for which this information is processed is very slow and redundant. It usually takes ten minutes or more for an employee to complete a task that can be handled in a second by the platform.

The rate of success for each trail, by average, is greater than 70% with multiple distractions adding in, such as zip code and phone numbers. All experiments and evaluations, we intend to stimulate the real working situations, since this is the primary goal of the platform. We even modify harder situations to accommodate the possibility that any emails are not formatted ideally or content contains errors in syntax. Nevertheless, the success rate is acceptable and keeps increasing as more updates are made on the algorithm and model.

The rest of the paper is organized as follows: Section 2 gives the details on the challenges that we met during the experiment and designing the sample; Section 3 focuses on the details of our solutions corresponding to the challenges that we mentioned in Section 2; Section 4 presents the relevant details about the experiment we did, following by presenting the related work in Section 5. Finally, Section 6 gives the conclusion remarks, as well as pointing out the future work of this project.

2. CHALLENGES

In order to build the tracking system, a few challenges have been identified as follows.

2.1. Regular expression implemented to extract prices

One challenge is the regular expression implemented to extract prices. Since each email contains surcharges, models need to identify those surcharges and add them together. For the first models, the algorithms do not identify that extra information, but only retrieve the first price it detects. For example, the email will contain a main price for a particular shipment of $40.00. However, it will also list other surcharges for any additional need such as fuel and other service fees. The first model only extracts the first price it sees, abandoning other costs. For revision, I add components from regular expressions so that whenever the model sees a dollar sign in the email, it knows the next string is a cost that needs to be picked up to store. This time the issue is fixed. All numbers with a dollar sign will be grabbed and accumulated to calculate a final price. Even with some kinds of constraints (for any price must start with a dollar sign as a signal), this method is mostly successful because it models how humans perceive and comprehend an email.

2.2. Classifying emails to different users

Another challenge is classifying emails to different users. Since there will be multiple users to send emails to different destinations, the data structure of the system must be efficient enough to store and locate emails. Even though the system does not report an error when emails are not directed to a specific destination, it is mostly due to the fact there is only one tester on the platform. In real situations in which numerous emails will be processed and results distributed to each end user, managing these emails is extremely crucial as it is individual work that needs to be monitored. To this end, we develop the idea of "quote-id", "minor-id", and "user-id". Minor-id keeps track of different vendors. Quote-id records each quote submission in general. User-id records from whom the quote is submitted. Once these labels are created, emails can be managed more structurally for business and intensive use.

2.3. Price comparison

The last challenge is price comparison. The comparison's goal is to extract, among multiple costs, the median one. The difficulty of this problem is not as high as others. However, many scenarios need to be considered because only one implementation can handle this task. For example, many vendors will return the same price. And there are chances that even numbers of quotations are returned, making it impossible for the model to choose a price to display. To solve this issue, algorithms will display all prices with minor-id's and recommend the relatively higher price to the users. At this time, human participation is needed to make a decision for the machine model, opting into the best choice based on their experience. This is the only process in which human power is needed. But it will not decrease the efficiency overall because it can connect to later workflow, which requires human participation, nonetheless.

3. SOLUTION

The entire platform is built in two parts: user interface and server. In the user interface, employees input information from an inquiry [13]. After filling out all information needed, it will be sent to the server that is going to compile it into a professional email and send it individually to vendors. These vendors will be labeled with an id, respectively, and stored in an email-list. Before sending each email, the server will assign it with a unique id (quote-id + minor-id + user-id) so that responses given back can be distributed to each end user correctly. To continue, once an email is back from a trucking company, it will be fed into the model. There it will be decomposed and analyzed in smaller components. After several emails are back, the comparison model will be prompted to extract the most favorable cost, which will then be listed as the recommended one in the query results. To ensure stability and security of each process in the workflow, other prices will also be listed for employees to confirm before making a decision. The same idea applies to other quotes. After submitting multiple inquiries and receiving some quotations, that information will be organized under the "My" tab. Each inquiry is labeled as the time it is submitted to the server so that employees have an easier time to identify how many and when they submit an inquiry [14]. For a company, managers can have an intuitive idea of the performance of individual employees on this particular workflow.

