EagleBot: A Chatbot Based Multi-Tier Question Answering System Using Deep Learning

Muhammad Rana and Mehdi Allahyari
Department of Computer Science, Georgia Southern University
Email: {mr07520, mallahyari}@georgiasouthern.edu

Abstract—**In this paper, we propose a chatbot based Question Answering (QA) approach to tackle questions asked in the university domain. For faster retrieval of answers to the questions posed, we used a three-tier system architecture and significantly improved the performance of the system by adding some linguistic and domain knowledge with the BERT model. User satisfaction based on our system exhibit impressive results. We designed this system in such a flexible manner that it can be replicated easily into other domains with careful execution. Another contribution from this paper is the creation of a new question-answering data set for the university domain.**

Index Terms—**QA System, BERT, BiLSTM, Sentence Embedding, Chatbot**

I. Introduction

One of the big challenges at every university or college is how to answer the vast amount of students' questions in a fast and efficient manner. Students usually check the university website, send an email, make phone calls or meet in person with the appropriate staff to get their information. Although these methods work, they are limited in certain ways: (a) since the number of university employees are restricted; (b) university staff are not available around the clock and more importantly (c) students mostly require to carry out extra work such as navigating the university website to extract the desired information from there. The same challenge exists in many other domains where we need constant human assistantship to tackle user queries. So, we propose a chatbot approach, which we have called Eaglebot, to answer the variety of questions that students usually ask on the university domain. For the case study, we use the Georgia Southern University (GSU) website.

Chatbots are increasingly gaining popularity in the university domain for different tasks. Jill Watson, the Georgia Tech teaching assistant chatbot [1] has demonstrated the strong viability of chatbots in the educational domain. Jill Watson has shown promise as an alternative to teachers in the near future. In 2016, a Boston based EdTech startup named Admithub gained huge success by launching a chatbot in Georgia State University for reducing Summer Melting [2] and helping high school students on their transition period to college [2].

Besides these two applications, we have also seen chatbots deployed in university domain for answering questions for a specific course by using previous years' chat discussion board as training data [3]. All these aforementioned chatbot systems work for a very specific domain and don't deal with the whole university domain. As of now, as to the best of our knowledge, there is no such integrated system for answering all types of questions asked in the university domain. Our experiment shows that it's possible to build a chatbot for handling the whole domain using our proposed architecture.

II. Approach

In our project, we build a three-tier system architecture to tackle three different groups of questions asked in a specific domain. We classified them as; I) QA on Structured Data, II) QA on FAQ Data and III) QA on Unstructured Passage Data. For example: retrieving the name of the courses taught by a specific teacher from a course table can be considered as type I, retrieving answers from a frequently asked question list can be considered as type II and retrieving answers from any document in the web within that domain can be considered as type III. We use Dialogflow's [4] NLU engine to understand the user's query and identify the entities and intent of the question, which helps us to select the route for finding the answer. For answering from type II (FAQ module), we use both TF-IDF and Sentence Embedding based model Infersent [5] and compare the retrieval accuracy and runtime. Finally, for answering type III queries, we use the BERT model and compare the performance with our baseline method BiLSTM. Additionally, we added some semantic changes to our data set to further improve the system.

A. Type I: Answering Question from Structured Tabular Data

Dialogflow is a conversational agent building platform from Google. It is a web-based platform that can be accessed from any web browser [4]. We use Dialogflow to capture the intent of a user query, the entity (the most important pieces of tokens in a text) within the query and the context of a query during multi-turn conversation.
Question: Is **Dr. X** teaching **CSE 101?**
Intent: "Course and Teacher Related Search"
Entity: "teacher_name": "Dr. X", "course_id": "CSE 101"

For example, from the above question our system maps the question as a type I question with the intent type of *"Course and Teacher Related Search"* and extracts two entity *"teacher_name"* and *"course_id"*. Then our Flask powered backend converts this question to a MongoDB search query like below to extract the desired result from the database,

