INSTITUTO SUPERIOR DE ENGENHARIA DO PORTO
MESTRADO EM ENGENHARIA INFORMÁTICA




Chatbot de Suporte para Plataforma de Marketing Multicanal

LUÍS ANDRÉ DA SILVA MADURO FERREIRA
Outubro de 2019

POLITÉCNICO
DO PORTO

Instituto Superior de Engenharia do Porto

Multichannel Marketing Platform Supporting Chatbot System

Luís André da Silva Maduro Ferreira

A thesis submitted in fulfillment of the requirements
for the master's degree in Informatics Engineering

Advisor: PhD Ana Madureira

Supervisor: PhD Ivo Pereira

Porto, October 13th, 2019

Abstract

E-goi is an organization which provides automated multichannel marketing possibilities. Given its system's complexity, it requires a not so smooth learning curve, which means that sometimes costumers incur upon some difficulties which directs them towards appropriate Costumer Support resources. With an increase in the number of users, these Costumer Support requests are somewhat frequent and demand an increase in availability in Costumer Support channels which become inundated with simple, easily-resolvable requests. The organization idealized the possibility of automating significant portion of costumer generated tickets with the possibility of scaling to deal with other types of operations.

This thesis aims to present a long-term solution to that request with the development of a chatbot system, fully integrated with the existing enterprise modules and data sources. In order to accomplish this, prototypes using several Chatbot management and Natural Language Processing frameworks were developed. Afterwards, their advantages and disadvantages were pondered, followed by the implementation of its accompanying system and testing of developed software and Natural Language Processing results.

Although the developed overarching system achieved its designed functionalities, the master's thesis could not offer a viable solution for the problem at hand given that the available data could not provide an intent mining model usable in a real-world context.

Resumo

A E-goi é uma organização que disponibiliza soluções de marketing digital automatizadas e multicanal. Dada a complexidade do seu Sistema, que requer uma curva de aprendizagem não muito suave, o que significa que os seus utilizadores por vezes têm dificuldades que os levam a recorrer aos canais de Apoio ao Cliente. Com um aumento de utilizadores, estes pedidos de Apoio ao Cliente tornam-se frequentes e requerem um aumento da disponibilidade nos canais apropriados que ficam inundados de pedidos simples e de fácil resolução. A organização idealizou a possibilidade de automatizar uma porção significativa de tais pedidos, podendo escalar para outro tipo de operações.

Este trabalho de mestrado visa apresentar uma proposta de solução a longo prazo para este problema. Pretende-se o desenvolvimento de um sistema de chatbots, completamente integrado com o sistema existente da empresa e variadas fontes de dados. Para este efeito, foram desenvolvidos protótipos de várias frameworks para gestão de chatbots e de Natural Language Processing, ponderadas as suas vantagens e desvantagens, implementado o sistema englobante e realizados planos de testes ao software desenvolvido e aos resultados de Natural Language Processing.

Apesar do sistema desenvolvido ter cumprido as funcionalidades pelas quais foi concebido, a tese de mestrado não foi capaz de obter uma solução viável para o problema dado que com os dados disponibilizados não foi possível produzir um modelo de deteção de intenções usável num contexto real.

Index

Figure Index

Table Index

Acronyms

AHP	Analytic Hierarchy Process
ANN	Artificial Neural Networks
BOW	Bag-Of-Words
CRUD	Create, Read, Update, Delete
FFE	Fuzzy Front End
LSTM	Long Term-Short Memory
NER	Name Entity Recognition
NLP	Natural Language Processing
NLTK	Natural Language Toolkit
PCA	Principal Component Analysis
RNN	Recurrent Neural Networks
SMS	Short Message System
SVM	Support Vector Machine
SWOT	Strengths, Weaknesses, Opportunities, Threats
TFIDF	Term Frequency-Inverse Document Frequency

1 Introduction

Following an increased migration of economic activity to the internet, several businesses and systems have been created to provide services for the growing number of web-based operations. Learning to use these systems, as the learning process for any other, can be difficult and time consuming. If the system has a steep learning curve, the resulting predictable user experience may cause him/her to outright give up on the opportunity. This is the problem which lead E-goi to resort to automated processes as a solution to simple and often repeated customer issues. Such a system would require extensive research and experimentation into available frameworks and their use in an intelligent and incorporated chatbot system. This thesis aims to design and implement such a system which is easily usable and maintainable at the corporate level.

1.1 Context

Given the explosion in internet user numbers and internet connected devices, observable since the introduction of the World Wide Web in the 1990's, a lot of businesses and professionals connected to the practice of marketing have shifted towards e-marketing and away from the concept of traditional marketing. E-marketing became largely held as the practice of advertising and selling products through the internet. Even if both paradigms have similar intents of predicting and meeting their costumers' needs and expectations, e-marketing can make a few good arguments for the ever-growing amount of business presence in the digital world. E-marketing is usually cheaper and easier to plan, is capable of reaching a higher number of potential customers and of doing so at a faster rate (although it can be said that user data leaks are far less likely to happen in traditional marketing) [1]. The main factor in the mass enterprise acceptance and focus on e-commerce stems from its ease of incorporation into an enterprise's system and data-flow.

Parallelly, with the aforementioned increased connectivity between humans and internet, new avenues of possible costumer interaction and presentation were created, like social networks [1]. Modern marketing became far more than just brand, service or product promotion. A more current definition would include concepts like the active brand management, direct customer interaction, data and statistic collection and trend prediction.

E-marketing became the basis for costumer nurturing and fidelity. Incorporation in a computer system allows for the recording and access of user activities. Statistical and data elaboration routines are implemented to support these routines and to give a better notion of what is the best step to take next, always to maximize user satisfaction and comfort. As a result, there is an increased competitivity in the area of marketing. It is becoming necessary to reach to more customers, quicker and with a more personalized touch to survive as a business. Marketeers and executives are therefore investing more and more into internal or external computer and information systems.

This environment resulted in opportunities for some enterprises like E-goi, who provides a multichannel platform for customizable and automated business-to-client marketing messages.

1.2 Organization

The project was funded and created by a Matosinhos-based organization, E-goi. E-goi created by Miguel Gonçalves and currently standing as the leading Software as a Service multichannel marketing automation platform in Portuguese territory.

In order to distribute and propagate client defined marketing campaigns and communications, E-goi employs an array of communication channels which include email, sms, voice audio messages, push notifications and soon Facebook Messenger and Whatsapp. Access and utilization to the platform is given according to the following list of possible plans [2]:

- 5001 - If a user's contact base does not exceed 5001, the user is free to utilize E-goi E-mail Marketing and Marketing Automation functionalities free of charge. In this plan, up to 15,000 emails are allowed per month;

- Starter Plan - Allows up to 10,000 unique contacts and now allows for unlimited push notifications and emails;

- Pro Plan - Allows up to 100,000 unique contacts and improved resource utilization and customer service options;

- Corporate Plan - This broad paradigm requires discussion of the client's necessities and of resulting price.

As noticeable by the variety of scope between the plan options, E-goi seeks to provide intuitively accessed and utilized solutions to clients of all scopes, from large corporations to micro-enterprises [3]. Currently, E-goi has over 380,000 registered clients from all over the world, including from native country Portugal, Spain, Brazil, Colombia, up to total count of more than 50 distinct countries. Some of the most notorious client enterprises include IKEA, Fnac, Sephora, Altice, Sonae Group, Porto Editora among others.

1.3 Problem

The introduction to a new system can be difficult. The user learning process of a system can be so lagging that user churn is created. Some users often resort to the system's tutorials (if the developer has made them available) but more often than not the main impulse continues to be to resort to the company's customer support channels. In E-goi's case, with the sheer number of users (more than 380,000 accounts with an average of 250 new accounts registering each day, as of February of 2019), the costumer support team is frequently overwhelmed with help requests. The team has verified that most of the user concerns are simple and repeated, with more complex and intricate issues, the ones that might warrant human comprehension interaction, being very rare.

To aid this situation E-goi theorized the development of a chatbot system to automatically solve a large portion of easily solvable customer supports issues (which would substitute the already existing tutorial string-match program). Firstly, a chatbot system can be defined as "artificial intelligence (AI) program that simulates interactive human conversation by using key pre-calculated user phrases and auditory or text-based signals" [4], requiring the study and incorporation of Natural Language Processing [5]–[7]. The necessity for an automated conversational agent increases as the company grows in scale and their daily login number and customer base escalate. Basic requirements would include the chatbot to be completely integrated within the enterprise's system, the satisfactory usage of available knowledge basis, user intent mining and subsequent problem solving or pertinent offer, as well as too be capable or learn from experience (given a user satisfaction inquiry).

Furthermore, given the project's growth potential, the devised solution would require to be easily deployable as a Service in the future, providing chatbot instances for paying customers. The solution can also be required to one day incorporate domain actuation capabilities, in order to provide automated intent-based actions.

1.4 Objectives

In order to improve customer service output and maximize customer retention, E-goi envisioned the research, implementation and deployment of an intelligent, learning and communicationally robust chatbot as a first line of help towards frequent issues felt by the company's customers. This chatbot should be scalable to be multiplatform (able to communicate with the user through multiple channels) and multilingual, as well as completely integrated in the overall enterprise system. The utilized models, frameworks and developed components should be arranged in a modular fashion, facilitating their substitution, testability and overall debugging [8].

The chat-bot's internal structure must guarantee the processing and analysis of the user's message, the accurate detection of the user's intent and mood, the creation of an

adequate response and the external storage of user input, detected intent and generated response.

The complete list of the project's results and objective observable behaviors is as follows:

- Analysis of the state of the art on the scientific areas encompassed by the problem;
- Analysis and experimentation of relevant technologies and frameworks;
- Development and testing of multiple easily-retrainable intent mining models;
- Development and testing of the encompassing module and communication channels;
- Analysis of the existing enterprise systems and subsequent integration;
- Adequate storage of the user's communications (anonymously), detected mood and intent, response and bot evaluation;
- Analysis and evaluation of the system's integration, models and channels.

1.5 Approach

The approach utilized to comprehend the proposed problem, theorize and an implement a viable solution could be divided in the following sequential phases:

1. Bibliographic research regarding the pertinent state of the art with focus on the following topics:
    - Overall study of the definition of Machine Learning and basic techniques;
    - Study of learning techniques and approaches and their applicability to the master's thesis as a possible solution;
    - Study of context-relevant frameworks and commercial examples.
2. Value analysis of the proposed solutions;
3. Gathering and subsequent manipulation of existing data for analysis and for the system's learning process;
4. Analysis of the main decisions required to fully model and realize the proposed solution, according to good practices in software engineering and researched technologies;
5. Implementation of a fully integrated, operational chatbot system prototype;

6. Planification of an evaluation and simulation process to which the developed system would be subjected, including utilized metrics such as the percentage of correct determined intents, number of recognized intents and number of user requests;

7. Appraisal of the success rates and user satisfaction of each utilized algorithm;

8. Gathering and description of encountered setbacks and how they affected the previously designed solution.

1.6 Document Structure

In order to maximize accessibility and user understandability, this thesis has been divided into multiple chapters:

The current chapter provides an introduction (Section 1) to the context and environment of the project. It starts the thesis by providing an explanation of the why the need for such a solution occurred. It follows up by describing the organization that supports the project, its devised problem statement and objectives. Finally, the section presents the approach followed during project realization and this document structure description.

The next chapter consists of a Literature Review (Section 2) used to provide some insight into the scientific areas and underlying topics which comprise the problem's context. For this master's thesis, these topics include machine learning and (applied to) Natural Language Processing. Additionally, an analysis on available tools for machine learning systems, bot-centered user communication and exploratory data analysis shall be presented.

An analysis (Chapter 3) shall then be made on the value and innovation generated by the proposed solution (including possible alternatives). In order to do this, several methods and models shall be used, including: Business Model Canvas [9], Analytic Hierarchy Process (AHP) [10] and Porter's Value Chain [11].

This is followed up by the design section (Chapter 4) which provides different high-level technological alternatives for the problem's solution. Additionally, an analysis shall be made on the available data and data structures utilized on the machine learning processes and exploratory data analysis.

The particularities of the development process shall be detailed in a following chapter (Chapter 5) where each is delimited development stage is listed and explained, culminating in the results of the functional tests made to the final solution.

The master's thesis conclusions are present in Chapter 6 where a mention on Future Work and a personal Feedback on the thesis experience are also made.

2 Literature Review

This chapter includes relevant information regarding the scientific areas which encompass this thesis' problem and through which an adequate solution may be devised. These areas include Machine Learning, Natural Language Processing and chatbot implementations (in both an enterprise and academic context).

2.1 Machine Learning

Although the notion of Machine Learning was introduced by Alan Turing in 1950 [12], the term Machine Learning and its first practical example were only presented in 1959 by Arthur Lee Samuel [13]. In this work, Samuel described the up and coming field of study as a: "*Field of Study that gives computers the ability to learn about without being explicitly programmed*".

Tom Mitchell expanded Samuel's definition with his 1997 work *Machine Learning* [14]: "*Computer programs learns from Experience E with respect from some task T and performance measure, if its performance on T, as measured by P, improves with Experience E.*". As an example, for a spam classification program, the Task T is classifying emails as spam or not, the Experience E are records of spam/not spam emails and their characteristics and the performance P is the number of correctly predicted emails. In this same work Mitchell describes as, similarly, to living biological beings, an increase in available data (Experience) would likely yield better results, provided there would still be room for improvement. Mitchell [15] also suggested three types of self-interactable types of research focuses in Machine Learning. These included the emphases in Task-Oriented studies, emulation of Cognitive capabilities and theoretical analysis.

Machine Learning projects have recently warranted headlines as being capable of winning against the best human players of a game, much like the 1997 defeat of world chess

champion Garry Kasparov at the hands of IBM's Deep Blue [16]. In 2015, AlphaGo [17] managed to defeat a 18 time world champion at the game of Go, and in 2018, OpenAI Five [18] defeated a team of top-rated professional players at the Game of DOTA 2 and AlphaStar [19] won against professional players at the game of StarCraft II.

Essential to a Machine Learning implementation (and to the explanation of their functioning) are the notions of features and labels, which can be described as:

- Feature – Individual measurable characteristic or attribute of the dataset's entries (e.g. in a house price classification problem, based on house age and space, these two are features);

- Label – Attribute utilized to classify an entry of data based on the combination of its features (e.g. in the aforementioned house price example, the predicted final price is the label).

Machine Learning approaches can be classified as being Supervised, Unsupervised, Semi-Supervised or Reinforcement techniques [20]. The following sections shall be dedicated to an overview of Supervised and Unsupervised techniques with a definition of the Semi-Supervised and Reinforcement approaches, as well as the particular subsection of Deep Learning, due to their particular adoption in NLP related tools and frameworks.

2.1.1 Supervised Learning vs Unsupervised Learning vs Reinforcement Learning

This section includes a brief explanation of the Machine Learning Techniques within the Supervised/Unsupervised gradient and within Reinforcement Learning. Additionally, relevant previous academic examples shall be mentioned [20]:

1. Supervised – For a Supervised Learning approach, it is expected that each entry of data has been previously labelled. By utilizing these entries as reference points, each generated model should be able to learn how to currently identify new entries. Supervised Learning problems are divided into two subsections based upon the type of variable it predicts. Classification problems are suited to predict discrete values, while Regression problems ascertain continuous values [21]. Some examples of Supervised Classification problems have been solved resorting to techniques like Support Vector Machines (SVM) [22]–[24], Decision-Trees [25], Random-Forrest approaches [26] or Naïve-Bayes networks [27], [28]. Supervised Regression solutions, on the other hand, are divided by the application of Linear [29] and Ridge Regression [30] techniques;
2. Unsupervised – Unsupervised Learning is based upon the classification of homogeneous data [31]. The objective of Unsupervised Learning is the mapping of entries by their feature values, creating a dynamic where proximity in this representation can be equalled to feature similarity (Clustering). Like Supervised Learning, Unsupervised techniques are utilized in two categories: Dimensionality Reduction (reduce the number of considered features of a dataset by calculating the least relevant ones) and Clustering problems [31]. Some relevant Dimensionality

Reduction techniques have been presented like Locally Linear Embedding [32] and Principal Component Analysis [33]. Clustering techniques include the K-means algorithm [34], Hierarchical Clustering [34], Density-Based Spatial Clustering of Applications with Noise [35] and Mean-Shift Clustering [36];

3. Semi-Supervised – Given that it shares the goal of minimizing the classification error, the Semi-Supervised philosophy is considered as belonging to the supervised category [37]. The difference is that along with a previously labelled sample, these solutions incorporate learning from a larger unlabelled sample. Some previous works refer how classification using unlabelled data in conjunction with a small amount of labelled data can result in a considerable improvement in learning accuracy over Unsupervised Learning, without the time and computational expenditure of Supervised Learning [37]. Some notable examples can include Semi-Supervised solutions based on Neural Network generated models [38], Gaussian Fields [39] and even NLP word vector representation models [40];
4. Reinforcement Learning – In this kind of machine learning, the operational agents attempt to find the optimal way to accomplish a particular goal or improve their performance on a given task. This approach is based on positive or negative reinforcement of the current model's iteration in order to achieve the highest possible final cumulative reward [41]. By incorporating this notion of cumulative reward, the agent relies both on learnings from past feedback and on the exploration of new avenues which outcome is a larger payoff. The study and application of Reinforcement Learning techniques has been applied in several subject areas of computation such as Swarm Intelligence [42], Game Theory [43], Genetic-based Algorithms [44] or Simulation-based Optimization [45].

2.1.2 Deep Learning

The field of Deep Learning has been described as: "allows computational models that are composed of multiple processing layers to learn representations of data with multiple levels of abstraction. These methods have dramatically improved the state-of-the-art in speech recognition, visual object recognition, object detection and many other domains such as drug discovery and genomics. Deep learning discovers intricate structure in large data sets by using the backpropagation algorithm to indicate how a machine should change its internal parameters that are used to compute the representation in each layer from the representation in the previous layer." [46] and as being "our best shot at progress towards real AI" [47]. It can be shortly described as mimicking the neuroscience of biological brain synapses [48] with the utilization of exceedingly large neural networks for classification and feature learning (automatic feature discovery and extraction from unclassified raw data). It does this by achieving a hierarchical learning of features where higher-level learned features are defined in terms of lower-level features [49]. Some deep learning techniques shall now be presented:

- Artificial Neural Networks (ANN) – Artificial Neural Networks have been studied for many years trying to accomplish human-like performance in intelligent tasks like speech and image recognition. An ANN is formed through the interconnection o simple computational elements (Figure 1) [50]. These elements, or neurons, are

interconnected through mathematical weights and they "activate" based on an internal threshold function, the incoming weights of previously activated connected neurons and a neuron-specific bias value. An ANN is typically constituted of three sections (as seen in Figure 1): an input Layer where inputs are introduced, a hidden computation layer (comprised of several sub-layers) and an output layer which transforms the hidden layer activations into an understandable result. A highlightable technique in ANN is Backwards-Propagation, where after the information has run through an ANN, the algorithm goes back through each connector weight and adjusting them based on the error's partial derivative.

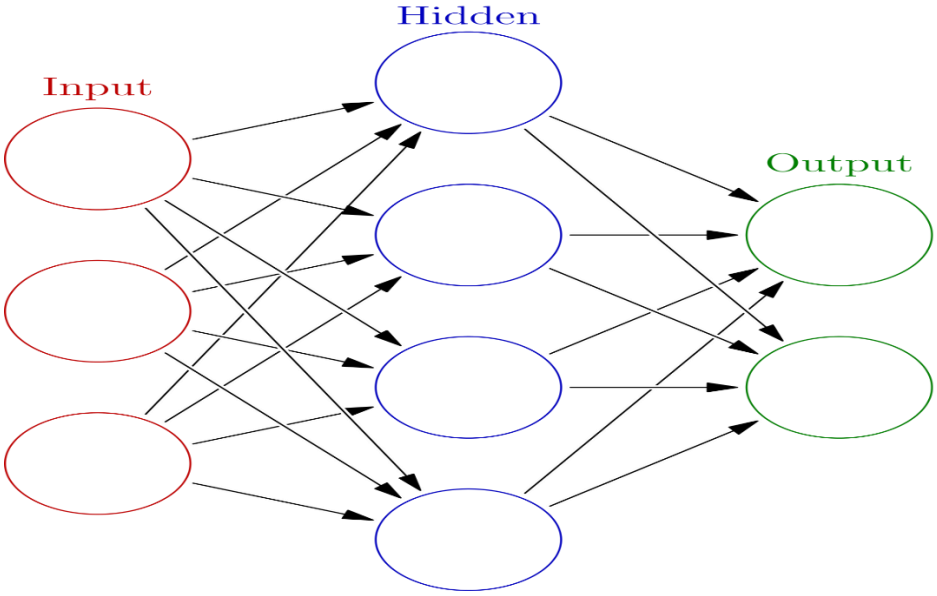

Figure 1 - Artificial Neural Networks Representation [51]

- Recurrent Neural Networks (RNN) – These are a class of ANN where connected neurons form a cyclic graph [52], allowing for the occurrence of information reaching a neuron more than once (Figure 2). Due to this particularity, it generates the notion of" internal memory" in the algorithm as a node loops its output back into the network. This characteristic makes this algorithm ideal for Machine Learning problems with sequential data (e.g. time series, speech recognition, audiovisual recognition).

- Long Short-Term Memory (LSTM) Networks – These are an extension of a RNN where the "memory" (which can be read, written or deleted) of each neuron is extended, making it suitable for learning tasks where long-time gaps may occur between relevant experiences. Each of these memory units decides if it passes, stores or deletes information through its three different gates (therefore called a gated cell), aptly called as input, a forget and an output gate (Figure 3). Over several time steps, these gates determine what information should be added or deleted (forgotten) based on their learned importance (algorithm learns over time to adequately adjust the attribution of importance, meaning that a LSTM learns over time what information is relevant or not).

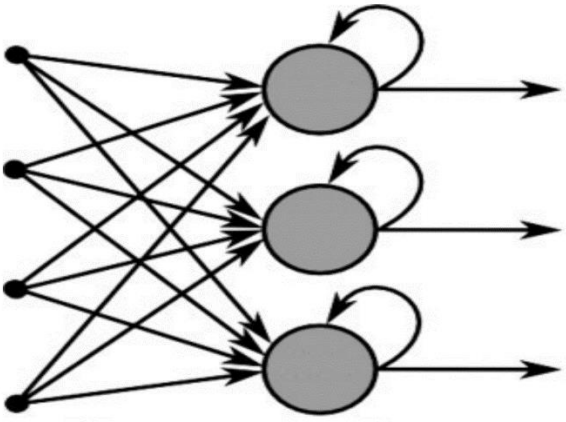

Figure 2 - Recurrent Neural Network Representation [53]

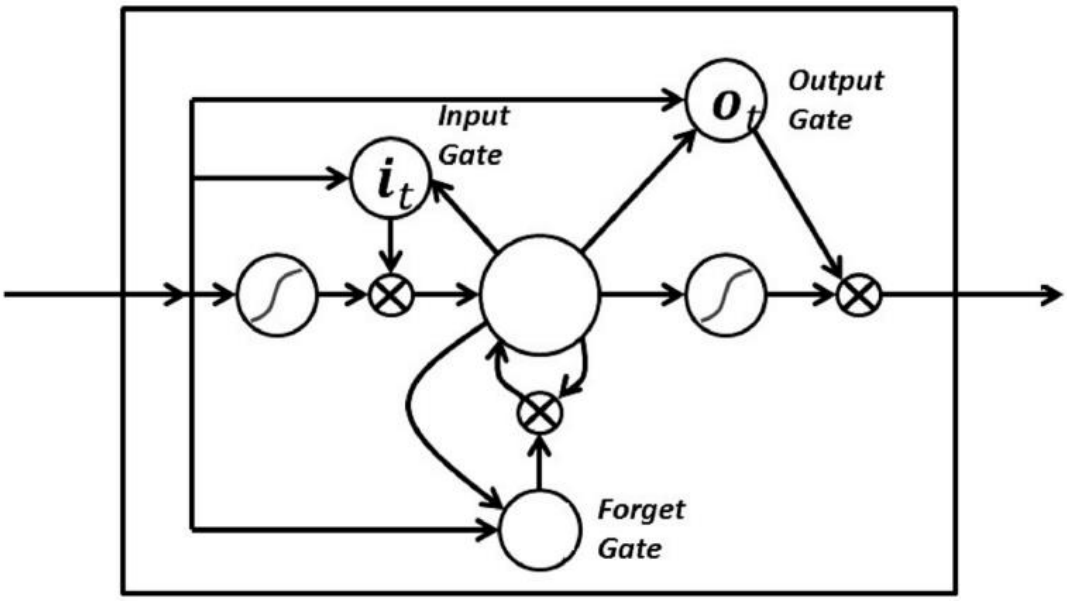

Figure 3 - LSTM gated cell [53]

2.2 Natural Language Processing

Natural Language Processing (NLP) is a field of study that is centered around interactions between human language and computer-based systems. It focuses on efficiently getting computers ever closer to our intuitive understanding of natural language. The notion of Artificial Intelligence based language processing can be attributed to Alan Turing's 1950 publication Computational Machinery and Intelligence [12] where the now called The Turing Test was introduced. The end goal of this test (indistinguishability between a computer and human interaction) could only be attained through some advancements in NLP. Some of these advancements shall be listed in the following section as they are pertinent to the

problem at hand as they could provide a means for the envisioned automated conversation agent to understand and reply to real-world customers:

2.2.1 Tokenization

Tokenization refers to the process of dividing a text string into a sequence of tokens according to its punctuation or blank spaces (to a word or sentence level where languages with different types of spacing require an adaptation of the model) [7]. Usually in the preprocessing phase, tokenization is utilized to reduce noise (everything that isn't a standard letter or number within a certain language) and the substitution/removal of stop words (very common words which are of little value to the meaning of a sentence).

