Requirements & Design Specification

For

the Design of a Cognitive Research Assistant

29-March-2019

V0.1

Prepared by:

Poornima R Parasuraman

**Table of Contents**

[Revision History 3](#_Toc4756978)

[1 Introduction 4](#_Toc4756979)

[1.1 Overview 4](#_Toc4756980)

[1.2 Goals and Objectives 4](#_Toc4756981)

[1.3 Scope 4](#_Toc4756982)

[1.4 Definitions 5](#_Toc4756983)

[2 General Design Constraints 5](#_Toc4756984)

[2.1 Product Environment 5](#_Toc4756985)

[2.2 User Characteristics 6](#_Toc4756986)

[2.3 Mandated Constraints 6](#_Toc4756987)

[3 Nonfunctional Requirements 6](#_Toc4756988)

[3.1 Operational Requirements 6](#_Toc4756989)

[3.2 Performance Requirements 7](#_Toc4756990)

[3.3 Security Requirements 7](#_Toc4756991)

[3.4 Other Quality Attributes 7](#_Toc4756992)

[3.5 External Interface 7](#_Toc4756993)

[3.5.1 User Interface 7](#_Toc4756994)

[3.5.2 Storage Interface 7](#_Toc4756995)

[4 Functional Requirements and Design overview 7](#_Toc4756996)

[4.1 Design approach for component : Natural Language Interpreter 8](#_Toc4756997)

[4.2 Design approach for component: Search, Ranking and Download 8](#_Toc4756998)

[4.3 Design for component: User interface 9](#_Toc4756999)

[4.4 Design for component: Feedback 9](#_Toc4757000)

Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| **Version** | **Date** | **Name** | **Description** |
| 0.1 | 29-March-2019 | Poornima | Initial Draft |
|  |  |  |  |

# Introduction

## Overview

This document details the design approach for the building of a cognitive assistant called the Cognitive Research Assistant or the CRA. The function of the CRA will be to accept a request from the user, ascertain the type of request, perform a search/searches, determine if the selected material is relevant, produce a cache of relevant materials for the user to review at a later point in time and return the result (the cache and a link to it.)

There can be two types of research requests –

1. Serious – academic / professional research
2. Casual – outside of academic pursuit e.g. reviews of products, places etc.

A serious request leads to a search for published research papers/white papers while a casual request will be a more general search on the internet. The systems used for the search may be different. The results returned will a combination of links, documents/articles or a subset of them. The CRA may also request the user to be more specific if the search results are too large or may return ‘Result not found’ if there are no results.

## Goals and Objectives

The goal of this project is to design a cognitive assistant (called the Cognitive Research assistant or CRA) who will assist its users in gathering information. More specifically, the assistant which assist the user with gathering published reading material on a topic which the user can then peruse to achieve their end objective.

The objective is to produce a cognitive assistant who function is both supplementary and complementary to that of the user. The assistant will be used by the user when the user requires the task to be completed parallelly and/or in the background while the user themselves are engaged in other activities.

## Scope

The scope defines the boundaries of the product – what it will and will not do.

In Scope-

* The product will produce a cache of relevant material for the user based on the initial request and every request, thereof.
* For the current project, owing to time constraints, the functionality of the product will be limited to a single topic of choice.
* A barebones UI will be provided. The UI will also incorporate a simple feedback mechanism.

Out of Scope-

* Modifying search on the fly if user terminates and resends a request.
* Producing a coherent digest of the material
* Producing an independent summary of the concept.
* Results verified for correctness
* Managing and maintaining storage system resources
* Anything else not in scope.

## Definitions

**Cognitive Assistant –** an agent which can assist with and perform tasks deemed intelligent by humans

**Product** – what is being described here; the software system specified in this document.

**Project** – activities that will lead to the production of the product described here as well as the requirement for COS521 which led to this idea.

**User** – the person or persons who will interact with the product.

**Use case** – describes a goal-oriented interaction between the system and an actor. A use case may define several variants called scenarios that result in different paths through the use case and usually different outcomes.

**Scenario** – one path through a user case

**Actor** – user or other software system that receives value from a user case.

**Role** – category of users that share similar characteristics.

**Developer** – the person or organization developing the system, also sometimes called the supplier.

**Stakeholder** – anyone with an interest in the project and its outcomes. This includes clients, customers, users, developers, testers, managers and executives.

**Controls** – the individual elements of a user interface. For example, the following are user interface controls: button, check box, dropdown list.

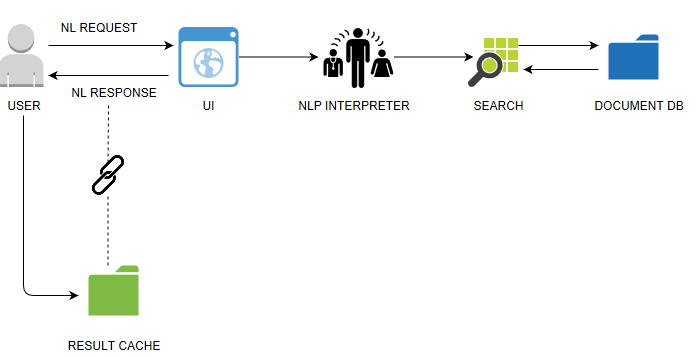
# General Design Constraints

## Product Environment

The product will consist of the following components –

1. A user interface for communication
2. NLP interpreter
3. Search module for information search and gathering
4. A decision-making module to determine relevance of information gathered to the request made
5. A storage system to store and access results
6. A feedback mechanism to improve performance and storage system to record and retain feedback
7. A user who believes it is worthwhile to use the CRA

The CRA will interface with an internal or external document repository as well as a result storage system.