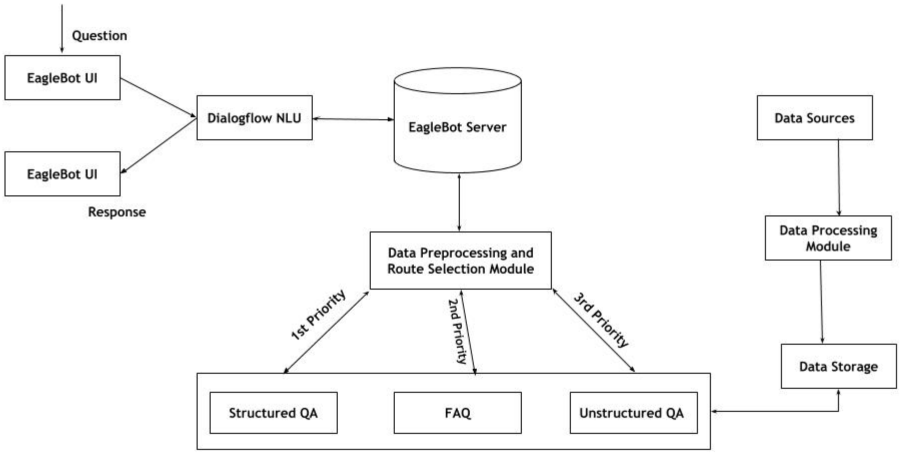

Fig. 1: High Level Architecture of EagleBot

where we have pre-stored many frequently-searched tables.

```
collection.find(”Title”: ”$regex”: ”.*” + course_id + ”.*”, ”teacher_name”: ”$regex”: ”.*” + teacher_name + ”.*”,”Title”: 1, ” teacher_name ”: 1, ”course_name”: 1, ”_id”: 0)
```

Finally, from the returned query result we create a rich response message to show the result to the user. And, if the question doesn't match with Type I, it automatically proceeds to Type II and III.

B. Type II: Answering Question from EagleBot FAQ KB

If the question doesn't match with Type I, our system pursues the answer from our Frequently Asked Question Knowledge Base (FAQ KB). We use Dialogflow's Default Fallback Intent to select this route. Here, in this module, our system tries to extract the answer using TFIDF with cosine similarity and Infersent model.

*1) **TFIDF & Infersent**:* Term frequency-inverse document frequency (tf-idf) gives the idea of how important a word is to a document [6]. We use tf-idf and cosine similarity to find out the closest question with an answer from our FAQ KB. Besides, tf-idf we also experiment with Infersent model, published by Facebook research [5]. Inferesent is a sentence embeddings method that provides semantic representations for English sentences. It is trained on Natural Language Inference (NLI) data and generalized for many NLP tasks. We use it for encoding our FAQ KB and to answer type II questions.

C. Type III: Answering Question from Unstructured Passages

If EagleBot doesn't find a Type II answer with a certain confidence level, it indicates to the system that it needs to dive into the final module. In this module, the workflow can be divided into Two main parts. *1) Candidate Document Retrieval* and *2) Answer Span Retrieval*.

*1) **Candidate Document Retrieval**:* For candidate document retrieval portion we use Elasticsearch search engine. We pre-index the texts extracted from all the 28,000+ webpages of GSU with their links. When a user's query comes to this module, the system fetches the top n=5 to 10 documents as the most probable candidate documents. Next, these candidates are appended together and finally fed along with the user query into the deep learning models for predicting the most probable answer span.

*2) **Answer Span Retrieval**:* For answer span prediction we use the BiLSTM model described in Reading Wikipedia to Answer Open-Domain Questions (Danqi Chen et al.) [7] as our baseline model. But, our system gives better performance by using BERT [8]. After some fine-tuning using BERT and some linguistic updates, our system gives us the best result.

BiLSTM: Our baseline BiLSTM model for machine reading is inspired by the recent success of DrQA, where their document reader model showed impressive performance on machine comprehension of text (Identifying answer span of a question from a list of selected documents). Given a question q consisting of l tokens $q1, \ldots, ql$ and a document or a small set of documents of n paragraphs where a single paragraph p consists of m tokens $p1, \ldots, pm$, the system uses the BiLSTM model to predict the span of tokens that is most likely the correct answer [7]. First, the model uses the paragraph to construct a feature vector using word embeddings and few more feature extraction techniques i.e. exact match, token features, aligned question embedding. The system then feeds the feature vector constructed from the paragraph to a multilayer BiLSTM model to create the paragraph encoding. The same technique is applied for constructing the question encoding. Finally, the system trains two different classifiers using the paragraph encoding