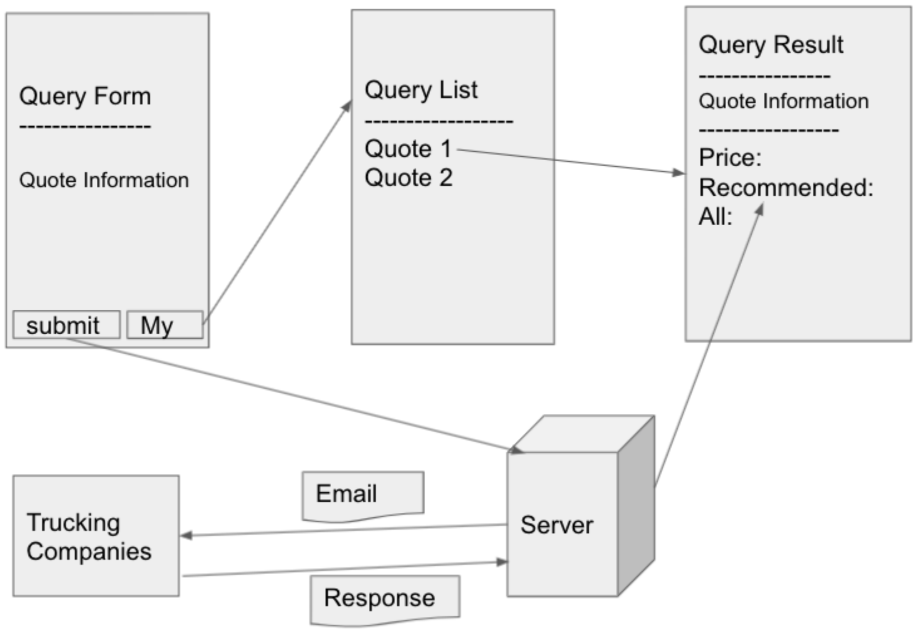

Figure 1. Overview of the project

For the user interface, I employ the flutter SDK to implement a number of widgets, including text fields, buttons, list-view, and alert-logs. Text field will collect information about user input. Buttons will be prompted to send requests to the server. List-view will be used to display quote names. Alert-logs will be used to display specific information about a quotation. After a user logs in to the system, information about an inquiry will be provided. When clicking the confirm button, all information above will be sent to a specific function implemented in the server. In that function, information will be stored in a json dictionary. Then an email is prompted in that function to send the inquiry to different email addresses listed in the email list. After that, the server will scan the inbox every few minutes, checking if any of the sent emails are responded to. If there are, another server function will fetch this email and input it to a function that will parse the price from the email. This process is the most important part of the platform. I use regular expressions and the NLTK package to accomplish this task. Examining over fifty email samples, we develop a pattern of expressions in the emails. Since all samples have common characteristics like prices and other professional words, regular expressions can accommodate those and pick up the price after a particular signal string. Even if sometimes the target is lost because of over-complicated clauses and exceptional expressions with numbers, the success rate for the processing is satisfying overall. Most regular quotations can be withdrawn correctly. After a price is retrieved, information about the inquiry will be accessed and sent back to the user interface. There, users can check the status of each inquiry they make, including the information they have input previously, all prices with vendor number, and a recommended cost derived from the algorithm. The user will make an independent choice from that point, pertaining to which party the company shall opt in with.

Above is the workflow of this platform, the data structure is presented below. Since it is used by employees, I set individual users at the top of the structure. From there, each user will develop independent quotes. Each quote contains its respective information. Until now, this structure is perceived as the most efficient and easy to manage one. Since we do not have to delete or transplant information in this particular dataset, this structure is the most secure and stable one to maintain and sustain. Future modification is definitely needed; however, only addition will apply, which is the most safest measure to take. Other data structures might also apply in this project, but it will not be as efficient and stable as the current one because data has to be transferred over to another point and deleted in its current place, which makes it not ideal as there are dangers of losing data.