2.2.2 Stemming & Lemmization

Stemming is the process of reducing inflected (or sometimes derived) words to their stem, base or root form—generally a written word form. As an example, the stem of the words: "Stems", "Stemming", "Stemmed", "and Stemtization", the result would be a single word "stem" [54].

Lemmatization is a slight variation of the notion of Stemming. The major difference between these is, that, stemming can often create non-existent words, whereas lemmas are actual words [55]. Examples of Lemmatization are that "run" is a base form for words like "running" or "ran" or that the word "better" and "good" are in the same lemma, so they are considered the same.

2.2.3 Sentiment Analysis

These field of study includes techniques which allow for the identification and quantification of affective states and subjective information. It is accomplished upon training a model to identify and quantify text polarity (positive, negative, neutral) and classify emotional states (e.g. happy, sad or excited). Several factors contribute to have been credited as being responsible for the rise in academic interest in Sentiment Analysis techniques. Some of the factors include the rise of machine learning methods in natural language processing and information retrieval as well as an increase in available training datasets through social media and the World Wide Web [56]. Some commercial interest has been raised over Sentiment Analysis since it allows for the mining on opinions in increasingly competitive global markets [56]. Some solutions have been presented based on Tree Kernel generated models [57], based on word vectorization models [58], [59] and Convolutional Neural Networks [60].

2.2.4 Named Entity Recognition

Name Entity Recognition (NER) is a process which objective is to locate and classify named entities within text. These entities include markable (names of organizations, persons,

locations, date and time mentions and mentions of currencies/percentages) and unmarkable ones (common nous, names of groups and laws derived from a person's name, adjectival forms of location names and non-percentage number usages) [61]. Some notable NER systems include the DARPA-funded MUC-7 system, initially devised to identify entities in military and civil distress reports [61]. Some more recent examples of NER architectures include solutions based on Deep Learning Techniques [62]–[66].

2.2.5 Machine Translation

Machine Translation pertains to the practice of decoding meaning from a source text and re-encode it into the target language, either through direct translation or resorting to an intermediate language (Figure 4). There are rule-based (resorting to pre-organized dictionaries) [67], statistics-based (resorting to statistical methods) [68], hybrid (resorting simultaneously to rule-based and statistical similarity methods) [69], example-based (resorting to analogies in already translated sentences and tokens) [70] and neural-based approaches (resorting to Deep Learning Techniques) [71].

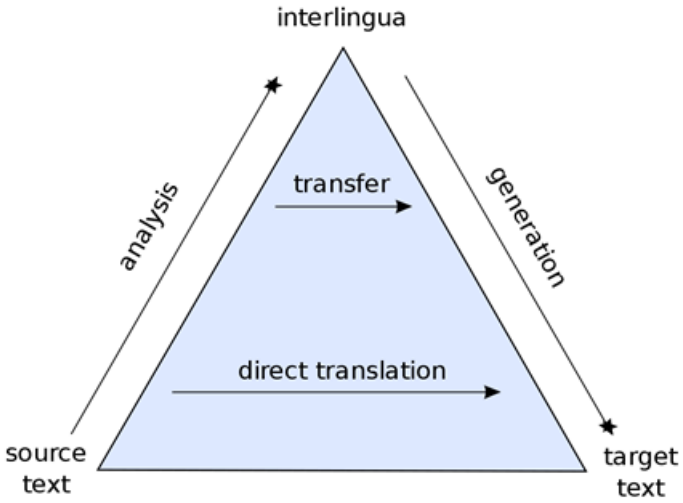

Figure 4 - Steps of Machine Translation [72]

2.2.6 Bag-Of-Words (BOW)

Bag-Of-Words (BoW) is a popular model for representation of text documents and their similarities. Its aims to quantify similarities in meaning by categorizing objects through transformation of the text into a meaningful vector (or array) of numbers [73] (Figure 11). The intuition behind the BoW model is that documents are similar if they have similar content (including word count) thus being likely to have similarities in intrinsic meaning as well. By combining these vectors in a n-dimensional space, some unsupervised clustering algorithms (e.g. K-means) can be applied to sort similar texts.

	a	characters	China	computer	destroyer	in	missile	sequence	south	stored	string	the	U.S.	word
Doc1	1	1	0	1	0	1	0	1	0	1	1	1	0	1
Doc2	1	0	1	0	1	1	1	0	1	0	0	1	1	0

	a	characters	China	computer	destroyer	in	missile	sequence	south	stored	string	the	U.S.	word
Doc1	4	1	0	2	0	1	0	1	0	1	2	2	0	1
Doc2	2	0	2	0	2	1	1	0	1	0	0	4	1	0

Figure 16 - Representation of a Bag-Of-Words Vector [74]

2.2.7 N-gram Representation

An n-gram (previously called shingle) is a contiguous subsequence of a text's tokens. For an n-gram size representation of 4 (4-gram) there is the text example token list [75]:

$$(a, rose, is, a, rose, is, a, rose)$$

Yields as 4-grams:

$$\{(a, rose, is, a), (rose, is, a, rose), (is, a, rose, is)\}$$

Adequate n-gram modelling (utilizing n-grams instead of tokens) can suppress one of the flaws of BoW. Since BoW is an order-less representation of a document's content where only the count of each individual word is considered, it lacks n-grams' representation of added similarity through grammatical ordering.

2.2.8 Skip-gram Representation

Skip-grams are a technique used in the field of speech processing. Similarly to n-grams which are formed by allowing adjacent sequences of words (bigrams, tri-grams, etc.), skip-grams allow tokens to be "skipped" [76]. Skip-grams include a skip distance k which allows a total of k or less skips to construct the n-gram. As such, a 4-skip-n-gram includes 4 skips, 3 skips, 2 skips, 1 skip, and 0 skips combinations. A provided example shows [76] how the sentence tokens:

$$(insurgents, killed, in, ongoing, fighting)$$

Yields as bi-grams:

$$\{(insurgents, killed), (killed, in), (in, ongoing), (ongoing, fighting)\}$$

And as 2-skip-bi-grams:

$$\{(insurgents, killed), (insurgents, in), (insurgents, ongoing), (killed, in),$$

$$(killed, ongoing), (killed, fighting), (in, ongoing), (in, fighting), (ongoing, fighting)\}$$

2.2.9 Word Embedding

Word Embedding is a method for distributed semantic representation by generalizing the semantic context of words into low-dimensional vectors [77] (Figure 5). Their generalization power has been instrumental in the boost in performance of NLP systems (e.g. on Sentiment Analysis, Translation [Figure 6], Intent Classification tasks).

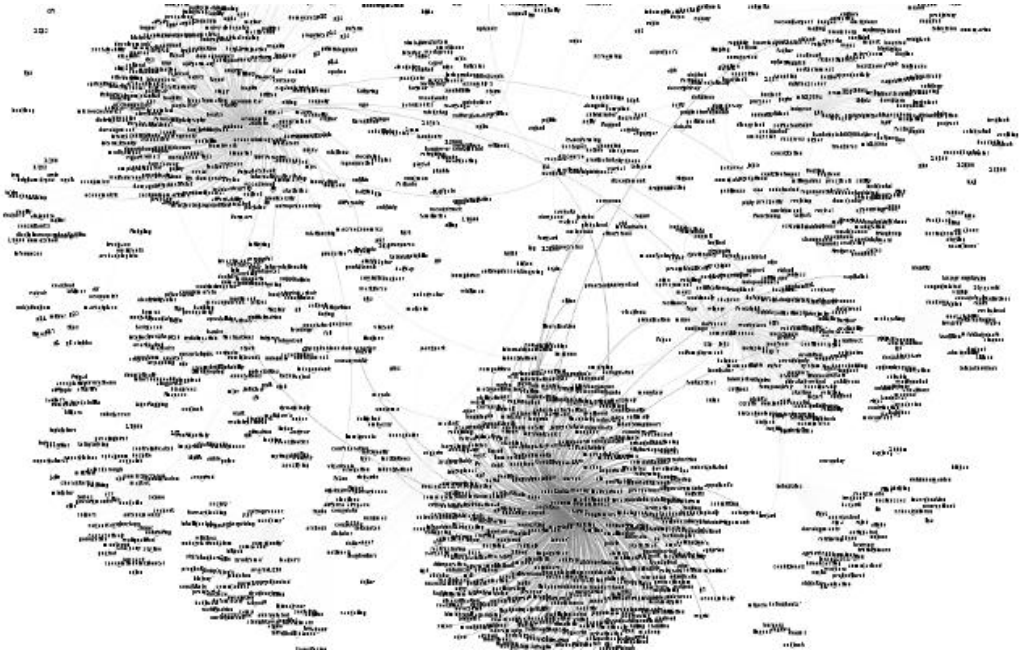

Figure 5 - Word Embedding Representation of Semantic Proximity [78]

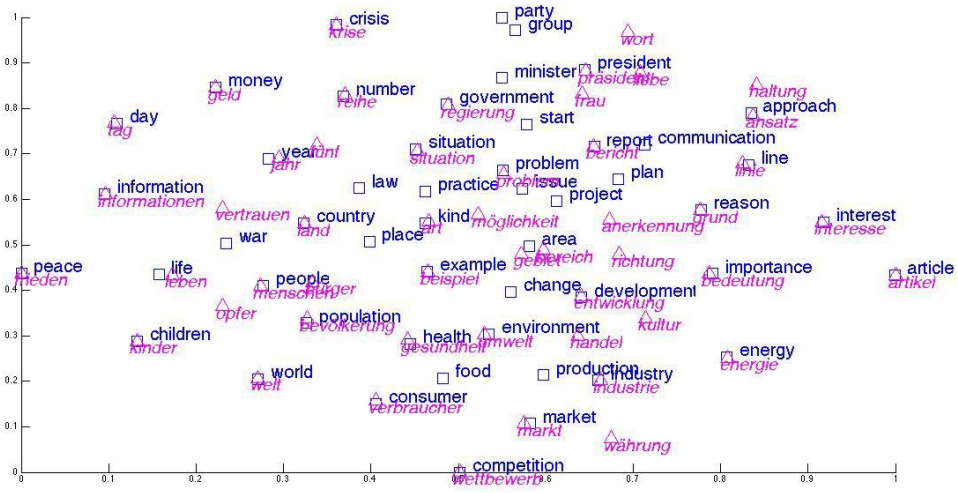

Figure 6 - Word Embedding Representation of Semantic Similarities between English and German terms [79]

The main limitation of a Word Embedding representation is that the whole of a word's meaning is put into a single atomic representation. However, some recent solutions have been presented to minimize this [77].

2.2.10 TFIDF Bag-Of-Words Cosine Similarity

Term Frequency-Inverse Document Frequency (TF-IDF) calculates the relative frequency of words in a document compared to the inverse proportion of that word over the entire document corpus [80].

$$TF = \frac{Number\ of\ times\ term\ t\ appears\ in\ a\ document}{Number\ of\ terms\ in\ the\ document} \quad (1)$$

$$IDF = 1 + log\left(\frac{N}{n}\right) \quad (2)$$

$where\ N\ is\ the\ number\ of\ documents\ and$

$n\ is\ the\ number\ of\ documents\ a\ term\ as\ appeared\ in.$

$$TFIDF\ Vector = TF * IDF \quad (3)$$

Intuitively, this calculation determines how meaningful a term is within a document. Words that are common in a single or a small group of documents tend to have higher TFIDF numbers than common words such as articles and prepositions. This comes in contrast with the BOW approach which gives higher scores to those much frequent words, which do not give any informational content. Additionally, BOW gives comparatively higher scores to larger documents.

Cosine similarity, however, is a measure of similarity between two non-zero vectors common utilized to model relative meaning proximity between the TDIDF values of textual documents (represented here as doc1 and doc2):

$$Cosine\ Similarity\ (doc1, doc2) = doc1\vec{} * doc2\vec{} / ||\text{doc1}|| * ||\text{doc2}|| \quad (4)$$

2.3 Technological Review

In this section the existing available state-of-the-art NLP Frameworks and Libraries shall be presented. An enumeration of attempts at chatterbots throughout the years will also be included, followed by a list of current chatterbot development frameworks.

2.3.1 NLP Frameworks

This section shall be dedicated to the describing of some of the available NLP libraries.

2.3.1.1 Natural Language Toolkit (NLTK)

The Natural Language Toolkit is an open-source and community-driven platform for Python-based NLP solutions with access to over 50 corpora and other lexical resources [81]. Some highlightable current capabilities of NLTK include interfaces to several Stemming and Lemmatizing Frameworks, sentence tokenization, Part-Of-Speech (if the word is a noun, pronoun, verb, adverb, adjective, conjunction, preposition or interjection) Tagging and even some Classification modelling functions [82].

2.3.1.2 fastText

fastText is an open-source, free and lightweight text classification library developed by Facebook's Artifical Intelligence lab [83]. It allows for the learning of text representations and classifiers. It is used for supervised and unsupervised neural networks with the purposed of obtaining Word Embedding Models. Some models have been trained for several known languages.

2.3.1.3 Word2Vec Model

Word2Vec is a two-layer neural net developed by Google researchers [84] which processes a text corpus and creates an appropriate vectorized representation of its terms. It does this by resorting a Continuous Bag-of-Words or a skip-gram methodology, with the latter being credited as producing more accurate results on larger datasets [85].

2.3.1.4 GloVe

GloVe is a Stanford-developed unsupervised learning algorithm [86] for obtaining vector representations for words. Given a corpus, GloVe training is "performed on aggregated global word-word co-occurrence statistics from a corpus, and the resulting representations showcase interesting linear substructures of the word vector space" [87].

2.3.1.5 Gensim

Gensim is a free, open-source Python library which allows unsupervised semantic modelling from plain text through TFIDF bag-of-word vector representation [88].

2.3.1.6 Spacy

SpaCy is a Cython NLP platform that offer neural-based modelling for tasks like multilingual Part-Of-Speech tagging, text parsing, NER and word vectorization (among others) [89].

2.3.1.7 Scikit-learn

Sickit-learn is a Python-based library with built-in Classification, Regression, Clustering, several Dimensionality Reduction algorithms, Model/Feature Selection and Text Pre-processing [90].

2.3.2 Chatterbot Oriented NLP Frameworks

Several platforms have been developed in order to provide easily integrated Natural Language Processing to chatterbot instances. This section includes an enumeration and description of several of these platforms.

2.3.2.1 Rasa NLU

Rasa NLU is a chatbot oriented (built to be incorporated by chatbot frameworks via dedicated channels) that offers supervised word-embedding training and representation from plain text entries. The models are trained to offer intent classification and NER capabilities. It does this by allowing the definition and incorporation of the text's language, corpora and existing NLP pipelines like Spacy's [91].

2.3.2.2 Dialogflow

Dialogflow can be introduced as Google's cloud-based machine learning service [92]. It provides contextual-based voice and conversation intent, entity and context-mining as a request response. It provides adapting entity and intent mining results for each ongoing conversation depending on the previously determined contexts. Additionally, it provides several out-of-the-box connectors like the ones for Google Assistant and Facebook Messenger [92] as well as personal sites and applications.

2.3.2.3 IBM Watson

Through the Watson project and Deep-Learning, IBM provides an open-platform for several chatbot oriented capabilities. Of these capabilities particular highlight should be given to Watson's Language Translation, Natural Language Understanding (entity, keyword and semantic role extraction), Machine Learning and Personality Analysis (linguistic analytics to determine an individual's personality characteristics and mood) due to their usefulness to this project [93].

2.3.2.4 Microsoft LUIS

Microsoft LUIS (Language Understanding) is a cloud-based API service that provides voice and text access to ready-to-use deep-learning models for conversational intent and entity mining. The platform is easily connected to other Microsoft applications [94].

2.3.2.5 Wit.ai

The Wit.ai platform provides an API service that provides customizable intent and entity mining capabilities, sentiment analysis, language detection through text or audio input [95]. Wit.ai provides a simple integration process into Facebook Messenger and personal sites and applications.

2.3.3 Chatterbot Frameworks

Over the years, ever since Weizenbaum's 1967 attempt with ELIZA [96], several attempts were made at a systems which could mimic actual human interaction. In the recent past, many multinational companies have doubled down on this customer interaction approach. Some examples include Apple's Siri [97], Google's Home Assistant [98], Amazon's Alexa [99], or Microsoft's Asian market bound Xiaoice [100]. While these systems have a larger scope than a common chatbot system, the advantages [4] of the intelligent automatization of customer interaction are not lost on many businesses. Chatterbots (or chatbots) solutions which have been described as an "artificial intelligence program that simulates interactive human conversation by using key pre-calculated user phrases and auditory or text-based signals. Chatbots are frequently used for basic customer service and marketing systems that frequent social networking hubs and instant messaging (IM) clients " [101], inspired communities to develop specific frameworks for their programming and maintenance. Some examples of free and open-source frameworks, considered to be a part of a possible solution for this thesis, shall now be presented.

2.3.3.1 Botpress Framework

Botpress is an open-source bot building platform built using TypeScript. It supports multiple parallel chatbot instances with multiple dedicated channels to NLP providers (Rasa NLU, Google's DialogFlow, Microsoft's LUIS) and communication platforms (Facebook Messenger). The available documentation [102] shows a structured divided into 4 communicating models:

- User Interface (Chat) - Composed by the means utilized to receive input/transmit output from and to the user. Usually composed by third-party software (Chat that installed the bot);
- Input Analysis - Composed by the channels utilized to communicate with the user interface as well as the NLU component. The NLU Component (Natural Language Understanding) determines the intent of the user's communication;
- Logic and Dialog Management - Comprised of a single component, a Dialog Manager utilized to determine which conversation flow is best for the conversation.
- Chat Response Formation - Contains two components:
    - Content Element Layer - Capable of determining the best terms to construct the response text
    - Content Rendering Component - Capable of constructing the response with the given terms.

Its indications on conversation are editable through a dedicated .json file. Although Botpress offers a visual flow editor, on embeddable UI is given. After some testing, it was noticed that each instance is programmed to be completely asynchronous, even ignoring sleep() instructions.

2.3.3.2 Rasa Stack Framework

Rasa Stack is a project composed of two separately available components, Rasa Core and Rasa NLU [103], the former being dedicated to user communications and conversational flow while the latter is used for intent mining. Both are trained (and debugged) separately

with different commands; however, the Rasa NLU training component requires an NLP pipeline to be configured between both components. This pipeline should include indication to the target language, available json-based datasets and utilized ML frameworks and libraries.

While this framework offers basically the same group of pre-programmed communications channels (and the possibility to add your own) as the one mentioned in Botpress, it does not allow for the usage of another NLP provider. It does not offer, however the management of multiple parallel instances and a visual conversational flow editor. Conversational flow is instead manually defined in a specific Rasa Core file, as is the definition of intents, the programming of custom actions to be enacted upon a certain intent as well as the endpoints configurations.

An User Interface for Rasa Stack is provided in a different project [104]. However, since its dedicated communication channel does not allow for any type of configurability, no intermediate cleaning or parsing operations can be made.

2.3.3.3 Botkit Framework

Botkit is a now Microsoft-based chatbot building and management platform [105]. It allows for the upkeep and configuration of multiple parallel instances complete with a visual conversation flow editor and an embeddable and editable User Interface. It also allows for the conversation flow to be defined in a manual and centralized form: Botkit relies on Controller instances which provide a listener to a pre-defined set of intents and the action to be enacted upon their detection. Unlike Rasa and Botpress, Botkit can only resort to external providers for the chatbot's NLP requirements. As such, the framework has made available a number of connectors (much like the previous frameworks) to NLP frameworks and communication channels.

2.4 Summary

Throughout this section, available frameworks and their used learning techniques have been enumerated and explained. A case can be made for the current state some of them as being the basis for an adequate solution of this Thesis' problem. After some Proof-of-Concept project realization of each of the three considered frameworks and accompanying plugins and channels the results were compared. The frameworks are compared by the functionalities and tools they provide such as an implementation of multiple communications channels (Facebook Messenger, Slack, REST communications…), possibility of implementation of a custom communication channel, built-in NLP service, NLP service provider communication channels, offering of an embeddable UI, visual conversation flow editor, debugging/log generating capabilities and multiple simultaneous chatbot instances management. (Table 1).

Table 1 - Comparison of available functionalities and tools of the considered bot framework

	Botkit	Rasa Stack	Botpress
Multiple Channels	x	x	x
Custom Channels	x	x	x
Built-In NLP		x	x
External NLP Provider Channels	x		x
Embeddable UI	x	x	
Visual Flow Editor	x		x
Debugging/Logging	x	x	x
Multiple Instances	x		x

It was noticeable that Botpress would possibly be excluded. Although it can be said all options had similar learning curves and provided adequate documentation, Botpress did not allow for any type of synchronous process behavior as its back-end was programmed to carry out programmed tasks as soon as possible (even ignoring sleep() instructions). As such, the level of control one would have on the solution's operations would be greatly diminished. As for the other two, with the same source information as basis for supervised learning (which for both cases required equal effort in information gathering and dataset preparation), both achieved comparable results in classification attempts. As such, some more consideration would be required to conclude on a framework to commit on moving forward. Such considerations would be made following an analysis on the project's potential enterprise value.

As for the NLP framework that would be utilized if the chosen chatterbot framework does not have capable built-in NLP faculties, frameworks like Dialogflow, IBM Watson and Microsoft LUIS were disregarded as per project restrictions (frameworks require a paid subscription). The final choice landed on Rasa NLU based on community size [105] and quantity of available documentation.

3 Value Analysis

Businesses and ideas are often defined by how innovative they are. In terms of a business it can mean how much can they distinguish themselves from competitors with their ideas and products, possibly attracting potential investors. In the search for innovation, businesses are looking to make a more educated guess on how valuable a devised concept or idea can be. This notion of value became subject of study for methods to more accurately and safely (some projects can seem like a good idea but end being impossible or extremely costly to implement or divulge) determine it. The current chapter seeks to explore such an analysis with the aforementioned value analysis methods pertaining the thesis project and possible solution variants.

3.1 Innovation/New Concept Development

Necessity is more often than not the mother of invention. Many businesses are made successful upon a single concept or idea which fills a void in a given market. In terms of costumer needs and preferences these concepts, simply put, introduce something new and desirable. It is imperative that costumer feels the need to make and affordable purchase, contributing to the survival of the business, a situation where there are benefits for consumer, provider and possible intermediate links.

Before we reach a commercial model there is typically a long and iterative process that defines the final offer. This process is usually divided into 3 parts: Fuzzy Front End, Iterative Product Development and Commercialization (Figure 7):

The Fuzzy Front-End stage is the starting point of the project where opportunities and concepts are respectively identified and devised before formal product development and delinearization of commercialization tactics. Unlike the two stages that succeed it, Fuzzy Front End isn't a linear process but rather a non-sequential, self-feeding stage based on discovery, pontification and validation. The lack of a coherent and well-established process model meant that the flow of this stage was based solely on the performance and cohesion of the team. A solution was introduced with the work of Peter A. Koen and co-authors [108] where past FFE experiences in a corporate setting allowed for the introduction of a substage

model for the process. While it still wasn't a linear model, as they believed it to be inadequate for the task at hand, the New Concept Development (NCD) [109] was introduced. The NCD model divides the Front-End Stage of Innovation into 3 parts: the "engine", the "wheel" and the "rim" (Figure 8).

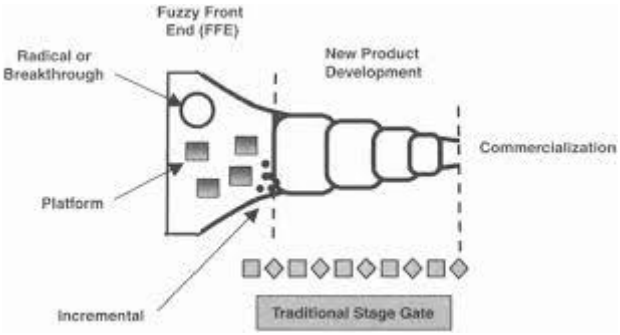

Figure 7 - Traditional Iterative Innovation Process [109]

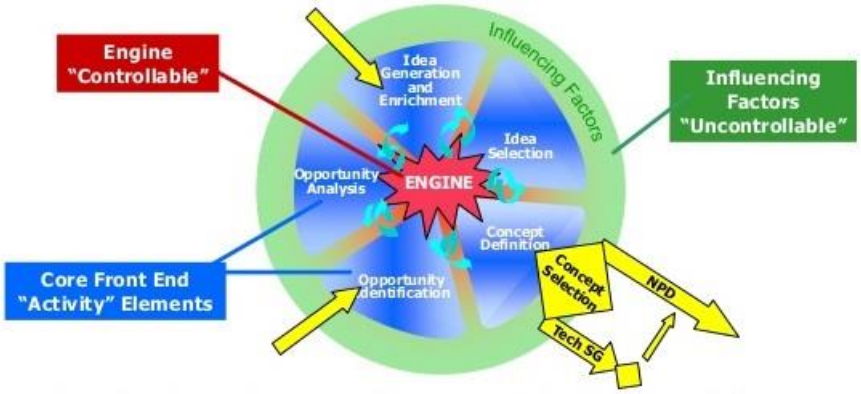

Figure 8 - Representation of the New Concept Model Stages [108]

These terms will now be described:

- The "engine" – represents the internal corporate elements that directly or indirectly influence the flow between "wheel" stages;
- The "wheel" – contains 5 important Front-End activity elements:
    - Opportunity Identification – one of the activities where a project concept may start. It pertains to the discovery of a business or technological gap that exists between the current market situation and a foreseeable future;
    - Opportunity Analysis – Analysis of a is identified opportunity;
    - Idea generation and enrichment – one of the activities where a project concept may start. It refers to the generation of the most basic and embryonic form of a product or service, including some possible alternatives;
    - Idea selection – Selection of the principal features and methods of obtaining them as a costumer benefit;

- Concept Definition – realization of a well-defined product or service that includes its principal features and costumer benefits.

