Requirement Elicitation using Natural Language Processing

Sharoon Nasim1, Zainab Zahid1*, Nosheen Sabahat1
Department of Computer Science Forman Christian College Lahore, Pakistan

Email: 213506267@formanite.fccollege.edu.pk

ABSTRACT:
This paper is the outcome of the research conducted to investigate the affective requirement engineering techniques proposed and used for developing software projects. We have assessed traditional methods and proposed an approach that covers various aspects for generating a successful project. An NLP-based model is designed that takes input from the user and gives the output in the form of a text document after processing it. We have set a 62% similarity index to achieve the maximum requirements of the required system. These requirements, in return, help the developers to develop the product with more functionality and productivity.

KEYWORDS: Requirement Engineering, Cemantic Analysis, Natural Language Processing, Requirement Elicitation.

1. INTRODUCTION

As the world is progressing, we are now experiencing a digital shift. People are now shifting their businesses to software solutions rather than relying on traditional systems. With the advancement in technology, many projects have been developed, but only a few of them became successful. These are the ones that hit the bull’s eye and fulfill their objectives. On the other hand, most projects failed to achieve their goals as they lacked clear direction and user-centered functionality. One of the primary reasons behind this is the incomplete or vague requirements by the client.

This shows the communication channel that is followed by the developers and customers. These traditional methods create ambiguities that result in vague and out-of-scope requirements. Further inexperienced developers miss the vital functionalities. Table I shows Components & descriptions involving communication between two parties & constraints.

Requirement Engineering (RE) is considered an essential part of numerous kinds of programming advancements. There are various aspects that cause a software project to fail. Unfortunately, many software solutions are not able to meet the desired objectives. Moreover, existing businesses are shifting towards digital solutions to create their digital presence. Incomplete or vague software requirements are a common reason for such failure. The motivation behind the research is to help such business owners, who have little or no knowledge about the required project, in this way they can convey their thoughts in a precise fashion and get the product that fits their needs.
In this research, we will assess some of the leading causes behind the failure of such startups and businesses. Furthermore, we will propose a solution to cater these problems related to requirement gathering.

Given below are the research questions:

1) What are the factors that are leading to the failure of software projects?
2) How to overcome these problems in an effective manner using Natural Language Processing?

2. RELATED WORK

Online shopping and digital presence have flourished as a massive domain in the modern era. COVID-19 has further fueled the shift toward online buying and selling. Researchers are playing a prominent role in finding ways that can enhance customer experience. Shervin Malmasi states in his paper that IR and NLP play a vital role in many software projects; these include product description, customer review processing, product search, recommendations, and sentiment analysis [1]. Shubhi Jain et al. states that customer review processing, product search, recommendations, and sentiment analysis are an aid to improve customer experience. Besides all this customer-side improvement, the seller’s point of view is still not given due importance, due to which they are shifting to cross-border platforms [2]. Umar et. al claimed that many businesses are looking for solutions to achieve success. This goal could only be achieved through effective and functional products [3]. To do so, software engineers need to elicit the requirements of users and build solutions that fit their needs. During the requirement elicitation process, many requirements are skipped to be catered at a later stage. [3]

L. Zhang highlights the requirement engineering process for web application development, he states that this process plays a vital role and is used to determine the success of the product [4]. C. Jones concluded from his study that the success or failure of any system relies on how the requirements were gathered and what was the quality of those requirements [5]. and most of the time, they are missed due to the human-to-human communication gap. Jane Coughlan et al., and C. W. Aybuke Aurum et al. describes one of the major issues during the requirement elicitation phase is the communication gap between developers and customers, the less experience of a developer with the required field is also a constraint. These problems are based on personalities, cognitive aspects, tools, and techniques [6][7].

The effects of the poor process of requirement elicitation, is discussed by Abudllah Mohd Zin et al. He states that in this process we seek, uncover, acquire and elaborate our requirements. This, in return, helps to generate a computer-based system. This system may consume several months and is an essential part of the software development phase [8] but if it is done poorly, then it may cost budget-overrun, rework, and even some failures.

R. Feldman and O. Bin Lin et al. state different techniques used for sentiment analysis, this area is considered as the most demanding research area of computer science. It has been implemented by using NLP techniques, sentiment analysis has been applied to a wide range of software engineering tasks [9][10]. So far, the applications of sentiment analysis have been seen in extracting and analyzing the emotions of users [11][12].

After reviewing all previous work, we have planned a computer-based NLP model that can process user input and produce functional requirements after applying semantic analysis, we are proposing this solution that will help in the requirement elicitation process.

3. METHODOLOGY

In our model, the user’s requirements will be filtered out by a machine and will ultimately remove the human-human communication gap. We propose a framework that will use NLP and

| Table 1: Components & Their Description in Tradition System used by Software House |
|-----------------------------------|----------------------------------|
| **Component**                     | **Description**                  |
| Client’s Input                    | Existing System details and desired objectives |
| Channel/ Medium                   | Meeting, Video Calls, Document    |
| Sender & Receiver                | Customer and Developer           |
| Developer                         | Who is going to develop the software solution |

LGU Research Journal of Computer Science & Information Technology 7(1) LGURJCSIT
semantic analysis [10] to bridge the gap between customers and developers. In this model, the customer will communicate directly with the machine, the system will then process the user response, list down the Nouns and Verbs from the provided description, features, and limitations. Afterward, a semantic analysis will list the relevant nouns and verbs that are 50% similar to the title and relevant keywords. These words (Nouns and Verbs) are searched from the knowledge base and all the resulting requirements are again processed through the semantic analysis and only those requirements are chosen that are 62% similar to the title and keywords of the user response. Less similarity measure has been adjusted because of limited knowledge base 827 requirements. Those resulted requirements have been written to the text file as an output. The resulted requirements can be analyzed by the developer. There would be some requirements that will enhance the developer’s perspective regarding the functionality scope. A document will then be generated that will be proposed to the customer. The customer will go through the document that contains a solution to his needs. Once approved by the customer, it will then be sent to the developer who will read this documentation. The components of the proposed system are summarized in Table II.

