|  |
| --- |
| COM3001 Final Year Project |
| Developing a Chatbot to Answer Wikipedia Queries |
| University of Surrey |

|  |
| --- |
| Alex Turner  13 February 2020 |

# Declaration of Originality

I confirm that the submitted work is my own work. No element has been previously submitted for assessment, or where it has, it has been correctly referenced. I have also clearly identified and fully acknowledged all material that is entitled to be attributed to others (whether published or unpublished) using the referencing system set out in the programme handbook.

I understand that the University may submit my work as a means of checking this, to the plagiarism detection service Turnitin® UK. I confirm that I understand that assessed work that has been shown to have been plagiarised will be penalised.

If in completing this work I have been assisted with its presentation by another person, I will state their name and contact details of the assistant in the ‘Comments’ text box below. In addition, if requested, I agree to submit the draft material that was completed solely by me prior to its presentational improvement.

© Copyright Alex Turner, January 2020

# Acknowledgments

# Abstract

Abstract here

Contents

[Declaration of Originality 1](#_Toc32263577)

[Acknowledgments 3](#_Toc32263578)

[Abstract 4](#_Toc32263579)

[List of Figures 7](#_Toc32263580)

[List of Tables 8](#_Toc32263581)

[Glossary 9](#_Toc32263582)

[Abbreviations 10](#_Toc32263583)

[1 Introduction 11](#_Toc32263584)

[1.1 Overview 11](#_Toc32263585)

[1.2 Objectives 11](#_Toc32263586)

[1.3 Report Structure 11](#_Toc32263587)

[2 Literature Review 12](#_Toc32263588)

[2.1 Chatbots 12](#_Toc32263589)

[2.2 Natural Language Processing 12](#_Toc32263590)

[2.3 Machine Learning 12](#_Toc32263591)

[2.4 Datasets 12](#_Toc32263592)

[2.5 Programming Languages 14](#_Toc32263593)

[2.6 Existing Solutions 14](#_Toc32263594)

[2.7 Conclusion 14](#_Toc32263595)

[3 Problem Analysis 15](#_Toc32263596)

[4 System Requirements 16](#_Toc32263597)

[4.1 Requirements Gathering 16](#_Toc32263598)

[4.2 Software Requirements 16](#_Toc32263599)

[Functional Requirements 16](#_Toc32263600)

[Performance Requirements 17](#_Toc32263601)

[5 Design 18](#_Toc32263602)

[5.1 Methodology 18](#_Toc32263603)

[6 Implementation 19](#_Toc32263604)

[7 Testing 20](#_Toc32263605)

[8 Project Evaluation 21](#_Toc32263606)

[9 Conclusion 22](#_Toc32263607)

[9.1 Overview 22](#_Toc32263608)

[9.2 Conclusions 22](#_Toc32263609)

[9.3 Future Work 22](#_Toc32263610)

[9.4 Final Statement 22](#_Toc32263611)

[10 Ethics 23](#_Toc32263612)

[11 References 25](#_Toc32263613)

[12 Appendix 27](#_Toc32263614)

# List of Figures

[Figure 1: DBpedia Ontology instances per class [13] 13](#_Toc32265715)

[Figure 2: DBpedia extraction framework [12] 14](#_Toc32265716)

# List of Tables

# Glossary

# Abbreviations

HCI

IoT

# Introduction

## Overview

## Objectives

## Report Structure

# Literature Review

In the past decade, conversational chatbots have seen a surge in popularity. The virtual assistant, such as Google Assistant and Amazon Alexa, are now entering our homes with Internet of Things devices. In 2017, Google Assistant was installed on over 400 million devices [1]. Furthermore, specialized chatbots have seen an influx within banking, retail, and healthcare. Chatbots represent a trend towards using natural language in the realm of human-computer interaction (HCI). This literature review will explore how chatbots are implemented, their benefits, and how this project can innovate within the chatbot space.

## Chatbots

When discussing chatbots, the beginning of their history is usually cited as Alan Turing’s 1950 article “Computing Machinery and Intelligence” [2], wherein Turing describes a test to determine whether a human evaluator can distinguish between a human and a machine during a natural language conversation. This test became known as the Turing test, and asks the question ‘Can machines think?’. However, the goal of many chatbots is not to create true artificial intelligence, but rather to using pattern matching and conversational responses to mimic the responses of a human.

