



RAILBOT: A Railway (IRCTC) Chatbot

Sahaya Sakila.V¹, Akshat Shrivastava², Md Arman Ansari³, Laxmi Kanta Kar⁴, Mukul Kumar⁵

Assistant Professor¹, Student^{2,3,4,5}

Department of CSE

SRM IST, Ramapuram, Chennai, India

Abstract:

Chatbots are in high demand in today's industry world. It presents a new way for individuals to interact with computer systems. Traditionally, to get a question answered by a software program involved using a search engine or filling out a form. It allows a user to simply ask questions in the same manner that they would address a human. The technology at the core of the rise of the chatbot is natural language processing ("NLP"). It can be improved by integrating it into the organization's enterprise software, allowing more personal questions to be answered. The platforms where chatbots are deployed include Facebook Messenger, Skype, and Slack, Telegram. This paper successfully explained and implemented a chatbot which can be used to get some basic information related to Indian Railway such as PNR, train status and Seat Availability, etc. It saves the time of the user as they can directly fetch the information while conversing with the Chatbot using text or voice in the above mentioned deployed methods. This new proposed system will help the common passengers and visually impaired people.

Keyword: Chatbot, NLP, Facebook Messenger, Computer software, Search engine.

I. INTRODUCTION

Indian Railways is the fourth largest railway network in the world, transporting millions of passengers across the length and breadth of the country. Equally mind boggling is the traffic on the web portal for booking train tickets, fetching PNR, live train status, Indian Railway Catering and Tourism Corporation (IRCTC). An average of 13 lakh tickets are booked per day. Over 1 million people visit the site to check their ticket status. While there are certain other travel aggregator sites that allow users to book train tickets, IRCTC is still the preferred method. Despite its shabby looks, it gets the work done. Besides booking ticket, some of the most commonly performed actions by travellers are Check the Passenger Name Record (PNR) status also known as ticket status, check if a waitlisted ticket has been confirmed, check running status of train, In India trains run late, passengers just want to know by how much and check the train schedule, which waste time of the passengers. Even to get the above-mentioned information from an application using platforms like Android and iOS, the user has to download an application which consumes internet bandwidth as well as waste time. Thus, the passengers can save their time and internet data by using the Chatbot by just messaging using app such as Facebook Messenger, Telegram, Google Assistant, Skype which is generally pre-installed.

II. LITERATURE SURVEY

In this section we discuss the different methodologies review or literature review and motivation outcomes from it.

A. Android Based Educational Chatbot for Visually Impaired People

The purpose of the mentioned android application is to provide educational based Chatbot for visually impaired people. It will give an answer to the educational based queries asked by the

visually impaired people. They can easily launch the application with the help of google voice search. Once the application is open, it will give a voice instruction to use an application. Output will be provided in voice form as well as in text form. So normal people can also use this application.

B. Artificial Intelligence Applied to Challenges in the Fields of Operations and Customer Support

Traditionally, Mastercard has been seen as simply a payment processor, and while that remains a large part of the business Mastercard also offers hundreds of products to financial institutions and others in its ecosystem around the world. Consequently, there is a need for large operations and customer support teams. Their systems accumulate data and, in this paper, they present four different examples of applying Artificial Intelligence to this data to solve business problems.

C. Climebot: An Argumentative Agent For Climate Change

While climate experts have agreed that global warming is real, this consensus has not reached all the society levels. Our aim is to develop a conversational agent able to explain issues related to global warming. The developed chatbot relies on textual entailment to identify the best answer for a statement conveyed by a human agent. To enhance the conversational capabilities, they employed the technical instrumentation provided by the API.AI framework. To exploit domain knowledge, the agent uses climate change ontologies converted into an adequate format for the API.AI model. Hence, we developed a Climebot, which is an argumentative agent for climate change based on ontologies and textual entailment.

D. Executable Semantics of Recursively Nestable Dialog Flow Specifications for Web Application

Information systems for the support of complex business process are often equipped with web-based front-ends to allow

convenient user access. To produce executable specifications of the users' interactions with such web-based applications, we use a visual language that enables developers to model their complex dialog structures. In this paper, we introduce the formal semantics of the core constructs of this. Dialog Flow Notation: its syntax in terms of invariants about the permitted elements and their relations and show how any words of the language (i.e. any syntactically correct dialog box specifications) can be mapped to a deterministic pushdown automaton whose behavior dense the notation's semantics. This gives us and other tool developers a formal basis for the design and implementation of tools and frameworks that mirror the precise meaning of all DFN constructs.

