University of Essex – Computer Science

Initial Report

Open Domain Question & Answer System

Osama Rahman – 1304349

John Gan

Diego Perez Liebana

Table of Contents

Table of Contents 2

Introduction 3

Project Goals 4

Core Objectives: 4

Secondary Objectives (To be completed if possible / have time): 4

Personal Objectives: 4

Goal Summery 5

Development 5

Natural Language Processing 5

Adapt Intent Parser by Mycroft 5

Natural Language Tool Kit (NLTK 3.0) 5

Databases / Knowledgebases 6

DBpedia 6

Voice API’s 6

SpeechRecognition 3.4.6 6

Information Retrieval 6

Planning and Project Management 7

Waterfall Methodology 7

Gantt Chart 7

Research 8

Factoid vs Complex Questions 8

Similar Systems 8

Information Retrieval (IR) Research 9

Natural Language Processing Research 10

Knowledgebase vs Database 10

References 11

# Introduction

In modern computing it seems that the next wave of consumer and commercial innovation will be in artificial intelligence. The field, while not new, has recently spiked in popularity. With companies such as Google and Apple competing head to head with their *personal assistants*, and IBM researching and developing Watson, it is an exciting time to be in the field of artificial intelligence.

Open domain question & answer systems are a form of artificial intelligence which, when given a question, will return an answer. Currently, no system has a truly open domain, artificial intelligence isn’t quite there yet. However, many systems such as Siri, Google Now, and IBM’s Watson, have a substantial domain and will be able to answer a considerable amount of questions. Usually these questions will be trivial such as “What is the weather like today?”, or “Who is the current prime minister of the United Kingdom?” Open ended questions such as “Why did Brexit happen?” are not likely to be answered as the answer would have to be meaningful, and would rely on more than textual information to be formulated.

In this project I intend to create an open domain question and answer system, likely with more of a niche. Alongside this I intend to implement basic general questions one would find themselves asking Siri or Google now, questions such as “What is the time?”, or perhaps simple mathematics such as “What is six multiplied by two?”. If possible I will be implementing an open source voice API so the user can speak to the system, as opposed to communicating the question via text. This will be discussed in further details below.

# Project Goals

## Core Objectives:

* Create a user friendly GUI to input and output text and data
  + It is important to create a user interface so the end user feels comfortable using the application. I will be using websites such as Dribbble to gain inspiration while also basing the design on systems such as Siri and Google Now
* Parse natural language user input into a readable query
  + This is needed for the system to retrieve information from the knowledgebase/ database. A user will ask a question, this will then be translated using Natural Language Processing into a computer readable query to retrieve the answer. I will be using an open source natural language processing module.
* Use the query to retrieve data from a database/knowledgebase
  + Once translated into a query, from natural language, the system will use this query retrieve the relevant data from a knowledgebase/database.
* Parse the information from the database/knowledgebase into a natural language reply
  + Once the data has been retrieved it will be slightly modified to make the systems answer more human like. The answer will be similar to the reply one would expect from an actual human.

## Secondary Objectives (To be completed if possible / have time):

* Implement a voice API to allow a user to ask a question through speech
* Implement a large data dump, to make give the system a more open domain
* Implement a way to answer basic maths questions
* Implement a weather API for the system to report back weather
* Implement a world clock API for the system to report global times

## Personal Objectives:

* Enhance my programming knowledge and skills
* Further educate myself in the field of artificial intelligence
* Enhance my project management skills and knowledge
* Enhance my professional skills

## Goal Summery

To summarise the project goals, the core objectives are the ones that will be implemented. Combined, these objectives will allow for a relatively open domain question and answer system. The secondary objectives will be implemented if it is possible, and I am able to do so within the deadline. I have also outlined some personal objectives, seeing as how this is my final year project and I will not only be showing what I have learnt over the years, but I will also be learning throughout the project.

# Development

I will be using some open source software, mostly for non-core modules, however I will also be using open source software for core modules, such as the natural language processing. My main development will be with the information retrieval.