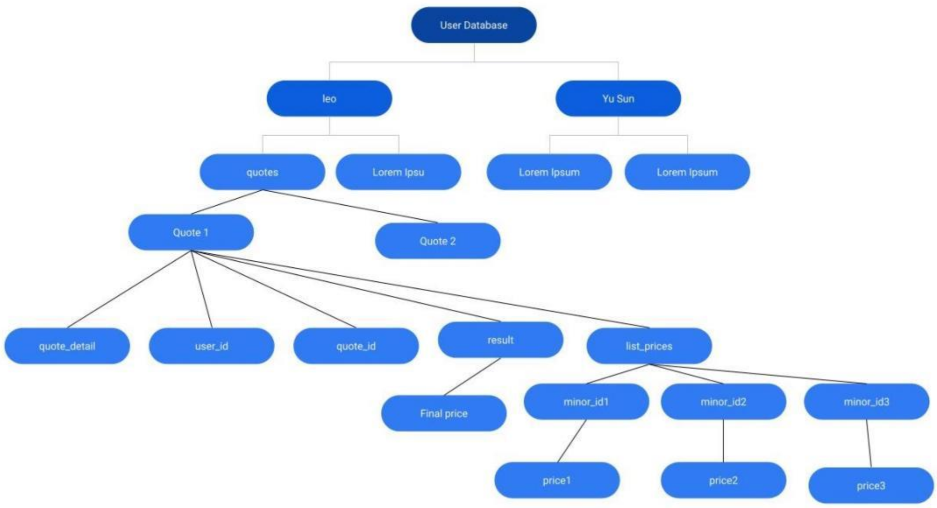

Figure 2. Data structure

4. EXPERIMENT

4.1. Experiment 1

We developed a platform to receive quotes from different logistics industry employees, extract all quote information and automatically send customized emails to the different trucking companies. Our solution reduces misleading communication between trucking companies and logistics industry employees since there are mediators between them. Also, it reduces the amount of time to send the information to trucking vendors since emails are sent instantaneously after logistics industry employees send the quote requests.

For our experiment, we utilized fifty quote samples that were sent to a group of three people. Also, these fifty samples were sent to our platform. For some of the samples, we sent them every minute while others were sent 15 - 20 minutes between them. For the trucking vendors, we used 5 email accounts to simulate trucking vendors.

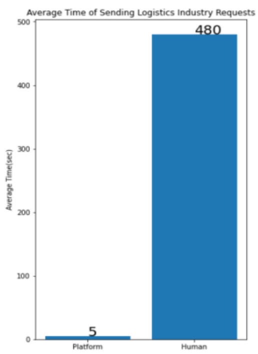

Figure 3. Average time of sending logistics industry requests

The above chart bar shows that the average time of the platform is less in comparison to the human method. Our platform method takes about 5 seconds to send the emails to the simulated trucking vendor emails while the human method takes on average 480 seconds to send the emails to the vendors.

4.2. Experiment 2

We use Natural Language Processing and Python regular expressions to extract the price from the email. Natural Language Processing is a powerful tool that can extract the parts of speech of a sentence. Thus, we use the NLTK package to create a grammar that contains a regular expression of part of speech, so we can extract the correct part of the tree that contains the description and price of the service. Then, we use Python regular expressions to select the desired price of the service. By doing this method, we were able to analyze and extract information from the email.

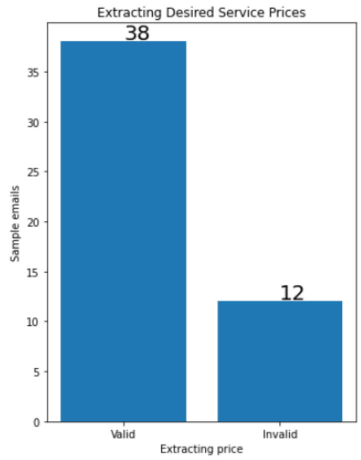

Figure 4. Extracting desired service prices

In our experiment, we use fifty sample emails to extract the prices and fetch the desired service. These email samples were obtained from real responses from the truck companies which contain diverse patterns. These patterns were used to verify if NLTK grammar was correctly designed, so description and the price of truck services were extracted from the emails. After processing and extracting the info from the fifty email samples, we obtained that our system could analyze and extract 38 of the email samples correctly which corresponds to 76% of the samples while 12 samples of the emails were returned incorrectly.

In experiment 1, as we expected, our platform works more efficiently than the human method since the time of extracting and sending the emails to trucking vendors takes about 5 seconds on average while the human method takes about 480 seconds. Also, our platform reduces the amount of mistakes that can occur by sending the quotes to the trucking vendors since there are no intermediates between the trucking vendors and logistics industry employees.

Natural Language Processing is a useful tool to extract sentences that contain part of speech patterns. In experiment 2, after we used regular expressions and NLTK packages, we observed that the success rate of our system was 76%. Even though the description and prices were extracted incorrectly from some of the sample emails since they did not follow the part of speech patterns of our grammar or the system selected the incorrect service price, we could state that the success rate of our system is acceptable.

5. RELATED WORK

Freightos is also an online platform that automates the requests of inquiries [1]. In their work, they compare multiple aspects between the manual process and digital process of logistic workflow. Their production (application), compared to our application, exhibits a more thorough automation for the entire workflow. While it is more narrowly applicable, our application targets a specific client group, employees. This aspect of our work is both a strength and major difference compared to Freightos. As a full-scale automated platform, Freightos sometimes cannot obtain a proper cost for customers, unlike our platform that employs some degree of human resources to better assist customers.

Freightquotes achieves the same function as our application does [2]. One main difference, however, is the source of input to the platform. Freightquotes' target client is customers who will submit their quotes directly to obtain a price from trucking companies. Our application, instead, prompts employees to initiate the submission of prices to trucking companies and then send back costs to customers. Even though it seems to be an unnecessary step, this algorithm can reduce error in the system that may cause more laborious service processes.