- The "rim" - represents the external market elements that directly or indirectly influence the flow between "wheel" stages.

Application of the various parts and flow of the model may result in the discovery of previously unknown and overlooked factors. In addition, the model forces an analysis within the context of the corporation and encompassing environmental market. By applying the 5 elementary controllable facets of the model, we can have a better look into the problem that motivated this thesis:

3.1.1 Opportunity Identification

The business opportunity for this thesis' project comprised several factors. The most obvious was the fact that a user friendly, quick and NLP-incorporated conversational agent would vastly reduce the amount of easily resolvable tickets which flood E-goi's Customer Support queues. As we can see from the customer plan options offered [2], the best it can do is 15 hours between ticket request and response. By diminishing the amount of influx, the project would aim to reduce the amount of time needed to respond to all of the costumer's issues. In short, the project could provide an increase in user satisfaction and appreciation for the product as well as contributing to the maximization of employed human resources (as we'd be reducing the amount of trivial issues to resolve). In increasing user satisfaction, it would be expectable that an increase in user retention would follow as well as a better public reputation and better profit returns.

Additionally, such a system would have amazing growth capabilities. By having a conversational bot framework, one can envision other possible uses and environments. For example, the bots can be sold as part of a Software as a Service program, where specific user dialogue flow and context domains can be modified by the user. Another example could be bots for internal use in E-goi helping around with logistic and planning tasks. Finally, the bots can be programmed to be more than merely suggestive tools. Specific actions and parameter requirements could be linked with E-goi's context domain, creating a bot that instantiates and creates based on intent alone.

Given the all the reasons mentioned above, this thesis' project can become a selling point for E-goi a possible source of major user experience improvement.

3.1.2 Opportunity Analysis

In order to make a sufficient analysis of the opportunity realization described above, an accurate enumeration and measurement of its predictable benefits and costs. The SWOT analysis approach (Strengths, Weakness, Opportunities, Threats) [110] was chosen for this section as it encompasses a description of a concept's advantages and disadvantages, both internal and external (Figure 9).

Figure 9 -SWOT Analysis [111]

This particular developed SWOT analysis is as follows [110]:

- Strengths –The realization of an adequate solution for the opportunity identified in the previous section. In its ideal state the solution should provide an easier to adapt and to scale, working solution for the enterprise;

- Weaknesses – The proposed solution would have some inherited costs to it. Besides the development time requirements which include planning as well as technological choice and implementation, an adaptation period will be mandatory from both developers and users. Possible reworks could also be mentioned and added to the list of predictable solution costs;

- Opportunities – The reusability of the project's technological capabilities is envisioned to provide multiple avenues of differentiation and growth to the platform as enumerated in the previous section;

- Weaknesses – Only low chance threats to the project's conclusion and long-term health were identified. These include market related corporate bankruptcy of disappearance. In the technological aspects, possible stagnation of the utilized technologies could be deduced as a to the future health and adaptability of the solution.

After a review of the project's possible advantages and disadvantages and respective pondered weighting, both motivated by internal corporate and external factors, a likely positive trade-off was predicted, with good chances of having a profitable and adaptable solution for the enterprise.

An additional factor is the favorable conclusion was the ever-growing need for a means to diminish the influx of costumer support tickets, which is caused by the growing number of clients.

3.1.3 Idea Generation and Enrichment

Following the appraisal as a viable and desirable avenue, this section includes the definition of initial embryonic features and possible alternate feature concepts and characteristics. The first step would involve a search for techniques and architectures which would be included in the final solution as a basis for all its planned features. These features would require the development of User Dialog Management, User Dialog Interface Incorporation and Natural Language Understanding components. The possible alternatives for these components were:

- Development of these components from scratch;
- Usage of specific open-source tools and frameworks to implement the aforementioned processes;

3.1.4 Idea Selection

Given the sheer low-level understanding of machine learning and communication techniques require to execute the first option, the second option was naturally chosen. Additionally, in order to obtain a similar performance and accuracy with the first option, a large amount of code testing and reviewing would necessary. Utilization of open-source tools and frameworks also means that the community of developers and specialists around each one would be contributing to the project's success and longevity.

3.1.5 Concept Definition

The presented opportunity would be duly answered by the implementation of a Natural Language Understanding based Chatbot module. It would for a capable resolving costumer -based issues if its context domains and knowledge basis are well understood and incorporated. This would create indirect profit to the organization by improving customer response statistics and customer satisfaction. In a possible near future, the solution could be adapted to create actual direct profit with its sale as a Service besides possibly incorporating a more active stance, capable of execute previously user defined use cases.

3.2 Value of a Solution

This section aims to grant some understanding over the notions which the academia utilized to model the notion of Business Value, adapting them to justify why the development of the solution generated worth for the enterprise and its customers.

3.2.1 Notion of Value

When talking about a product, service or even a concept, the notion of value means different things depending on the intervenient. Widely used as a metric do evaluate a parameter

whose presence or absence is of particular note, as any other qualitative metric, its application and identification are highly subjective. For example, for a business service provider it would be more logical to recognize value in presentation and customer relations as difference in future profit may be a likely dependent variable, in the customer case's however he/she may place more value on the cost/effort require to acquire the service. Within the same interaction, the notions of value may differ or coincide completely, based on the overarching context [112].

Zeithaml's 1988 work introduced the four facets through which a user generates the notion of want (as he/she sees a worthwhile deal or opportunity) [113]. These facets included the value one sees in a low price, the presence/abundance of desirable characteristics, an advantageous quality/price relationship and an advantageous tradeoff between what is given and what is received. This framework will be utilized to map the value which is conferred upon the organization by allowing the development of this solution. Firstly, there is value in the solution in the low price (costs) of this solution, as the project was done under the context of an internship. Secondly, as the solution is being tailor made to the organization's needs and existing overarching system, there is little doubt that its incorporation and functionalities would be valuable. Thirdly, as the solution encompasses modern and innovative techniques as well as good software design practice, given its already presented low cost, it would be considered a good quality/price ratio. Finally, since the project has growth potential with its value expected to increase over time, it could be considered that such a project in exchange for an internship was a good tradeoff for the organization.

3.2.2 Perceived Value

The notion of Customer Perceived Value (or CPV for short), has been defined as "*the difference between the prospective customer's evaluation of all the benefits and all the costs of an offering and the perceived alternatives*" [114]. Total customer value is perceived as the monetary value of the bundle of all benefits (economic, functional, etc.) customers expect from a given idea. Be it because of the products, services, personnel, or even the image involved. Total customer cost is understood as the bundle of costs that customers expect to take in the effort to identify and acquire a specific deal [114]. Applied to a mathematical representation:

$$CPV = Total_{CostumerPerceivedValue} - Total_{CostumerCosts} \quad (5)$$

Applying this information to the work at hand, we get that the organization's (that in this case can be considered the customer) perceived value is given by the difference between the benefits they will reap from the solution, such as: an automated solution to most of the costumer's issues which the costumer support team faces, the solution's potential to be redefined as a Software as a Service possibility, among others.

And the costs they intake by trying to accomplish this solution: spend resources to find an intern, spend money, time and infrastructures to allow the intern to understand the problem, spend time of said intern to research and test approaches to the imagined solution, as well as develop the project's documentation artifacts.

3.2.3 Longitudinal Perspective of Value

Woodall's 2003 work introduced the notion of time variance in a person's perspective of value [115]. Woodall's notion of time-variant *Value of the Costumer* was divided in four stages, pre, during and post purchase as well as post use. Given the model's sequential nature, the author mentioned the "*both temporal and, consequently, cumulative, aspects*" of value, where the costumer's standing influences the expectations during and after future stages. The costumer's standing would be dependent upon the perceived sacrifices and benefits of the costumer. Applying this model to the project at hand, noting the benefits and sacrifices derived from the solution (Table 2):

Table 2 – Woodall's Longitudinal Value Perspective [115]

	Benefits	Sacrifices
Pre-Purchase	Expectation of a scalable, reusable and automated way to improve customer support response times.	Prerequisite of searching and understanding the very basics of a new technology in order to fully understand the advantages
Moment of the Transaction	Acquisition of an easily upgraded and adaptable valid solution which can be utilized for other purposes.	Costs related to learning and understanding the new solution fully (partially fixed through the produced documentation artifacts)
Post-Purchase	Full understanding of the solution as well as the way it can be changed and edited to adapt to new contexts	Work related to the preparation of contexts and possible datasets to retrain the solution in case of changes or evolution and contexts
After use	The solution's adaptation capabilities can let it keep updated with the current business situation until a possible better alternative in paradigms or technologies present themselves.	

3.3 Value Proposition

When pitching a any business proposal, be it a product or a service, it is important to preemptively define its value proposition. This valorization is a major part of the marketing effort for the product. It consists of the reasons of why a client should endorse your product or concept, as discernible from the term definition in the Cambridge dictionary which states it as: "*a reason given by the seller for buying their particular product or service, based on the value it offers customers*" [116].

A value proposition canvas is an organized and concise way to present the benefits of the a service or product based on the advantages it provides, the activities it now enables and the negative situations it corrects [117] (Figure 10):

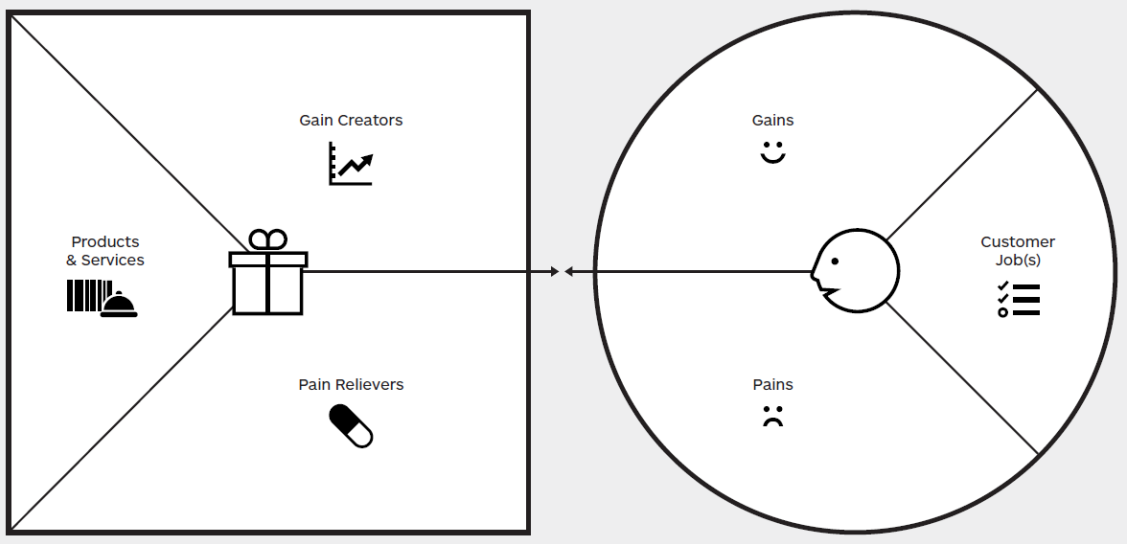

Figure 10 - Value Proposition Canvas [117]

Such a proposition has been developed for the task at hand:

- In terms of products and services, the project aims to yield a Natural Language Understanding incorporated chatbot generation and management system. The costumer jobs fulfilled by it includes automatic dynamic FAQ answering and issue solving;
- In terms of pain relievers, the project aims to be scalable, easily usable and maintainable as well as being capably of being fine-tuned to whatever cognitive domain or context. These characteristics aim to help with some "*pains*" that costumers could have with the system, these include: unease of understandability or usability with the tool, lack of performance due to lack of scalability as well as being given unreliable or unrelated results.
- In terms of gain generated by the project's gain creating facets, it can be said that its growth potential and enterprise system integration mean that it can serve some more profitable purpose in the future without needing many development cycles. Its retrainability results in a large context adaptability towards whatever contexts and flows are needed. Additionally, its utilization of free open-source technologies and its realization as part of a internship result in a very low cost for the enterprise to support.

3.4 Business Model Canvas

The Business Model Canvas is a model proposed as a standard through which the representation of a business model could be compact and efficient. Its aim was to make

understanding the proposal facets simple while enabling for a complete view of the project's standing and value generation. The author suggested its division into nine segments, namely: Key Partners, Key Activities, Key Resources, Value Propositions, Customer Relationships, Channels, Customer Segments, Cost Structure and Revenue Streams (Figure 11) [9]:

- Customer Segment includes the different types of people and businesses that the organization wants to serve and reach out to and serve. In the current scenario, we can include the company's registered users with whom the bots would interact with;
- Value Proposition includes the which might lead a segment of customers to choose our product or service over the competition. This could include the reduction of costumer churn and of costumer support queue times;
- Channels describes the communication avenues through which the organization contacts the customer segments in order to present a Value Proposition. In current context, an incorporation in the company's web portal is expected as well as a custom Web API;
- Customer Relationships includes various ways in which the organization manages and responds to the communications from each of its customer segments. For the problem at hand, the company (E-goi) is considered to be the main customer, with which a long-term relationship can be expected due to the lasting benefit the project can bring;
- Revenue Streams enumerates the sources of income from a product or service. In the current scenario an indirect source of revenue with the reduction of customer churn and customer support queue times are expected;
- Key Resources include the resources with are crucial the organization's Business Model success (essential to the creation and offering of the Value Proposition, to reach new markets, to keep relationships with targeted Segments and to generate avenue). In this case, we can include the resulting automated issue solving bots and a modular and repurposable software structure;
- Key Activities include the operations which must be undertaken by the organization in order to achieve success in its projects. In this case, they include the parsing of user interactions, development and use of multiple Natural Language Understanding models to determine user intent and resulting response. Additionally, it would be necessary to incorporate company data sources and records;
- Key Partners include the business relationships kept by the organization in order to optimize models, reduce risks and acquire resources. In this case, no external partners are identified, as all the work and planning are being made within the organization;
- Cost Structure pertains to all types of resources necessary to adequately create, publicize, maintain and distribute a given product or service. In this case this would include the hardware and software related expenses, as well as internship salary costs.

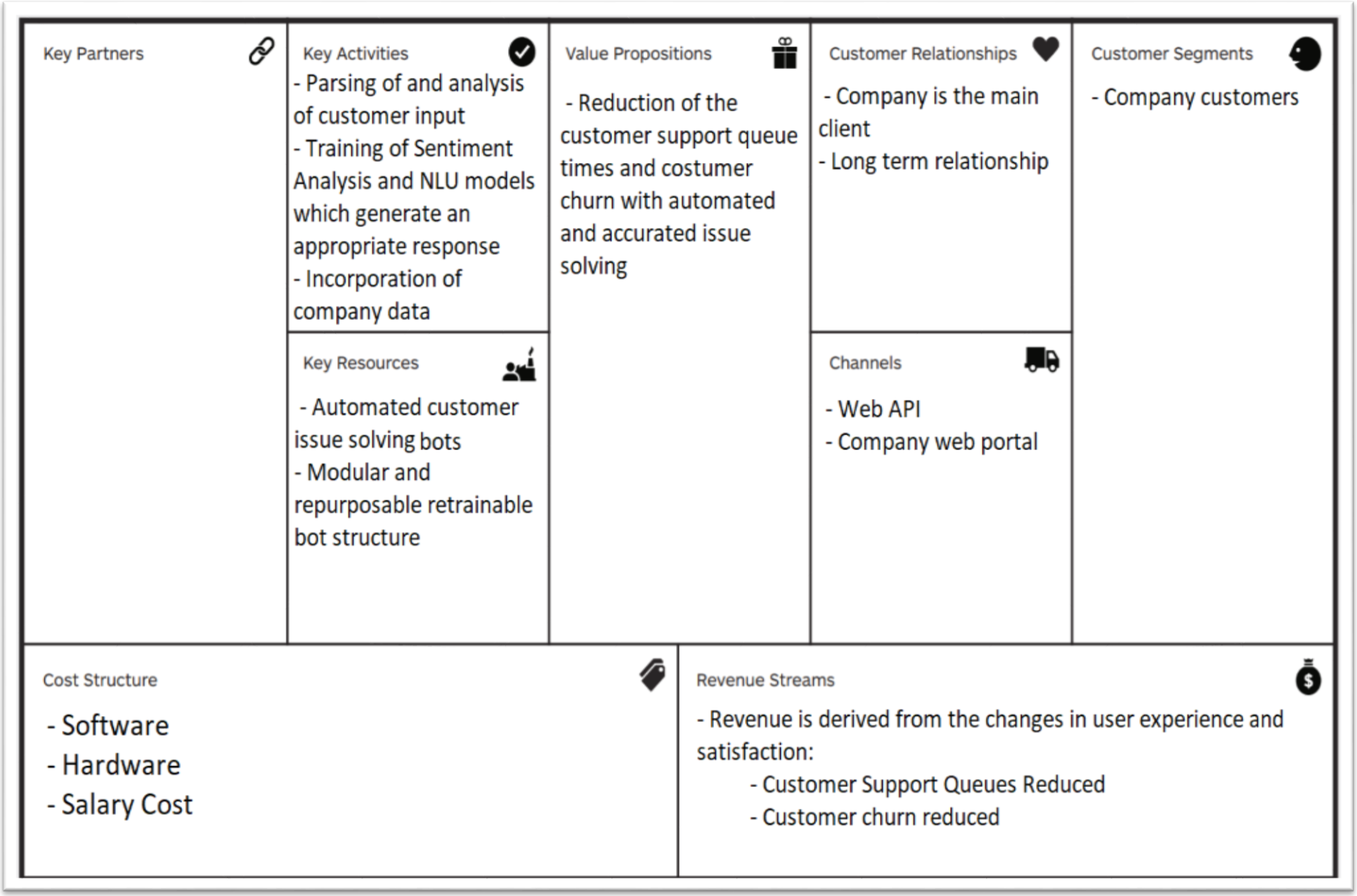

Figure 11 - Business Model Canvas

3.5 Porter's Value Chain

The Value Chain model was first introduced by Porter in his 1995 work Competitive Advantage [11]. In it, Porter mentions that "*Competitive advantage cannot be understood by looking at a firm as a whole. It stems from the many discrete activities a firm performs in designing, producing, marketing, delivering, and supporting its product.*". The proposed Value Chain model was designed with this statement in mind as it's a systematic tool that examines all organization activities and their interactions. Its goal is to separate and identify core business activities, their costs to the organization and their differentiation potential.

A Value Chain model is usually divided into the identification of primary activities and of supporting ones. Primary activities include those which relate to the physical production of a product, sale to the buyer and assistance. Support activities are those which back the primary activities and each other by obtaining and providing resources, as well as other necessary functions (Figure 12).

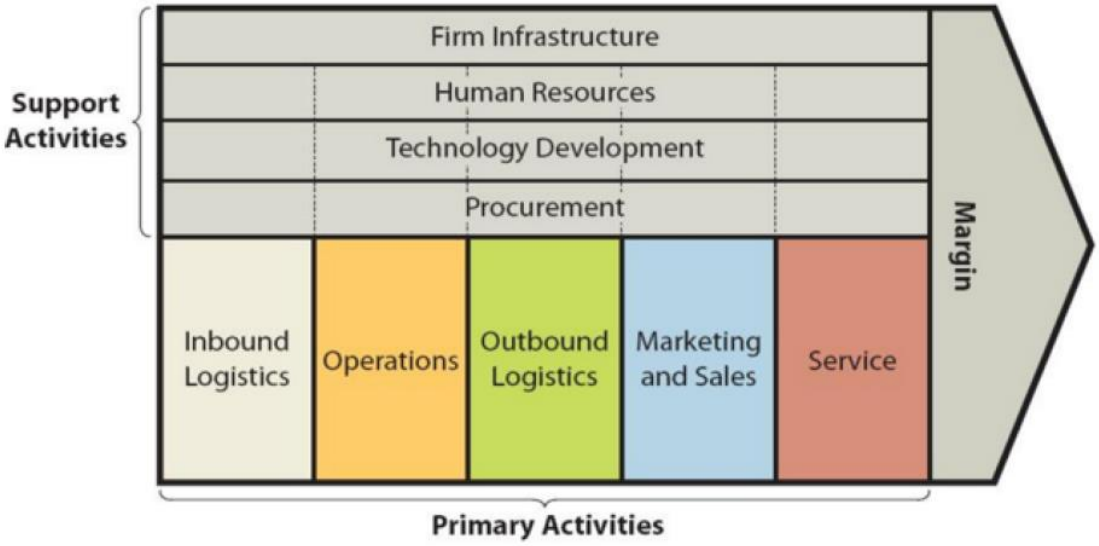

Figure 12 - Porter's Value Chain Diagram [118]

Porter mentioned four organizational operations which have a significant role in each suggested primary activity. These support activities operations which manage the organization's Infrastructure, Human Resources, Technological Development and Procurement (Purchasing). Infrastructural activities include the functions which allow the organization to maintain daily operations such as accounting, legal and administrative management. Human Resources activities include the recruiting, hiring, training, motivating rewarding and retention of its collaborators. Technological development activities include the management and protection of an organization's internal knowledge (technological advancements and cost reduction as well as technical expertise). Procurement activities include the financial management of the acquirement of essential resources.

The aforementioned activities support the primary functioning of the organization, which Porter divided into five categories: Inbound Logistics, Operations, Outbound Logistics, Marketing and Sales as well as Service. Inbound Logistics appertain to the related with receiving, storing, and propagating inputs to the product. In the current context, the collection, various data imputed by a user when using the various areas of the E-goi platform (contact lists, forms, etc.) in this area. The Operations category includes activities which convert inputs into final presentable outputs, such as the maintenance of a viable multi-channel marketing automation platform. Outbound logistics include the activities related with the collection, storage, and physical distribution of the product to consumers (since E-goi's product is intangible, its maintenance and upkeep can be included in this section). The Marketing and Sales category appertains to the processes which persuade clients to invest in an organization's product instead of the alternatives (in this case we can include Egoi's product and feature divulgement). The Service category is related to maintaining the value of your product or service to your customers, once it's been purchased. In the current context, one can include management of customer support channels and the various IT systems.

3.6 Analytic Hierarchy Process (AHP)

This method was first introduced in Tomas Saaty's 1980 work *The analytic hierarchy process: planning, priority setting, resources allocation* [119] as a complex decision support method. In terms of formal division, it is usually divided into three main steps: hierarchic division, priority definition and logic consistency. Each of the following subsections shall be dedicated to explaining an applying each of these steps to the problem at hand.

3.6.1 Hierarchic Division

This step divides the problem into levels of different hierarchical importance (usually objectives, evaluation criteria and alternatives). An application of Hierarchic Division to this Thesis 'problem can be seen in Figure 13. At the top of the hierarchy, there is the main objective of developing a fully integrated, scalable, retrainable and editable chatbot system. The next layer includes the evaluation criteria for the solution, which include full control over intents and response types, multiple registered data sources and channels and the capability to incorporate interactions on multiple languages. For the first three entries in our criteria set, two alternatives were identified: utilize a combination of existing chatbot management and NLP frameworks (which in turn divides itself into the alternatives of using either Botkit or Botpress with an NLP provider framework or using Rasa Stack Framework) or develop our own from scratch. As for the multilingual capabilities' criterion, each trained instance chatbot could be either trained for one (requiring a separate training process for each registered language) language or every single registered one.

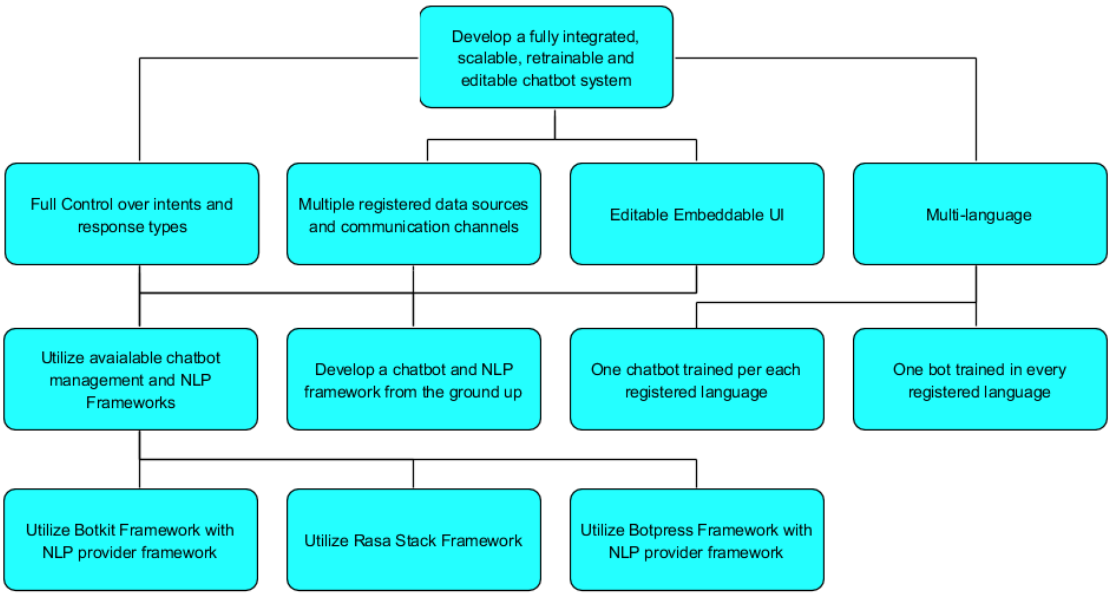

Figure 13 - AHP Diagram

3.6.2 Priority Definition

This step is done to establish and justify priorities between the criteria and alternatives presented in the previous section. In order to do this, it is required to establish a way of numerically identify which are the best alternatives to take regarding each identified criterion. The first step requires the relative evaluation of each set of alternatives with values ranging from one to nine as show in Figure 14. These results were obtained through internal meetings and debates with multiple intervenients.