## Natural Language Processing

This module will allow the user to interact with the system in the same way a human would interact with another human, through natural language. I have decided to use an open source project for this.

I have not decided which one I will be using yet, as I need to trial each one to see which would suit the needs of the project.

### Adapt Intent Parser by Mycroft

https://adapt.mycroft.ai/

This parser is an open source software library used to convert natural language into machine readable data structures. This parser is a used in the Mycroft AI system, and is used not only to parse natural language, but also to help create commands which will be used for the Mycroft AI system. [1]

### Natural Language Tool Kit (NLTK 3.0)

<http://www.nltk.org/>

This toolkit is specifically designed for Python programs. Out of all of the Natural Language tools available, I think this will be the best one for me to use. NTLK features 104 different corpora and lexical resources to translate natural language into machine readable data. [2]

## Databases / Knowledgebases

This is where the information will be kept, where the system will go to find an answer for a question. There are many open source databases / knowledgebases available, which are great for open domain question and answer systems. Ones I am considering to use are below. The main issue I will run into with this is the sheer size of the database/ knowledgebase. They can range from a few GB to a few TB. What this means is that the larger the size of the data, the longer it will take the system to find the answer.

### DBpedia

<http://wiki.dbpedia.org/>

DBpedia uses Wikipedia as its source of information. DBpedia is more of a knowledgebase than a database. This means that, while structured like a database, it is not a limited concrete technical solution. It will allow for more manipulation and accessibility of data, as opposed to an immediate retrieval of information.

## Voice API’s

I plan on adding voice functionality to the system. Creating one from scratch will prove to be too much work. My time would be better spent on the core functionality of the program.

### SpeechRecognition 3.4.6

[**https://pypi.python.org/pypi/SpeechRecognition/**](https://pypi.python.org/pypi/SpeechRecognition/)

This is a Python library for speech recognition. [4] With support for many SpeechRecognition API’s such as CMUSphinx, IBM Speech to Text, and many more this library should prove to be very resourceful.

## Information Retrieval

From my research I have an understanding on the theory and base principles of retrieving the data. In practice, however, I’m sure my development of this module will be slightly different. In essence all that will need to be done is to send a query to the knowledgebase/database, then have the system return the data in the form of a natural language answer. I don’t plan on using any open source projects for this section of the program, so I will be writing it entirely from scratch.

# Planning and Project Management

## Waterfall Methodology

I have decided to use the traditional waterfall project planning methodology. Currently I am on the system design part of the waterfall method. I have chosen this methodology as I appreciate the classic, straight forward approach to software development.

Considering I will be doing this project on my own, it will be easier to keep to the scheduled sequence of events. The requirements of my project are very clear, the system I am building has been built before, and I have read a lot of material on this subject matter so I know exactly how the end project should work. As with any software development project, no matter the methodology, things are likely to change. For example, some secondary goal modules may not end up being developed, or perhaps the way in which a specific module is required to be developed will change.

## Gantt Chart

Using a Gantt chart will allow me to manage the project and keep a timely schedule. This, along with my logbook, will also help me create the final report. What’s great about a Gantt chart is that it allows for deadlines to be met in a timely and much easier manner.