Question	Context	EagleBot Response
Who teaches csci 1302?	Structured Tabular Data (Type I)	The gym is open today, Friday (4/26/2019) from 6am to 9pm
Is gym open today?	Structured Tabular Data (Type I)	Dr. Mehdi csci 1302 a TR (2:00p-3:15p)
What is the deadline for i20 doc request?	From Eagle FAQ KB (Type II)	Deadlines for I-20 Requests: Students beginning in Summer (May): April 15 Students beginning in Fall (August): July 1 Most Probable URL: https://academics.georgiasouthern.edu/international/isss/prospective-students/applying-to-georgia-southern/
Where to mail my visa documents?	Where to Mail your Documents: ***You can mail all I-20 (F-1 Visa) documents to Ms. Tanya Brakhage (Statesboro) or Ms. Sara Nobles (Armstrong)***: You can contact our office at international@georgiasouthern.edu (type III)	You can mail all I-20 (F-1 Visa) documents to Ms. Tanya Brakhage (Statesboro) or Ms. Sara Nobles (Armstrong) Confidence: 89% Most Probable URL: https://academics.georgiasouthern.edu/international/isss/prospective-students/applying-to-georgia-southern/
Where to contact for scholarship related issues?	Apply for admission, apply for scholarships, and get to know your admissions counselor. ***Questions related scholarships please email to scholarships@georgiasouthern.edu. Questions related scholarships please contact 912.478.5391***	From FAQ KB: Questions related scholarships please email to scholarships@georgiasouthern.edu Confidence: 90 From BERT: ***Email gradadmissions@georgiasouthern.edu (Wrong Answer)*** Most Probable URL: https://admissions.georgiasouthern.edu/scholarships/

TABLE I: Sample questions and answers generated from EagleBot with the context and question type

and question encoding to predict two ends of the answer span.

BERT: BERT, which stands for Bidirectional Encoder Representation from Transformers is a method of pre-trainng language representations. This is the state of the art for pre-training contextual language representations. The key modules of BERT are two pre-training tasks. The first task Masked LM (MLM) aimed to break was the limitation of the traditional unidirectional models and leverage the power of bidirectional model. They mask out 15% of the words in the input and run the entire sequence through a deep bidirectional transformer encoder and then predict the masked words. The second task tries to learn the relationships between sentences. Given two sentences A and B, the model is trained to learn, if B is the actual next sentence that comes after A, or just a random sentence from the corpus.

BERT Fine-Tuning: Based on the original implementation of the multi-layer bidirectional transformer, BERT provides a few different model sizes. Considering our computational capacity, we choose BERT-Base, Uncased model.

- BERT-Base Uncased: L=12, H=768, A=12, Total Parameters=110M L: the number of layers (i.e., Transformer blocks); H: the hidden size; A: the number of self-attention heads.

We fine-tuned the BERT model using SQuAD 1.1(Stanford Question Answering) [9] dataset. We further fine-tuned the saved BERT model using our custom SQuAD like university domain Question Answering dataset to feed more specific domain knowledge into the BERT model.

III. Experiments

Our system requires three types of data. To test the whole system, we use Georgia Southern University(GSU) website data as our main data source. For supplying structured tabular data to answer Type I questions we extracted few frequent-searched table data using BeautifulSoup and stored those on MongoDB. To construct the FAQ KB for answering Type II questions, we extracted all the FAQ pages from GSU websites and stored them in memory. And, finally, for answering Type III questions we extracted all the webpages and indexed them into Elasticsearch. For training the BiLSTM model and fine-tuning the BERT model we used SQuAD 1.1 dataset, which is a reading comprehension dataset consisting of 100,000+ questions posed by crowdworkers on a set of Wikipedia articles. [9]

From the 28,000+ webpages of GSU, we select 50 webpages and generate 300 questions to ask EagleBot. We test the system with 20 volunteers and log all the conversation data with the testers. After each of the answers generated from EagleBot, we also capture user satisfaction level on that answer using three categories. *1. High, 2. Medium and 3. Low*. In total we have about 3,000 user conversation with EagleBot.

For creating the type I dataset, we analyzed Georgia Southern's historical search data. From the historical search data, we chose five topics and from those five topics we created about 30 intent types.

For the fine-tuning purpose, we use one NVIDIA GeForce GTX 1080 Ti. With larger batch sizes we exhaust the GPU's resources. So, we switch from batch_size= 32 to batch_size= 2 to reduce memory occupancy.

We also fine-tuned the BERT model to learn some domain-specific knowledge. One such example is; on many cases in university domain we encounter contact info pages, where the contacts are saved in the below format.

"Dr. ABC, Professor & Chair
IT 1000A, Statesboro Campus
(912) 478-XXXX
abc@georgiasouthern.edu"

We fine-tuned the model with many of these contact pages to train the model to learn how to retrieve the contact info from text like the one above.

IV. Results

We analyzed the results from the 3,000-conversation chat collected from EagleBot. Table I shows a few sample conversations generated from EagleBot including a wrong prediction from BERT (the last example).