One of the first programs to attempt the Turing test was ELIZA, created by Joseph Weizenbaum between 1964 and 1966 [3]. ELIZA consisted of a language analyser and a set of rules by which the ‘chatterbot’ followed. ELIZA used a script called DOCTOR was designed to simulate responses of a psychotherapist during a psychiatric interview – predominantly achieved by the therapist mirroring the responses of the patient [3]. ELIZA may be considered rudimentary and narrow by today’s standards, it forms the basis of our understanding of chatbots and human computer interaction, and how we can teach machines to mimic human characteristics in dialogue.

Another notable development in chatbots and natural language processing is ALICE (Artificial Linguistic Internet Computer Entity), originally implemented in 1995 by Richard Wallace [4]. The system won the Loebner Prize three times, a competition inspired by the Turing Test to judge how well a machine can mimic human responses. Although the prize itself was met with some criticism, Shieber critiques that the goal of the Turing Test is lost on the competition [5], ALICE provides a framework for many of the fundamentals we see in modern chatbots and AI.

Intelligent virtual assistants (IVA) are conversation agents that allow users to interact with services and Internet of Things (IoT) devices [6]. IVAs are ubiquitous in modern life, with most smartphones pre-equipped with a virtual assistant such as Google Assistant or Apple Siri. In many ways, IVAs incorporate many functions of chatbots, as well as providing additional features such as voice input and communication with IoT “smart devices”. ---

Industries are seeing a growing trend in chatbot integration in their business. Autodesk integrated IBM’s Watson Assistant [7] to process 100,000 user support conversations, reducing the resolution time of enquiries from 38 hours to 5.4 minutes [8]. Many technology companies offer AI cloud services, many of which allow the integration of chatbots including Watson Assistant [7] and Microsoft Azure Bot Service [9]. The next section will explore and discuss techniques for implementing chatbots and explore technologies that can be used.

## Chatbot Models

A chatbot usually consists of three key components – natural language processing (NLP), Generating a response given the context of a conversation is one of the fundamentals of a chatbot system. These models are usually rule-based or learning-based [7], and have their advantages and challenges which will be explored in this section.

A rule-based model uses pre-defined patterns in order to match an input to a response. This is seen in ALICE, which uses AIML to construct stimulus-response pairs [4].

* Open/closed domain
* Generative/retrieval
* NLP

### Neural Networks

### Pattern Matching

* AIML and other technologies

## Datasets

Typically, chatbots are divided into two groups, open-domain and closed-domain [8]. In an open-domain system, the conversation can go in any direction, and the user can talk to the chatbot about any topic. A closed-domain system is restricted to a narrower topic area or set of function – these are the chatbots we see most in real-world applications such as customer service and banking. For this project, the focus will be on a closed-domain system as the goal is to create a chatbot that can achieve a goal – these are often called Goal-Oriented (GO) Chatbots [8]. However, to create a GO chatbot, one must have a goal the chatbot should achieve, and a dataset from which to learn.

The Ubuntu Dialogue Corpus (UDC), is one of the largest public dialogue datasets available [9], consisting of 1 million multi-turn dialogues from users receiving technical support for Ubuntu-related problems [10]. **continue**

In terms of knowledge bases that lend themselves to the question and answer format, Wikipedia is the world’s largest collaboratively edited source of encyclopaedic knowledge [11]. In terms of size, it eclipses the size of the Encyclopaedia Britannica, its nearest rival, by a factor of ten [12] – as of 12 November 2019, there are over 5.9 million articles in English, and over 51 million articles in the 306 languages officially covered by the Wikimedia Foundation [13]. However, Wikipedia’s content is only fit for human reading [11] and is hard to process computationally. Many attempts have been made to formalise and structure this data, as seen in [11], [12], [14], but this review will focus one of these being DBpedia.

DBpedia is a crowd-sourced effort to extract structured content from various Wikimedia projects [15], including Wikipedia. The English version of the DBpedia knowledge base describes 4.58 million things, out of which 4.22 million are classified in a consistent ontology [], consisting of 320 classes described by 1,650 different properties [16]. This structure enables programs to process this data effectively, including a chatbot application. The size of the DBpedia Ontology is shown in Figure 1, which demonstrates the scale of the project, and how this might be effective for the project.