E.. Study on the Public Information Service Systems of Railway Industry

As we enter the 21st century, we experience one of the most important changes in our lives—information technology has been the important resource in the modern scientific and technology competition. So, the tide of Railway Informatization is increasingly rising. Railway information service system is the strategic resource element in the course of informatization. But railway information service in our country is not perfect yet, especially the information Service for the public. This paper first focus on the current situation of public information service in China, then proposed an information service system for railway industry to enhance the efficiency and transparency of information transmission. and then give some advices on how to operate the information service system to facilitate the information transmission in railway industry.

F. Home Automation using IoT and a Chatbot using Natural Language Processing

Home automation - controlling the fans, lights and other electrical appliances in a house using Internet of things is widely preferred in recent days. In this paper, we propose web application using which the fans, lights and other electrical appliances can be controlled over the Internet. The important features of the web application is that firstly, we have a chatbot algorithm such that the user can text information to control the functioning of the electrical appliances at home. The messages sent using the chatbot is processed using Natural Language processing techniques. Secondly, any device connected to the local area network of the house can control the devices and other appliances in the house. Thirdly, the web application used to enable home automation also has a security feature that enables certain users to access the application. And finally, it also has a functionality of sending an email alert when intruder is detected using motion sensors.

III. REQUIREMENT

The requirement for this project include some software specifications which are: -

1. Heroku server- This is an open server which can be used to create the link between the chatbot and railway API.
2. Heroku CLI & Git – This is a command line interface for working and deploying the local repository to the Heroku server which take the help of Git or Github desktop.
3. Python IDE- As the backend is programmed in Python, therefore this IDE helps to script and debug the test.py

file for fetching the JSON from Railway API and getting it back to the chatbot using webhook.

4. Dialog Flow (API.ai) - This online service allows us to design an Artificial Intelligence Chatbot through defining the intent/activity and entity and fetching the information from Heroku using fulfillment/webhook.

5. Facebook Developer Page or token- To enable this chatbot to be used by the passengers, the administrator has to get access to Facebook Developer platform or any other required messenger token for deploying the chatbot into their application.

6. Facebook Messenger/Telegram/Slack-Enables the user to message the input through keyboard and mic in the form of voice and text and get the output in the same two formats.

IV. SYSTEM ARCHITECTURE

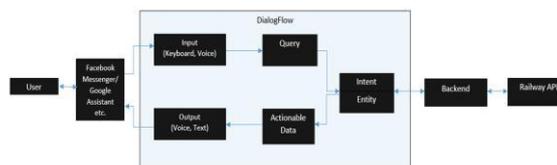


Figure.1. System Architecture

1. Facebook Messenger/Slack/Telegram/Google Assistant Enable the user to give the input through keyboard and mic in the form of voice and text and get the output in the same two formats.

1.1) Input- Example- To get the PNR details the user can give the input as “PNR 85761234”

1.2) Output- Example- The above query given in the input module will be responded as “PNR 85761234, NAME-AKSHAT SHRIVASTAV, DOJ-1/1/2018, TRAIN NO. -13552, CUR STAT- RLWL/051”.

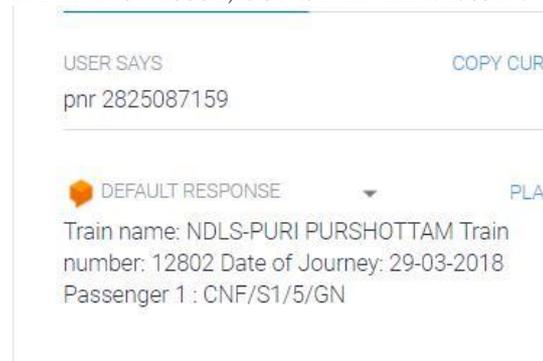


Figure.2. User Input and Output

2. DialogFlow(API.ai) -

This online service allows us to design a rich Artificial Intelligence Chatbot through defining the intent/activity and entity.

2.1) Intent- Defines the activity which is performed by identifying the query given by the user. Intent is the described activity defined in the dialog flow. It can fetch the information from webhook using fulfilments.

2.2) Entity- The database of the required real-world object that is provided by the developers.

2.3) Fulfilment- The intent which has a particular action name can use the fulfilment for connecting the webhook with the Heroku server.

3. Backend-

The Backend that is used to connect and fetch the information between the Railway API and dialog flow.

3.1) Heroku server- This is an open server which is used to create an application to store the backend database and creating the link between the chatbot and railway API.