4. PROPOSED FRAMEWORK
Our proposed solution converts speech to text and processes it with the NLP technique. Lastly, it outputs the relevant requirements. The prototype is divided into different modules, i.e., interactive module, speech to the text module, and text process. Figure 2 shows the flow chart of our proposed system.

**Figure 2: Flow chart of the System’s technical Details**

<table>
<thead>
<tr>
<th>Component</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>User Response</td>
<td>User inputs the title, description, features, limitation and relevant keywords</td>
</tr>
<tr>
<td>Text Processor</td>
<td>Process the User Response and generate list of nouns and verbs.</td>
</tr>
<tr>
<td>Semantic Analysis</td>
<td>List down the relevant Nouns and verbs to search relevant requirements from knowledge base.</td>
</tr>
<tr>
<td>Knowledge Base</td>
<td>Knowledge base of 827 requirements have been maintained by using SQLite database.</td>
</tr>
<tr>
<td>Semantic Analyzer</td>
<td>List out the related requirements</td>
</tr>
<tr>
<td>Output Requirements</td>
<td>Resulted requirements are written into a text file</td>
</tr>
</tbody>
</table>
semantic analyzer to check their similarity with the title and keywords provided by the user. Only those requirements are chosen that are 62% similar.

4.2. Tool Evaluation
Our tool gives two different options to users. The users can copy and paste the title, description, and keywords into the system. Secondly, with the interactive module, the system has four pre-fed questions i.e.,
1) What is the title of your system?
2) Enter description of your system
3) Enter features of your system
4) Enter limitations of your system
The user answers these questions, and then these answers are added into a list and then this list has been given to the system

Figure 3: Command Line Use of the System

Figure 3 shows the results of the input in CMD. A total of 520 requirements were found in the database/ knowledge base related to the description and features. As a result, 20 software requirements were shortlisted after applying semantic analysis further detail is shared in the following sections. The similarity of each requirement is printed against each it. We have adjust a 62% similarity index at this stage because the knowledge base contains 827 requirements of previous existing systems. The similarity indexes can be raise to 80 to 90%, when a more detailed knowledge base is maintained in the future.

Figure 4: Direct User Input through Text Editor

This GUI enables the user to input his requirements by typing them into the recognized text section, whereas an interactive module has been also implemented in which the system asks questions from the user to get input, further a speech to text feature has been also added to the system where user can speak in English and system will recognize and provide the spoken text, whereas user can also provide audio notes that can be converted to the text. Figure 4 shows that for using this approach, a user must provide the title for his product in the first line and relevant keywords in the last line. This will help the system to provide more relevant requirements.

Figure 5: Interactive Module - Asking Each question separately

Figure 5 shows the interactive module. Our system asks the user, and the user then answers the questions that follow. Once the complete details are entered, the user clicks on “OK,” and the input is processed.

5. RESULTS AND ANALYSIS
In this section, we will discuss the results obtained from our proposed system.

Figure 6: Generated text file as output

Figure 6 shows the auto-generated text file as output after pro- cessing. The similarity score of each requirement is displayed against it.

Figure 7: Manual Evaluation of the Provided results
Figure 7 shows manual evaluation, where “y” shows that the resulted requirements are relevant and “r” depicts that partially relevant requirements. To find the relevancy score, or how accurately our system provides the relevant requirement we have used the formula that is given below:

\[
Relevancy\ Score = \frac{\text{No. of } y's + \frac{\text{No. of } r's}{2}}{\text{No. of Resulted Requirements}}
\]  

\(r = \text{Partial Relevant Requirements}\)  
\(y = \text{Relevant Requirements}\)

This formula considers “y”-requirements as a whole and “r”-requirements as half because they are partially relevant. The accuracy for providing relevant requirements lies between 66 to 96%.

Figure 8: User input processed after interactive mode

According to the figure above, after one of the users’ inputs, an 82.5% accuracy rate was found for Health Care System. The output files generated from the system could be studied by the client/user. He can see if the requirement of the design fits his need. He can also edit the criteria if needed, otherwise he can send this file to the concerned person for the development of the software.

6. CONCLUSION

After the detailed study and careful analysis of our proposed model, we have concluded that such a system has never existed before, therefore we have worked on this idea to aid the developers and customer to develop a product with a competitive functionality. This system gives more insight to the developer and client about the desired product as compared to the traditional methods used for data gathering and requirement elicitation. The results obtained from this model are highly beneficial for both the expert software developers and less experienced or beginners, as our system processes input and provides result from the existing knowledge base, will help the developer and client to understand the required functionality and features that could be introduced in the system to enhance its productivity.

A more extensive knowledge base could be maintained for future work, and the similarity index will then be changed from 62% to 80 or 90%. Further, more techniques can be implemented for semantic analysis. This will create a massive difference in the accuracy and relevancy of the output.

REFERENCES


Parallel and Distributed Systems, Malaysia, 2009.


