Title: *CA4010:* Search Technologies - Research and Development Project Report

Course: Computer Applications and Software Engineering *Year 4*

Students:

Festus Ikenna:	14383451	ikenna.ejike2@mail.dcu.ie
Michael Collins:	16313231	michael.collins38@mail.dcu.ie
Nigel Guven:	14493422	nigel.guven2@mail.dcu.ie
Shaun Carey :	16450454	shaun.carey26@mail.dcu.ie

Word Count: 3319

Date of Completion: 12th December 2019

Date of Submission: 17th December 2019

Table Of Contents

1. Disclaimer

A report submitted to Dublin City University, School of Computing for module CA4009: Search Technologies, 2019/2020. I understand that the University regards breaches of academic integrity and plagiarism as grave and serious. I have read and understood the DCU Academic Integrity and Plagiarism Policy. I accept the penalties that may be imposed should I engage in practice or practices that breach this policy.

I have identified and included the source of all facts, ideas, opinions, viewpoints of others in the assignment references. Direct quotations, paraphrasing, discussion of ideas from books, journal articles, internet sources, module text, or any other source whatsoever are acknowledged and the sources cited are identified in the assignment references.

I declare that this material, which I now submit for assessment, is entirely my own work and has not been taken from the work of others save and to the extent that such work has been cited and acknowledged within the text of my work. By signing this form or by submitting this material online I confirm that this assignment, or any part of it, has not been previously submitted by me or any other person for assessment on this or any other course of study.

By signing this form or by submitting material for assessment online I confirm that I have read and understood DCU Academic Integrity and Plagiarism Policy (available at: http://www.dcu.ie/registry/examinations/index.shtml)

Name(s):

Date: 11th December 2019

2. Abstract

This report is written and designed to detail a specification for an E-Commerce Chatbot search system.

It will employ innovative AI search mechanisms in order to understand and evaluate user requirements, empowering business-client communications and marketing, and provide the means to invigorate online business solutions.

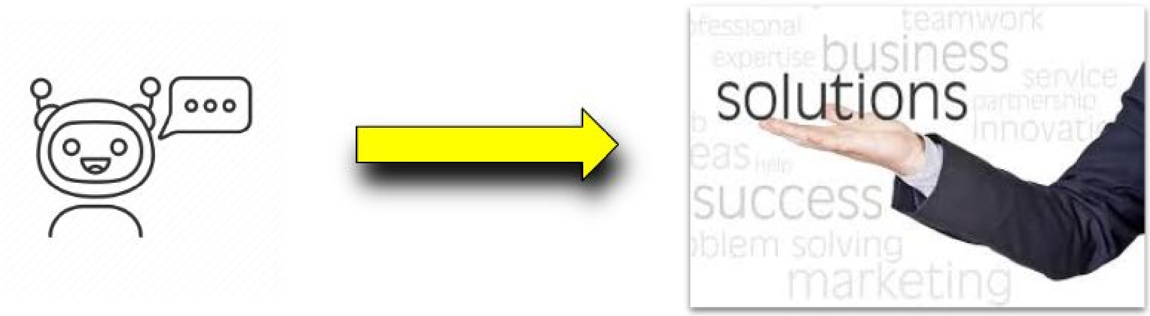
The system, in short, will allow the user to ask a chatbot any questions that they may have regarding a product. The bot will then use a combination of semantic search and recommender systems to provide answers or solutions to the user's query.

We will explore user analysis, and have developed ideas for implementation and evaluation, along with an example of a mock-up interface, all of which will be further detailed in this report.

3. Introduction

The motivation behind this chatbot search system came from our combined love of online shopping, and realising that a system such as this may actually be extremely helpful to e-commerce customers, whilst being quite an unfilled niche at the moment. Currently, if you come across a product online that you are interested in, you are forced to read through a plethora of product descriptions, reviews, and other spread out information, to eventually find what information you may be looking for. Very often, users are opening separate windows entirely to look up information regarding the product.

The aim of this system is to cut that time spent searching at least tenfold, while also halting the need to navigate through multiple different windows to finally find what you may need. If you have a question on a product, you may simply ask the embedded chatbot, which will then give you an answer using its underlying search technologies. It is quite similar to getting an answer to a specific question from a staff member in-person.