3.2) Heroku CLI & Git – This is a command line interface for deploying and debugging the test.py to the Heroku server which take the help of Git or Github desktop. This server enables us to use Heroku logs and deploying methods.

3.3) Python- The backend is programmed using Python version 2.7 and pip. This is used to fetch the JSON object from the Railway API and responding to the Dialogflow using webhook, flask.

4. Railway API -

Responsible for providing the information as JSON object like PNR status, Train between the stations, running status, etc. The module works with a unique API key provided by API providers such as erail.in and railwayapi.com.

V.PROPOSED SYSTEM

The chatbot will be fast and concise. It is dummy friendly. It gives confirmation of the user query eliminating misunderstanding. The conversational interface of a chatbot is suited ideally for the purpose of fetching the information of the Indian Railways and there is a lot of scope for improving and fine tuning the services provided. It saves device memory space, time and internet data usage. The chatbot will also enable the user to get the response of Small talks.

VI. IMPLEMENTATION

1. First, create an account on Dialogflow(api.ai) and Heroku

2. Second, using Dialogflow create an app and follow the basic instructions and choose the version as APIV1.

2.1 Creating entities – To create entities go to the entity tab in the side Navigation Bar and is specify the variables which has to be used in the intents such as pnr_no, train_no, sys.date.

2.2 Creating Intents- Specify the intent name and describe the expressions or templates that a user can ask. Specify the parameter using the entities and create an action name, which has to be the same defined in the Python file.

2.3 Fulfilment – Refer section 3-4 of this implementation column to create the backend. For any given intent the webhook can be enabled which fetched the data from the Heroku server using the same parameter and action name described in the intent.

2.4 Required – The small talks can be enabled using the prebuilt feature of Dialog flow. The webhook has to be connected with the Heroku server using the HTTP and token.

3. Install Git and Heroku CLI using the help of documentation available in the Heroku and Git website (install other required

packages like pip and files like proc file, etc). <https://devcenter.heroku.com/categories/deployment>

4. After configuring Git and Heroku CLI login into Heroku account. Create an app and specify the default language to python.

4.1 Using python create a webhook to connect with the Dialogflow , fetch the information from Railway API using HTTP and python request, store the JSON object and return the response message [https:// devcenter. heroku. com/ categories/ pytho n.](https://devcenter.heroku.com/categories/python)

```
1 # -*- coding: utf-8 -*-
2 from flask import Flask
3 from flask import request
4 from flask import make_response
5
6 app = Flask(__name__)
7
8 railway_key = 'c3d22b27'
9
10 @app.route('/webhook_methods/POST')
11 def webhook():
12     req = request.get_json()
13     print req
14     res = process_request(req)
15     res = json.dumps(res, indent=4)
16     r = make_response(res)
17     r.headers['Content-Type'] = 'application/json'
18     return r
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
580
581
582
583
584
585
586
587
588
589
590
591
592
593
594
595
596
597
598
599
600
601
602
603
604
605
606
607
608
609
610
611
612
613
614
615
616
617
618
619
620
621
622
623
624
625
626
627
628
629
630
631
632
633
634
635
636
637
638
639
640
641
642
643
644
645
646
647
648
649
650
651
652
653
654
655
656
657
658
659
660
661
662
663
664
665
666
667
668
669
670
671
672
673
674
675
676
677
678
679
680
681
682
683
684
685
686
687
688
689
690
691
692
693
694
695
696
697
698
699
700
701
702
703
704
705
706
707
708
709
710
711
712
713
714
715
716
717
718
719
720
721
722
723
724
725
726
727
728
729
730
731
732
733
734
735
736
737
738
739
740
741
742
743
744
745
746
747
748
749
750
751
752
753
754
755
756
757
758
759
760
761
762
763
764
765
766
767
768
769
770
771
772
773
774
775
776
777
778
779
780
781
782
783
784
785
786
787
788
789
790
791
792
793
794
795
796
797
798
799
800
801
802
803
804
805
806
807
808
809
810
811
812
813
814
815
816
817
818
819
820
821
822
823
824
825
826
827
828
829
830
831
832
833
834
835
836
837
838
839
840
841
842
843
844
845
846
847
848
849
850
851
852
853
854
855
856
857
858
859
860
861
862
863
864
865
866
867
868
869
870
871
872
873
874
875
876
877
878
879
880
881
882
883
884
885
886
887
888
889
890
891
892
893
894
895
896
897
898
899
900
901
902
903
904
905
906
907
908
909
910
911
912
913
914
915
916
917
918
919
920
921
922
923
924
925
926
927
928
929
930
931
932
933
934
935
936
937
938
939
940
941
942
943
944
945
946
947
948
949
950
951
952
953
954
955
956
957
958
959
960
961
962
963
964
965
966
967
968
969
970
971
972
973
974
975
976
977
978
979
980
981
982
983
984
985
986
987
988
989
990
991
992
993
994
995
996
997
998
999
1000
```

