

# Ontology-Based Question Answering System for Banking Domain

W.M.D.Weerabahu, M.A.G.A.S.Amalika, H.K.C.Dulakshi, T.C.Sandanayake  
and G.U.Ganegoda

*Faculty of Information Technology, University of Moratuwa, Sri Lanka*

weerabahu.wmd@gmail.com, agragamage93@gmail.com,

chathurya993@gmail.com, tcsandanayake@gmail.com, upekshag@uom.lk

## **Introduction**

Bank Assistants help people providing answers to user queries in the banking domain. This study is to develop a digital assistant for banking domain as a chat application. Though it is trending in the world, the concept is still new to Sri Lankan Banking Sector. In call centres, card centres and enquiry desks of the bank, human intervention is insufficient. Furthermore, it takes a considerably longer period of time to process a single request. This results in a waste of time and ultimately makes customers unhappy. Hence, it tends to reduce the quality of the customer service (Kulkarni et al. 2007). The aim of the research is that the customer can interact with mentioning their queries in English and it can resolve their queries with an appropriate response in return.

This answer generation system is one of the major modules of a digital assistant designed to formulate answers for a set of Frequently Asked Questions (FAQs) and corresponding answers of the card centre of the bank. The system uses an ontology as the knowledge base to store the answers for FAQs. The digital Assistant extracts unique keywords out of the given user query and it's used to execute the ontology query to retrieve the answer. This answer generation system formulates a complete answer by defining a proper linguistic structure appending the ontology answer into it. The ontology answer is not a complete sentence but a fraction of a sentence and it should be added to create a grammatically complete sentence. As the research objectives, this system defines a suitable question pattern and match it with generic keywords to generate an appropriate answer to the given question. The answer template is created by defining an appropriate linguistic structure filling the gap with ontology query result.

## **Literature review**

Natural language generation has played a vital role in Natural Language Processing and widely used for developing digital assistants (Reiter 1995). They are developed using technologies such as rule-based, pattern-based, AI and Machine Learning. New approaches usually need a large set of dataset whereas pattern-

based approaches allow handling limited dataset. Ontology-based answer generation with template-based approaches is not frequently found out. One way of getting the ontology result is, extracting all questions that have classes from the data set while checking the cosine similarity with user query (Kulkarni et al. 2007). The answer to the most similar question is chosen and respond to the user. There is no linguistic structure definition presented in this approach, rather retrieve the whole answer saved in the ontology.

Template-based systems are natural language generating systems that map non-linguistic input directly (i.e. without intermediate representations) to the linguistic input directly. It contains gaps; well-formed output results if the gaps are filled (Reiter and Robert 1997). More precisely when the gaps are replaced by linguistic structures that do not contain gaps.

For instance, a simple template-based system might start out from a semantic representation saying that the 306 train leaves Aberdeen at 10:00 am: Departure (train306, location and, time1000), and associate it directly with a template such as [train] *is leaving* [town] *now*, where the gaps represented by [train] and [town] are filled by looking up the relevant information in a table (Reiter 1995), (Reiter and Robert 1997). Some template-based systems use grammar to aid linguistic realization.

Regular Expression rewriting is another method that can be used in Question Answering Systems. Regular expression (RegEx) substitutions are a very succinct, efficient, maintainable, and scalable method to model many NL subtasks of the QA task. The basic text substrings, such as the target or named entities, are recognized using regular expressions and replaced with an angle-bracket-delimited expression. For example, the target is marked as <TARGET>. More commonly, a named entity *e* of type *t* is replaced with <t\_es> (Keřseljand Cox 2004). Furthermore, this process involves in passage retrieval, target marketing, Question category marking and finally answer matching. Combining several features of some template-based approaches, an ontology-based answer generation is proposed.

## **Methodology**

This research was based on the ontology-based application developed to generate an answer. The proposed system is designed to handle cases where the dataset is limited and it contains questions that frequently repeat the same question patterns. For instance, what is a web card, what is a set plus card, what is the joining fee for a web card, How to apply for a web card, How to apply for a credit card etc? Datasets with a small amount of data often find it difficult to be designed with the latest cutting-edge technologies. In practice, a good amount of data is needed to

train a model, in order to be sure of the model's ability to generalize. Therefore, it is very unlikely that a small dataset such as two hundred like datasets can be effectively used to train and get a valid response. In fact, Frequently Asked Questions (FAQs) do not contain millions of data. On the other hand, rather than just filling some values such as the gap to fill the ontology answer in a text, it is necessary to define a grammatically complete and sound sentence as the final output. Considering the facts, a solution is proposed by defining a proper linguistic structure to generate an answer using natural language processing (NLP) techniques. The proposed solution does not focus on formulating answers based on the semantics of the sentence but defining a syntactic structure of the sentence.