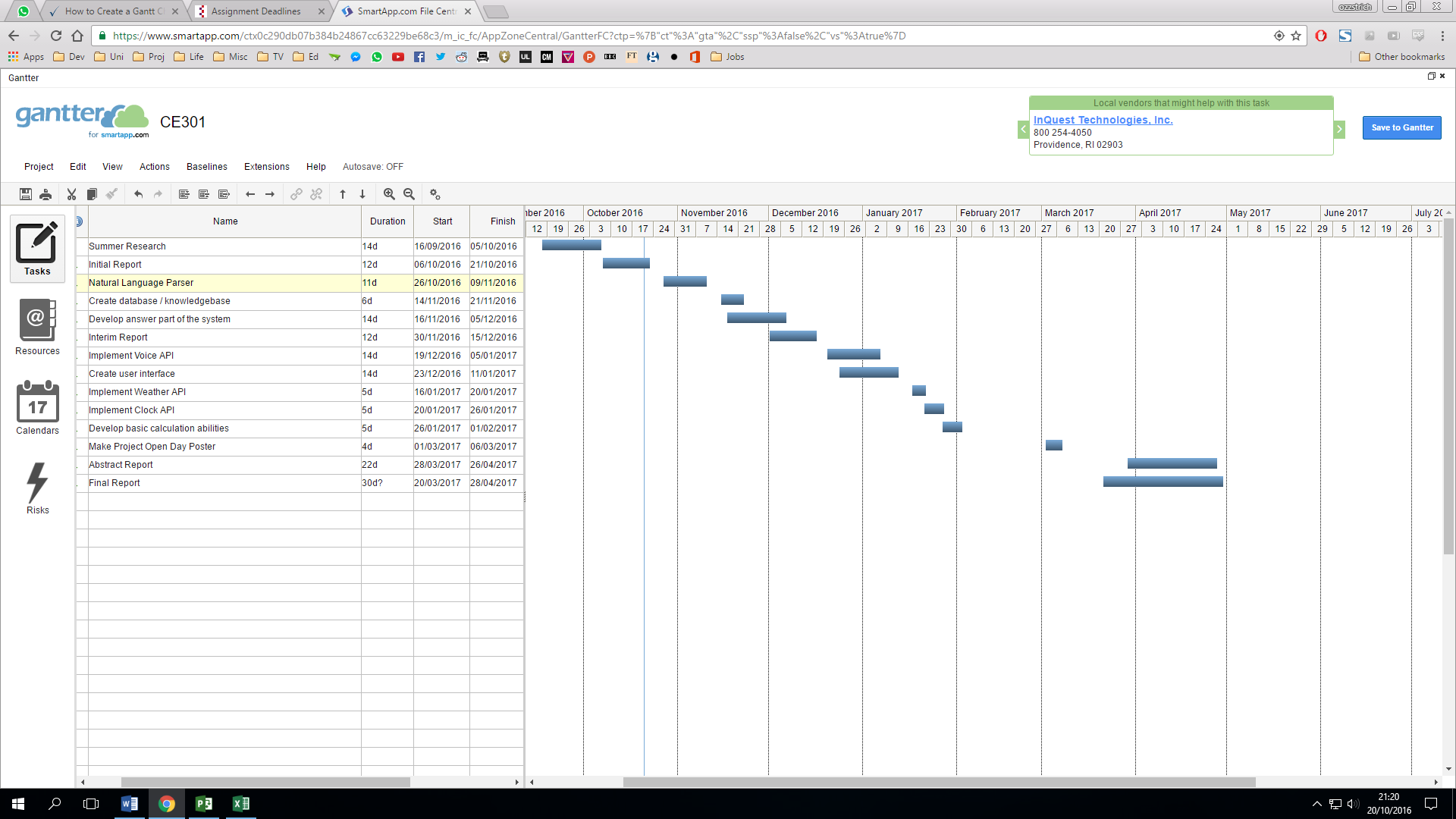


Figure 1. Project Gantt Chart

Above is the proposed Gantt chart for the project. If previous software development projects are any indication, the dates on this Gantt chart are likely to change over time. Some aspects are likely to take longer, while some may take shorter than expected. Over time this Gantt chart will be updated and modified to show progress. While creating this Gantt chart other projects and deadlines were taken into account, which is why there are a few gaps between stages. These gaps also allow for other development stages to overlap, should they need to.

# Research

## Factoid vs Complex Questions

With question and answer systems the types of questions can be split into two main categories, factoid and complex. As the name suggests, factoid questions are answered with short fact based textual answers. Factoid questions are simple; they are questions such as “Where is Apple Computer based?”[5]. Complex questions are ones which don’t rely on solely on facts. To use a previous example “Why did Brexit happen?” is a complex question as it relies on more than facts.