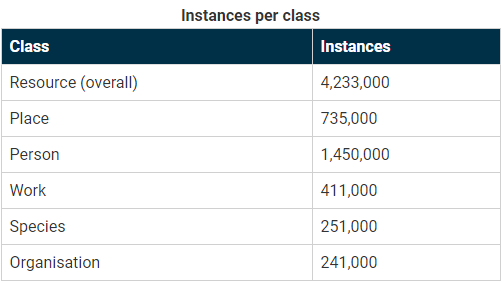


Figure : DBpedia Ontology instances per class [17]

The DBpedia extraction framework is responsible for extracting data from Wikipedia into a structured knowledge base, an overview of which is shown in Figure 2. This extraction is structured into four phases, as described in [16]:

**Input:** Wikipedia pages are read from an external source, either from a Wikipedia dump, or using the MediaWiki API.

**Parsing:** Each Wikipedia page is parsed, which transforms the source code of the Wikipedia page into an Abstract Syntax Tree (AST). An AST is a tree representation of the syntactic structure of the source code.

**Extraction:** The Abstract Syntax Tree of each page is forwarded to the extractors. There are many types of extractors, which will later be described, which extract data such as labels, images and infoboxes. Each extractor takes an AST as input and yields a set of Resource Description Framework (RDF) statements. These are XML statements which describe properties and values of resources.

**Output:** These RDF statements are written into sinks, which receive the data.

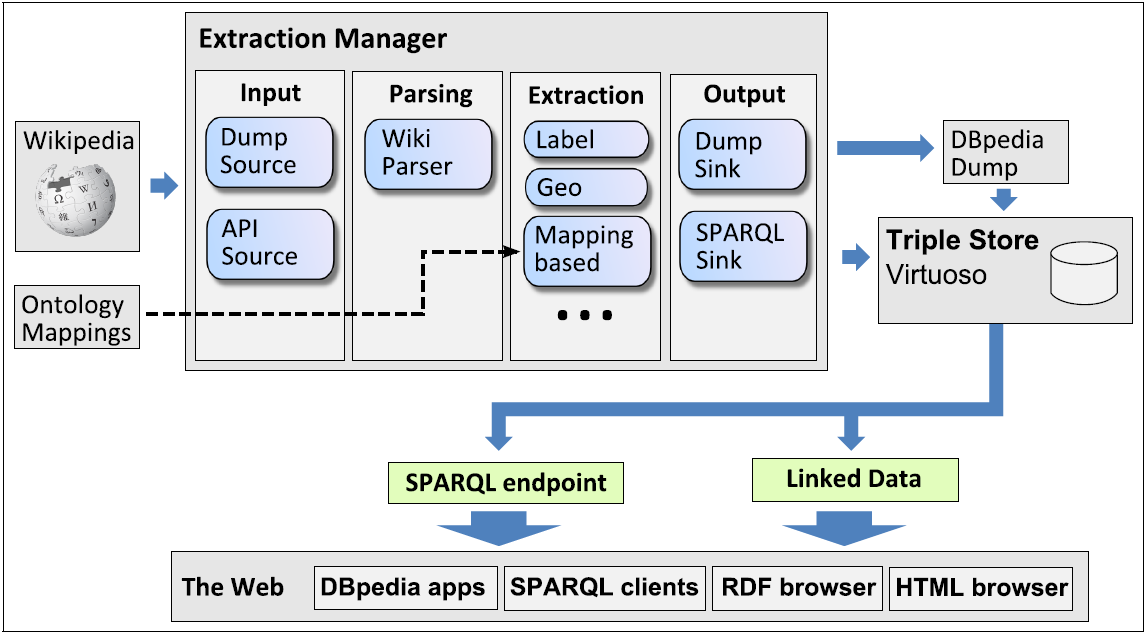


Figure : DBpedia extraction framework [16]

* Ontology
* Endpoint (SPARQL)

## Programming Languages

* Java
* Python
* Libraries

## Chatbots

* AIML
  + ALICE framework
  + <https://ieeexplore.ieee.org/document/7810979>
* Alternatives

## Web Interface

* Vs standalone/command line
* Web technologies
  + JSP
  + Spring

## Existing Solutions

* Google Search/Assistant
* DBPedia chatbot
  + Analysis
* Mitsuku

## Related Works

* AIML and SPARQL papers

## Conclusion

# System Analysis

* 1. Existing Solutions

1. Feasibility Study
2. Requirements
   1. Requirements gathering
   2. Functional requirements
   3. Non-functional requirements
   4. Requirements prioritisation
3. Planning

## Feasibility Study

* feasibility

## System Requirements

### Functional Requirements

* Intro

|  |  |  |  |
| --- | --- | --- | --- |
| **ID** | **Requirement** | **Assessment** | **Priority** |
| F1 | **User Interaction** | | |
| F1.1 | The system should allow the user to interact with the chatbot through the web browser | The chatbot is shown | Essential |
| F1.2 | The user should be able to send queries to the chatbot |  | Essential |
| F1.3 | The user should be able to see the response of the chatbot in the webpage |  | Essential |
| F1.4 | The user should be able to see the conversation with the chatbot in the webpage |  | Essential |
| F2 | **Basic Queries** | | |
| F2.1 | **Basic Person Queries** | | |
| F2.1.1 | *‘Who is X’*  The application should take a query about a person and respond with a description of that person |  |  |
| F2.1.2 | *‘When was X born’*  The application should take a user query about the birthdate of a person and return the birthdate of that given person. |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

# System Design

4.