## User Characteristics

The user is the starting point of this system. The CRA receives its context and input from the user i.e. it is not a continuously running system. It becomes active when a user requests its assistance. The user is responsible for requesting the assistance of the CRA and providing the Natural Language Request. The quality of the results depends somewhat on the quality of the request, so this part of the burden falls on the user.

The user will also provide feedback for the results returned by the system. This helps the system adjust its performance measurement as well as its function accordingly.

The user will access results as required based on the link provided by the system, the user will provide the storage resources required to record results.

## Mandated Constraints

The final project needs be to be delivered by April 15.

# Nonfunctional Requirements

The system should be able to return results in an appropriate amount of time or alternatively, keep the user updated of the status of the task. The system should also limit results to a reasonable number of documents so as to not overwhelm the user with results. Results will be returned either as pdf or text files or a combination.

## Operational Requirements

The input to and output of the system will be visual. There will be no audio component.

## Performance Requirements

It is assumed that the user would request assistance with tasks whose results would be used at a later point in time i.e. the user will not wait online for the results. So, system performance is not of immediate concern. However, when the UI is initiated it should be come up fairly quickly (<5 seconds).

## Security Requirements

It is assumed that the user provides a storage location to the system with appropriate access permissions for the user, so the system itself has no requirements in this area. The system will not return results in any file formats which are known to harm the user’s device (\*.zip etc.).

## Other Quality Attributes

Time and size constraints will be established so the assistant doesn’t run forever. There will also be termination points defined in case the system isn’t able to find any results or finds too many results. The quality of the results would depend somewhat on the quality of the request i.e. if valid terms from the natural language (English) were used.

## External Interface

### User Interface

The user interface will be web-based and built in HTML. For this project, it will be a basic one and will be form-based. The user enters a request for information and a link to the location where results should generally be stored, the system sends a short summary of the results it gathered – the number of documents returned, the search terms used and a link to ranked documents so user can review these first.

Once results are returned, the user can provide feedback for the operation. The UI will also provide a link for feedback. The system will maintain a history of requests so the user can provide feedback when they are ready.

### Storage Interface

The storage interface is inbuilt in the technologies used to develop the system. The storage resources will be provided by the user. Eventually, the document location would be world-wide web in which case web interfacing must be taken into consideration. But for the scope of this project, the document repository will also be provided by the user.

# Functional Requirements and Design overview

The following would be main functional requirements of the CRA –

1. Provide a visual interface to the user for interaction
2. Accept a natural language request from the user
3. Parse and interpret the input.
4. Initiate a search of the main search term in the input.
5. Identify relevant results and download them to the result store
6. Return a message with the summary of the operation and a link to the results
7. Ask/remind the user for feedback
8. Incorporate received feedback into its processes (performance) and evolve

Design overview:

This system will be using a client – server model with the UI being the client and the actual processing system residing on the server. The User Interface will be developed in HTML and Javascript will include calls to Python. Python will act as the processing component behind the web server. For the purposes of this project, both the web server and the client will be hosted on the local machine. This may be the paradigm for the actual application if one was to be developed outside of this course.

## Design approach for component : Natural Language Interpreter

ID: S01

Component: Natural Language Interpreter

Description: Parse and interpret the input

This component will be developed in Python. The decision to use Python was made due to its rich and extensive libraries available for Natural Language. It is also easily interfaced with storage and the UI. The NLTK libraries and any other NLP libraries available will be used as needed. The output of this stage will be input to component S02.

## Design approach for component: Search, Ranking and Download

ID: S02

Component: Search, Ranking and Download

Description: Initiate a search of the main search term in the input. Identify relevant results and download them to the result store

This component will be developed in Python. The decision to use Python was made due to its capabilities in web scraping, search and ranking. It is also easily interfaced with storage and the UI. This component involves search and more importantly, ranking based on relevance. This has already been achieved by others in the Python community and some of the code is available publicly for use. This project will leverage what is already available to use legally and will build upon it. The libraries to be used will be determined and finalized before project delivery as this depends greatly on contextual performance.

The download/copy functionality available in Python will be used to store the obtained results.

The output of this step will be passed on to component U01.

## Design for component: User interface

ID: U01

Component: User Interface

Description: Provide a visual interface to the user for interaction, Accept a natural language request from the user, Return a message with the summary of the operation and a link to the results, Ask/remind the user for feedback

This component will be developed in HTML and Javascript. The landing page of the UI will contain two links – Send a Request and Give Feedback. If the user clicks on ‘Send a Request’, the user will be redirected to a form-based page and will receive the natural language request from the user. Once the initial request has been sent, the page will transition to a ‘chat window’ in which the results will appear. The results link and operation summary will be sent back in Natural language. After this message, a request/reminder for feedback will be sent to the user.

If the user chooses to provide feedback, there will be a feedback link in the HTML positioned below the chat window. Alternatively, the user can choose to give feedback at a later point in time and click on ‘Give feedback’ on the landing page to give feedback.

The feedback page will contain pre-decided questions which could potentially affect performance and the user will be able to give a rating for each question on a scale of (1-5) or from ‘very satisfied to very dissatisfied’ or similar. Only one rating can be chosen for each question.

## Design for component: Feedback

ID: S03

Component: Feedback

Description: Incorporate received feedback into its processes (performance) and evolve

Once received, the ratings are stored along with the corresponding contexts, search terms. This stored ratings will then be used to adjust system parameters which will decide what keywords are used in the search(NLP component), whether the search should be narrowed or broadened, how the results are ranked (search and ranking), how many results are returned and improvement from past performance.

This part is still under thought.