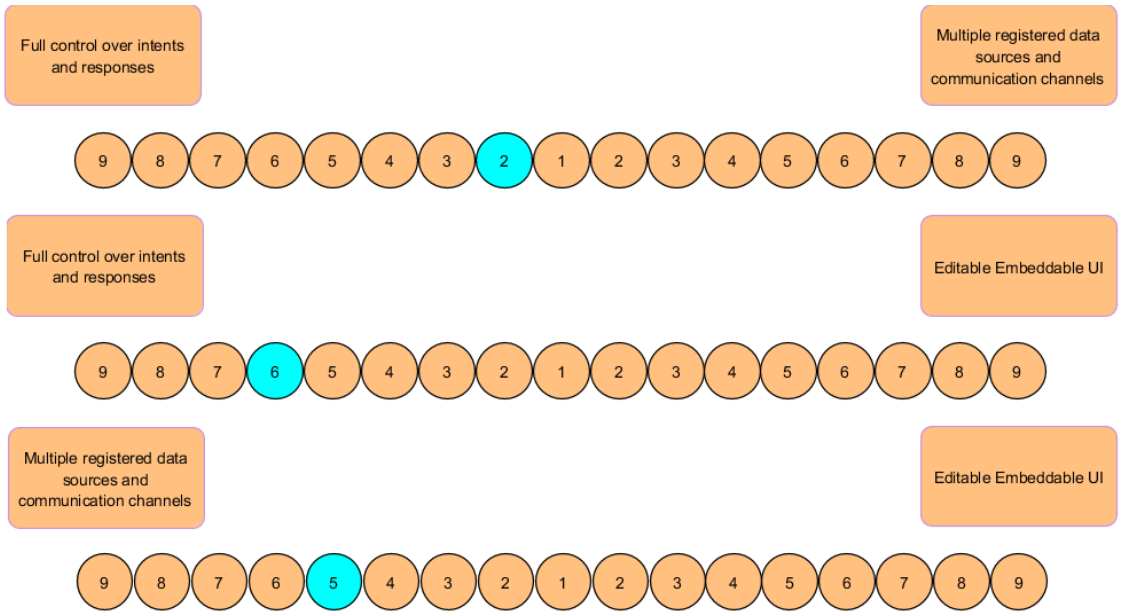

Figure 14 – Relative Criteria Comparison Results

Afterwards, a filtering of obvious alternatives not to take was made, where it was considered that it was necessary to utilize existing frameworks (less testing and research needed) and that each language would require one specifically trained chatbot (training all together would unbalance the training set), essentially simplifying the previous AHP Diagram to the following shown in Figure 15.

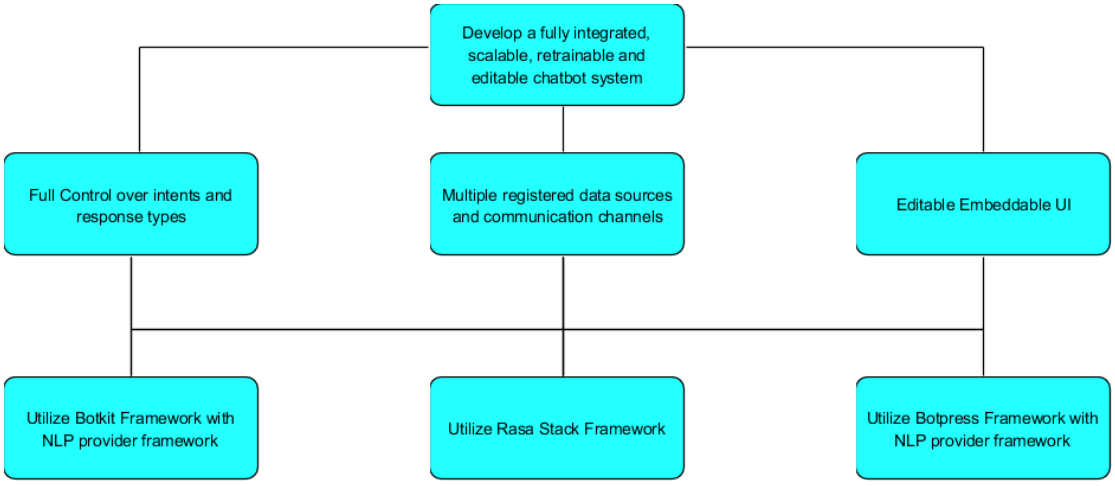

Figure 15 - Simplified AHP Alternatives

Similar evaluations were made for each Alternative in regards to the three criterions considered to be the fulcrum of the solution's evaluation. As per the last attempt, the results were obtained through internal meetings and debates with multiple intervenients (Figure 16, Figure 17 ,Figure 18).

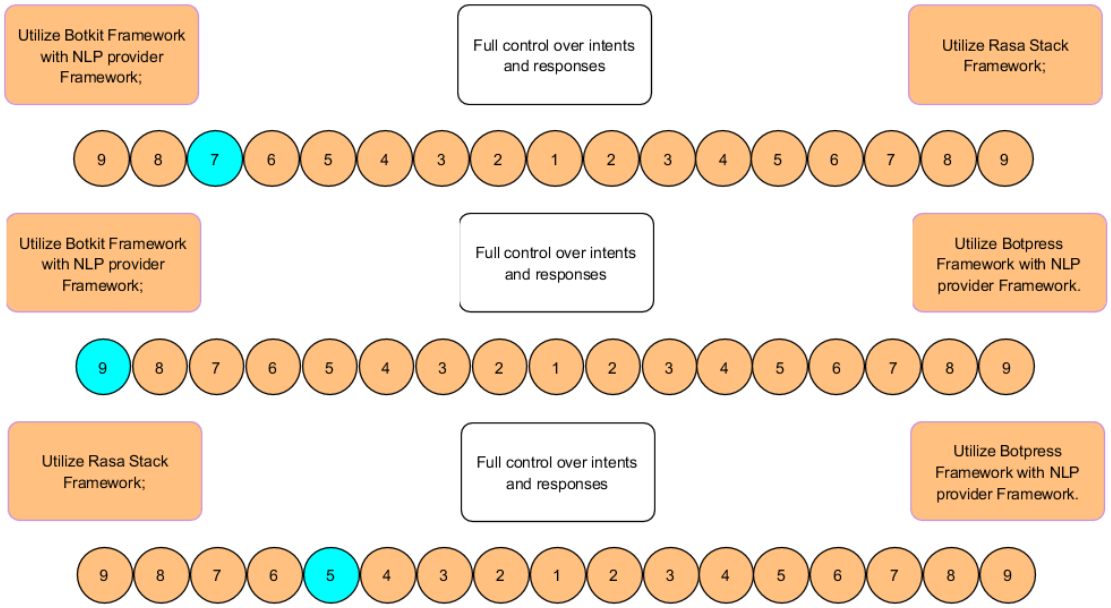

Figure 16 - Alternative Comparison Results According to the First Criteria

The three alternatives were weighted based the amount of control each provided in conversational flow and underlining programming. It was observed that Botpress didn't allow

for any methods which required synchronous behaviour. The Rasa Stack and Botkit prototypes revealed that both frameworks did not have such a flaw. The differentiation factor between these two pertained to Rasa Stack inability to programmatically change the user's input string, which Botkit allowed.

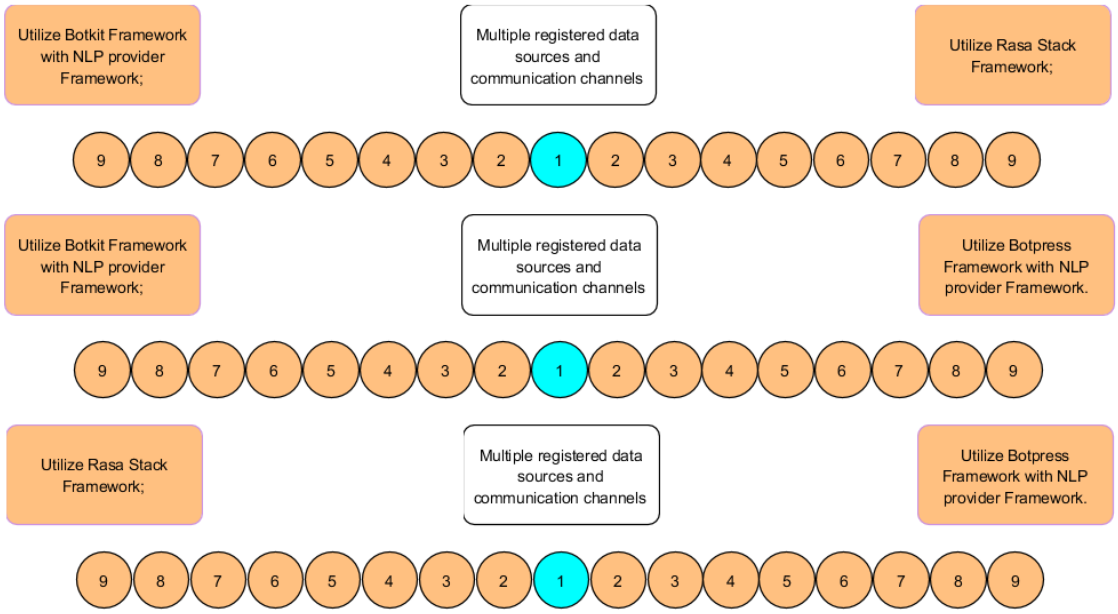

Figure 17 - Alternative Comparison Results According to the Second Criteria

In terms of communication channels, all frameworks provided support for the same platforms.

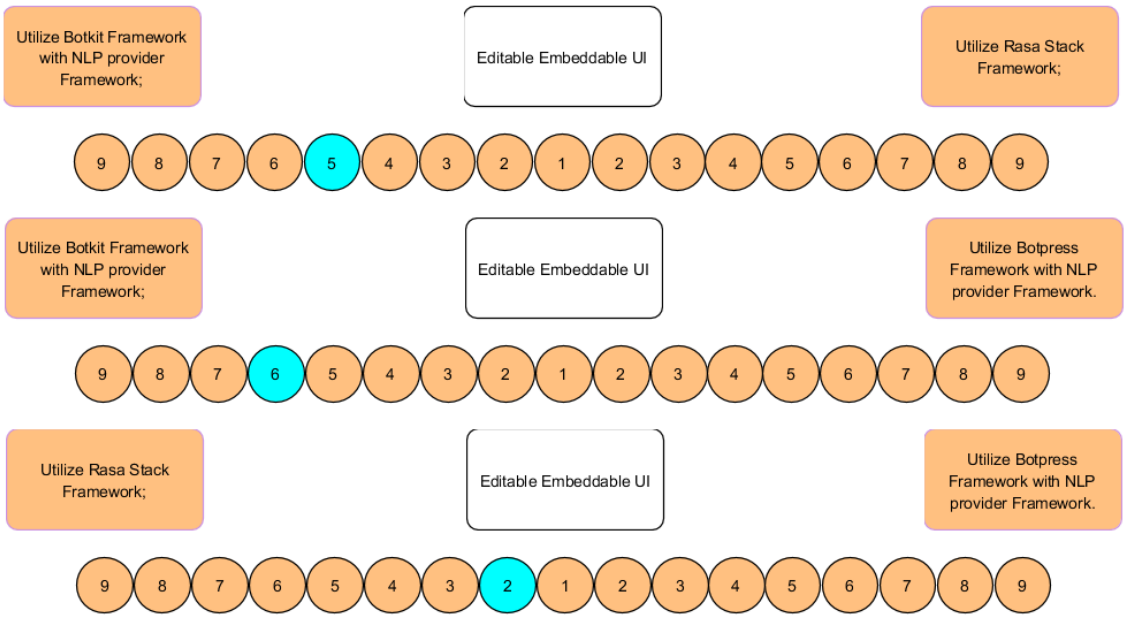

Figure 18 - Alternative Comparison Results According to the Third Criteria

All alternatives provided customizable and embeddable User Interfaces. The scores for each alternative was given by Front-End experts at the organization after a detailed presentation.

The mentioned results will be processed through AHP's weight computation method, first to numerically determine the relative importance between criteria and between the alternatives (for each criterion). The weight computation method is given by the following table (Table 3):

Table 3 - Weight Computation Process Table [120]

	A	B	C	Weight (Eigen Value)
A	*aa/sumA*	*ab/sumB*	*ac/sumC*	(*aa/sumA* + *ab/sumB* + *ac/sumC*)/*NumberOfColumns*
B	*ba/sumA*	*bb/sumB*	*bc/sumC*	(*ba/sumA* + *bb/sumB* + *bc/sumC*)/*NumberOfColumns*
C	*ca/sumA*	*cb/sumB*	*cc/sumC*	(*ca/sumA* + *cb/sumB* + *cc/sumC*)/*NumberOfColumns*
Column Sum	*sumA* = *aa+ba+ca*	*sumB* = *ab+bb+cb*	*sumC* = *ac+bc+cc*	

The following tables will contain the results of this method when applied to the determined relative numerical priorities (Table 4, Table 5). For legibility's sake, each criteria and alternative will be represented as an upper-case letter. In the case of criteria, full control over intents and responses will be represented as A, Multiple registered data sources and communication channels will be represented as B and Editable Embeddable UI will be represented as C. As for alternatives, utilization of Botkit with and NLP framework will be represented as D, utilization of Rasa Stack Framework will be represented as E and utilization of Botpress with an NLP framework will be represented as F:

Table 4 - Criterion Weight Computation

	A	**B**	**C**	Weight (Eigen Value)
A	1	2	6	0.575
B	1/2	1	5	0.3431
C	1/6	1/5	1	0.0819

Table 5 – Alternatives' Weight Computation by Criterion

A				
	D	E	F	Weights (Eigen Value)
D	1	7	9	0.7504
E	1/7	1	5	0.1897
F	1/9	1/5	1	0.0599
B				
	D	E	F	Weights (Eigen Value)
D	1	1	1	0.33
E	1	1	1	0.33
F	1	1	1	0.33
C				

	D	E	F	Weights (Eigen Value)
D	1	5	6	0.7225
E	1/5	1	2	0.1741
F	1/6	1/2	1	0.1033

Using the now known priorities between alternatives for each criterion, obtaining the global priority between each alternative can now be calculated. This is done by computing each the sum of the products between criterion the alternative weights. As an example, if for every alternative U there is a calculated relative weight (U1, U2 and U3) for each considered criterion (criteria weights represented as C1, C2 and C3). Then the alternative's global weight is thusly calculated, revealing that utilization of the Botkit framework with an NLP provider can be considered as the best option:

$$GW_U = U1 * C1 + U2 * C2 + U3 * C3 \ (6)$$

The following table represents the results of such calculations (Table 6):

Table 6 - Global Weights Table

	A		B		C		Global Weight
X	0.575 * 0.7504	+	0.3431 * 0.33	+	0.0819 * 0.7225	=	0.60505
Y	0.575 * 0.1897	+	0.3431 * 0.33	+	0.0819 * 0.1741	=	0.23771
Z	0.575 * 0.0599	+	0.3431 * 0.33	+	0.0819 * 0.1033	=	0.15723

3.6.3 Relative Priority Consistency

The decisions made in the above AHP method are based upon the pretense that the intervenients were consistent in their evaluations. As such, it becomes necessary to determine if the judgements made in the previous section (Figure 16, Figure 17, Figure 18) were not entirely subjective. This is done by calculating a Consistency Ratio (CR) value lesser than 0.1 [120]. This result is obtained by dividing the Consistency Index (CI) term with the consistency index of a random matrix denominated RI (whose tabled values can be checked at Table 7):

$$CR = \frac{CI}{RI} \ (7)$$

Table 7 – Tabled IR Values by n dimension matrices

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0.00	0.00	0.58	0.90	1.12	1.24	1.32	1.41	1.45	1.49	1.51	1.48	1.56	1.57	1.59

The CI term is obtained by utilizing the criteria comparison matrix (denominated A) and its resulting Eigen Value column (denominated x) in order to induce. the value of λ:

$$A * x = \lambda * x \ (8)$$

$$\lambda \cong 3.027$$

This value is then utilized in the formula:

$$CI = \frac{\lambda - \mathrm{n}}{n-1} = \frac{3.027 - 3}{3 - 1} \cong 0.013$$

With every needed term attained, the value of CR can now be calculated:

$$CR = \frac{0.013}{0.58} \cong 0.02$$

Since 0.02 > 0.1, it can be considered that the evaluations were consistent.

3.7 Summary

Given that the numerical evaluations of the defined criteria and alternatives were proven to be consistent, the calculation of the global weights for each alternative can be considered valid. These results show that the utilization of the Botkit framework alongside an NLP provider framework is the best technical solution for the defined requirements.

4 Design

This chapter will present the various engineering inherent requirements for this thesis' development. A possible solution structure shall be document afterwards, complete with considerations to possible alternatives in each case. It will be concluded after some information is provided on the data which shall be modelled for the solution. The need for each section was concluded according to the work of Nuseibeh ("Requirements Engineering : A Roadmap.") [121].

4.1 Requirement Elicitation

A Requirement Engineering process always starts with Requirement Eliciatation. The former consists of the interpretation, analysis, modelling and validation of a project's success requirements [121]. Several suggested techniques were utilized as choice processes through which we can infer the initial set of requirements for this project (e.g. prototyping with the aforementioned frameworks and constant group meetings), both functional and non-functional. The conceived requirements include:

- Functional - Behaviors or functions a system and/or component must be able to perform:
  1. The system should be able to allow for retrainable NLP models;
  2. The system should be able to communicate through several specification channels, with the possibility to add more;
  3. The system should be able to allow multilingual communication;
  4. The system should have editable and adaptable UI;
  5. The system should be flexible with the definition of its actions (should not have a preset unchangeable structure through which it would handle user input and subsequent response);
  6. The system should be able to gather information from multiple defined sources.
- Non-functional – Aspects which will characterize a system's performance of a function of behavior like performance, or reliability, which are not as easily tested or evaluated:

1. Perform a comprehensive state of the art research regarding the relevant topics encompassed by the problem;
2. Perform research regarding available free and open-source bot frameworks and perform a Proof-of-Concept for each identified viable options;
3. Perform an analysis in order to detect the various functional requirements needed to ensure an adequate solution;
4. Perform an analysis of available enterprise data sources and its possible adaption to the project;
5. Create a test plan to check the system's accuracy and effectiveness;

4.2 Domain Modelling

The notion of domain modeling can be defined as the appropriate representation of various concepts inherent to a given problem, complete with the enumeration of underlining attributes and the relation between them. As defined by Evans in 2004, a domain model is a: "*a rigorously organized and selective abstraction of the knowledge in a domain expert's head*" [122].

For this master's thesis, the following domain was considered (Figure 19):

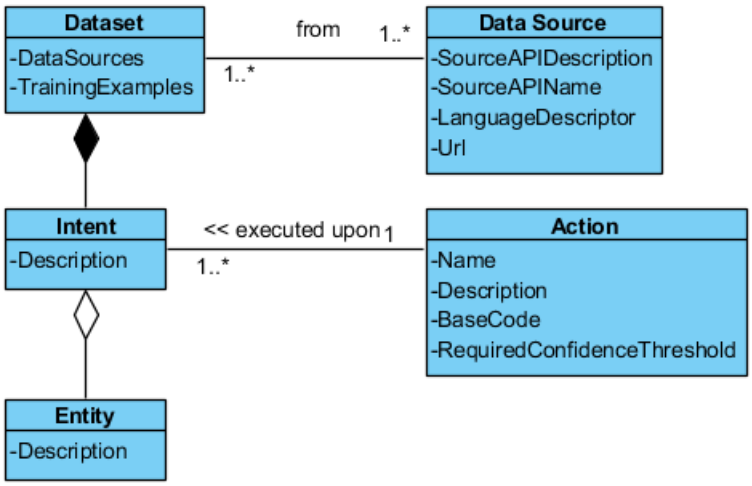

Figure 19 - Domain Model

The domain model mainly includes the notion of a registered data source and the datasets generated through them. Individual training example rows of the dataset were also considered as part of domain, but they were scrapped as such granularity was not needed. A dataset would instead be associated with composite intents it contains, which in turn would aggregate any number of associated entities.

4.3 Solution Component Structure (Enterprisal Modelling)

After a thorough analysis of the requirements needed for the flexibility of the system, which would help guarantee long-term project viability, it was defined that an ideal solution (given the organization's current situation) would require this component structure (Figure 20):

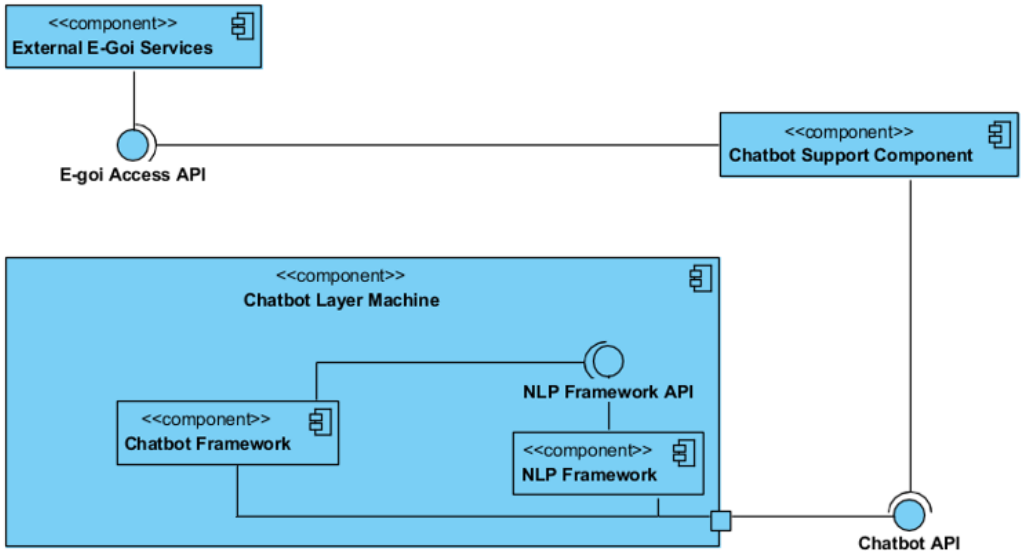

Figure 20 - Project Component Structure

The first component, Chatbot Layer Machine will be comprised of the updated instances of the Chatbot and NLP Framework solutions, the internal NLP Framework API stands as an imposition of the NLP Framework. The second component, External E-goi Services stands as a generalization of the remaining organization's Services and Resources, all accessed through a single dedicated API. The third and final component, Chatbot Support Component is the component responsible for the non-framework operations required by the system such as action and data source managing.

4.4 Process Modelling

The term Process Modelling has been described as "*an abstract description of an actual or proposed process that represents selected process elements that are considered important to the purpose of the model*" [123]. In an organizational context, it serves to provide "*abstraction within the domain of software development range from the detailed process steps executed entirely on a machine, to the larger-grained human processes involved in executing a life-cycle phase*"[123]. It is used to map sequence of activities required to achieve a goal, minding the flow and restrictions between them.

In this case, given that most processes and communications are granted by whatever NLP and chatbot frameworks are used, the following process was deemed as relevant (Figure 21):

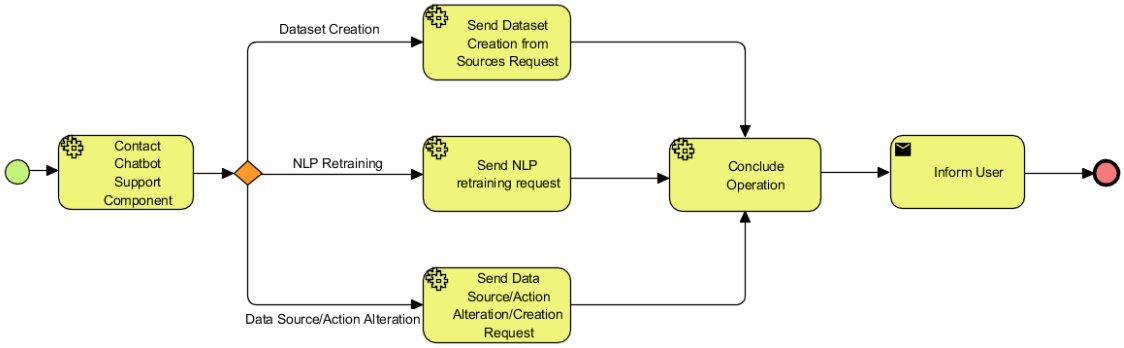

Figure 21 - Project Business Process Diagram

The flow starts with a contact to the support component (Figure 20), from which the user can execute a creation of a new dataset based on the current registered data sources , request the retraining of the chatbots models as well as execute CRUD operations on the registered data sources and actions.

