

Recruitment Chatbot

Hanusha Sabbavarapu (Matriculation Number:225018)

University of Trento

Via Sommarive, 9, 38123 Povo,Trento TN

hanusha.sabbavarapu@studenti.unitn.it

Abstract:

A chatbot, at its most basic, is computer software that mimics and interprets human interaction (spoken or typed), enabling users to converse with digital gadgets as if they were speaking to real people. Chatbots may be as basic as one-line programs that respond to straightforward questions, or they can be as complex as digital assistants that learn and develop over time to provide even more individualized service as they acquire and analyze more data. Chatbots increase operational effectiveness and save costs for organizations while providing convenience and extra services to both internal staff and external clients. They lessen the need for human engagement while enabling businesses to quickly address a wide range of client inquiries and difficulties. It also allows a firm to expand, customize, and be proactive all at once—a key differentiation. For instance, a company can only serve a certain number of customers at once while using just human labour. Human-powered firms are constrained in their capacity for proactive and individualized outreach since they must concentrate on standardized models in order to be cost-effective.

Here we have developed a Recruitment bot which will help HRs Speed up the recruiting process and also helps candidates to get their updates without staying in suspense. Recruitment chatbots help organizations and HR to ease the recruitment process and help the candidates to be updated without utilizing the time of HRs. Many candidates never hear back after one touch point as recruiting team also struggles to communicate and keep updated. It may increase the productivity of the whole process by 30-40%.

1. Introduction

A) Model Type:

The bot works in different modes such as task-based and question-answering modes. As it's a Recruitment chatbot, Candidates can get updates regarding their process and also other open positions in the company. And the bot also asks basic questions like the candidate's details and experience etc.

B) Domain:

As we have read in the above matter the bot is based on the Human resources Domain as it helps in Recruiting candidates and helping them get updates. The Bot is trained for IT sector recruitment processes. It can handle the basic greeting and the questions related to domain-like

- Can I know about open positions?
- I applied for a job and would like to know when I'll hear back.
- I haven't heard about my application.
- greetings, do you have jobs openings?

C) User Group:

The chatbot will be used by any firms which are indulged in recruiting, so can say it has a wide scope. But mainly the HR and Candidates who are willing to apply for new opportunities would be the potential end user of the system.

2. Conversation Design

a). The dialogues that the system can Handle:

It can handle Basic greetings, direct questions and the screening process. It can check the candidate's Position, and status and also gives info about the process.

- o Can I know about open positions?
- o I applied for a job and would like to know when I'll hear back.
- o I haven't heard about my application.
- o greetings, do you have jobs

b). Interaction Properties:

- i) **Initiatives:** It starts the conversations and also gives recommendations to the user regarding choices.
- ii) **Acknowledgement:** It repeats the name if I ask for status and if the process is time taking then ask to keep patience as well.
- iii) **Ground-taking:** It helps user by asking the relevant questions and giving proper scenarios to choose.
- iv) **Fall-back Responses:** As a fallback, it repeats the question or asks the user to rephrase the question. If it doesn't have an answer it says to try later till the info gets updated. Added **FallbackClassifier** in config files.
- vi) **Error Recovery:** It tries to redirect the user to help options.

3. Data Descriptions and Analysis

a) The dialogue data used to train model:

- i. **How you collected it:** I have manually entered the intents and then as I keep chatting with it generates the intents and then I retrain it.
- ii. **How you generated it:** Manually added the data intents and conversations and as I keep chatting with it generates the intents and then I retrain it.
- iii. **Complete Data Analysis on the size, vocabulary, #entities, #slots, #turns per dialogue:** As it trains on spacy which is a powerful for NLP tasks it handles the dialogue intents. But I have trained more than 5 instances for each converse variation.

b). The NLU data used to train the model:

Yaml was used as it's a unified and extendable way to manage all training data, including NLU data, stories and rules. Used the basic greets provided by RASA and added the utterance like name status to get the updates trained by the model using stories

c). The domain data:

Manually added the data and then kept chatting with it generating the intents and then I retrain it. Created the Main instance to save the Candidate's name status and positions manually. Referred to some websites to get a crisp of questions to be asked.

4. Conversation Model

- a) **Rasa Based system:** It is a Rasa Base system as it is an open-source chatbot framework based on machine learning and deep learning for training the intents and data.
- b) **The response policy:** The bot keeps asking the question if user has any issue or query and based on the query the chat revolves around the intents and trained NLU.
- c) **The fallback policy:** As a fallback, it repeats the question or asks user to rephrase the question. If it doesn't have an answer it says to try later till the info gets updated. Added **FallbackClassifier** in config files.
- d) **The NLU model, tokenizer, featurizer, extractor:** We use the Pipeline for training which is Spacy based and gets trained on TensorFlow and keras to save the modular changes recommended by Spacy. Rasa is already a customized chatbot which uses NLU to build custom AI.
- e) **Features:** Once the intents are created the RASA featurizes and extracts the intents for training

5. Evaluation

a) Results on extrinsic & intrinsic & human evaluation performed

I. Natural Language Understanding:

- As I have used interactive mode for training, it keeps saving the chat as entities and we can train the same to make the bot more mature.
- In start training the bot worked well but it improves the replies as we chatted more and trained with more data
- As the task is completed the bot greets to say bye etc.

II. Dialogue Management components:

- The response policy: The bot keeps asking the question if the user has any issue or query and based on the query the chat revolves around the intents and trained NLU.
- The fallback policy in case of lack of data or error recovery: As a fallback it repeats the question or asks the user to rephrase the question. If it doesn't have an answer it says to try later till the info gets updated. Added **FallbackClassifier** in config files.

b) The correct interpretation of errors and analysis:

- The major error the bot can make is when it is not trained with the relevant intents.
- If we ask 1 questions repeatedly the model is about to give wrong output. And also, to stop the recurrent chat we have to implicitly add the stop word to the loop.
- Some format issues in CMD run and package dependency errors with RASA are faced.
- Some errors are faced due to inconsistent space or continuation of intent keyword and intent name
- Spelling error, Out of Scope messages needs to be handled.
- Flows and misclassification need more data for correction.

6. Conclusion

Successfully setup the Rasa Architecture and created the intents and stories to train the Spacy and TensorFlow based models. And get to learn many things like actions, forms, rules, regex, synonyms, interactive learning, config files, pipelines, and so much more.

Rasa is a fantastic conversational AI tool. It is a wonderful option tool because of its adaptability and depth of customization. The bot we created for this post was quite basic.

The Recruitment Bot was able to give the status of candidates and also ask the relevant questions from the Candidate.

Output:

```
>> Your input -> hi
Hello! How can I help?

>> Your input -> my name is Hanusha
Hi Hanusha! Let me check that for you
Yes, your application has been rejected. Comments"We need someone with more than 5yrs exp."

>> Your input -> is here any other opening
What is it that I can help you with? Open position or check you application status

>> Your input -> is there any machine learning positions
Are you looking for a technical or a business role?

>> Your input -> technical
machine learning engineer and ML product success engineer are the open positions.

>> Your input -> machine learning engineer
Great! Let me start the process with some basic questions.
Can you type your full name

>> Your input -> Hanusha Sabbavarapu
What is the location are you looking for?

>> Your input -> Chennai, India
Sorry! Not available at that location. Currently position is based at Noida.
Are you willing to relocate?

>> Your input -> yes
Great!
Can you tell me about you last Role

>> Your input -> Associate ML engineer
Interesting!
What was you last qualification?

>> Your input -> Bachelor in IT
Great! We will get back to you.
Do you want more help?

>> Your input -> No
Thanks! Have a Great Day!
```