We plan on using quite a wide range of search technologies in this system. Natural language processing will be carried out to break down user queries into machine readable queries, which will be explained in more detail later. In addition, a combination of semantic search and Okapi BM25 will be used to find relevant answers to these aforementioned queries. Finally, we plan on having a recommender system in place to predict trends and then recommend perhaps more relevant items to the user based on their queries and item history.

A small constraint for us may include the fact that whilst getting swift and accurate answers to queries will be a huge benefit, there would still be a lack of actual human interaction at the business end to address more serious issues such as damage to an item. Companies may feel as if there won't be as much need for basic customer support staff should a system like this become popular, thus users may be waiting for a longer time for support should they have a bigger issue which requires human interaction.

4. User Analysis

Naturally, it will be the users that shape our product over time. With such an intuitive design, we wanted the chatbot to be accessible by all types of users. Given that we aim to have our product be able to be injected into any e-commerce website to increase usability, we have to assume our target audience to be as wide and diverse as possible. Our audience will be dependant upon the target audience of the platform(s) that we will be hosted on. To refine the target audience we will have to do so on a case by case basis.

To get a picture of this, below is a sample of the target audience for Amazon's e-commerce platform:

Type of segmentation	Segmentation criteria	Amazon target customer segment
Geographic	Region	More than 100 countries
	Density	Urban and rural
Demographic	Age	18 and older
	Gender	Males & Females
	Life-cycle stage	Bachelor Stage young, single people not living at home Newly Married Couples young, no children Full Nest I youngest child under six Full Nest II youngest child six or over Full Nest III older married couples with dependent children Empty Nest I older married couples, no children living with them Empty Nest II older married couples, retired, no children living at home Solitary Survivor I in labour force Solitary Survivor II retired
	Occupation	Students, employees and professionals

fig. 1.1

From the above data we can estimate that we have a very wide target audience. With this comes drawbacks and advantages. The main disadvantage would come with designing the user interface and experience. As our users can range from 18 years old and upwards, we can assume that we would be dealing with people with widely ranging technical capabilities and thus we would need to keep functionality simple and streamlined. With this comes the further drawback of extra time and capital needing to be spent on user research and testing. However there are a number of advantages, namely have a more total customers and a larger safety net. A higher number of customers means a higher ROI, when this is paired with a larger safety net due to there being more potential customers it allows for some flexibility in product design and especially during the early stages and during design refinement.

With all of this knowledge in mind, we decided upon using a well-known and intuitive style of interface. We are all familiar with Facebook, and more specifically its chat window when in the browser. Users of all ages and backgrounds can interact with a window like this with relative ease. Given the design of the back-end of our product, the user can interact with our service as naturally as talking to a friend.