4.5 Data Modelling

Data models support data and computer systems by providing the definition and format of data utilized in the various components of a system [124]. It grants some understandability on the structure of the information which was deemed necessary to store (as well as their relationships and attributes).

For this project, data comes from the registered organization sources, whose content is processed into a work training file discernible by the NLP Framework. Additionally, besides the safekeeping of actions to be performed and previously mentioned data sources, it was considered necessary to store user and systems usage statistics, user information, detectable intents and their entities (Figure 22).

As for noteworthy attributes and relationships: every action is executed after a certain confidence threshold is matched on one of its associated intents. Each intent can have any number of underlining speech entities. Each utilization statistics table (for the system and for each user) stores binary columns of statistical measurements by granularity of day, week, month and year. Lastly, a copy of each action's base code and another of the dataset's training examples are also store in binary format.

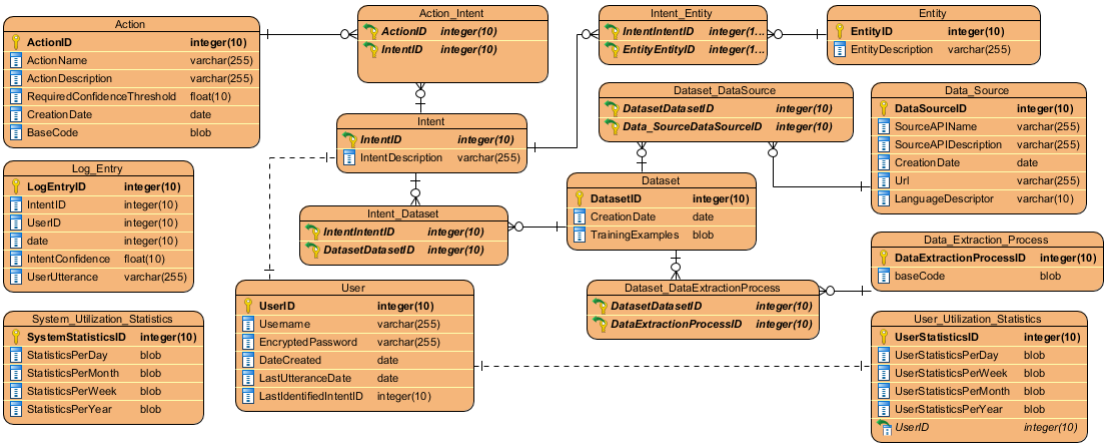

Figure 22 - Project Entity Relationship Diagram

4.6 API Design

The developed APIs, which contribute to a modular project structure, should be able to provide the following functions (which will be described as individual use cases moving forward with an accompanying Sequence Diagram):

1. Generate NLP Model (Initiate NLP Framework Training Process);
2. Manage Chatbot Actions;
3. Manage Data Sources;
4. Manage Data Extraction Processes;
5. Manage Datasets;
6. Manage Log Entries;
7. Manage Entities;
8. Manage Intents;
9. Get Current Chatbot NLP Model Identifier Labels;
10. Change Active Chatbot NLP Model;
11. Chatbot Framework Instance Start;
12. Chatbot Framework Instance Status;
13. Chatbot Framework Instance Termination;
14. Terminate Current NLP Framework Training Process;
15. NLP Framework Start;
16. NLP Framework Termination;
17. NLP Framework Server Status.

4.6.1 Generate NLP Model (Initiate NLP Framework Training Process)

For this use case, which can be interpreted has generating an NLP Model to be used by a Chatbot Instance, it would be necessary to start a training process utilized by the NLP Framework upon collaborator initialization. Upon receiving this request through the designated Support Component, said component would execute an NLP Framework control

script through the intermediate Chatbot API. If at the start of the process, there is already a registered non completed training process, the user is warned. When the training process is started, it is registered in the system Data Base by PID. When the training process is complete, it is registered by a completion time stamp in the system Data Base. If by for reason (e.g. lack of physical memory required to complete the process) the system should warn the collaborator. The Use Case design is represented by the following Sequence Diagram (Figure 24):

4.6.2 Manage Chatbot Actions

For the management of Actions, the following operations are considered:

1. Creation of new Actions;
2. Reading of a certain Action;
3. Reading of all registered Actions;
4. Update of a registered Action;
5. Deletion of a registered Action.

For the Creation of a new Action, it would be expected of the system to register an input Action's information and to add it to the Chatbot Framework registry (through script) as represented by the following Sequence Diagram (Figure 25):

For the Reading of a certain Action, it would be expected of the system to retrieve from the information of an Action from the Data Base based on the input identifier as represented by the following Sequence Diagram (Figure 23):

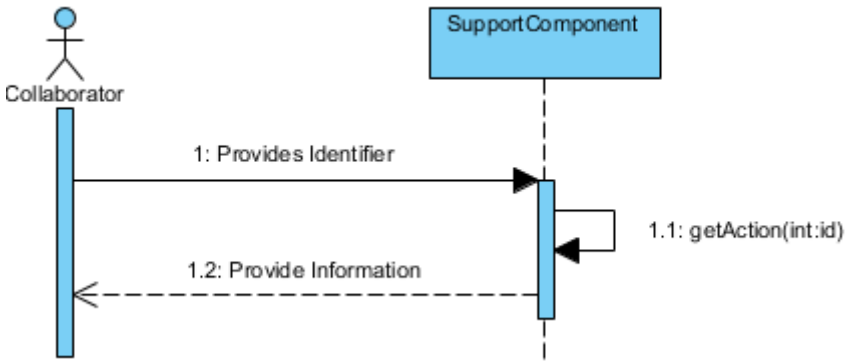

Figure 23 –Action Reading Sequence Diagram

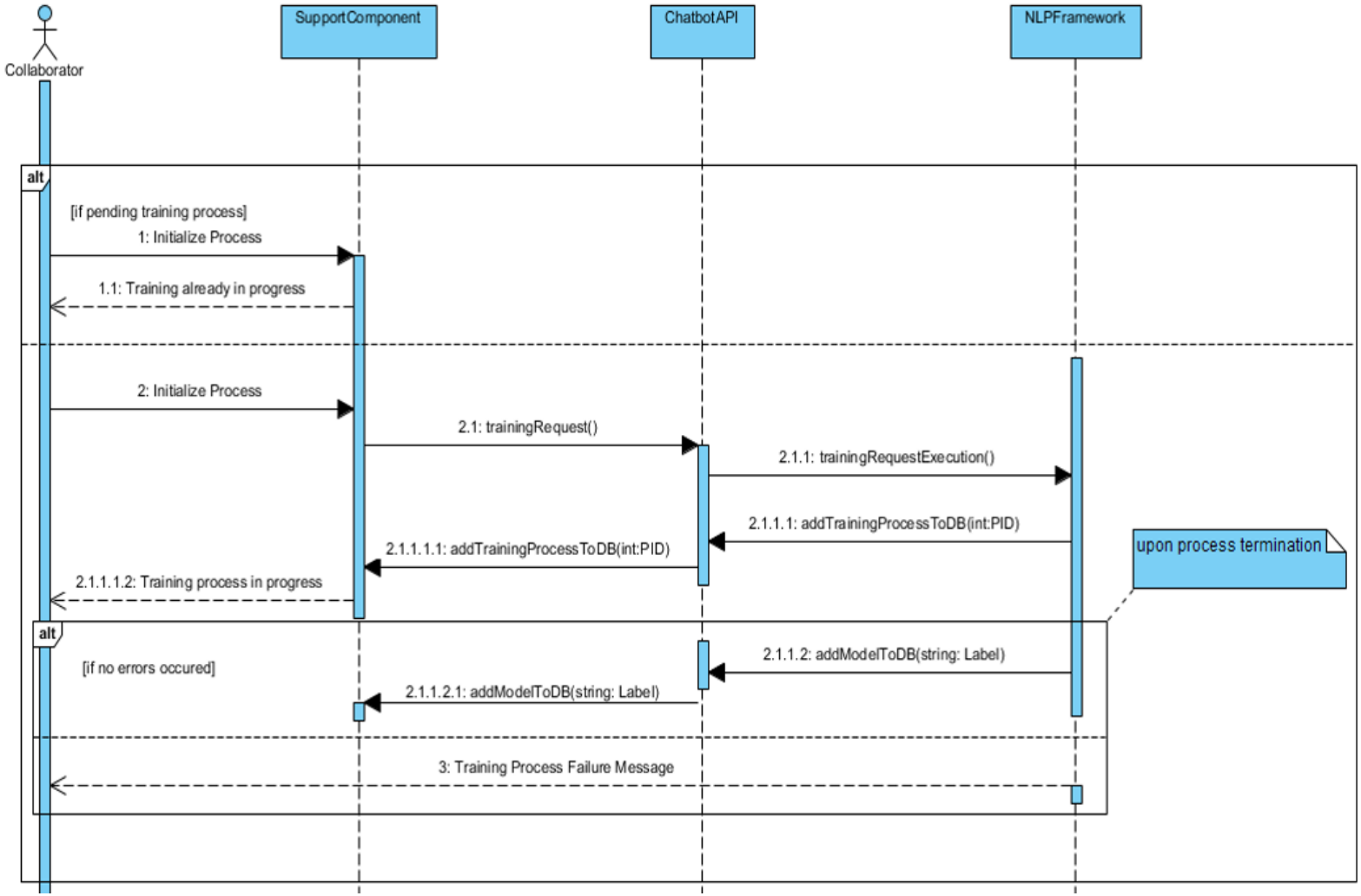

Figure 24 - Generate NLP Model Sequence Diagram

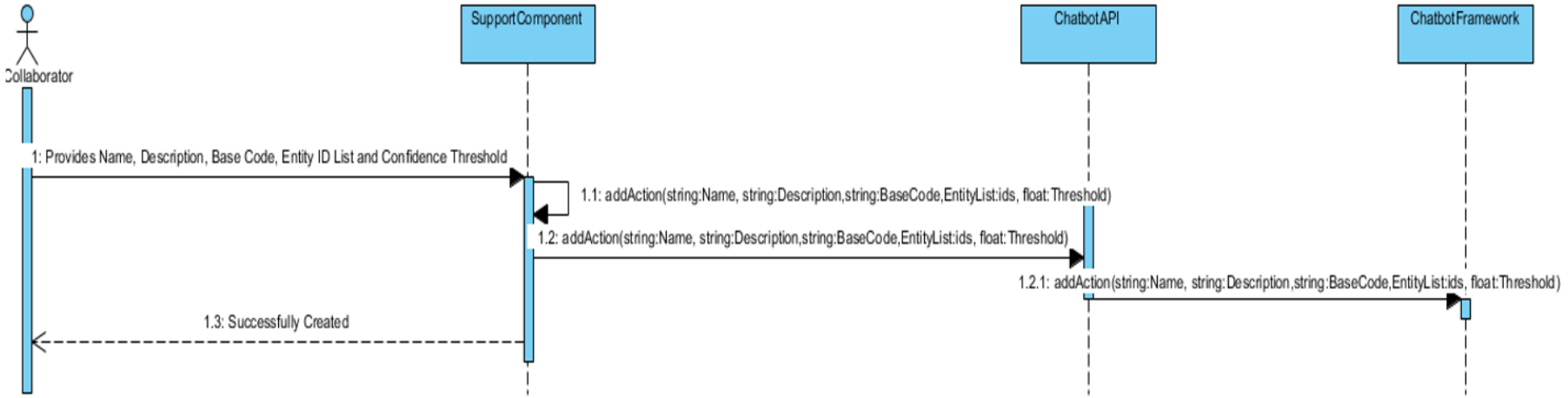

Figure 25 - Action Creation Sequence Diagram

For the Reading of all registered Action, it would be expected of the system to retrieve the information of all Actions from the Data Base as represented by the following Sequence Diagram (Figure 26):

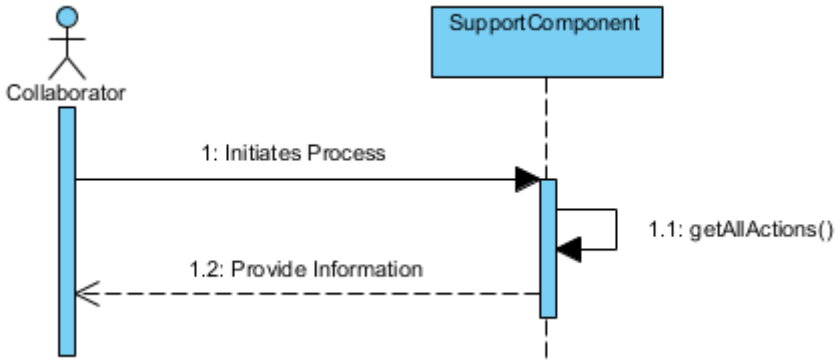

Figure 26 - All Action Reading Sequence Diagram

For the Update of a certain Action, it would be expected of the system to alter the registered information of an Action based on the input fields as represented by the following Sequence Diagram (Figure 28):

For the Deletion of an Action, it would be expected of the system to delete an identified Action's information from the Data Base and from the Chatbot Framework Registry as represented by the following Sequence Diagram (Figure 27):

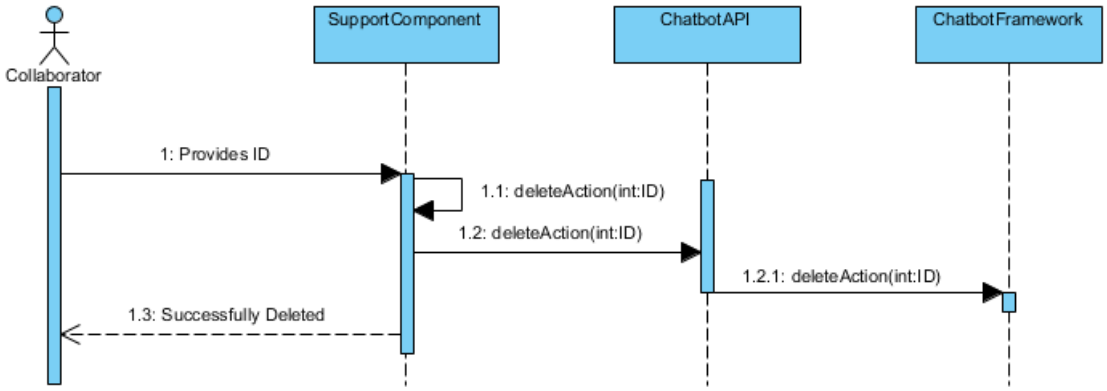

Figure 27 – Action Deletion Sequence Diagram

Figure 28 - Action Update Sequence Diagram

4.6.3 Manage Data Sources

For the management of Data Sources, the following operations are considered:

1. Creation of new Data Source;
2. Reading of a certain Data Source;
3. Reading of all registered Data Sources;
4. Update of a registered Data Source;
5. Deletion of a registered Data Source.

For the Creation of a new Data Source, it would be expected of the system to register an input Data Source information as represented by the following Sequence Diagram (Figure 29):

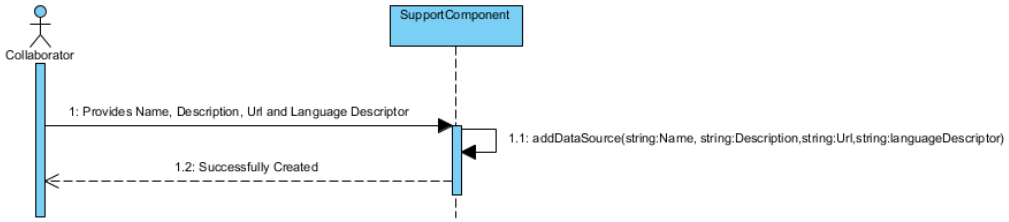

Figure 29 – Data Source Creation Sequence Diagram

For the Reading of a certain Data Source, it would be expected of the system to retrieve the information of an Data Source from the Data Base based on the input identifier as represented by the following Sequence Diagram (Figure 30):

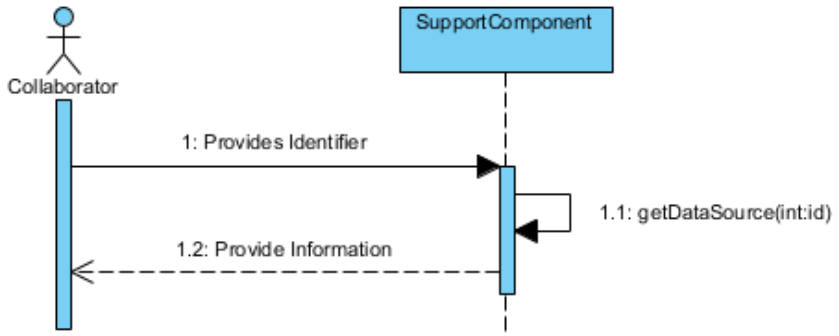

Figure 30 - Data Source Read Sequence Diagram

For the Reading of all registered Data Sources, it would be expected of the system to retrieve the information of all Data Source from the Data Base as represented by the following Sequence Diagram (Figure 31):

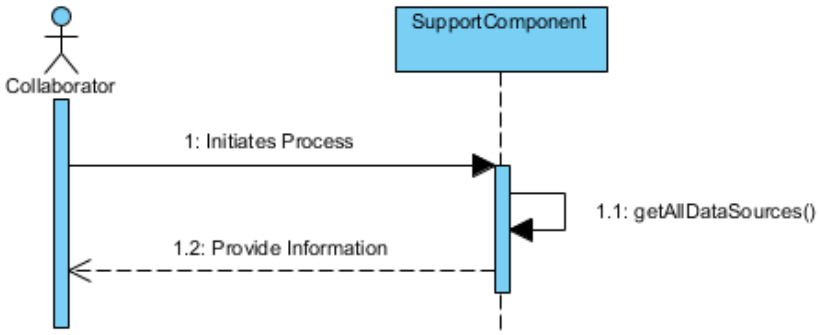

Figure 31 – All Data Source Read Sequence Diagram

For the Update of a certain Data Source, it would be expected of the system to alter the registered information of an Data Source based on the input fields as represented by the following Sequence Diagram (Figure 32):

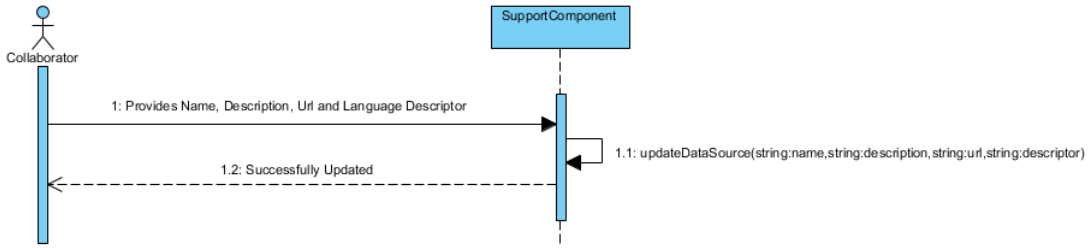

Figure 32 – Update Data Source Sequence Diagram

For the Deletion of an Action, it would be expected of the system to delete an identified Action's information from the Data Base as represented by the following Sequence Diagram (Figure 33):

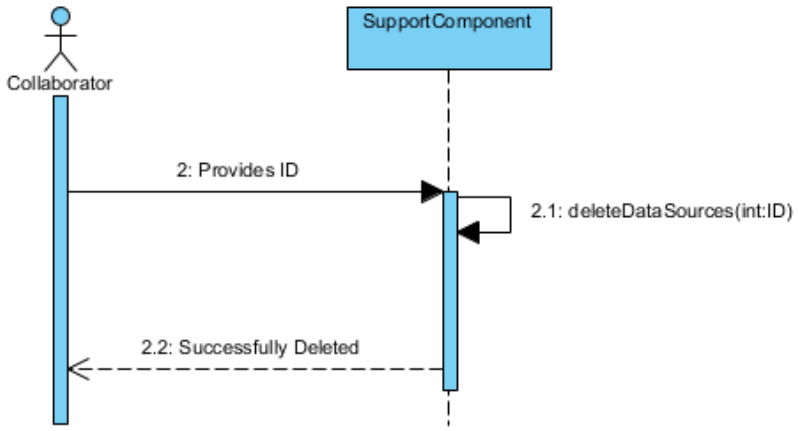

Figure 33 - Delete Data Source Sequence Diagram

4.6.4 Manage Data Extraction Processes

For the management of Data Extraction Processes, the following operations are considered:

1. Creation of new Data Extraction Process;
2. Reading of a certain Data Extraction Process;
3. Reading of all registered Data Extraction Processes;
4. Update of a registered Data Extraction Process;
5. Deletion of a registered Data Extraction Process.

For the Creation of a new Data Extraction Process, it would be expected of the system to register an input Data Extraction Process information as represented by the following Sequence Diagram (Figure 34):

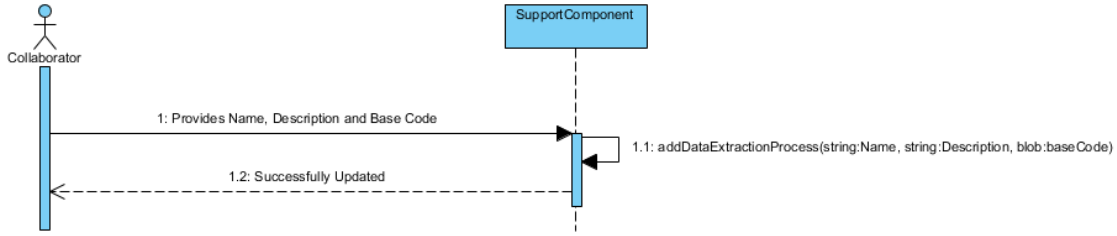

Figure 34 - Create Data Extraction Process Sequence Diagram

For the Reading of a certain Data Extraction Process, it would be expected of the system to retrieve Base the information of a Data Extraction Process from the Data Base on the input identifier as represented by the following Sequence Diagram (Figure 35):

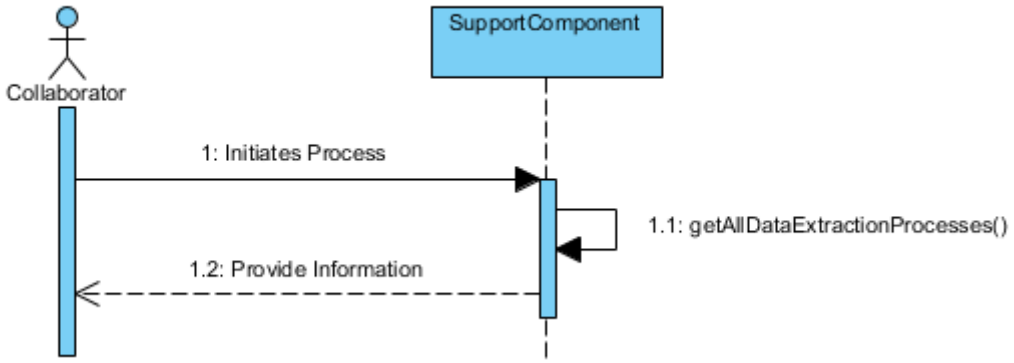

Figure 35 – All Data Extraction Process Retrieval Sequence Diagram

For the Reading of all registered Data Extraction Process, it would be expected of the system to retrieve from the Data Base as represented by the following Sequence Diagram (Figure 36):

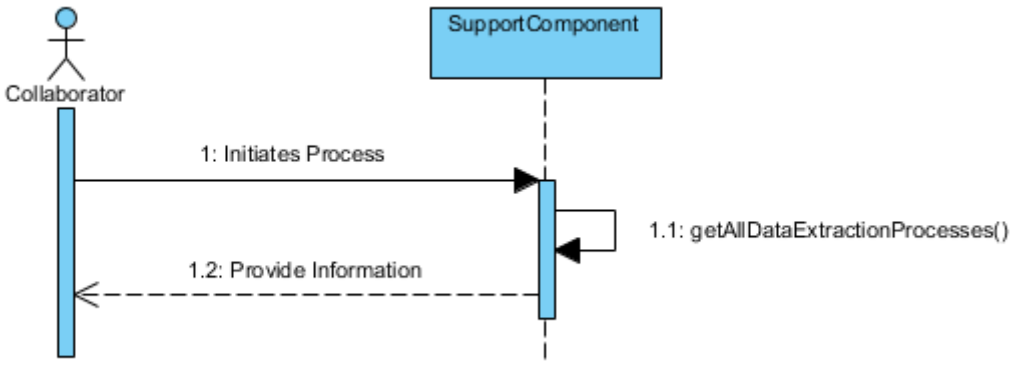

Figure 36 - Data Extraction Process Retrieval Sequence Diagram

For the Update of a certain Data Extraction Process, it would be expected of the system to alter the registered information of a Data Extraction Process based on the input fields as represented by the following Sequence Diagram (Figure 38):

For the Deletion of a Data Extraction Process, it would be expected of the system to delete an identified Data Extraction Process's information from the Data Base as represented by the following Sequence Diagram (Figure 37):

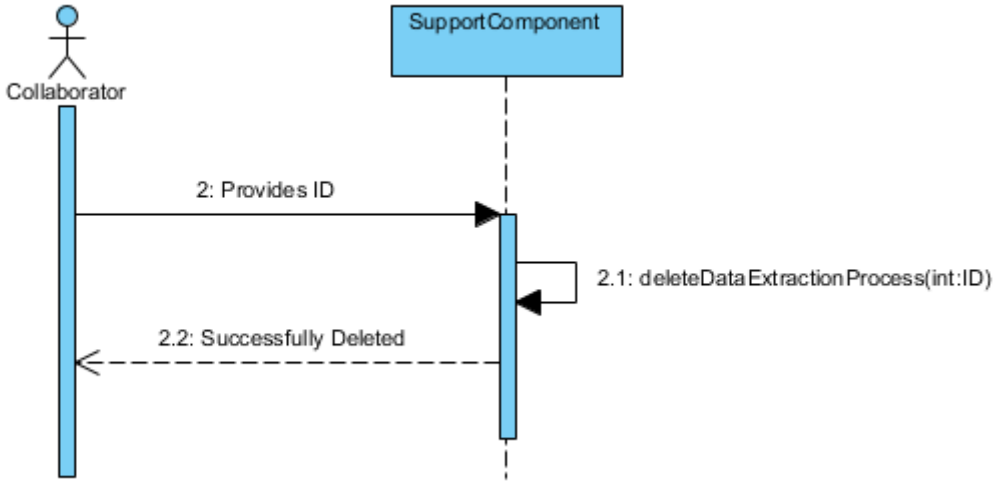

Figure 37 – Data Extraction Process Deletion Sequence Diagram

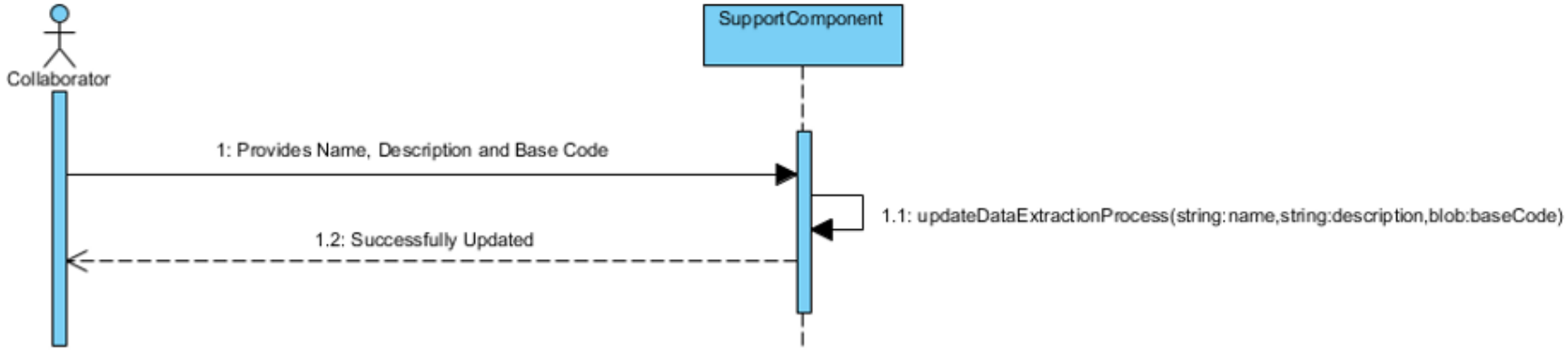

Figure 38 - Data Extraction Process Update Sequence Diagram

4.6.5 Manage Data Sets

For the management of Data Sets, the following operations are considered:

1. Creation of new Data Set;
2. Reading of a certain Data Set;
3. Reading of all registered Data Sets;
4. Deletion of a registered Data Set.