QA Retrieval Technique	**Execution Time**
From Structured Tabular Data	<= 1.0 sec
From FAQ KB	<= 1.5 sec
From FAQ KB + Using BiLSTM	<= 2.0 sec
From FAQ KB + Using BERT	<= 4.8 sec
From FAQ KB + Using BERT + Using BiLSTM	<= 5.8 sec

TABLE II: Answer Retrieval Time Using Different Techniques

User Satisfaction Level Calculated from EagleBot	**Percentages (%)**
High	52
Medium	25
Low	23

TABLE III: User satisfaction level on EagleBot response

Table II validates the use of three-tier architecture. It shows that if the answers match with any of type I or type II questions, the retrieval time is very low.

We log all the conversations including user feedback on that answer. The table shows in 77% of the cases the users report their satisfaction level as medium and above.

Retrieval Technique	**Accuracy (avg. %)**
Structured Tabular Data Retrieval	78
From FAQ KB	87
From Passages Using fine-tuned BERT	65
From Passages Using BiLSTM	48

TABLE IV: Accuracy From EagleBot

Table IV shows a few interesting results. It shows that if the question is of type I or type II, there's a high chance that EagleBot will retrieve the correct answer. With proper fine-tuning and some linguistics changes, even BERT-base model performs better than our baseline BiLSTM model.

V. Ongoing Works

We fine-tuned the BERT model with SQuAD 1.1 dataset and our own custom dataset created from the GSU website. There is a lot of room for improvement by fine-tuning heterogeneous data sources. For example, we are now experimenting with other university websites, SQuAD 2.0, CuratedTREC and others.

There are several open-source BERT models available in Github. BERT-large models significantly outperform our used BERT-base model. So, we are planning on using an ensemble technique, assigning more weights to predictions from BERT-large than BERT-base and combining the final probability. We also plan on considering BiLSTMs' results for the final probability. In other words, if we find an answer on both BiLSTM and BERT, we can automatically bump up the probability of that answer being correct. [10]

BERT provides several answers to a single question with different probabilities. We experienced many cases, where the later answers were more detailed and precise than BERT's predicted best answer. Displaying the most desired answer for a specific user from all the candidate answers is an interesting problem yet to be solved.

VI. Conclusion

In this paper, we demonstrated our initial findings using our purposed three-tier question answering system architecture for faster answer retrieval. Our result showed the validity of the posed architecture. We also contributed a new dataset in the university domain for future researchers working on this issue. Our fine-tuned BERT model outperforms the baseline BiLSTM and BERT-base model by a good margin. With careful execution, this system can be replicated into any domain which needs a smart auto assistant system.

References

[1] A. K. Goel and L. Polepeddi, "Jill watson: A virtual teaching assistant for online education," Georgia Institute of Technology, Tech. Rep., 2016.

[2] L. C. Page and H. Gehlbach, "How an artificially intelligent virtual assistant helps students navigate the road to college," *AERA Open*, vol. 3, no. 4, p. 2332858417749220, 2017.

[3] D. Feng, E. Shaw, J. Kim, and E. Hovy, "An intelligent discussion-bot for answering student queries in threaded discussions," in *Proceedings of the 11th international conference on Intelligent user interfaces*. ACM, 2006, pp. 171–177.

[4] Wikipedia contributors, "Dialogflow — Wikipedia, the free encyclopedia," https://en.wikipedia.org/w/index.php?title=Dialogflowoldid=882038711, 2019, [Online; accessed 27-April-2019].

[5] A. Conneau, D. Kiela, H. Schwenk, L. Barrault, and A. Bordes, "Supervised learning of universal sentence representations from natural language inference data," *arXiv preprint arXiv:1705.02364*, 2017.

[6] J. Ramos *et al.*, "Using tf-idf to determine word relevance in document queries," in *Proceedings of the first instructional conference on machine learning*, vol. 242. Piscataway, NJ, 2003, pp. 133–142.

[7] D. Chen, A. Fisch, J. Weston, and A. Bordes, "Reading wikipedia to answer open-domain questions," *arXiv preprint arXiv:1704.00051*, 2017.

[8] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," *arXiv preprint arXiv:1810.04805*, 2018.

[9] P. Rajpurkar, J. Zhang, K. Lopyrev, and P. Liang, "Squad: 100,000+ questions for machine comprehension of text," *arXiv preprint arXiv:1606.05250*, 2016.

[10] W. Zhou and Jiang, "Ensemble bert with data augmentation and linguistic knowledge on squad 2.0." *http://web.stanford.edu/class/cs224n/reports/default/15845024.pdf*, 2018.