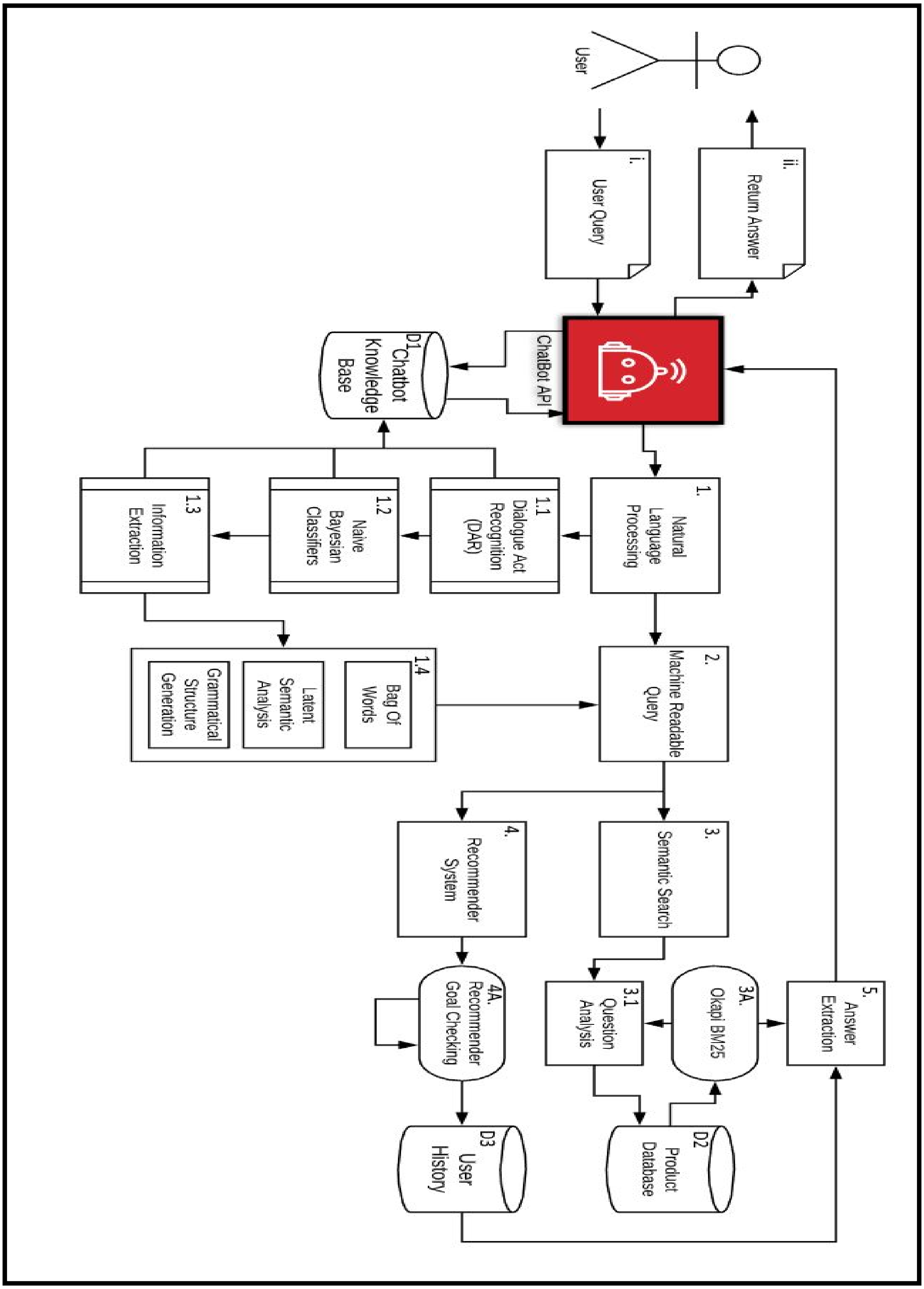
The objective of the system from the user's perspective will be to streamline the user's interaction with the e-commerce platform, from browsing for a product up until the purchasing stage. The system should be able to answer questions about a product that the user may have. These questions may range in complexity from what colours a particular phone comes in to what specification a certain laptop has and what other laptops have similar specifications within the same price range.

The system should also fluidly suggest similar products if the user is not showing the desired amount of interest or suggest products based on previous search history if the user seems totally unsure on what they would like to purchase. This functionality should function organically and not behave too clunky, thus we think less is more in this context. Given the interface that the user is interacting with we do not want the chat to be bombarded with information which may not be important to the user while still maintaining engagement with the platform.

The requirements of the system should be to display particular items that are available on the host platform, and to display relevant information about such products when asked by the user. The system should also be able to answer abstract questions within the scope of the platform's products and services. The system should also be able to interact with the shopping cart if necessary, whether this be adding/removing items, or suggesting deals that are present on the platform. This last one is important as it can increase customer satisfaction and thus increase their retention, benefitting both the user and the host business, which ultimately benefits our system and us.

5. Scientific Functional Description

5.2 High Level System Architecture



5.2 Functional Architecture Description

5.2.1 Natural Language Processing

Natural Language Processing plays a big role in the chatbot system. It breaks down user queries into machine readable formats. Once a Chatbot receives a query from a user, it processes the query using NLP. The NLP processor we have designed is made up of four phases to generate a machine readable packet. They are listed below in order of how a query is processed:

1.1 Dialogue Act Recognition

Dialogue Act Recognition is utilised to determine the function of the text or sentence that a user has inputted as a query.

In dialogue act recognition systems, a corpus of sentences (training data) is labeled with the function of the sentence, and a statistical model is built which takes in a sentence and outputs its function. (Chatbot: Architecture, Design and Development,[pp:9,10] Jack Cahn, 2017).

This essentially means that a DAR system receives the query and analyses it as, for example a open/close ended statement, a question or an opening phrase. Our idea here is that the DAR creates a table with which the next step can examine.

1.2 Naive Bayesian Classifiers

Similar to data mining approaches, Bayesian classifiers cross-examine the table created by DAR with all stored query data inside of the Chatbot database. The naive Bayesian classification examines the probability of a query with every possible sequence of Dialogue acts.