For the Creation of a new Data Set, it would be expected of the system to extract training examples from registered Data Sources as per the selected Data Extraction Process. Afterwards it would be necessary to register the created Data Set information after represented by the following Sequence Diagram (Figure 39):

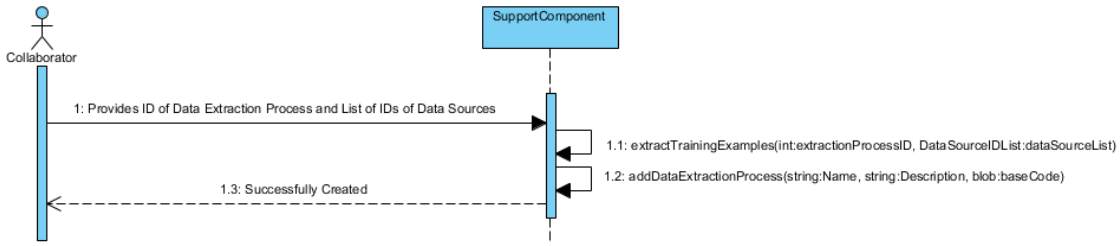

Figure 39 - Data Set Creation Sequence Diagram

For the Reading of a certain Data Set, it would be expected of the system to retrieve Base the information of a Data Set from the Data Base on the input identifier as represented by the following Sequence Diagram (Figure 40):

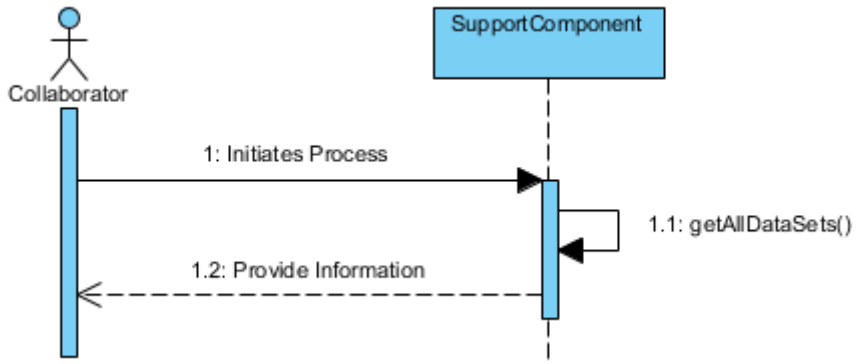

Figure 40 - Data Set Retrieval Sequence Diagram

For the Reading of all registered Data Set, it would be expected of the system to retrieve from the Data Base as represented by the following Sequence Diagram (Figure 41):

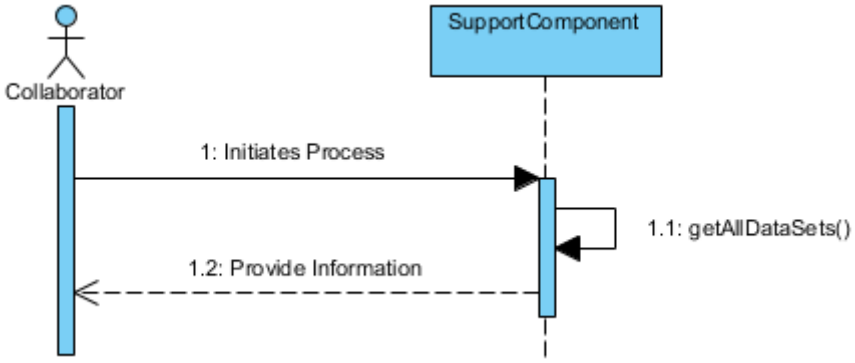

Figure 41 - All Data Sets Retrieval Sequence Diagram

For the Deletion of a Data Set, it would be expected of the system to delete an identified Data Set's information from the Data Base as represented by the following Sequence Diagram (Figure 42):

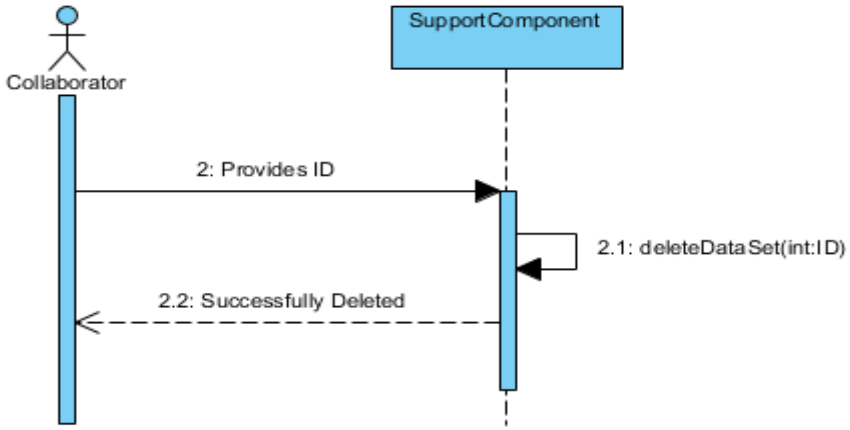

Figure 42 - Data Set Deletion Sequence Diagram

4.6.6 Manage Log Entries

For the management of Log Entries, the following operations are considered:

1. Creation of new Log Entry;
2. Reading of all registered Log Entries.

For the Creation of a new Log Entry, it would be expected of the system to register an input Log Entry information after represented by the following Sequence Diagram (Figure 43):

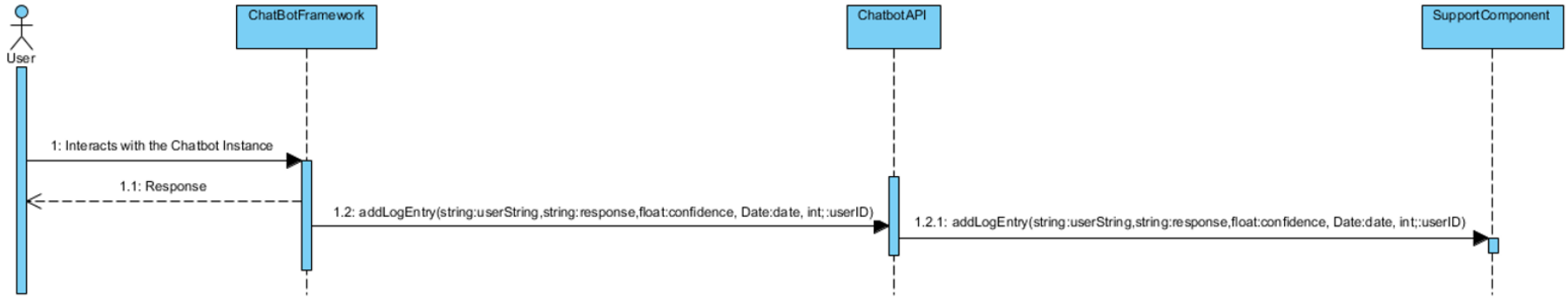

Figure 43 - Log Entry Creation Sequence Diagram

For the Reading of all registered Log Entries, it would be expected of the system to retrieve from the Data Base as represented by the following Sequence Diagram (Figure 44):

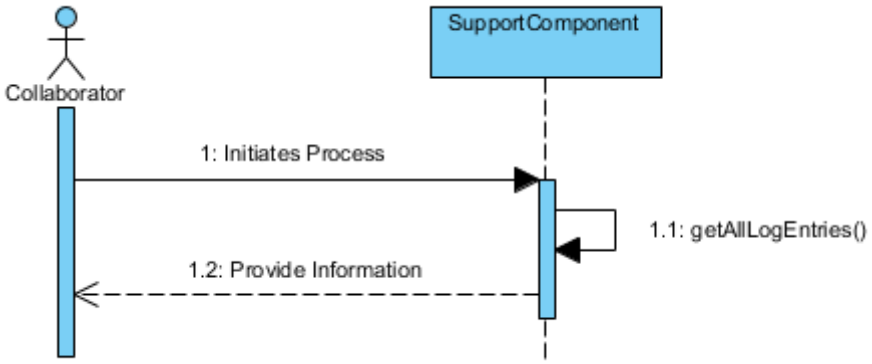

Figure 44 – All Log Entries Retrieval Sequence Diagram

4.6.7 Manage Entities

For the management of Entities, the following operations are considered:

1. Creation of new Entity;
2. Reading of a certain Entity;
3. Reading of all registered Entities;
4. Update of a registered Entity;
5. Deletion of a registered Entity.

For the Creation of a new Entity, it would be expected of the system to register an input Entity information after represented by the following Sequence Diagram (Figure 45):

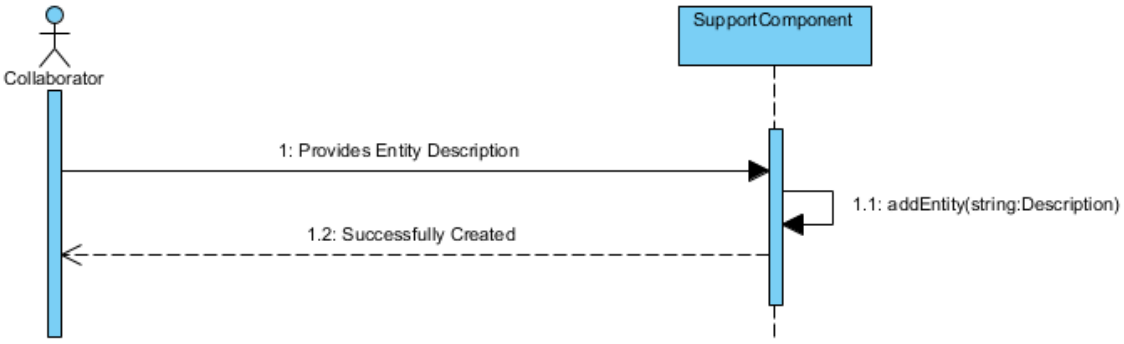

Figure 45 - Entity Creation Sequence Diagram

For the Reading of a certain Entity, it would be expected of the system to retrieve Base the information of an Entity from the Data Base on the input identifier as represented by the following Sequence Diagram (Figure 46):

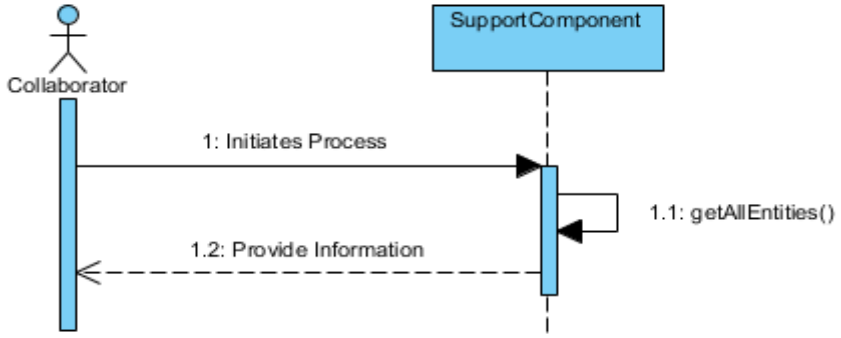

Figure 46 - All Entity Retrieval Sequence Diagram

For the Reading of all registered Entities, it would be expected of the system to retrieve from the Data Base as represented by the following Sequence Diagram (Figure 47):

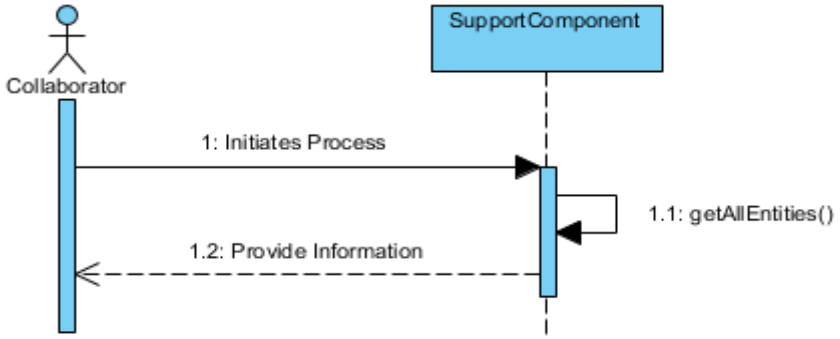

Figure 47 - Entity Retrieval Sequence Diagram

For the Update of a certain Entity, it would be expected of the system to alter the registered information of an Entity based on the input fields as represented by the following Sequence Diagram (Figure 48):

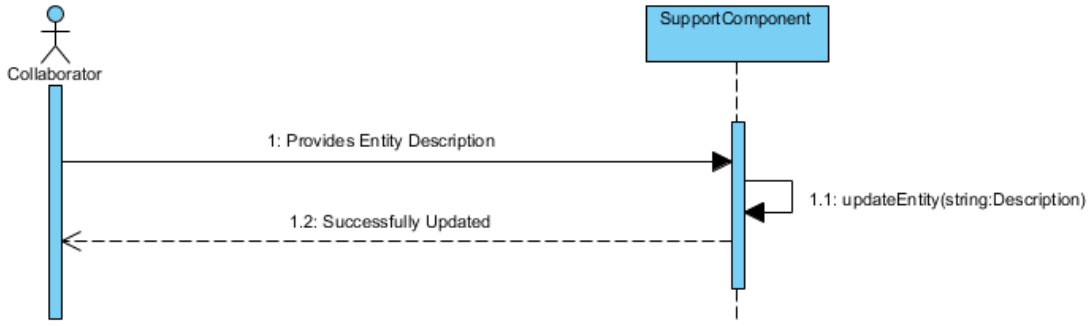

Figure 48 - Entity Update Sequence Diagram

For the Deletion of an Entity, it would be expected of the system to delete an identified Entity's information from the Data Base as represented by the following Sequence Diagram (Figure 49):

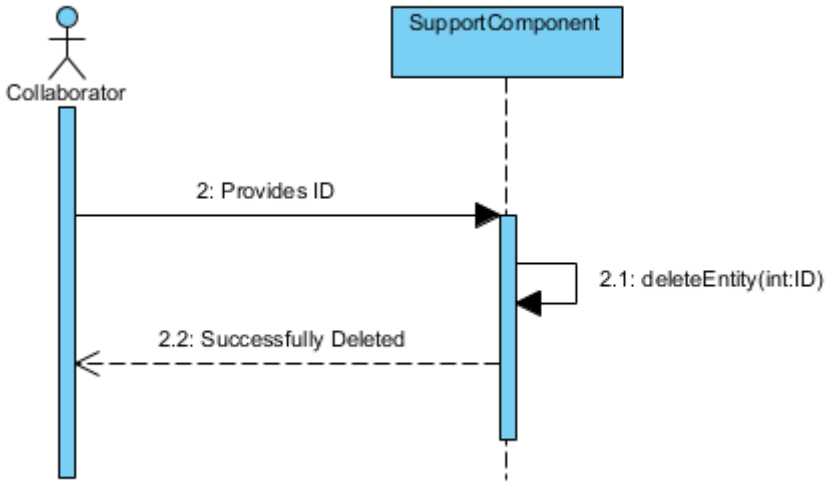

Figure 49 - Entity Deletion Sequence Diagram

4.6.8 Manage Intents

For the management of Intents, the following operations are considered:

1. Creation of new Intent;
2. Reading of a certain Intent;
3. Reading of all registered Intents;
4. Update of a registered Intent;
5. Deletion of a registered Intent.

For the Creation of a new Intent, it would be expected of the system to register an input Intent information after represented by the following Sequence Diagram (Figure 50):

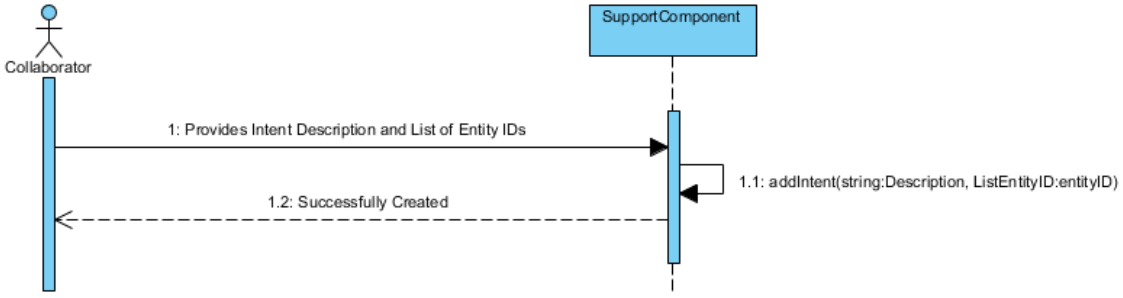

Figure 50 - Intent Creation Sequence Diagram

For the Reading of a certain Intent, it would be expected of the system to retrieve Base the information of an Intent from the Data Base on the input identifier as represented by the following Sequence Diagram (Figure 51):

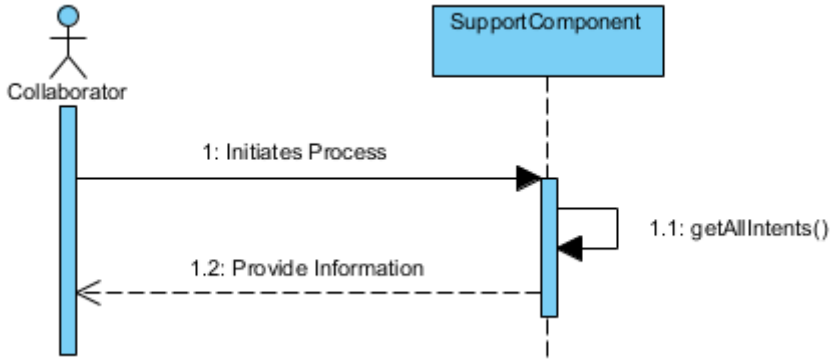

Figure 51 - Intent Retrieval Sequence Diagram

For the Reading of all registered Intents, it would be expected of the system to retrieve from the Data Base as represented by the following Sequence Diagram (Figure 52):

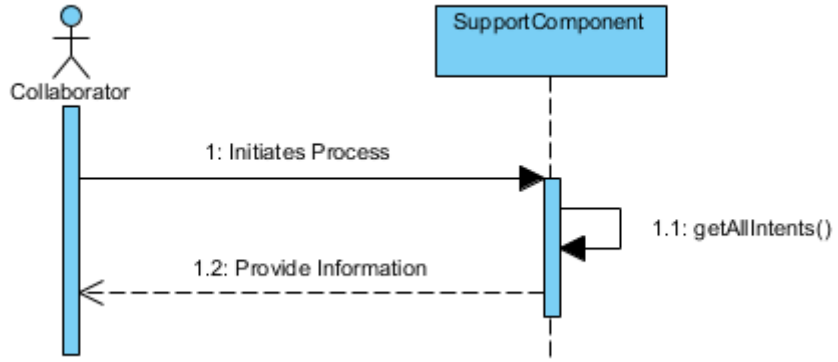

Figure 52 - All Intents Retrieval Sequence Diagram

For the Update of a certain Intent, it would be expected of the system to alter the registered information of an Intent based on the input fields as represented by the following Sequence Diagram (Figure 53):

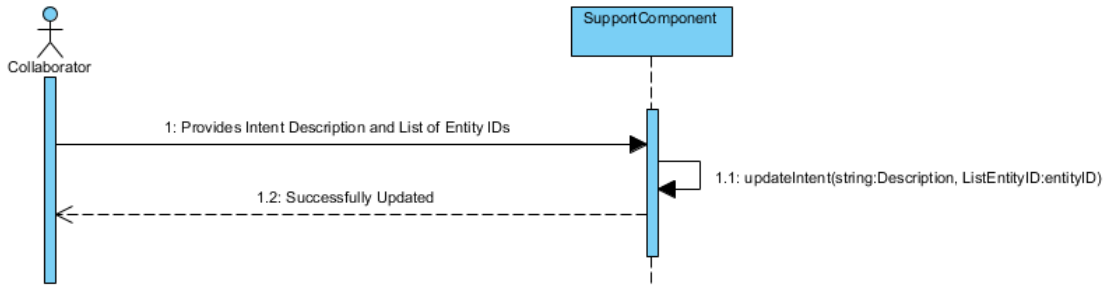

Figure 53 - Intent Update Sequence Diagram

For the Deletion of an Intent, it would be expected of the system to delete an identified Intent's information from the Data Base as represented by the following Sequence Diagram (Figure 54):

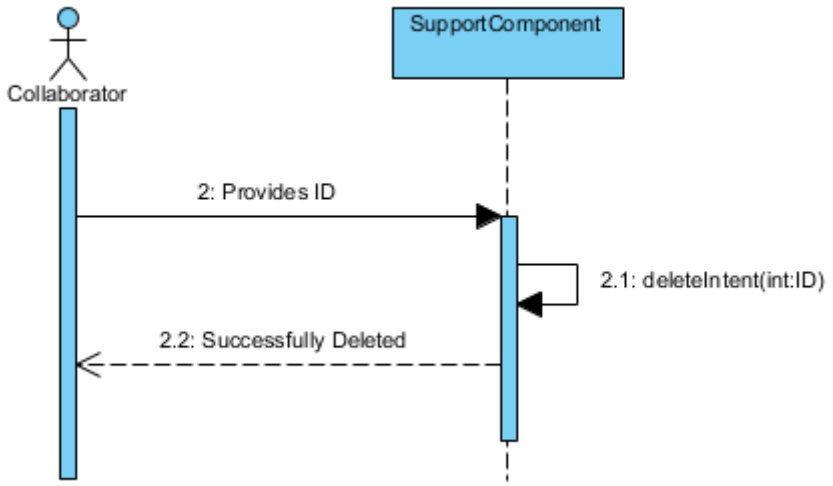

Figure 54 - Intent Deletion Sequence Diagram

4.6.9 Get Current NLP Framework Model Identifier Labels

For the retrieval of identifier labels for all the generated models present in the NLP Framework registry, the system would be required to read and process all available labels (through script) as represented by the following Sequence Diagram (Figure 55):

