Automation of the Training of a Chatbot for a Customer Service

A comparison between the chatbot building platforms Amazon Lex, IBM Watson and Google Dialogflow

Jürgen Nützel, jn@4fo.de
CEO 4FriendsOnly.com AG
4FriendsOnly.com Internet Technologies AG

- Spin-off von Fraunhofer IDMT & Tech. University Ilmenau
- CEO, lecturer and main owner
- E-Commerce experts (> 10 years)
- We focus also on
  - Mobile development
  - Cloud-computing (AWS since 2013)
  - Digital transformation
- Amazon AWS partner since 2017
- AWS user group lead
- Speaker associated partner
Why we came up with chat services?

A Customer from Salzburg asked for
Electro Wholesale, Gebrüder LIMMERT AG from Salzburg

... asked us for replacing the telephone support by an online chat.

We develop and run their B2B online shop on AWS: www.limmert.com
Setup for our Service Chat

Node.js Server uses MongoDB to store the chats

User Interface in Shop: Angular PWA (Progressive Web App)
Integration of a Chatbot

We still store the complete chat (including the bot’s answers)

Customer’s input is sent to the bot

The Service/Hotline is able to observe the bot’s answers
We were looking for a bot platform?

We found several cloud based platforms
# 6 Chatbot Building Platforms

<table>
<thead>
<tr>
<th>Platform</th>
<th>Owner</th>
<th>URL</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dialogflow</td>
<td>Google</td>
<td>dialogflow.com</td>
</tr>
<tr>
<td>wit.ai</td>
<td>Facebook</td>
<td>wit.ai</td>
</tr>
<tr>
<td>Language Understanding service (LUIS)</td>
<td>Microsoft</td>
<td><a href="http://www.luis.ai">www.luis.ai</a></td>
</tr>
<tr>
<td>Watson Assistant</td>
<td>IBM</td>
<td>cloud.ibm.com/docs/services/assistant</td>
</tr>
<tr>
<td>Amazon Lex</td>
<td>Amazon</td>
<td>aws.amazon.com/lex</td>
</tr>
<tr>
<td>SAP Conversational AI (Recast.ai)</td>
<td>SAP</td>
<td>cai.tools.sap</td>
</tr>
</tbody>
</table>
The German Speaker Project

A voice assistant platform »Made in Germany«

https://www.speaker.fraunhofer.de/
Start of implementation phase on 04/01/2020
4FriendsOnly.com AG is associated partner

Supported by:
Federal Ministry for Economic Affairs and Energy
on the basis of a decision by the German Bundestag

APPIS 2020, January
Dr. Jürgen Nützel, jn@4fo.de, 4FriendsOnly.com Internet Technologies AG
Amazon Lex

We found it in Amazon Web Services (AWS)
But it supports only English
**Amazon Lex sample bot**

**Intents**
A particular goal that the user wants to achieve

**Utterances**
Spoken or typed phrases that invoke your intent

**Slots**
Data the user must provide to fulfill the intent

**Prompts**
Questions that ask the user to input data

**Fulfillment**
The business logic required to fulfill the user’s intent

---

**BookHotel**

I’d like to book a hotel.

Sure, which city?

**New York City**

What date do you check in?

November 30th.

Are you sure you want to book the hotel in New York City?

Yes.

Thank you. The reservation went through successfully.
The intent `BookHotel` in the Amazon Web Services Console

- **Sample utterances**
  - Book a `{Nights}` night stay in `{Location}`
  - I want a make hotel reservations
  - Book a hotel

- **Lambda initialization and validation**
  - The business logic required to validate the user’s input

- **Slots**
  - Data the user must provide to fulfill the intent
  - Slot type

  1. `Location`
     - AMAZON.US_CITY
     - Slot type: Built-in
     - Question: What city will you be staying in?
  2. `CheckInDate`
     - AMAZON.DATE
     - Slot type: Built-in
     - Question: What day do you want to check in?
  3. `Nights`
     - AMAZON.NUMBER
     - Slot type: Built-in
     - Question: How many nights will you be staying?
  4. `RoomType`
     - RoomTypeValues
     - Slot type: 1
     - Question: What type of room would you like, queue

- **Confirmation prompt**
  - Questions that ask the user to input data
  - Confirmation prompt
Test dialog in the AWS console

This phrase was accepted but was not in the list of sample utterances.

The fulfillment could be sent to a Lambda function.
IBM Watson Assistant

It supports several language.

Not so easy to start with. But we like it.
User examples are assigned to intents

Watson Assistant extracts entities from the user's phrases

Like in Lex user examples may be annotated

The user entity @City:
Google Dialogflow

It is easy to start with.
We made a pizza order bot (in German)
Specify a system or a developer entity

Phrases you can expect from users, that will trigger the intent.

- In Dialogflow training phrases are assigned to intents.
- Dialogflow’s intents have parameters not slots
- All platforms use the same principles
We connected the Dialogflow pizza order bot with our chat system (server + UI)
We need to connect with external systems.