The idea behind using a Bayesian approach to DA models is to find the probability of every possible sequence of dialogue acts DA that could represent a sentence or utterance and find the dialogue act sequence with the highest probability of occurring (Chatbot: Architecture, Design and Development,[p:11] Jack Cahn, 2017). The formula for this is displayed below:

$$DA_{Max} = argmax_{DA}\, P(DA|U) = argmax_{C}\, \frac{P(DA)P(U|DA)}{P(U)}$$

$$DA_{Max} = argmax_{DA}\, P(DA) * P(U|DA) \text{ [16]}$$

1.3 Information Extraction

Once a query has been assigned a probability, the chatbot must be able to examine the intent of a query by understanding the actual words inside the queried sentence. Of course, machines do not understand the actual meaning of words, rather they understand types and descriptions of words. The goal is to develop a grammatical structure which focus on the relationships of words and store it all in the chatbot database so that the chatbot may learn simultaneously.

There are three steps to creating a relational dependency grammar and they are listed below:

A. Bag of Words:

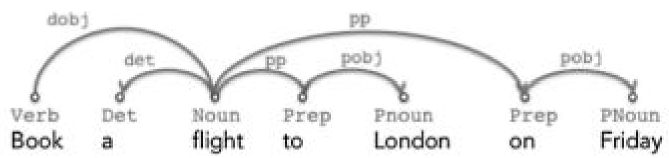
Sentence, structure, order and syntax must be ignored for this step. A table of words with all of their frequencies are stored inside this 'bag'.

B. Latent Semantic Analysis:

In Latent Semantic Analysis, we create a matrix from the bag of words where each row represents a unique word, each column is a query and the value of the cell in question is the frequency of the words in the document. This refines the bag of words and enables the creation of grammatical data structures

C. Grammatical Data Structure Generation:

Sentences and utterances can be stored in a structured way in grammar formalisms such as context-free grammar (CFGs) and Dependency grammars (DGs). For the context of this project we assume we will be developing Dependency grammars as these focus on the relationship between words(Chatbot: Architecture, Design and Development,[p:14] Jack Cahn, 2017). An example of this is listed below:



5.2.2 Okapi BM25

In finding items related to a search query, the chatbot search engine must be able to find relevant content inside of a vast array of products. With the combination of semantic query analysis and Okapi BM25, our aim is to find relevant items that are specific to a user query. Through IR-based Question Analysis received from the NLP ALgorithm, which is knowledge based, items are retrieved from the vast list of products.

These are sent through an Okapi BM25 algorithm which reduces the set of retrieved items into a ranked list of relevant items. Obviously the items at the top of the list are the items that the user is looking for. However, our plan is to implement a minimum confidence threshold for the items so that if items do not support this threshold, then the items will not pass through to answer extraction.

As an alternative to TF-IDF, Okapi BM25 weighting scheme follows a probabilistic approach to assign a document with a ranking score given a query. We propose an adaptation of such a model by assigning each tag with a score (weight) given a certain item. Our third profile model has the following expressions:

$$u_{m,l} = bm25_{u_m}(t_l) =$$

$$= \frac{u_{m,l} \cdot (k_1+1)}{u_{m,l} + k_1\left(1-b+b \cdot {|u_m|}/{avg(|u_m|)}\right)} \cdot iuf(t_l),$$

$$i_{n,l} = bm25_{i_n}(t_l) =$$

$$= \frac{i_{n,l} \cdot (k_1+1)}{i_{n,l} + k_1\left(1-b+b \cdot {|i_n|}/{avg(|i_n|)}\right)} \cdot iif(t_l) \, ,$$

Where b and k1 are set to the standard values of 0.75 and 2 respectively (Content-Based Recommendation in Social Tagging Systems, [p:238], Ivan Cantador and Alejandro Bellogin, 2010).

5.2.3 Recommender System Goal Checking

Recommender systems are powerful analytical devices that predict trends of items that may be of interest to a user. Recommender systems are now a common module in many E-commerce websites like Amazon and Ebay where they provide users with items similar to those a user may have bought before. We require the chatbot to be able to access user transaction history in order to deduce if a user may require a similar item.

In order to perform a decisive analysis of a users history and find relevant items, the recommender system must satisfy four goals. The goals of an operational RS are typically:

- Relevance: recommend items relevant to the user
- Novelty: recommend new items relevant to the user which they have not seen before.
- Serendipity: recommend unexpected or surprising items which the users find relevant
- Diversity: if a diverse list of items are recommended, there is a greater chance one of them will be relevant to the user.