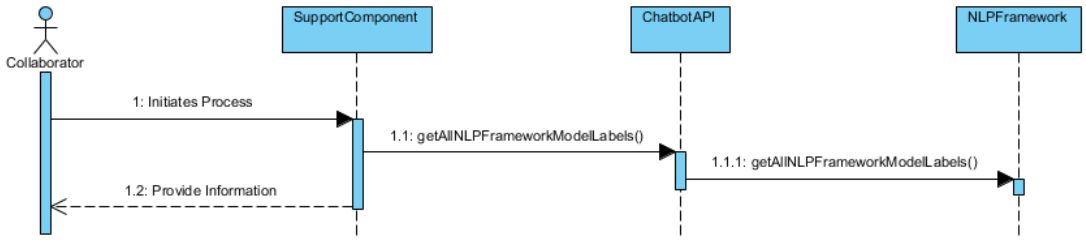

Figure 55 – Get Current NLP Framework Model Identifier Labels Sequence Diagram

4.6.10 Change Active Chatbot NLP Model

With this Use Case, it is expected for a collaborator to be able to change the NLP Model (generated and present by the NLP Framework) which the Chatbot Framework would utilize when communicating to requesting users. It would be expected of the system to be able to modify the active model identification pointers (indicative of the model which would be run and served through a API Service to the Chatbot instance) as represented by the following Sequence Diagram (Figure 56):

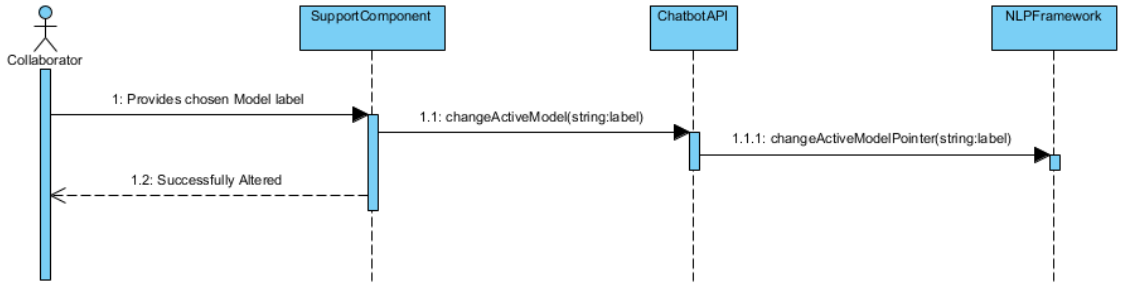

Figure 56 - Change Active Chatbot NLP Model Sequence Diagram

4.6.11 Chatbot Framework Instance Start

With this Use Case, it is expected for a collaborator to instantiate the utilized Chatbot Framework for public use. It would be required of the system to verify if an instance is already running at the time of request (warning if so) and running an initialization script if no instance is running. The following Sequence Diagram represents the implemented operation (Figure 57):

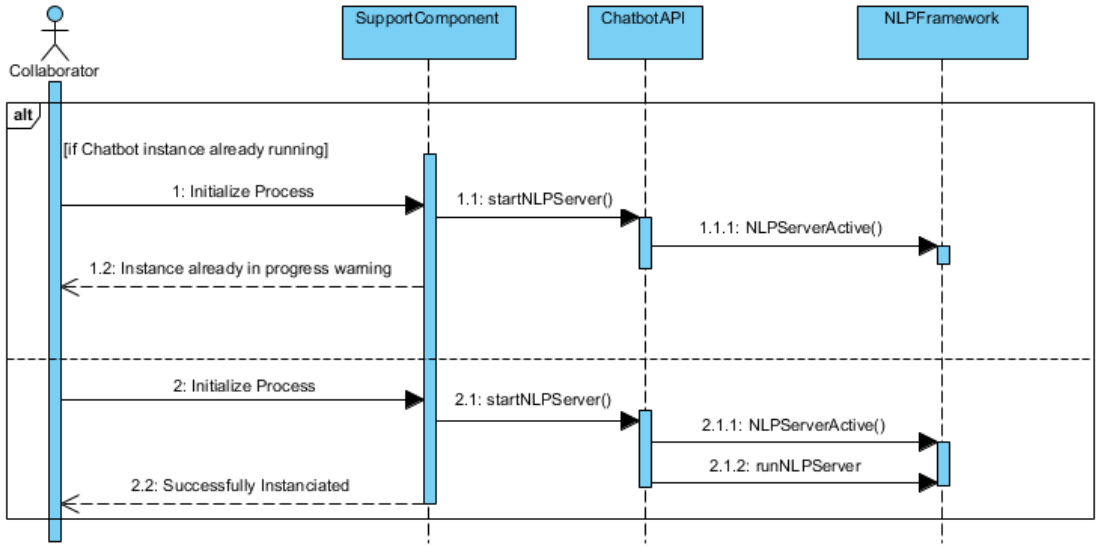

Figure 57 - Chatbot Instantiation Sequence Diagram

4.6.12 Chatbot Framework Instance Status

With this Use Case, it is expected for a Collaborator to verify if there is an active instance of the Chatbot Framework as represented by the following Sequence Diagram (Figure 58):

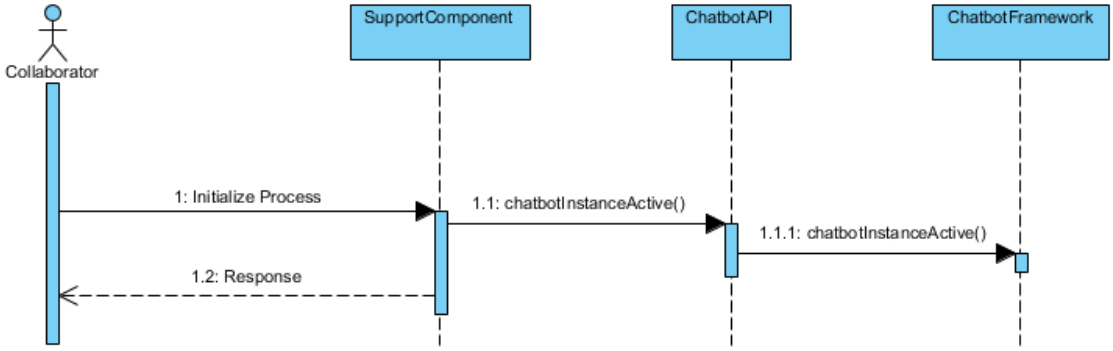

Figure 58 - Chatbot Framework Instance Status Sequence Diagram

4.6.13 Chatbot Framework Instance Termination

With this Use Case, it is expected for a Collaborator to terminate an active instance of the Chatbot Framework as represented by the following Sequence Diagram (Figure 59):

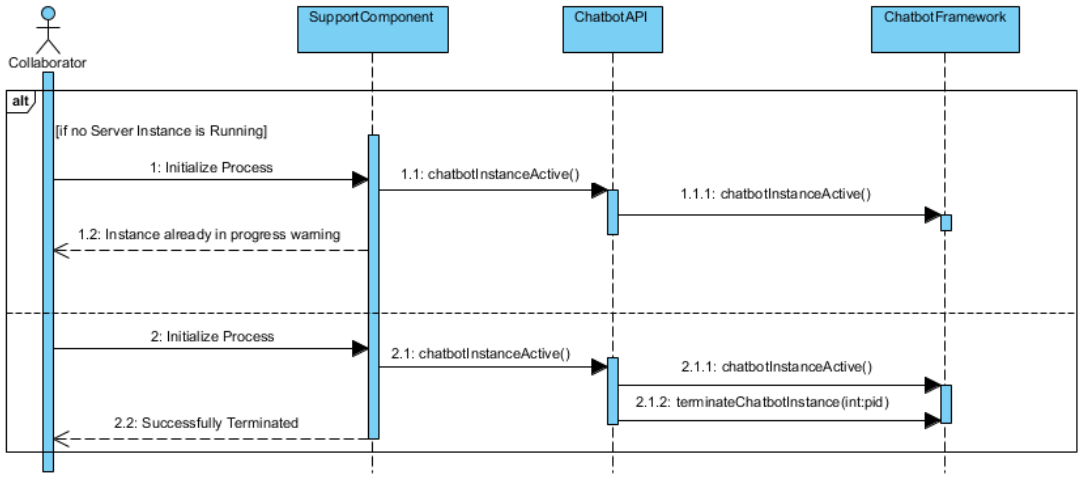

Figure 59 - Chatbot Framework Instance Termination Sequence Diagram

4.6.14 Terminate Current NLP Framework Training Process

With this Use Case, it is expected for a Collaborator to terminate an active NLP Framework Training process as represented by the following Sequence Diagram (Figure 60):

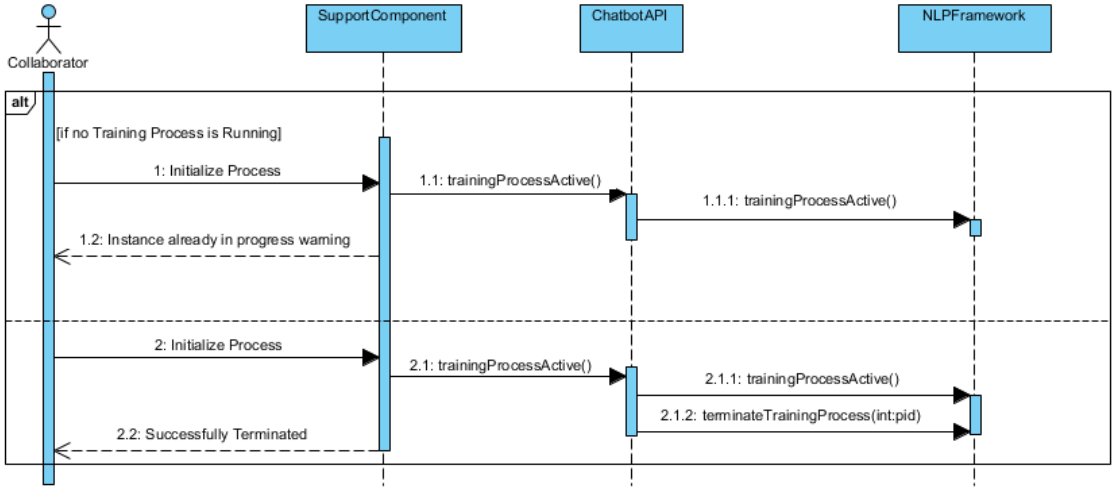

Figure 60 - Terminate Current NLP Framework Training Process Sequence Diagram

4.6.15 NLP Framework Server Start

With this Use Case, it is expected for a collaborator to instantiate an NLP Framework Server utilized by the Chatbot Framework. It would be required of the system to verify if an instance is already running at the time of request (warning if so) and running an initialization script if no instance is running. The following Sequence Diagram represents the implemented operation (Figure 61):

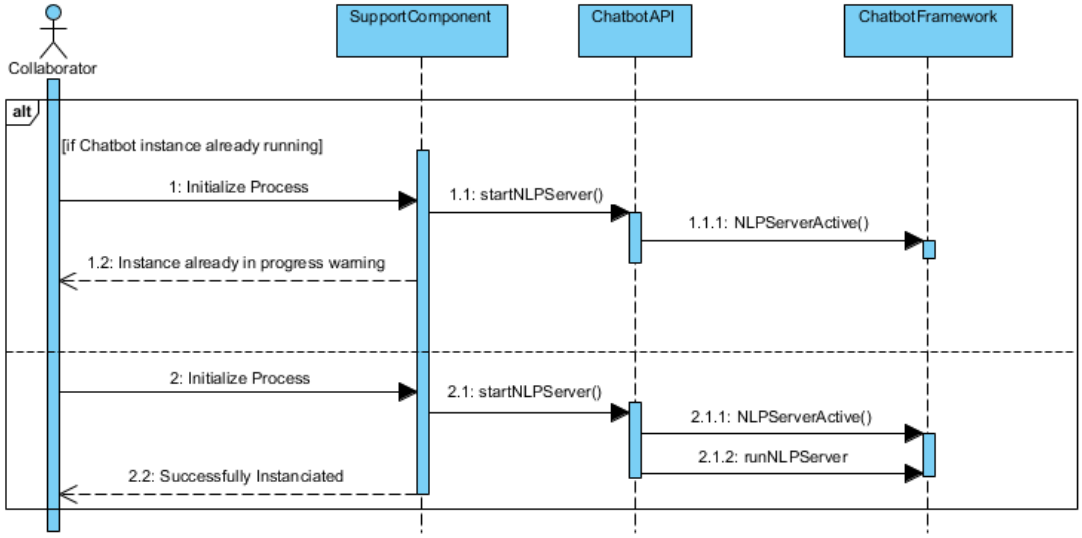

Figure 61 - NLP Server Instantiation Sequence Diagram

4.6.16 NLP Framework Server Status

With this Use Case, it is expected for a Collaborator to verify if there is an active NLP Framework Server instance as represented by the following Sequence Diagram (Figure 62):

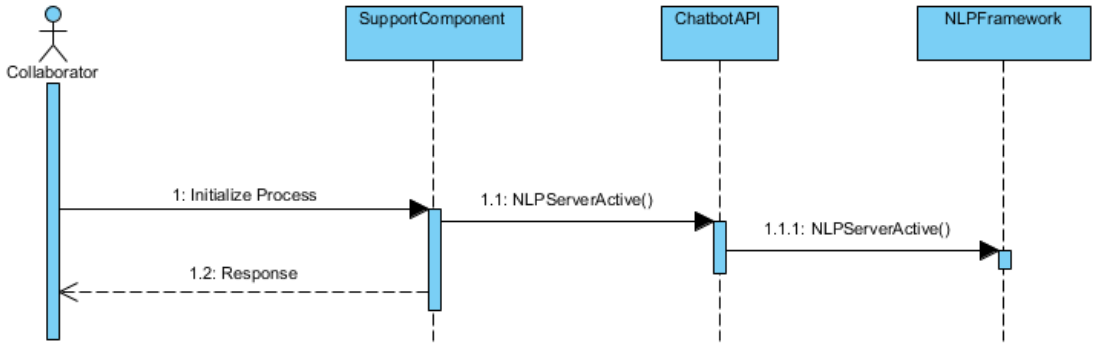

Figure 62 - NLP Framework Server Status Sequence Diagram

4.6.17 NLP Framework Server Termination

With this Use Case, it is expected for a Collaborator to terminate an active NLP Framework Server instance as represented by the following Sequence Diagram (Figure 63):

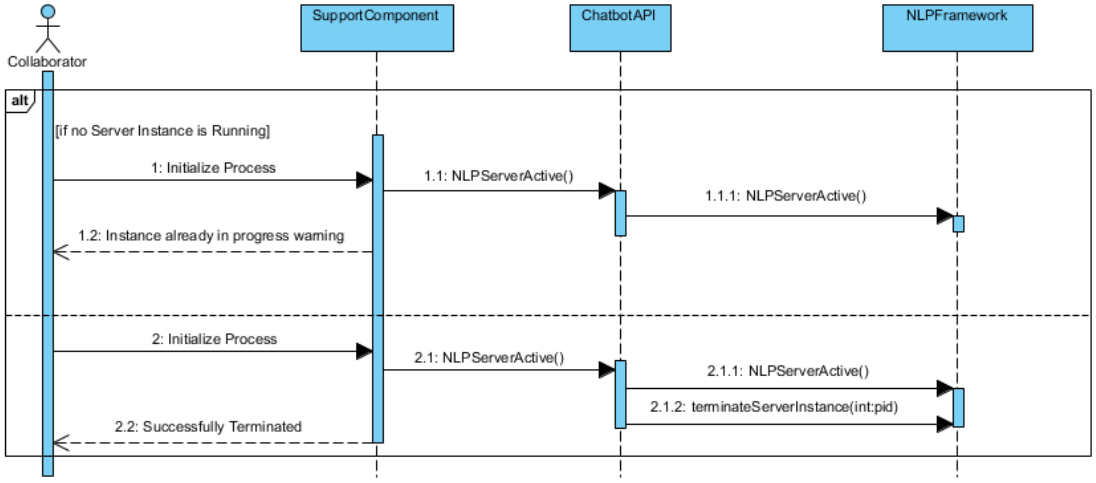

Figure 63 - NLP Server Instance Termination Sequence Diagram

4.7 Summary

This chapter included requirement gathering performed in order to obtain both functional and non-functional requirements necessary to the solution of this work's problem. Afterwards, the core concepts to the problem's domain were presented and briefly described.

The required businesses processes were later identified, as well as component structure of the projected solution. Afterwards, the structure of the data, which is used by the defined solution, as well as its origin, are presented. Finally, the sequence logic for all operations which would be directed through the designed API's and components were shown and discussed.

5 Development

This section will include the considerations required to fully develop the previously proposed solution. Development time past Literature Review, Value Analysis and Design was subdivided into data gathering and pre-processing, as well as implementation and testing. Data Gathering and Data Pre-processing processes will be shown separately, and the test results enumerated and discussed.

5.1 Data Gathering

In terms of pre-existing data of past user interactions and their intent, it must be mentioned that the company site included a customer support option which redirected to the Customer Support team. Before a ticket could be directed to the Customer Support team, a customer could enquiry the system through a quick search option which did a string match operation between the user string and the company's existing tutorial and FAQ articles. When retrieved, the system would enquire if the quick search was helpful in any way and all inquiry information would be recorded. This consisted on the only pertinent data available to create and NLP-based model to and their intents within the company's service domain.

The Data Gathering process consisted of providing an extraction, transform and load routines for these records made available through a pre-existing dedicated RESTful API service. The extracted tuples consisted of thousands of user utilization records with columns pertaining costumer information, costumer string, language indicator, resulting articles (as well as their keywords) and search usefulness. Many of these features were useless to the problem at hand and were removed as well as any tuples that were considered invalid by the user. The resulting sets could be described as log entries of the user string and resulting articles information separated by language. It must be mentioned that from an ideal list feature intent, not all of them were present in the records.

5.2 Data Pre-processing

Before any model could be trained with the aforementioned generated extraction sets, several different transformations were made to achieve a set of NLP associations between user string with an underlying intent and its present entities. For every previous tuple, several data entries were created associating the user string with each individual detected verb and noun as an intent along with the detected entities as such:

$$"Send\ campaign\ to\ my\ associated\ contacts."$$

Yielded as a result:

```
1.  {
2.     "rasa_nlu_data": {
3.        "common_examples": [{
4.          text:"Send campaign to my associated contacts.",
5.          intent:"Send campaign",
6.          entities:["campaign","contacts"]
7.        },
8.        {
9.          "common_examples": [{
10.         text:"Send campaign to my associated contacts.",
11.         intent:"Send contacts",
12.         entities:["campaign","contacts"]
13.       }],
14.    }
15. }
```

Several transformations were done to the base process in order to try and include more and increase their overall confidence detection levels. Each of these transformation hypotheses included a separate generation procedure complete with the formation of a JSON file for the usage of the NLP Framework with a total of one hundred and twenty-six thousand training examples:

- Stemmatization of the user string, intents and entities – each word present is stemmatized;
- Lemmatization for the user string, intents and entities – each word present is lemmatized;
- Manual addition of additional entities and business specific words - selected entities and business specific words are introduced to the training example on match;
- Stemmatization of the user string, intents and entities with manual addition of additional entities and business specific words – each word present is stemmatized, including the selected entities and business specific words which are introduced to the training example on match;
- Lemmatization of the user string, intents and entities with manual addition of additional entities and business specific words - each word present is lemmatized, including the selected entities and business specific words which are introduced to the training example on match;

- Inclusion of known entity synonyms to the entity list of each case – utilizing a synonym dictionary, entity synonyms were added to the training example on match;
- Stemmatization of the user string, intents and entities with inclusion of known entity synonyms to the entity list of each case - utilizing a synonym dictionary, entity synonyms were added to the training example on match. Afterwards, every word in the training examples is stemmatized;
- Lemmatization for the user string, intents and entities with of known entity synonyms to the entity list of each case - utilizing a synonym dictionary, entity synonyms were added to the training example on match. Afterwards, every word in the training examples is lemmatized;
- Stemmatization of the user string, intents and entities with manual addition of additional entities and business specific words. Additional inclusion of known entity synonyms to the entity list of each case - each word present is stemmatized, including the selected entities and business specific words (as well as their synonyms) which are introduced to the training example on match;
- Lemmatization of the user string, intents and entities with manual addition of additional entities and business specific words. Additional inclusion of known entity synonyms to the entity list of each case - each word present is lemmatized, including the selected entities and business specific words (as well as their synonyms) which are introduced to the training example on match.

5.3 Testing

This section includes the results of several types of functional tests made to the developed system. It will include Functional, Acceptance and Integration testing.

5.3.1 Smoke Testing

Smoke tests are defined as: *"A subset of all defined/planned test cases that cover the main functionality of a component or system, to ascertaining that the most crucial functions of a program work, but not bothering with finer details."* [125].

A round of smoke tests, much like a checkup on the general health of the developed system, is done by the documenting the sequential evaluation of critical functionalities. In this case, it may include the verification of the correct high-level execution of each mentioned component, for instance.

5.3.1.1 Support Component Smoke Tests

- Initiate NLP Framework Training Process.....Success
- Manage Chatbot Actions.....Success
- Manage Data Sources .....Success
- Manage Data Extraction Processes .....Success

- Manage Datasets ..... Success
- Manage Log Entries ..... Success
- Manage Entities ..... Success
- Manage Intents ..... Success
- Get Current Chatbot Model Identifier Labels ..... Success
- Change Active Chatbot NLP Model ..... Success
- Chatbot Framework Instance Start ..... Success
- Chatbot Framework Instance Status ..... Success
- Chatbot Framework Instance Termination ..... Success
- Terminate Current NLP Framework Training Process ..... Success
- NLP Framework Start ..... Success
- NLP Framework Status ..... Success
- NLP Framework Termination ..... Success

5.3.1.2 Chatbot API Smoke Tests

- Initiate NLP Framework Training Process ..... Success
- Manage Chatbot Actions ..... Success
- Get Current Chatbot Model Identifier Labels ..... Success
- Change Active Chatbot NLP Model ..... Success
- Chatbot Framework Instance Start ..... Success
- Chatbot Framework Instance Status ..... Success
- Chatbot Framework Instance Termination ..... Success
- Terminate Current NLP Framework Training Process ..... Success
- NLP Framework Start ..... Success
- NLP Framework Status ..... Success
- NLP Framework Termination ..... Success

5.3.1.3 NLP Framework Smoke Tests

- Initiate NLP Framework Training Process ..... Success
- Get Current Chatbot Model Identifier Labels ..... Success
- Terminate Current NLP Framework Training Process ..... Success
- NLP Framework Start ..... Success
- NLP Framework Status ..... Success
- NLP Framework Termination ..... Success

5.3.1.4 Chatbot Framework Smoke Tests

- Manage Chatbot Actions ..... Success

- Change Active Chatbot NLP Model ..... Success
- Chatbot Framework Instance Start ..... Success
- Chatbot Framework Instance Status ..... Success
- Chatbot Framework Instance Termination ..... Success

5.3.2 Acceptance Testing

Acceptance tests are defined as: "*Formal testing with respect to user needs, requirements, and business processes conducted to determine whether or not a system satisfies the acceptance criteria and to enable the user, customers or other authorized entity to determine whether or not to accept the system*." [125].

As for the project's acceptance tests, a list of predicted user expectations was defined each Use Case as well as the final user experience. Some of these requirements include quick response times and correct information storage for each individual use case as well as having most intents being accurately predicted, high confidence (over 75% in an initial phase) in predicted intents and a wide range of known E-goi contextual intents for each tested NLP model.