## Similar Systems

There are many open domain QA systems that have been, and are being, developed. Popular commercial systems include Google Now, Siri, and Cortana. These are not only QA systems, but also act as a personal assistant, allowing the user to control the device through the system.

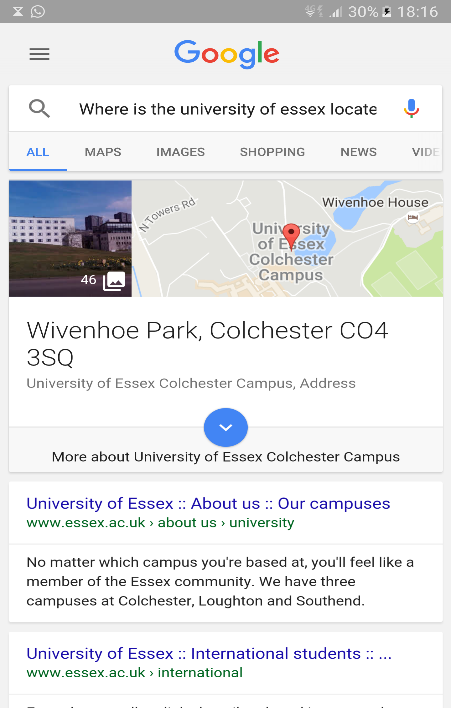
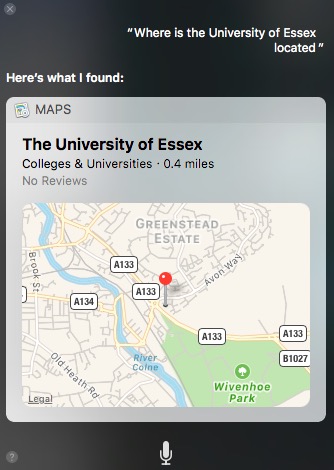


Figure 2.1 Google Now, on Android

Figure 2. Apple’s Siri, on MacOS

As you can see both of these systems can quite easily return answers to factoid questions. The style of these systems are also quite similar, in that they return the location on a card based format.

A more relatable system would be MIT’s START Natural Language Question Answer System (pictured below). As you can see in the above image, this is a web based system, which is able to answer factoid questions. This system uses natural language annotation to connect users to information sources [6]. START was one of the initial QA system on the web, having been connected in December 1993.