(Recommender Systems slides,[p:5] DCU Loop, Gareth Jones, 2019)

We propose that our recommender system must extract an item and has some degree of confidence that an item is of relevance, satisfies these goals and then sends them as a list of items to the user. How the recommender system will decide if an item satisfies these goals is displayed below.

In the case of a correlation based algorithm prediction on product ‘P’ for customer ‘C’ is computed by calculating a weighted sum of co-rated items between C and all his neighbours and then by adding C’s average rating to that. This can be expressed by the following formula:

$$C_{\mathrm{P}_{pred}} = \overline{C} + \frac{\sum_{J \in rates} (J_{\mathrm{P}} - \bar{J}) r_{CJ}}{\sum_{J} |r_{CJ}|}$$

(Application of Dimensionality Reduction in Recommender Systems - A Case Study, [p:6], Badrul Surwar and George Karypis).

5.2.4 Analysis of Design

- The technological algorithms are well suited to the whole design of this chatbot. NLP and semantic search are fundamental modules of chatbot and overall artificial technology.

- The system was originally planned to function like a data warehouse. We decided to change the architecture to better match an embedded Chatbot API over the presentation layer of a web interface.

- With regards to the recommender system, it is not as necessary to be included in the design and can be removed if there are important time and memory constraints. However, the inclusion of the recommender system does increase the likelihood of user interaction with the chatbot interface and therefore we believe it is fundamental to the overall UI experience.
- Some limitations we expect regards to the accuracy of the algorithms we have chosen. Perhaps BM25 might not be suitable for this type of project whereas a vector space model might be convenient. We do not know as we are not developing an actual implementation.

- We assume that products on a website database are made of information cards. This is so our semantic search algorithm can examine the items in the database and retrieve content efficiently.

6. Evaluation

Unique to this project, evaluation metrics need to take into consideration both the underlying algorithms as well as user satisfaction with the chatbot interface. The less technical user interaction can be measured using bot analytics provided by 'Botanalytics' whose software provides statistics and user feedback mechanisms which can all help in order to improve the functionality of the chatbot. Important evaluation metrics we must examine are user retention and informative engagement.

Botanalytics is a conversational analytics tool focused on analysing engagement and retention measurement for chatbots. It's available for bot on many conversational UI platforms, including Facebook Messenger, Slack, Amazon, Kik and more. Developers need chatbot analytics to improve their bots, so we empower them to learn more about their user's behaviour with easy-to-use tools (https://botanalytics.co/what-is-botanalytics).

We plan to examine the performance of our three main algorithms individually. In this way we can fix each algorithm independently of the others and therefore make it easier to refactor changes to the modular architecture without disturbing the overall infrastructure.

For Natural Language Processing, we can utilise the highly favourable BLEU metric. The Bilingual Evaluation understudy evaluates the quality of text which has been machine translated from one machine to another. Quality of translation is a major goal from the algorithm. We plan to use test sentences to feed into the chatbot and BLEU should analyse the outputted packet that is received from the chatbot API.

The BLEU metric scores a translation on a scale of 0 to 1, but is frequently displayed as a percentage value. The closer to 1, the more the translation correlates to a human translation. Put simply, the BLEU metric measures how many words overlap in a given translation when compared to a reference translation, giving higher scores to sequential structures of words. (https://kantanmt.com/whatisbleuscore.php).

Our recommender system must be able to achieve its goals. In order to evaluate the system, we realise that the approach must view recommendation as a ranking of relevant items. One such algorithm we may use is Normalised Discounted Cumulative Gain or NDCG.

A recommender system may return some items and as we are trying to evaluate it, we need to compute how good the list is in terms of relevant, interesting items. Each item is assigned as relevance score which is a non-negative number. Given that there are many products in a commercial database, this might be tricky to implement. This non-negative number is the gain. We add up each of the scores, the cumulative gain.

