



Building Robyn: A Natural Language Interface to Database

System for Medicine

Ashutosh Rai
Earlham College

ABSTRACT

Found in abundance in various platforms such as smartphones, computers, web platforms, popular commercial software and so on, natural language interface to database(NLIDB) systems are ubiquitous in today's world. However, despite the rapid growth of the field, the limited availability of resources to learn about building such systems creates a high entry barrier. Additionally, absence of an NLIDB system that is dedicated to medical health, which is a topic of utmost importance, persists as well. Robyn is a software (web application) that attempts to solve both these issues that is built with easily accessible software tools and architecture. Robyn implements AIML as the NPL engine, SQLite as the database engine, and Python to put all the components together as well as provide the interface for the users. Bottle, a micro web-framework written in Python is used to host Robyn on the server-side

INTRODUCTION

A natural language interface to database (NLIDB) is a system that allows the user to access information stored in a database by typing requests expressed in some natural language (e.g. English)[1]. It allows technical actions to be activated without the rigidity of technical sentences and syntaxes, which is replaced by a barrierless natural conversation. Modern Intelligent Personal Assistants(IPA) like Siri, Cortana and Google Now could be considered some popular instances of such a system. However, these are rather general purpose programs that complete tasks and answer questions related to a range of fields rather than focusing on a particular field. This project will present Robyn, a natural language interface to database system with medicine as the domain of knowledge.

After much testing and analysis (which is discussed later in the paper) of various tools, Python, AIML and SQLite were chosen to build Robyn. AIML (Artificial Intelligence Markup Language) is an XML-compliant language used for creating natural language software agents. It was used to handle the natural language processing for the interaction. A SQLite backend was used for the database while Python was used to put all the tools together, provide an interface and handle the queries. With the choice of tools for Robyn set, various designs for Natural Language Processing, storing database, handling queries and putting together the different tools were explored before choosing a hierarchical pattern based design for natural language processing and marking of queries returned from the AIML processing with a special identifier to distinguish queries from other conversation. Python handled the queries, setting up the statements, communicating with the SQLite database and then returning the results to the user. Limited context is maintained as well, using AIML to store the topic of the current conversation (i.e which disease). Bottle was used as the web framework to host Robyn on the server-side.

HIGHER LEVEL WORKFLOW

The workflow in Robyn works like this in general:

1. First, via a text based interface, Robyn provides a platform for users to communicate and ask questions in a natural language (English in this case). The input could be general phrases such as "Hello!", "How are you?", or questions related to medical health, in which Robyn specialized such as "What is AIDS?" or "What are the symptoms of common cold?".
2. The input would then be analyzed and parsed. If it is a general phrase such as "How are you?" then an appropriate response such as "I'm fine, how about you?" would be chosen. However, if the input is a medicine related question then necessary query is chosen/formed.
3. The query then would be applied against the right database.
4. After the correct data is abstracted, the data is then put in a sentence in a natural language (English for Robyn), maintaining the flow of the natural language.
5. Finally, the response, both the general phrase or medicine specific, is then output in the text based interface