Figure.3. Python Code for Webhook Fulfilment

5. Integration- For integrating to Facebook Messenger or Telegram or any other applications (refer the Dialogflow documentation), refer the following steps. [https:// dialogflow. com/docs /integrations/](https://dialogflow.com/docs/integrations/)

5.1 Create a Facebook account and developer account . After login in create an app and integrate it to Dialogflow using a generated token. In the webhook section enable message and message postback.

5.2 Telegram – Create a chatbot user using BotFather and get the token using which integrate it to the Dialogflow.

VII. TESTING

The below screenshots show the testing and bugs of the chatbot.



Figure.4. Small Talk with the Chatbot on Telegram

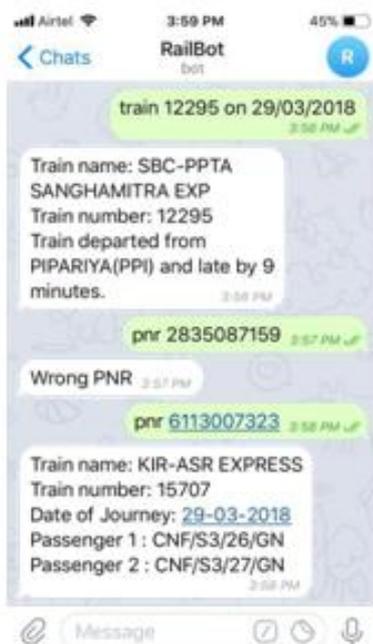


Figure.5. Getting PNR and Train Status with the Chatbot on Telegram

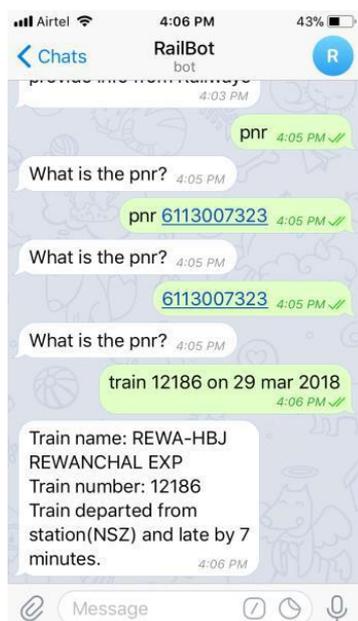


Figure.6. Some bugs in the Chatbot on Telegram

VII. CONCLUSION

This paper successfully explains and implemented chatbot which can be used to get the required information from Indian Railway. In the future, we will fix the bugs and include the railway complaints and booking information. The key to effective chatbot use is to make sure they enhance your customer’s experience. This chatbot will benefit the passengers by saving time, memory space and internet usage.

ACKNOWLEDGEMENT

We want to thank our guide Ms. Sahaya Sakila.V, A.P, SRM Institute of Science and Technology, Ramapuram Campus,

Chennai, India for his continuous support and his encouraging and fruitful advises in accomplishing this task.

VIII. REFERENCES

- [1]. Naveen Kumar M, Linga Chandar P C, Venkatesh Prasad A, Sumangali K-“Android Based Educational Chatbot for Visually Impaired People”, Issue, 2016
- [2]. Ravi Santosh Arvapally, Hasan Hicsasmaz, Wally Lo Faro-“Artificial Intelligence Applied to Challenges in the Fields of Operations and Customer Support”, Issue, 2017.
- [3]. Daniel Toniuc and Adrian Groz- “Climebot: an argumentative agent for climate change”, Issue, 2017
- [4]. Sören Blom, Matthias Book, Volker Gruhn- “Executable Semantics of Recursively Nestable Dialog Flow Speciation for Web Applications”, Issue, 2008
- [5]. Guo Chunfang , Wu Hao, Sun Zhongyue Guan Zhongliang- “Study on the Public Information Service Systems of Railway Industry”, Issue, 2010
- [6]. Cyril Joe Baby, Faizan Ayyub Khan, Swathi J. N.- “Home Automation using IoT and a Chatbot using Natural Language Processing”, Issue , 2017