In order to view the most relevant items, we may sum the gains by a logarithmic function which is discounting. These new values are not comparable for a single user so we normalise the values to retrieve a set of numbers with the closer-to-zero items being less relevant. The higher the score the more relevant the item. (http://fastml.com/evaluating-recommender-systems/)

The formula for Normalised Discounted Cumulative Gain is as as follows:

$$DCG_n = \sum_{i=1}^{n} \frac{rel_i}{\log_2^{i+1}},$$
$$NDCG_n = \frac{DCG_n}{IDCG_n},$$

(https://www.mathworks.com/matlabcentral/fileexchange/65570-normalized-discounted-cumulative-gain-ndcg)

7. Conclusion

To conclude, with over 2.5 billion people using instant messaging platform (i.e Whatsapp, Viber, Messenger, etc.), it has become increasingly important for E-commerce websites (such as Amazon and Ebay) to utilise the effectiveness of a chatbot and adapt it into their platform.

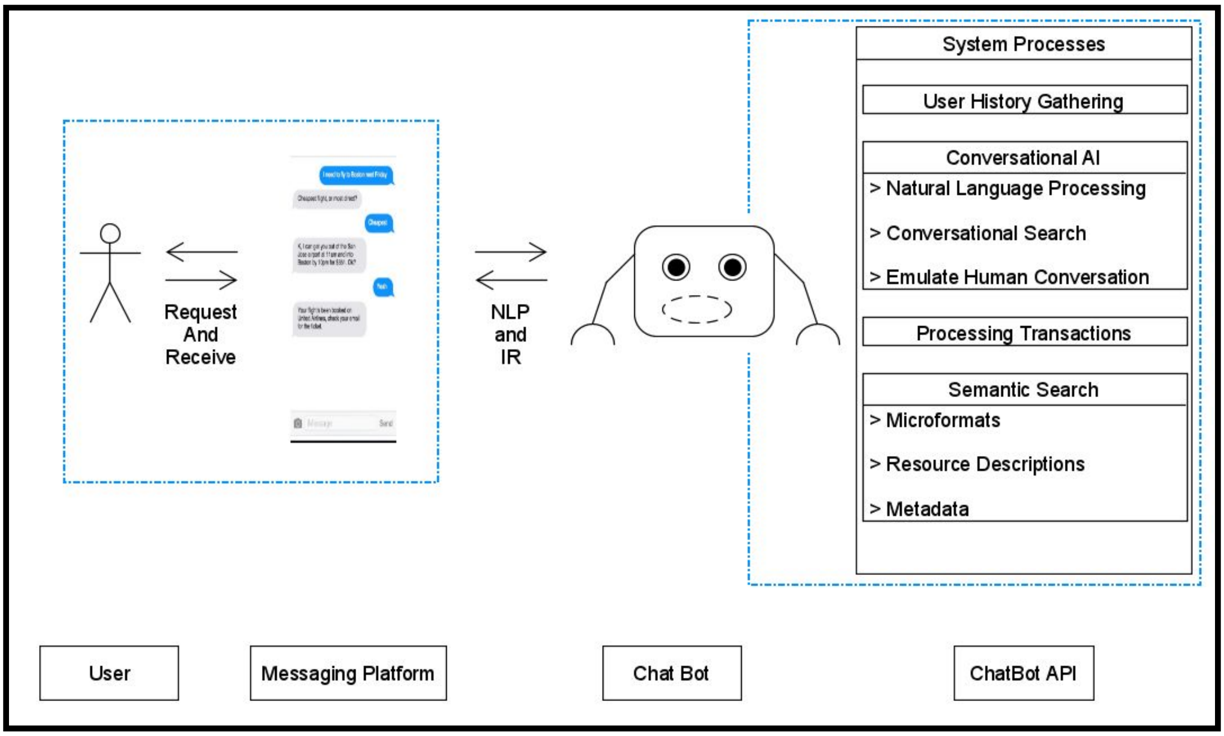
According to a Facebook survey, **50%** of customers more likely to buy from a business that uses chat implementations and only **5%** of online retailers use chatbots. By 2020 it is proposed that **80%** of people will be using chatbots (according to Oracle Survey).