## Methodology

* Software development methodology with justification

## Design Experimentation

# Implementation

# Testing

# Project Evaluation

# Conclusion

## Overview

## Conclusions

## Future Work

## Final Statement

# Ethics

# References

|  |  |
| --- | --- |
| [1] | R. Chandra, “How Google Home and the Google Assistant helped you get more done in 2017,” 5 Jan 2018. [Online]. Available: https://blog.google/products/assistant/how-google-home-and-google-assistant-helped-you-get-more-done-in-2017/. [Accessed 5 Nov 2019]. |
| [2] | A. M. Turing, “Computing machinery and intelligence,” *Mind,* vol. 59, no. 236, pp. 433-60, 1950. |
| [3] | J. Weizenbaum, Computer Power and Human Reason: From Judgment to Calculation, New York: W.H. Freeman and Company, 1976. |
| [4] | R. S. Wallace, “The Anatomy of ALICE,” in *Parsing the Turing Test*, Dordrecht, Springer, 2009, pp. 181-210. |
| [5] | S. M. Shieber, “Lessons from a Restricted Turing Test,” *Communications of the Association for Computing Machinery,* no. 37, pp. 70-78, 4 Apr 1994. |
| [6] | H. Chung and S. Lee, “Intelligent virtual assistant knows your life,” *arXiv preprint,* vol. arXiv:1803.00466, 2018. |
| [7] | H. Wang, Z. Lu, H. Li and E. Chen, “A dataset for research on short-text conversations,” in *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, 2013, pp. 935-945. |
| [8] | V. Ilievski, “Building Advanced Dialogue Managers for Goal-Oriented Dialogue Systems,” 16 Mar 2018. [Online]. Available: https://arxiv.org/pdf/1806.00780.pdf. [Accessed 13 Nov 2019]. |
| [9] | S. Kojouharov, “Ultimate Guide to Leveraging NLP & Machine Learning for your Chatbot,” Chatbots Life, 18 Sep 2016. [Online]. Available: https://chatbotslife.com/ultimate-guide-to-leveraging-nlp-machine-learning-for-you-chatbot-531ff2dd870c. [Accessed 13 Nov 2019]. |
| [10] | R. Lowe, N. Pow, J. Pineau and I. V. Serban, “The Ubuntu Dialogue Corpus: A Large Dataset for Research in Unstructed Multi-Turn Dialog Systems,” 4 Feb 2016. [Online]. Available: https://arxiv.org/pdf/1506.08909.pdf. [Accessed 13 Nov 2019]. |
| [11] | M. Völkel, M. Krötzch, D. Vrandecic, H. Haller and R. Studer, “Semantic Wikipedia,” in *Proceedings of the 15th international conference on World Wide Web*, Edinburgh, 2006. |
| [12] | O. Medelyan, D. Milne, C. Legg and I. H. Witten, “Mining meaning from Wikipedia,” *International Journal of Human-Computer Studies,* vol. 67, no. 9, pp. 716-754, 2009. |
| [13] | Wikimedia Foundation, “List of Wikipedias,” 17 Oct 2019. [Online]. Available: https://meta.wikimedia.org/wiki/List\_of\_Wikipedias. [Accessed 13 Nov 2019]. |
| [14] | F. Wu and D. S. Weld, “Autonomously semantifying Wikipedia,” in *CIKM '07 Proceedings of the sixteenth ACM conference on Conference on information and knowledge management*, Lisbon, 2007. |
| [15] | DBpedia, “About | DBpedia,” 2019. [Online]. Available: https://wiki.dbpedia.org/about. [Accessed 13 Nov 2019]. |
| [16] | J. Lehmann and et al., “DBpedia – A Large-scale, Multilingual Knowledge Base Extracted from Wikipedia,” *Semantic Web,* vol. 6, no. 2, pp. 167-195, 2015. |
| [17] | DBpedia, “Ontology | DBpedia,” 2019. [Online]. Available: https://wiki.dbpedia.org/services-resources/ontology. [Accessed 13 Nov 2019]. |

# Appendix