5.3.2.1 Quick response time and correct information storage

- Initiate NLP Framework Training Process ..... Success
- Manage Chatbot Actions ..... Success
- Manage Data Sources ..... Success
- Manage Data Extraction Processes ..... Success
- Manage Datasets ..... Success
- Manage Log Entries ..... Success
- Manage Entities ..... Success
- Manage Intents ..... Success
- Get Current Chatbot Model Identifier Labels ..... Success
- Change Active Chatbot NLP Model ..... Success
- Chatbot Framework Instance Start ..... Success
- Chatbot Framework Instance Status ..... Success
- Chatbot Framework Instance Termination ..... Success
- Terminate Current NLP Framework Training Process ..... Success
- NLP Framework Start ..... Success
- NLP Framework Status ..... Success
- NLP Framework Termination ..... Success

5.3.2.2 NLP Accuracy and Confidence

- Stemmatization of the user string, intents and entities ..... Failure
- Lemmatization of the user string, intents and entities ..... Failure
- Manual addition of additional entities and business specific words ..... Failure
- Stemmatization of the user string, intents and entities with manual addition of additional entities and business specific words ..... Failure
- Lemmatization of the user string, intents and entities with manual addition of additional entities and business specific words ..... Failure
- Inclusion of known entity synonyms to the entity list of each case ..... Failure
- Stemmatization of the user string, intents and entities with inclusion of known entity synonyms to the entity list of each case ..... Failure
- Lemmatization for the user string, intents and entities with of known entity synonyms to the entity list of each case ..... Failure
- Stemmatization of the user string, intents and entities with manual addition of additional entities and business specific words. Additional inclusion of known entity synonyms to the entity list of each case ..... Failure
- Lemmatization of the user string, intents and entities with manual addition of additional entities and business specific words. Additional inclusion of known entity synonyms to the entity list of each case ..... Failure

5.3.2.3 Wide range of correctly predicted intents

- Stemmatization of the user string, intents and entities ..... Failure
- Lemmatization of the user string, intents and entities ..... Failure
- Manual addition of additional entities and business specific words ..... Failure
- Stemmatization of the user string, intents and entities with manual addition of additional entities and business specific words ..... Failure
- Lemmatization of the user string, intents and entities with manual addition of additional entities and business specific words ..... Failure
- Inclusion of known entity synonyms to the entity list of each case ..... Failure
- Stemmatization of the user string, intents and entities with inclusion of known entity synonyms to the entity list of each case ..... Failure
- Lemmatization for the user string, intents and entities with of known entity synonyms to the entity list of each case ..... Failure
- Stemmatization of the user string, intents and entities with manual addition of additional entities and business specific words. Additional inclusion of known entity synonyms to the entity list of each case ..... Failure
- Lemmatization of the user string, intents and entities with manual addition of additional entities and business specific words. Additional inclusion of known entity synonyms to the entity list of each case ..... Failure

From the range of executed tests, it was perceptible that although the response times were acceptable for a real word costumer interaction, the range of answers provided by the NLP models as well as its accuracy and confidence weren't. Most test cases devised by organizational staff were not met with acceptable answers. In terms of expected range of predicted intents, many were not even present in the training examples generated from available data. In terms of accuracy, no data set could accurately predict half the devised test case intents. Finally, in terms of confidence, for all discussed data sets, the majority of the accurately predicted intents were predicted with very low confidence (below 20%), were confidences above 50% being a rarity. This deficiency was attributed to the lack of quality in available data from which the training sets were generated. Several problems were identified with said data: it was verified that only a fraction of intended intents was present and that the majority of intents which the enterprise would require to consider a valid solution weren't featured. Another major problem was the large portion of cases which were either unrelated with the business or were invalid uses of language. Another prevalent type of invalid cases consisted of entries which were abnormally long and who possessed a large amount of text unrelated to the intent the entry was used to transmit. As such, valid concise cases were few and far between and their usage in generating valid trainings proved insufficient.

5.3.3 Integration Testing

Integration tests are defined as: "*Testing performed to expose defects in the interfaces and in the interactions between integrated components or systems*." [125]. Integration tests differs from other test cases in the sense it focuses mainly on the interfaces & flow of data/information between the developed internal modules.

In this case, a round of integration tests will include the analysis of each communication between the presented components for each functionality.

- Initiate NLP Framework Training Process ..... Success
- Manage Chatbot Actions ..... Success
- Manage Data Sources ..... Success
- Manage Data Extraction Processes ..... Success
- Manage Datasets ..... Success
- Manage Log Entries ..... Success
- Manage Entities ..... Success
- Manage Intents ..... Success
- Get Current Chatbot Model Identifier Labels ..... Success
- Change Active Chatbot NLP Model ..... Success
- Chatbot Framework Instance Start ..... Success
- Chatbot Framework Instance Status ..... Success
- Chatbot Framework Instance Termination ..... Success
- Terminate Current NLP Framework Training Process ..... Success
- NLP Framework Start ..... Success
- NLP Framework Status ..... Success
- NLP Framework Termination ..... Success

5.4 Summary

Through this section the data gathering, and processing routines were discussed where an unprocessed batch of multilingual records were transformed into training sets appropriate for the chosen NLP Framework. Afterwards the results of each functional test on the developed solution and the produced training set variants are presented. From the executed smoke, acceptance and integration tests it is perceptible to that the developed overarching system is working as intended. However, acceptance tests show that all the considered hypotheses for training sets are not capable of having acceptable results in terms of predicted intents, prediction confidence and range of known intents. This was attributed to an obvious lack of quality examples to train the NLP models with.

6 Conclusions

The organizational background and motivations for this project was first introduced. The projects problem, objectives and solution approach were formally described. Subsequently, this document included a thorough literature review which included the explanation of relevant theoretical techniques and past examples, as well as an enumeration of the heavily considered frameworks for chatbot generation and NLP. The decision on the frameworks to be utilized was reserved to the subsequent section, after an extensive value analysis on the project and its projected functionalities. This analysis was done since the organization should not implement a solution if it didn't bring them any additional value over their current condition. Such effort was accomplished using various proven standards such as Porter's Value chain and the Business Model Canvas. The aforementioned framework conclusion was made using an analytic hierarchy process in order to establish a mathematical relationship between the types of solution which could be developed, while taking into account a set of criteria which should be present in the final result. It was concluded that a mix of Botkit framework and Rasa-NLU were best for chatbot management and NLP, respectively.

Proven that designing and implementing such a system would be a worthy utilization of resources, the enumeration of its functional and non-functional requirements followed. The required businesses processes were presented, as well as the necessary data modelling, component structuring and API design.

Following this, a description of the processes required to acquire and process the available data towards training data sets was made. Although every test on the system's functioning passed, acceptance testing failed on the overall system due to poor results, something attributed to the available data not being able to meet the necessary quality standards to provide the project with Intent Mining models which can be used in a real-world context.

6.1 Future Work

Some future work is required towards making this solution a viable one within a real-world enterprise situation. It is apparent that there is a need for more and better data, which adequately represents the totality of intents surmised by the organization, something not possible through the achieved processing and incorporation of previous costumer interaction.

A first alternative for the project moving forward would require the manual generation of a set of training examples which would cover a pre-defined list of intents which the organization would deem necessary.

A second alternative could include the changing of the initial demands for the solution (free, open-source frameworks) which would permit the opportunity to test and compare paid frameworks previously disparaged in the Literature Review Section (2). Additionally, if the projects demands were to change, some Machine Learning solutions could be developed and tested.

In terms of features, one can speculate on the utility of implementing and testing a Speech-to-test functionality for the system when it becomes viable.

6.2 Feedback

I would like to start of this section by expressing my gratitude towards E-goi for the opportunity to develop a master's thesis project for them, for my welcoming and for the effort and time which contributed towards a valid solution for this endeavor. However, as was mentioned in the previous section, I find it important to note that the difficulties found in providing a data set impacted the study in an undesirable manner. As described, the given data set did not meet the necessary quality standards required for the development of high-quality models. I also feel the necessity to point out that an alternative route should have been taken for the Intent Mining portion of the master's thesis. By circumventing the restrictions set upon the project, an alternative solution not based on third-party NLP frameworks but rather based on the development of a personal Intent Mining component learning from external NLP datasets could be presented, at least to provide a proof-of-concept for the Intent Mining portion of the master's thesis.

7 References

[1] M. T. P. M. B. Tiago and J. M. C. Veríssimo, “Digital marketing and social media: Why bother?,” *Bus. Horiz.*, vol. 57, no. 6, pp. 703–708, 2014.

[2] Unkown, “Planos de Assinatura.” [Online]. Available: https://www.e-goi.pt/precos/planos/.

[3] “Marketing Automation para Todos!,” 2019. [Online]. Available: https://www.e-goi.pt/. [Accessed: 26-Sep-2019].

[4] “Why use Facebook Messenger bots for business?,” 2018. [Online]. Available: https://blog.hootsuite.com/facebook-messenger-bots-guide/. [Accessed: 26-Sep-2018].

[5] R. Collobert and J. Weston, “A unified architecture for natural language processing,” *Proc. 25th Int. Conf. Mach. Learn. - ICML ’08*, pp. 160–167, 2008.

[6] P. Collobert, Ronan; Weston, Jason; Bottou, Léon; Karlen, Michael; Kavukcuoglu, Koray; Kuksa, “Natural Language Processing (Almost) from Scratch,” *October*, vol. 12, no. April, pp. 2493–2537, 2013.

[7] C. Manning, M. Surdeanu, J. Bauer, J. Finkel, S. Bethard, and D. McClosky, “The Stanford CoreNLP Natural Language Processing Toolkit,” *Proc. 52nd Annu. Meet. Assoc. Comput. Linguist. Syst. Demonstr.*, pp. 55–60, 2014.

[8] R. S. Pressman, *Software Engineering: A Practitioner’s Approach*, 6th ed. Mcgraw-Hill, 2005.

[9] Y. Osterwalder, Alexander; Pigneur, *Business model generation : A handbook for visionaries, game changers and challengers*, 2st ed., vol. 5, no. 7. John Wiley & Sons, 2010.

[10] T. L. Saaty, “Analytic Hierarchy Process BT - Encyclopedia of Operations Research and Management Science,” S. I. Gass and M. C. Fu, Eds. Boston, MA: Springer US, 2013, pp. 52–64.

[11] M. E. Porter, “Competitive advantage: creating and sustaining superior performance.,” *New York: Free Press*, 1985. .

[12] A. M. Turing, “Computing Machinery and Intelligence,” *Mind*, vol. 49, pp. 433–460, 1950.

[13] A. L. Samuel, *Some Studies in Machine Learning*. 1959.

[14] T. M. Mitchell, *Machine Learning*. Mcgraw-Hill, 1997.

[15] M. T. M. Carbonell J.G., Michalski R.S., *An Overview of Machine Learning*. Berlin: Springer, 1983.

[16] “Deep Blue,” 2019. [Online]. Available: https://www.ibm.com/ibm/history/ibm100/us/en/icons/deepblue/. [Accessed: 12-Jan-2019].

[17] “AlphaGO,” 2019. [Online]. Available: https://deepmind.com/research/alphago/. [Accessed: 12-Jan-2019].

[18] “OpenAI Five,” 2019. [Online]. Available: https://openai.com/five/. [Accessed: 12-Jan-2019].

[19] “AlphaStar: Mastering the Real-Time Strategy Game StarCraft II,” 2019. [Online]. Available: https://deepmind.com/blog/alphastar-mastering-real-time-strategy-game-starcraft-ii/. [Accessed: 12-Jan-2019].

[20] T. O. Ayodele, “Types of Machine Learning Algorithms.”

[21] G. James, D. Witten, R. Tibshirani, and T. Hastie, *An Introduction to Statistical Learning*. Springer, 2013.

[22] A. Rakotomamonjy, “Variable Selection Using SVM-based Criteria,” vol. 3, pp. 1357–1370, 2003.

[23] M. Q. Raza and A. Khosravi, “A review on artificial intelligence based load demand forecasting techniques for smart grid and buildings,” *Renew. Sustain. Energy Rev.*, vol. 50, pp. 1352–1372, 2015.

[24] J. A. K. Suykens and J. Vandewalle, “Least Squares Support Vector Machine Classifiers,” no. May 2014, 1999.

[25] R. Rastogi and K. Shim, “PUBLIC: A Decision Tree Classifier that Integrates Building and Pruning,” vol. 768, 1998.

[26] P. O. Gislason, J. A. Benediktsson, and J. R. Sveinsson, “Random Forests for land cover classification,” vol. 27, pp. 294–300, 2006.

[27] I. Rish, “An empirical study of the naive Bayes classifier,” pp. 41–46.

[28] A. Mccallum and K. Nigam, “A Comparison of Event Models for Naive Bayes Text Classification,” 1997.

[29] I. Naseem, R. Togneri, S. Member, and M. Bennamoun, “Linear Regression for Face

Recognition," vol. 32, no. 11, pp. 2106–2112, 2010.

[30] S. Le Cessie and C. S. Van Houwelinguen, "Ridge Estimators in Logistic Regression," *J. Am. Stat. Assoc.*, vol. 41, no. 1, pp. 191–201, 1992.

[31] C. M. Bishop, *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Springer-Verlag New York, Inc., 2006.

[32] J. C. Tenenbaum, V. De Silva, and J. C. Langford, "A global geometric framework for nonlinear dimensionality reduction," *Science (80-.).*, vol. 290, no. 5500, pp. 2319–2323, 2000.

[33] L. J. Williams and H. Abdi, "Principal Component Analysis," *Wiley Interdiscip. Rev. Comput. Stat.*, vol. 2, no. 4, pp. 433–459, 2010.

[34] J. Friedman, T. Hastie, and R. Tibshirani, "The elements of statistical learning," *Springer Ser. Stat. New York*, vol. 1, 2001.

[35] D. Birant and A. Kut, "ST-DBSCAN : An algorithm for clustering spatial – temporal data," vol. 60, pp. 208–221, 2007.

[36] D. Comaniciu and P. Meer, "Mean Shift : A Robust Approach toward Feature Space Analysis 1 Introduction," pp. 1–37.

[37] O. Chapelle and B. Schölkopf, *Semi-Supervised Learning*. .

[38] D. P. Kingma, D. J. Rezende, S. Mohamed, M. Welling, M. L. Group, and G. Deepmind, "Semi-supervised Learning with Deep Generative Models," pp. 1–9, 2009.

[39] X. Zhu, J. Lafferty, L. Cs, and C. M. U. Edu, "Semi-Supervised Learning Using Gaussian Fields and Harmonic Functions," 2003.

[40] L. Ratinov and J. Turian, "Word representations : A simple and general method for semi-supervised learning," no. July, pp. 384–394, 2010.

[41] R. S. Sutton and A. G. Barto, *Reinforcement learning: An introduction*, vol. 1. MIT press Cambridge, 1998.

[42] R. Eberhart and J. Kennedy, "A New Optimizer Using Particle Swarm Theory," pp. 39–43.

[43] V. Mnih *et al.*, "Playing Atari with Deep Reinforcement Learning," pp. 1–9.

[44] S. Mabu, "A Graph-Based Evolutionary Algorithm : Genetic Network Programming (GNP) and Its Extension Using Reinforcement Learning," vol. 15, no. 3, pp. 369–398.

[45] P. Abbeel and A. Y. Ng, "Apprenticeship Learning via Inverse Reinforcement Learning," 2002.

[46] Y. Lecun, Y. Bengio, and G. Hinton, "Deep learning," 2015.

[47] "Andrew Ng: Deep Learning, Self-Taught Learning and Unsupervised Feature

Learning," 2019. [Online]. Available: https://www.youtube.com/watch?v=n1ViNeWhC24. [Accessed: 22-Jan-2019].

[48] A. H. Marblestone, G. Wayne, K. P. Kording, and H. S. Scholte, "Toward an Integration of Deep Learning and Neuroscience," vol. 10, no. September, pp. 1–41, 2016.

[49] Y. Bengio, "Deep Learning of Representations for Unsupervised and Transfer Learning," pp. 17–37, 2012.

[50] R. P. Lippmann, "An Introduction ' to Computing with Neural Nets."

[51] "Artificial neural network," 2019. [Online]. Available: https://en.wikipedia.org/wiki/Artificial_neural_network.

[52] A. Graves, A. Mohamed, and G. Hinton, "SPEECH RECOGNITION WITH DEEP RECURRENT NEURAL NETWORKS Alex Graves, Abdel-rahman Mohamed and Geoffrey Hinton Department of Computer Science, University of Toronto," no. 3.

[53] "Recurrent Neural Networks and LSTM," 2019. [Online]. Available: https://towardsdatascience.com/recurrent-neural-networks-and-lstm-4b601dd822a5.

[54] B. Lovins, "Development of a Stemming Algorithm *," vol. 11, no. June, pp. 22–31, 1968.

[55] "Stemming and lemmatization," 2008. [Online]. Available: https://nlp.stanford.edu/IR-book/html/htmledition/stemming-and-lemmatization-1.html. [Accessed: 15-Jan-2018].

[56] B. Pang and L. Lee, "Opinion Mining and Sentiment Analysis," vol. 2, pp. 1–135, 2008.

[57] R. Passonneau, A. Agarwal, B. Xie, I. Vovsha, and O. Rambow, "Sentiment Analysis of Twitter Data," no. June, pp. 30–38, 2011.

[58] T. Mullen and N. Collier, "Sentiment analysis using support vector machines with diverse information sources," 2004.

[59] A. Pak and P. Paroubek, "Twitter as a Corpus for Sentiment Analysis and Opinion Mining," pp. 1320–1326.

[60] S. Poria, E. Cambria, and A. Gelbukh, "Aspect Extraction for Opinion Mining with a Deep Convolutional Neural Network," *Knowledge-Based Syst.*, 2016.

[61] E. M. Nrl, D. P. Nrl, and R. G. Nyu, "MUC-7 EVALUATION OF IE TECHNOLOGY : Overview of Results MUC-7 Program Committee," no. April, 1998.

[62] S. Subramanian, K. Kawakami, and C. Dyer, "Neural Architectures for Named Entity Recognition."

[63] E. F. Tjong, K. Sang, and F. De Meulder, "Introduction to the CoNLL-2003 Shared Task : Language-Independent Named Entity Recognition," pp. 142–147, 2003.

[64] J. P. C. Chiu and E. Nichols, “Named Entity Recognition with Bidirectional LSTM-CNNs,” vol. 4, no. 2003, pp. 357–370, 2016.

[65] C. N. dos Santos and V. Guimarães, “Boosting Named Entity Recognition with Neural Character Embeddings,” 2014.

[66] M. Habibi, L. Weber, M. Neves, D. L. Wiegandt, and U. Leser, “Deep learning with word embeddings improves biomedical named entity recognition,” 2017.

[67] D. Lonsdale, T. Mitamura, and E. Nyberg, “Acquisition of Large Lexicons for Practical Knowledge-Based MT,” pp. 251–283, 1995.

[68] D. Chiang, “A Hierarchical Phrase-Based Model for Statistical Machine Translation,” no. June, pp. 263–270, 2005.

[69] S. Oepen, E. Velldal, J. T. Lønning, P. Meurer, and V. Ros, “Towards Hybrid Quality-Oriented Machine Translation,” 1997.

[70] M. Nagao, “A FRAMEWORK OF A MECHANICAL TRANSLATION BETWEEN JAPANESE AND ENGLISH BY ANALOGY PRINCIPLE,” 1984.

[71] Y. Wu *et al.*, “Google ’ s Neural Machine Translation System : Bridging the Gap between Human and Machine Translation,” pp. 1–23.

[72] “Machine Translation,” 2019. [Online]. Available: https://en.wikipedia.org/wiki/Machine_translation.

[73] Y. Zhang and R. Jin, “Understanding bag-of-words model : a statistical framework,” pp. 43–52, 2010.

[74] “Text Classification using Neural Networks,” 2019. [Online]. Available: https://machinelearnings.co/text-classification-using-neural-networks-f5cd7b8765c6.

[75] A. Broder, S. C. Glassman, M. S. Manasse, and G. Zweig, “Syntactic clustering of the Web IS (A) n S (B),” vol. 29, 1997.

[76] D. Guthrie, B. Allison, W. Liu, L. Guthrie, Y. Wilks, and P. Street, “A Closer Look at Skip-gram Modelling,” pp. 1–4.

[77] J. Camacho-collados, “From Word to Sense Embeddings : A Survey on Vector Representations of Meaning,” pp. 1–46.

[78] “Word Embeddings for Sentence Classification,” 2019. [Online]. Available: https://towardsdatascience.com/word-embeddings-for-sentence-classification-c8cb664c5029.

[79] “Glossary of Deep Learning: Word Embedding,” 2019. [Online]. Available: https://medium.com/deeper-learning/glossary-of-deep-learning-word-embedding-f90c3cec34ca.

[80] J. Ramos, J. Eden, and R. Edu, “Using TF-IDF to Determine Word Relevance in Document Queries.”

[81] “Natural Language Toolkit,” 2019. [Online]. Available: https://www.nltk.org/.

[82] S. Bird, E. Klein, and E. Loper, “Natural Language Processing with Python,” 2019. [Online]. Available: https://www.nltk.org/book/. [Accessed: 21-Jan-2019].

[83] “fastText,” 2019. [Online]. Available: https://research.fb.com/fasttext/. [Accessed: 22-Jan-2019].

[84] T. Mikolov, K. Chen, G. Corrado, and J. Dean, “Distributed Representations of Words and Phrases and their Compositionality,” pp. 1–9.

[85] “A Beginner’s Guide to Word2Vec and Neural Word Embeddings,” 2019. [Online]. Available: https://skymind.ai/wiki/word2vec#sequence.

[86] J. Pennington, R. Socher, and C. D. Manning, “GloVe: Global Vectors for Word Representation,” 2014.

[87] “GloVe: Global Vectors for Word Representation,” 2019. [Online]. Available: https://nlp.stanford.edu/projects/glove/.

[88] “Gensim - topic modelling for humans,” 2019. [Online]. Available: https://radimrehurek.com/gensim/tut1.html#from-strings-to-vectors.

[89] “Industrial-Strength Natural Language Processing,” 2019. [Online]. Available: https://spacy.io/.

[90] “scikit-learn,” 2019. .

[91] “Rasa NLU: Language Understanding for chatbots and AI assistants,” 2019. .

[92] “Build natural and rich conversational experiences.” [Online]. Available: https://dialogflow.com/. [Accessed: 21-Mar-2019].

[93] “IBM Watson.” [Online]. Available: https://www.ibm.com/watson. [Accessed: 21-Mar-2019].

[94] “Language Understanding (LUIS).” [Online]. Available: https://www.luis.ai/home. [Accessed: 21-Mar-2019].

[95] “wit.ai.” [Online]. Available: https://wit.ai/. [Accessed: 21-Mar-2019].

[96] J. Weizenbaum, “ELIZA--A Computer Program For the Study of Natural Language Communication Between Man and Machine,” pp. 1–7, 1966.

[97] “Hey Siri: An On-device DNN-powered Voice Trigger for Apple’s Personal Assistant,” 2017. [Online]. Available: https://machinelearning.apple.com/2017/10/01/hey-siri.html. [Accessed: 25-Jan-2019].

[98] “Google Duplex: An AI System for Accomplishing Real-World Tasks Over the Phone,” 2018.

[99] A. Ram, R. Gabriel, and J. Nunn, “Conversational AI : The Science Behind the Alexa Prize,” pp. 1–18, 2017.

[100] B. Jiang Jie, “First AI-authored collection of poems published in China,” 2017. [Online]. Available: http://en.people.cn/n3/2017/0531/c90000-9222463.html. [Accessed: 25-Jan-2019].

[101] “Chatbot,” 2019. [Online]. Available: https://www.techopedia.com/definition/16366/chatterbot.

[102] “Guides and references for all you need to know about Botpress,” 2019. .

[103] “Rasa Stack starter-pack,” 2019. .

[104] “WEBCHAT COMPONENT FOR RASA STACK,” 2019. .

[105] “Botkit: Building Blocks for Building Bots,” 2019. .

[106] “rasa NLU vs Recast.AI.” [Online]. Available: https://stackshare.io/stackups/rasa-nlu-vs-recast-ai. [Accessed: 24-Mar-2019].

[107] “rasa NLU vs Wit.” [Online]. Available: https://stackshare.io/stackups/rasa-nlu-vs-wit. [Accessed: 24-Mar-2019].

[108] P. Koen, “What is the New Concept Development (NCD) Model,” 2018. .

[109] P. . A. G. S. S. Belliveau, *The PDMA toolbook 1 for new product development*. John Wiley & Sons, 2004.

[110] M. Kurttila, M. Pesonen, J. Kangas, and M. Kajanus, “Language Functions Revisited: Theoretical and Empirical Bases for Language Construct Definition Across the Ability Range (English Profile),” *For. Policy Econ.*, vol. 1, pp. 41–52, 2000.

[111] C. Ormand, “SWOT Analysis,” 2018. .

[112] A. Lindgreen and F. Wynstra, “Value in business markets: What do we know? Where are we going?,” *Ind. Mark. Manag.*, vol. 34, no. 7 SPEC. ISS., pp. 732–748, 2005.

[113] V. A. Zeithaml, “Consumer perceptions of price, quality, and value: a means-end model and synthesis of evidence,” *J. Mark.*, vol. 52, no. 2–22, 1988.

[114] K. L. Kotler, P.; Keller, “Creating Customer Value, Satisfaction, and Loyalty.” Pearson Education, Inc, 2009.

[115] T. Woodall, “Conceptualising ‘Value for the Customer’: An Attributional, Structural and Dispositional Analysis,” *Acad. Mark. Sci. Rev.*, no. 12, pp. 1–44, 2003.

[116] “Value proposition meaning in the cambridge english dictionary,” 2018. .

[117] “The Value Proposition Canvas,” 2018. .

[118] “Porter’s Theory of Value Chain,” 2019. .

[119] T. L. Saaty, “The analytic hierarchy process: planning, priority setting, resources allocation,” *New York McGraw*, vol. 281, 1980.

[120] E. Mu and M. Pereyra-Rojas, *Understanding the analytic hierarchy process*, vol. 3.

Springer, 2018.

[121] B. Nuseibeh and S. Easterbrook, “Requirements Engineering : A Roadmap.”

[122] E. Evans, *Domain-driven design: tackling complexity in the heart of software*. Addison-Wesley Professional, 2004.

[123] B. Curtis, M. I. Kellner, and J. Over, *Process modeling. Communications of the ACM,*. 1992.

[124] M. West, *Developing High Quality Data Models*. Shell International Limited, 2003.

[125] “ISTQB Glossary,” 2019. [Online]. Available: https://www.istqb.org/downloads/glossary.html. [Accessed: 28-Jan-2019].

[126] M. Baehr, *Evaluation methodology*. Pacific Crest Faculty Development Series, 2004.

[127] “Pearson Correlation.” [Online]. Available: https://libguides.library.kent.edu/SPSS/PearsonCorr. [Accessed: 12-Jul-2019].

[128] Unknown, “Paired Sample T-Test,” 2019. [Online]. Available: https://www.statisticssolutions.com/manova-analysis-paired-sample-t-test/. [Accessed: 09-Sep-2019].

[129] S. S. . Shapiro and M. B. Wilk, “An Analysis of Variance Test for Normality.” 1965.