This section illustrates the proposed methodology. For each question in the dataset, a question pattern is created and matched it with an answer template in order to formulate an appropriate answer for a given user query. For every question in the dataset, a regular expression pattern is created. Patterns are matched with a preconfigured set of unique keywords. If the question is matched with predefined regular expression pattern, an answer is generated by following a specific linguistic structure. Figure 1 illustrates the overall process carried out by the answer generation system. In order to define regular expression patterns for each question in the dataset, the user query should be entered into the system. Then it's converted into a regular expression pattern so that it can be stored in a database and retrieve the pattern whenever necessary. If a user query is matched with a particular regex pattern, then it formulates an answer for a predefined linguistic structure by filling the gap with ontology answer and output it to the user.

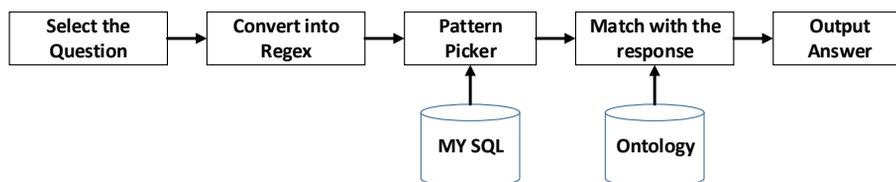


Figure 5: Process diagram of the system

The answer generation system has been designed to generate an answer for a given question in the dataset. It is necessary to go through the dataset and identify questions that follow a similar structure as they can be put into one category when creating regular expression patterns (Regex). Hence, it enables to reduce the number of regexes that should be created. Each and every question in the dataset has a defined regular expression pattern. Creating regular expression patterns involves several natural language processing techniques. Firstly, the question should be tokenized and get the lemma of the word by doing lemmatization. Then get the Part of Speech Tag (POS Tag) of each word and the outcome is used to create Regex. Regex Pattern is created in such a way that, it can be matched with

questions which follow the similar structure.

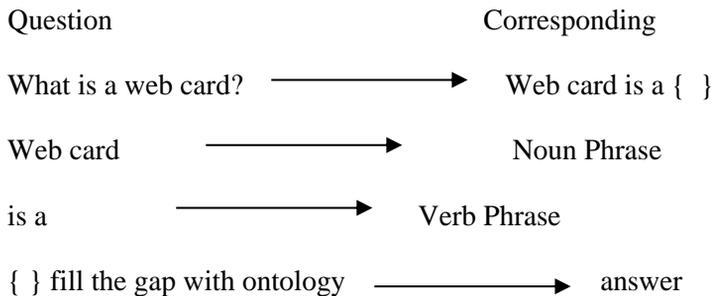
Depending on the question, the POS tag is varying. Figure 2 illustrates the meaning of each POS tag in the regular expression pattern created for the given scenario. Each question pattern must be identified correctly and create regex patterns accordingly as it can reduce the number of questions can be asked.

What is a web card? }  
 What is Set Plus Card? } <WP><VB><DT>?<JJ>\*<NN>+

WP	wh-pronoun	'what'
VB	Verb, the base form of 'is', be	
DT	determiner	'a'
JJ	adjective	'web'
NN	noun, singular	'card'

Figure 6: POS Tagging Operation

In order to map the regex with appropriate user question, it is needed to assign a set of keywords with corresponding regex pattern. The answer template is created using the chunking method of NLP which defines a proper grammar for the Noun Phrase (NP) and the Verb Phrase (VP) collectively define a linguistic structure for a sentence. Whenever a particular set of keywords are matched with a regular expression pattern, it formulates an answer creating a noun phrase and a verb phrase by filling the gap with answer retrieved from the ontology.



This method is efficient for questions and answers which follows a common structure as mentioned above. Better results can be obtained for a limited dataset as it is easy to go through each and every question and find out similar patterns among them. It was developed using python with NLTK libraries. Better results can be obtained if the coverage of different user questions is higher. If the system does not identify the user query, it kindly asks to contact a human agent. Figure 3 depicts how the system generates ontology-based answers to its end users.