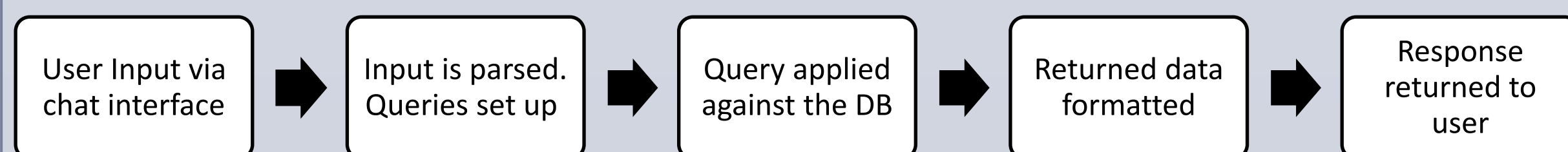


Figure 1. Robyn's workflow

SOFTWARE USED

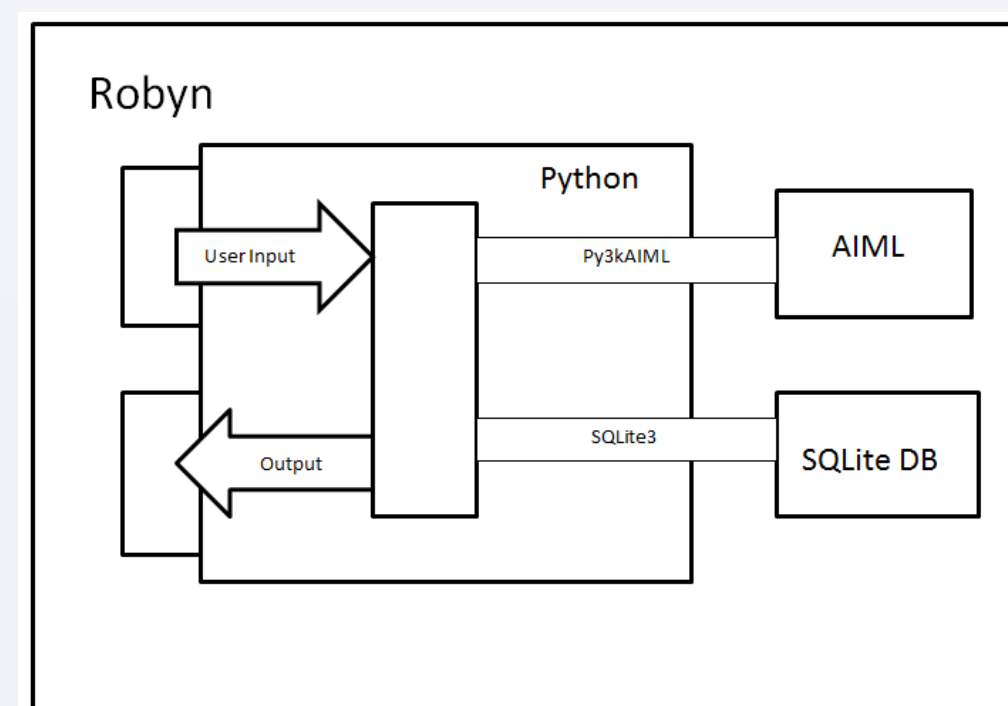


Figure 2. Robyn's architecture

- AIML: AIML is the heart of natural language processing part of Robyn. When the user inputs a text, AIML handles the response and provides necessary query that is applied against the database if necessary.
- SQLite: The portability, low barrier to accessibility and compactness of SQLite made it an excellent choice as the database engine for Robyn.
- Python: Python provides the interface to Robyn for users to interact, as well as connects Robyn to the AIML and database engine. The three main modules that were used in the Python scripts were Py3kAIML, SQLite3 and Bottle.
- HTML/CSS/JS & Bottle: These web development tools were used to provide Robyn with a web interface and make the interaction smooth and user-friendly. As Robyn's base file is written in Python, Bottle was used to host it on the server.

IMPLEMENTATION

Robyn was implemented in 5 main phases with each phase focusing on a particular component:

- Connecting Robyn with AIML/Database: Python provides reliable, minimal configuration modules for AIML as well as SQLite in the form of. Hence, integration of AIML and the database took minimal coding.
- Dialogue Handling: On a basic level, AIML responds to user inputs by matching the pattern, choosing the one that is higher in the hierarchy in case of multiple matches. Different types of statements are sent back by AIML depending on the whether the output requires data from the database or not. Special identifier(!) was used to differentiate SQL queries from natural sentences.

```

<category>
<pattern>HOW ARE YOU</pattern>
<template>
  Thanks for asking. I'm good, how about you?
</template>
</category>

<category>
<pattern>WHAT ARE THE CAUSES OF *</pattern>
<template>
  !SELECT description FROM facts
  WHERE id = (SELECT id FROM disease WHERE name='*')
  AND (fact = 'Causes');
</template>
</category>
  
```

Figure 3 and 4. AIML handling of the dialogues

- Handling Query Statements: When there the special character (!) is observed in the start of the response returned by the AIML engine, first the character is removed. Then the query statement is applied against the database, abstracting the necessary data and returning it to the UI screen
- Designing Web Interface: Robyn is a web-app, so for the web interface a simple chat screen with text box for user input was implemented using HTML/CSS/JS. Bottle was used on the server-side to host Robyn.

RESULTS

Robyn currently exists as a web-app locally on my machine. A test case in the form of a set of questions was formed. The result can be seen below. While this conversation is only a small set of questions asked from potentially thousands of questions, it should show the broad and depth of Robyn's knowledge about diseases as well as her ability to interact in context. As of now, Robyn's database has information such as definition, symptoms, causes, diagnosis and treatment about 4000 diseases.

You: Hi!
Robyn: Hi there!
You: How are you?
Robyn: Thanks for asking. I'm good, how about you?
You: I am fine!
Robyn: Glad to hear it. What's new?

You: What are the symptoms of flu?
Robyn: Common signs and symptoms of flu include fever over 101 F.....
You: What are the causes?
Robyn: Three strains of viruses cause the flu namely influenza A, B, and C.....
You: How can I diagnose it?
Robyn: A thorough physical exam is needed to diagnose the flu and a
You: What is bronchitis?
Robyn: Bronchitis is a condition that occurs when the inner walls lining
You: How can I treat it?
Robyn: A non-prescription cough medicine, plenty of rest, and lots

Robyn: Your Friend for Medical Health!