We used Lambda functions in Amazon Lex
AWS manages serverless code hooks. These node.js hooks query external data sources.
Node.js hook code in the AWS Lambda console

```javascript
import { get } from 'lodash';
import { SNS } from 'aws-sdk';
import renderHTML from 'react-html-parser';
import React from 'react';
import { render } from 'react-dom';

const lambda = new SNS();

function getDocumentationHook(event) {

  const message = {
    content: 'Do you need documentation for the "' + sessionAttributes.order + '" product you have ordered earlier?'
  };

  sessionAttributes.context = false;
  return response;

  if (slots.components === 'yes') {
    sessionAttributes.context = true;
    slots.number = 1;
    if (obj[i+1].attachments !== undefined) {
      let j = 0;
      let attachments = '\n';
      while (obj[i+1].attachments[j] !== undefined) {
        attachments += (j + 1) + " + obj[i+1].attachments[j].attachment + \n";
        j++;
      }
      //console.log(attachments);
      sessionAttributes.attachments = String(attachments);
      } else {
    sessionAttributes.attachments = "Sorry but this product does not have any documentation.";
  }

  if (sessionAttributes.order !== undefined) {
    let response = {
      sessionAttributes: sessionAttributes,
      dialogAction: {
    ```
A test dialog using the Lex UI

hey
Well, good day to you!(Fulfilled)

buy products
Which type of components would you like to order?(ElicitSlot)

kabel
How many of those do you need?(ElicitSlot)

2
What day do you want the kabel to be picked up?(ElicitSlot)

tomorrow
At what time do you want the kabel to be picked up?(ElicitSlot)

1 pm
Okay, your kabel will be ready for pickup by 13:00 on 2019-11-14 and will cost 10$. Does this sound okay?:(ConfirmIntent)

yes
Okay, I have ordered your 2 kabel for 13:00 2019-11-14(Fulfilled)

cool
Hell yeah!(Fulfilled)

and i want documentation

Do you need documentation for the kabel YM-J 3X1.5 grau PVC-Mantelleitung which you have ordered earlier?(ElicitSlot)

yes
1. https://media.immert.com/content/immert/HE_1101_02272797.pdf
2. https://media.immert.com/content/immert/HE_1102_02272797.pdf
(Fulfilled)
How the chat bot learns (semi-automatically)?

Some ideas only
Investigate manually the missed utterances in the monitoring tab

<table>
<thead>
<tr>
<th>Utterances</th>
<th>Count</th>
<th>Status</th>
<th>Last said date</th>
</tr>
</thead>
<tbody>
<tr>
<td>i need to receive component</td>
<td>2</td>
<td>Missed</td>
<td>December 19, 2019 at 2:58:51 PM UTC+1</td>
</tr>
<tr>
<td>Stupid bot</td>
<td>1</td>
<td>Missed</td>
<td>December 19, 2019 at 3:29:33 PM UTC+1</td>
</tr>
<tr>
<td>Thank you!</td>
<td>1</td>
<td>Missed</td>
<td>December 19, 2019 at 3:22:21 PM UTC+1</td>
</tr>
<tr>
<td>can i receive components?</td>
<td>1</td>
<td>Missed</td>
<td>December 19, 2019 at 3:29:03 PM UTC+1</td>
</tr>
<tr>
<td>i need to receive component</td>
<td>1</td>
<td>Missed</td>
<td>December 19, 2019 at 2:58:38 PM UTC+1</td>
</tr>
<tr>
<td>why don't you understand m</td>
<td>1</td>
<td>Missed</td>
<td>December 19, 2019 at 3:29:26 PM UTC+1</td>
</tr>
<tr>
<td>it would be great if i would h</td>
<td>1</td>
<td>Missed</td>
<td>December 19, 2019 at 3:27:35 PM UTC+1</td>
</tr>
<tr>
<td>can i order these things which</td>
<td>1</td>
<td>Missed</td>
<td>December 19, 2019 at 3:27:08 PM UTC+1</td>
</tr>
<tr>
<td>well give me components</td>
<td>1</td>
<td>Missed</td>
<td>December 19, 2019 at 3:28:31 PM UTC+1</td>
</tr>
<tr>
<td>what you are capable of</td>
<td>1</td>
<td>Missed</td>
<td>December 19, 2019 at 3:15:44 PM UTC+1</td>
</tr>
</tbody>
</table>
Lessons learned

• Several cloud bot building platforms are available
• All platforms have the same functional principle
• No AI knowledge is needed to work with
• The bots learn not automatically
• Most work is to be done to connect 3rd party systems

• Now we are able to offer our customers individual service bot solutions using IBM Watson Assistant, Google Dialogflow and Amazon Lex
Thank you

Dr. Jürgen Nützel
CEO
jn@4fo.de
www.4fo.de

Find me on LinkedIn