# Digital Assistant Platform for Customer Service

Team Esperanza

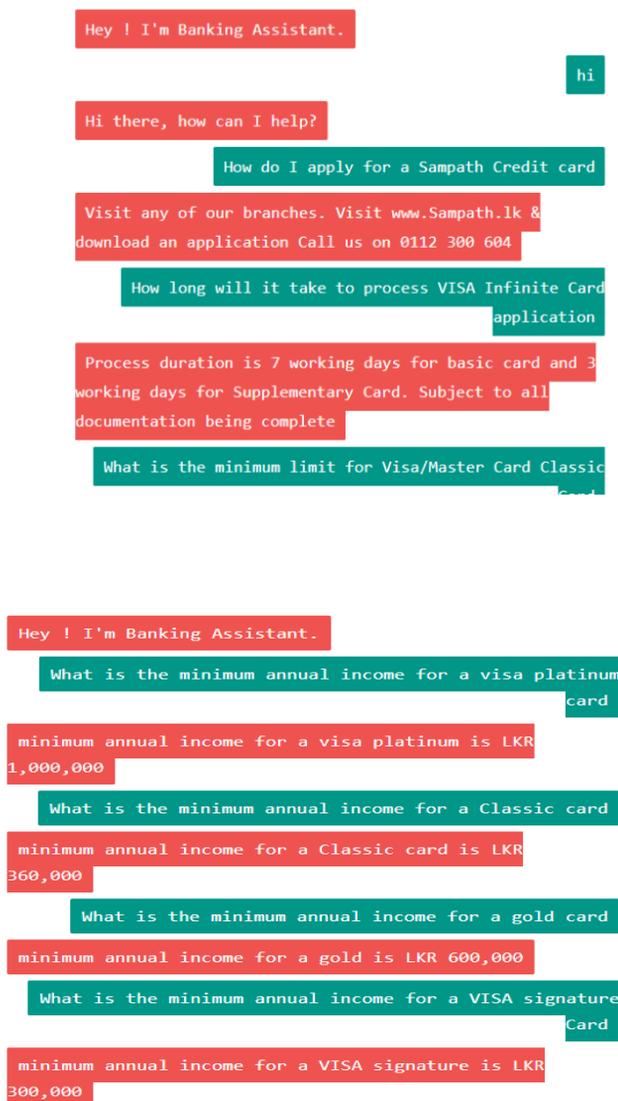
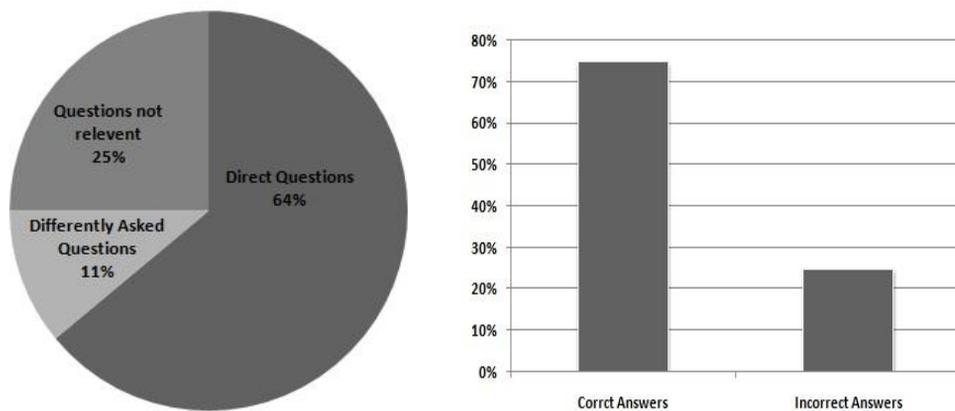


Figure 7: Chat Interfaces of the system

## Results and Discussions

The dataset contains 150 questions and corresponding answers. For the evaluation purpose, we have used 20 different types of questions which contains different ways of defining the grammar for the expected answer. In other words, each question has a different linguistic structure defined for the answer template. When the question pattern changes, so do the answer pattern. Furthermore, if we cover the number of different types of what and how question templates, then their corresponding answer templates changes as well. There are different types of

questions related to banking products such as credit cards, corporate card, web card and other types of cards included as the test data. This allows us to see how accurately the system can provide answers to given user questions. Test data contains direct questions which the system is able to provide answers. In fact, it consists of question which the system can't provide answers. Those are the questions out of the data set. On the other hand, test data contains similar questions which can be asked in different ways. Such questions also provided with an answer. All three kinds of questions are fed to the system and see the output against the real dataset. It's visible that, if we enter the question as mentioned in the dataset, most of the time it provides the accurate answer. 9 out of 10 questions were able to provide an exact answer. Following figure 3 shows the tested questions and expected the level of accuracy.



*Figure 8: Tested Questions and expected level of accuracy*

Questions out of the dataset are provided with the response as to contact a human agent. Similar question identification and unknown question result have a different level of accuracy. Out of 20 test data fed into the digital assistant, it could achieve 75% of overall accuracy level for the proposed answer generation system. It was visible that lengthy answers relatively process slowly than short answers. Yet, the response time is quite good as it does not interrupt the conversation.

## **Conclusions and Recommendations**

This template-based answer generation approach is a subsystem of a digital assistant designed to handle Frequently Asked Questions (FAQs) for a banking domain. This research approach addresses the issue of a limited dataset using a template based method while trying to combine several features of existing methods in order to generate an answer for banking domain. The research was used regular expression pattern creation and define an appropriate linguistic structure to formulate an answer for the given query. A new question can be added to the

system manually by checking whether it matches with an existing regex or not. If it matches, answer generation can be done by defining a linguistic structure while mapping with generic keywords. It would be easy to feed new questions if the question is defined using a format defined in the dataset. So that it enables to reduce the time-consuming process of pattern creation and defining grammar.

## **References**

Kulkarni, C.S., Bhavsar, A.U., Pingale, S.R., Kumbhar S.K. (2017) Bank chat bot – An Intelligent Assistant System Using NLP and Machine Learning | volume: 04|Issue: 05 | May -2017

Reiter, E and Robert, D., (1997) Building applied natural language generation systems. *Natural Language Engineering*, 3(1):57.87.

Ehud R., (1995) NLG vs. Templates. In Proc. 5th European Workshop on Natural Language Generation, pages 95.105, Netherlands.

Selj V., and Cox A., (2004) Question Answering using Regular Expression Rewriting, Dal TREC