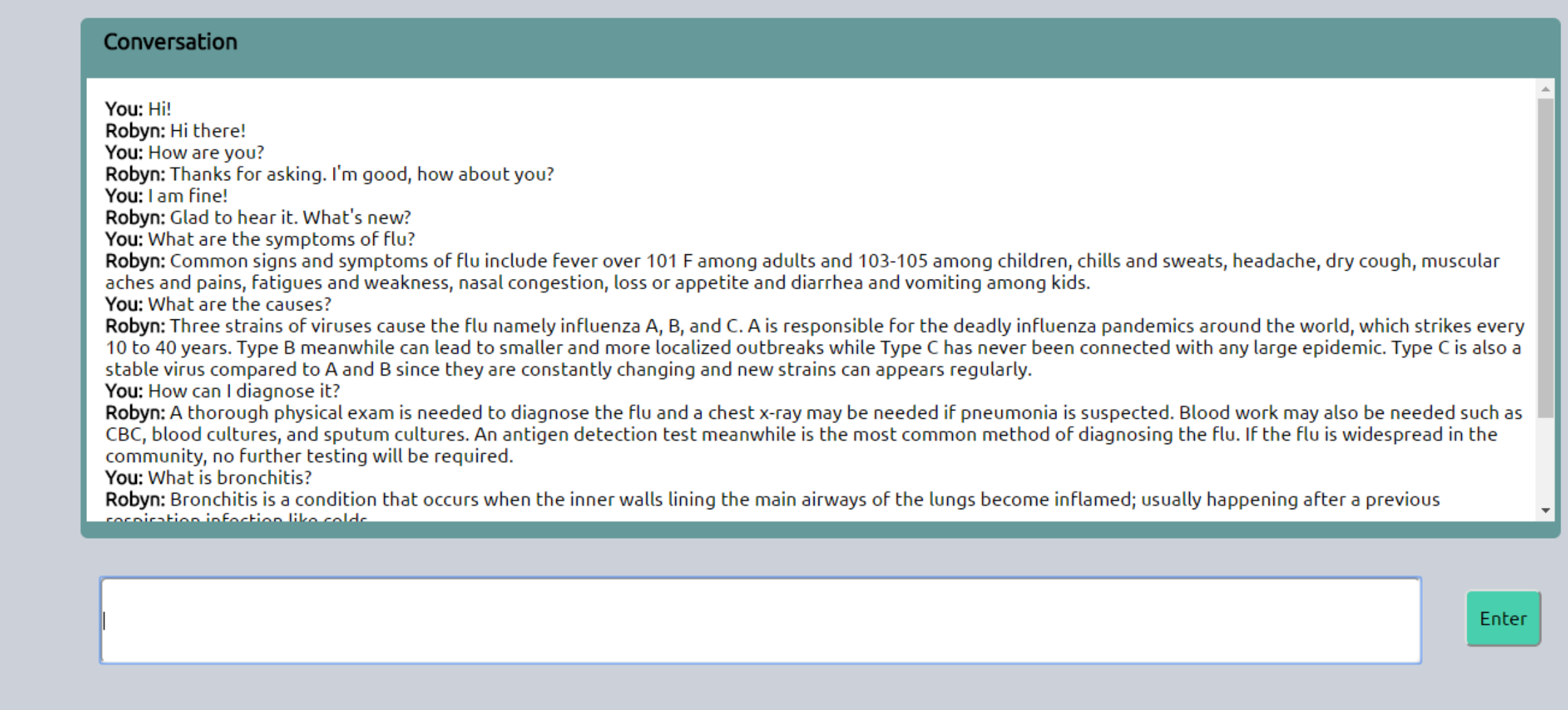


Figure 5. Robyn's web interface

CONCLUSION AND FUTURE WORK

Natural Language Interface to Database systems have always been a field of interest in the realm of Artificial Intelligence since it's invention. With the integration of such systems in popular commercial machines, like Apple's Siri and Microsoft's Cortana, as well as platforms, like Facebook it can be safely assumed that further development of NLIDBs will be continued. Robyn, an NLIDB with medical health as her domain of knowledge, is another example of such a system. The software used to write Robyn such as Python, SQLite and AIML are popular tools as well, which will hopefully make the barrier to learning more about such systems minimal. After some improvement, Robyn will hopefully be a valuable asset to refer to when talking about medical health.

For future, I have identified three main tasks:

- Setting up on a public server: Due to some problem with the libraries/module implementation, I have not been able yet to place Robyn's directory on the public server. However, once I get that done Robyn should be open and accessible to all.
- Natural Language Processing: While natural language processing of a system is something that will always be improvable, I wish to significantly focus on this aspect of Robyn. Along with expanding the range of conversations she can handle, I also wish to work on building and maintaining context of the conversations.
- Expanding the database: While the current database is fairly extensive with 4000 diseases, I wish to expand the database to include more related information. Possible expansion I am thinking about is database related to medicines/drugs, diet information, healthy habits and so on.

REFERENCES

[1] I. Androutsopoulos, G. D. Ritchie, and P. Thanisch. Natural language interfaces to databases—an introduction. Natural language engineering, 1(01):29–81, 1995.

ACKNOWLEDGEMENT

This work was possible through the guidance and support of my senior capstone advisers/mentors Dave Barbella, Xunfei Jiang and Charlie Peck. I would also like to thank the CS Department at Earlham for funding the project, and Adam Shwartz for his assistance with debugging. Finally, I would like to thank the chatbots.org forum community for their valuable discussions.