In Tung Nguyen's commentary research, one of the paper's points coincides with our future plan for the application [3]. In 3.1.1, it is said that forecasting demand in the industry is "a crucial part of business operation in every sector… to come up with the most reliable data of the upcoming period" (Nguyen 9). I intend to add another algorithm to our platform to analyze the quality and quantity of service provided by each vendor and selectively predict and choose those that are cooperative with us to collaborate for future work. One major difference between our work is that this paper is mostly theorized and researched-based while our project is in a real application scenario.

6. CONCLUSIONS

Overall, to reduce human resources needed for a logistic business when requesting repetitive inquiries to different trucking vendors, I design an automated platform on which employees can enter information to obtain a list of quotes from truckers [15]. After that information is entered, an email will be generated and compiled to send to a number of vendors for reply. For the platform will scan the inbox every once in a while, once replies are back, algorithms such as regular expression and natural language learning model will analyze the content of that email to extract a final price, taking extra fees into the account and calculations. After experimenting with a number of test cases, the system and algorithm proved to be effective for most professional business email. With a few editions on the implementation of regular expression, specifically, the program achieves greater accuracy and efficiency. Since the number of emails and users that will be served on the platform, each user is assigned an individual id. Each piece of email will be labeled with a major quote id and two minor id, indicating from which the email is sent and to which it is sent. In these ways, a well-constructed and mature data structure can be utilized for business purposes. Through experimentation, this data structure has also proved to be effective so far for current and future use in the near future. In conclusion, the current platform is mature and ready to put into real applications. It will alleviate employee's working pressure by reducing repetitions and unnecessary efforts.

Even though its platform is considered mature and functional, it has a few limitations that will hinder it from full scale automation in the future. Since the transportation system in logistics is divided into several regions across the continent, each region has several vendors that carry the service. In what way this system will manage inquiries to be sent to a proper set of vendors that are responsible for the region is one potential threat to current practicability. If that limitation is overcome, the platform is fundamentally practical across the globe and can better serve every employee in every logistic business.

The limitation above can be solved by setting up a database that stores all trucking companies. In this database, vendors are further divided by states in which they can provide service. Each time the user calls for a service, the platform will send email to the vendors that are responsible for the state. In this way, the platform is perfected.

REFERENCES

[1] Riedl, Jens, et al. “The Digital Imperative In Freight Forwarding.” BCG, 2018, https://image-src.bcg.com/Images/BCG-The-Digital-Imperative-in-Freight-Forwarding-Nov-2018_tcm108-207934.pdf.

[2] Le Blanc, Louis, and Laura Valentine. “FREIGHTQUOTE.COM: Value-Added Intermediation in Transportation.” FREIGHTQUOTE.COM: VALUE-ADDED INTERMEDIATION IN TRANSPORTATION, 2010, https://ctrf.ca/wp-content/uploads/2014/07/3LeBlancValentineFreightQuote.pdf.

[3] Nguyen, Tung. “Enhancing Logistics and Warehouse Management For a Startup Company: Challenges and Opportunities.” Theseus, 1 Jan. 1970, https://www.theseus.fi/handle/10024/172082.

[4] Gohberg, Israel, and Seymour Goldberg. Basic operator theory. Birkhäuser, 2013..

[5] Davidson, Donald. "Quotation." Theory and decision 11.1 (1979): 27.

[6] Tharp, Roland G., and Ronald Gallimore. "Inquiry process in program development." Journal of Community Psychology 10.2 (1982): 103-118.

[7] Murphy, Paul R., and A. Michael Knemeyer. "Contemporary logistics." (2018).

[8] Lambert, Douglas M., and James R. Stock. Strategic logistics management. Vol. 3. Homewood, IL: Irwin, 1993.

[9] Van Der Aalst, Wil, Kees Max Van Hee, and Kees van Hee. Workflow management: models, methods, and systems. MIT press, 2004.

[10] Plag, Ingo. "Productivity." The handbook of English linguistics (2020): 483-499.

[11] Moore, Richard. "Imitation and conventional communication." Biology & Philosophy 28.3 (2013): 481-500.

[12] Parker, Geoffrey, and Marshall W. Van Alstyne. "Platform strategy." (2014).

[13] Stalnaker, Robert. "Inquiry." (1984).

[14] Zeisel, John. "Inquiry by design." Environment/behavior/neuroscience in architecture, interiors, landscape, and planning (2006).

[15] Kleinbaum, David G., et al. Logistic regression. New York: Springer-Verlag, 2002.