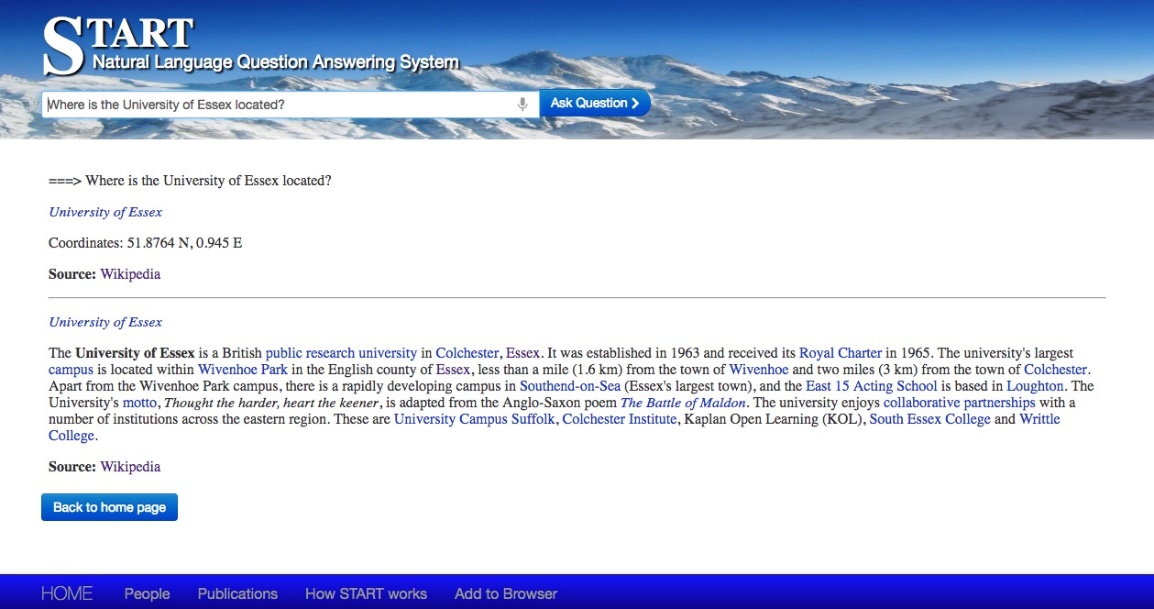


Figure 2.3 MIT’s SMART

### Information Retrieval (IR) Research

Some QA systems use certain IR techniques to retrieve information and return an answer. This is usually more complex than simply querying a database, and returning a string or value. The Open-Domain Textual Question Answering Techniques – UT Dallas (2003) report [7] indicates that a collection of documents must be indexed for information to be retrieved from then. In this context the collection of documents were being used as the knowledgebase. Different types of retrieval methods can be implemented, for example one would rank documents taking the question as an argument, then rank the passages found, then finally return an answer. This report also indicates that a QA system also needs to know which type of answer the user is excepting to receive. An example they use is the location of the Taj Mahal, does the user want to know about the Casino in Atlantic City, or the monument in India?

### Natural Language Processing Research

According to a document written by Himani Shukla and Misha Kakkar keywords must be extracted from a natural language string [8]. Take a question such as “Who directed Star Wars Episode VII”, this question is what's known as raw text. Once asked, the question will be tokenized, in that the words will be split and then the words will be tagged as nouns, verbs, adjectives and so on. The system will have to tag keywords such as “Star Wars Episode VII”, “Who”, and “directed”. From this a query will be formed, allowing the information to be retrieved from the database or knowledgebase.

### Knowledgebase vs Database

Compared to a database, a knowledgebase is used to store more complex, less structured information. Such information would include articles on subjects which are relevant to one’s question and answer system. Databases store information in a much more structured way, using tables and simple queries to retrieve the data. Using a database would be a good option for a simple open domain QA system, and a great option for a closed domain QA system. It could be argued that for a truly open domain QA system, a hybrid of the two would be the best option. Depending on how the initial stage of development goes for this project, it might be a better option to use a database instead of a knowledgebase. This is of course dependent on the data that will be used. If it it’s efficient to make an open domain QA system, then a knowledgebase is likely to be used, if however a specific domain is chosen and the system is closed domain, then a database may suffice. Using a database may prove to have limitations on the capabilities of the system, limitations a knowledgebase will not encounter.

# References

[1] Mycroft A (2016) Adapt intent Parser by Mycroft. Available at: https://adapt.mycroft.ai/ (Accessed: 20 October 2016).

[2] Natural Language Tool Kit (Open Source) (no date) Natural language Toolkit — NLTK 3.0 documentation. Available at: http://www.nltk.org/ (Accessed: 20 October 2016).

[3] DBpedia (2016) About. Available at: http://wiki.dbpedia.org/about (Accessed: 20 October 2016).

[4] Zhang, A. (1990) SpeechRecognition 3.4.6: Python package index. Available at: https://pypi.python.org/pypi/SpeechRecognition/ (Accessed: 20 October 2016).

[5] Klabunde, R. (2014) ‘Daniel Jurafsky/James H. Martin, speech and language processing’, Zeitschrift für Sprachwissenschaft, 21(1).

[6] *The START natural language question answering system* (no date) Available at: http://start.csail.mit.edu/start-system.php (Accessed: 20 October 2016).

[7] HARABAGIU, S.M., MAIORANO, S.J., PASCA, M.A. and University of Dallas (2003) ‘Open-domain textual question answering techniques’, *Natural Language Engineering*, pp. 231–267.

[8] Shukla, H. and Kakkar, M. (2016) *IEEE Xplore document - Keyword extraction from educational video transcripts using NLP techniques*. Available at: http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=7508096 (Accessed: 20 October 2016).