This report has outlined all key components of our unique E-Commerce Chatbot, in conjunction with addressing the needs of online shoppers and e-commerce enthusiasts (main target audience). Examined in this report are the following:

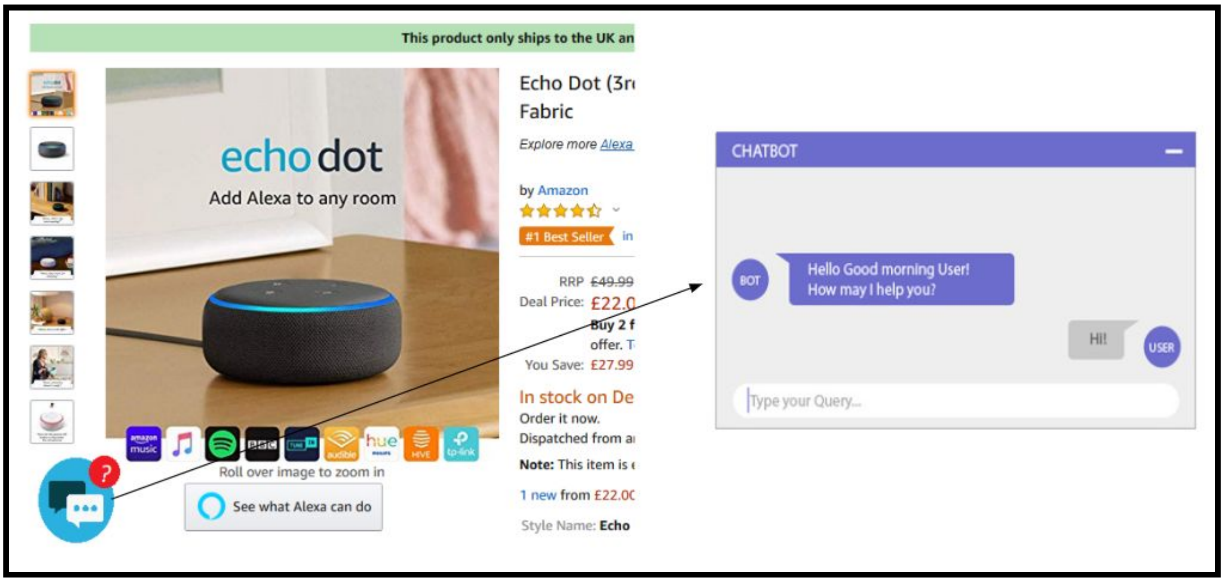
- ➢ The motivation and inspiration behind the project.
- ➢ User interaction with the system (with the use of real-world examples).
- ➢ A brief explanation and diagram of the Architectural design of the overall system using a High Level System Architecture whereby Natural Language Processing is utilised to break down user queries into machine readable format, Dialogue Act Recognition to break down a sentence or phrase to determine it's the function, Bayesian Classifiers (works in conjunction with Dialogue Act Recognition) in order to determine the probability of a query, Information Extraction to develop a grammatical structure and store the resulting relationship in the chatbot's database (which includes the three ways used to create relational dependency grammar), the use of Okapi BM25 weighting scheme to locate relevant contents and finally, powerful analytic devices such as the Recommender System Goal Checking to predict trends of items.
- ➢ The use of Bot Analytics (provided by Botanalytics) to measure user feedback and provide statistics on user interaction and the utilisation of the powerful BLEU metric for Natural Language Processing to evaluate the quality of text translated between computers and aid in reliability and efficiency of the chatbot.
- ➢ All components listed above have been analysed in detail throughout the report to ensure reliability, durability and efficiency of the overall chat platform, along with ensuring a successful project.

8. Appendices

8.1 High Level Design



8.2 Mock Up Interface



9. References

1. Anatomy of an ECommerce Search Engine
   Seminar by TechJam

2. ChatBot: Architecture, Design and Development
   Thesis by Jack Cahn

3. SuperAgent: A Customer Service Chatbot for ECommerce Websites
   Microsoft research provided by Lei Cui, Ming Zhou and Chaoqun Duan

4. Modern Information Retrieval: Concepts and Technology Behind Search
   Book by Ricardo Baeza Yates and Berthier Ribeiro-Neto

5. Application of Dimensionality Reduction in Recommender Systems - A Case Study Technical Report by Badrul Surwar and George Karypis

6. Content-Based Recommendation in Social Tagging Systems
   Research Thesis by Ivan Cantador and Alejandro Bellogin

7. Recommender Systems
   Search Technologies Presentation by Gareth Jones

8. https://www.mathworks.com/matlabcentral/fileexchange/65570-normalized-discounted-cumulative-gain-ndcg

9. http://fastml.com/evaluating-recommender-systems

10. https://kantanmt.com/whatisbleuscore.php

11. https://botanalytics.co/what-is-botanalytics

12. Fig. 1.1
    https://research-methodology.net/amazon-segmentation-targeting-positioning-widest-range-target-customer-segment/

-- END OF REPORT--