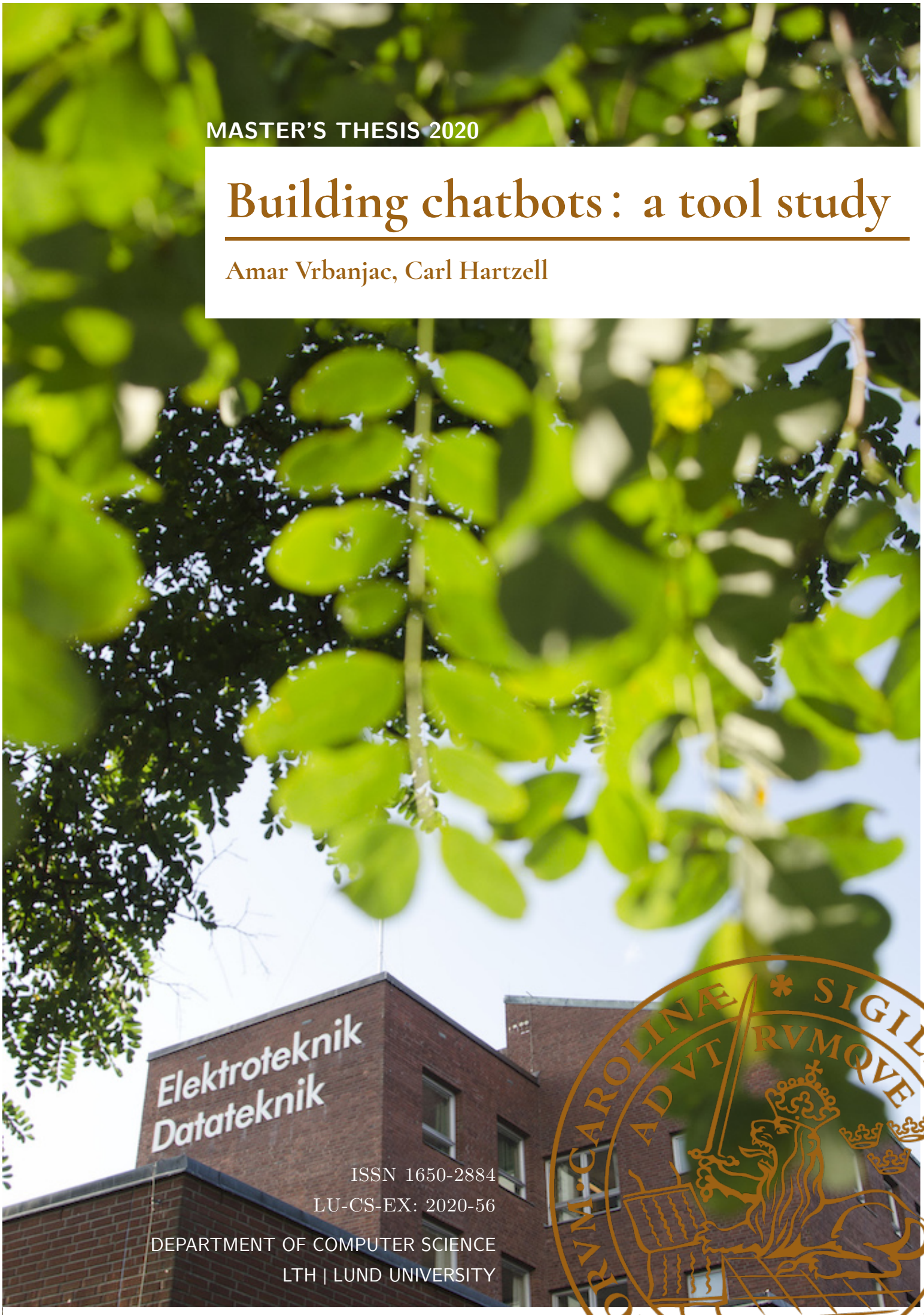


MASTER'S THESIS 2020

Building chatbots : a tool study

Amar Vrbanjac, Carl Hartzell

ISSN 1650-2884
LU-CS-EX: 2020-56

DEPARTMENT OF COMPUTER SCIENCE
LTH | LUND UNIVERSITY

EXAMENSARBETE
Datavetenskap

LU-CS-EX: 2020-56

Building chatbots : a tool study

Amar Vrbanjac, Carl Hartzell

Building chatbots : a tool study

Amar Vrbanjac
dic13avr@student.lu.se

Carl Hartzell
zba10ch1@student.lu.se

September 3, 2020

Master's thesis work carried out at Sinch Sweden AB.

Supervisor: Emelie Engström, emelie.engstrom@cs.lth.se

Examiner: Pierre Nugues, pierre.nugues@cs.lth.se

Abstract

The usage and applications of chatbots are growing at a rapid rate and there is a myriad of conversational platforms on the market, which facilitates the development of them. This study aims to help companies during the selection process of conversational platforms by answering the questions: which are the important aspects of developing chatbots and how should a company go about selecting the best conversational platform for their needs? We try to answer these questions by providing a set of 20 selection criteria, found and defined from conducting a literature review, experimentation with popular tools and consulting experts in the field. We also suggest a method for conversational platform selection, which is tested and validated for two common use-cases: a digital commerce bot and a marketing bot. Three conversational platforms have been evaluated in this study: Chatlayer, Dialogflow and Chatfuel.

Keywords: Chatbot taxonomy, Selection method, Chatbot criteria, Conversational platforms, Chatbot features, Decision method, Chatbot platforms

Contents

Chapter 1

Introduction

One of the early definitions of chatbots was introduced in 1950 by Alan Turing in his publication "Computing machinery and intelligence" in which he also introduced the now famous *Turing test*. In order to pass the Turing test, it should not be possible to distinguish between talking to a chatbot and a person [1]. In other words, a chatbot should be able to replace a human in a human-to-human conversation without the other user noticing they are now talking to a machine. The first example of a chatbot was ELIZA, developed by Joseph Weizenbaum in 1966, which used simple pattern matching and a template-based response mechanism to emulate the conversational style of a non-directional psychotherapist [2]. Another chatbot-like assistant was Clippy the Office Assistant which was included in Microsoft Office between the years of 1997 to 2003 as an interactive user's guide bot [3].

In recent years, there has been a renewed interest in chatbots. One big factor for this being that the way people communicate with each other have changed. In 2015, 6.1 billion people, out of a total human population of 7.3 billion, used an SMS-capable mobile phone and Facebook alone had over 1 billion users [4]. It is safe to say that people are more accustomed to short interactions today than they were just 10 or 20 years ago. This change in human behaviour suddenly makes chatbots a feasible investment for many companies where it is predicted that only specialized digital assistants, such as commerce and marketing chatbots, will generate a global revenue of $623 billion by 2020 [5].

Many companies have taken advantage of this growing interest in chatbots and have developed different tools and platforms which makes this technology easily available. Between the years of 2018 and 2024 there is an expected growth of 29% in the chatbot building tools segment alone [6]. Some examples of companies that provide these services are Google with Dialogflow, IBM with IBM Watson and Microsoft with Microsoft's Azure Bot Service. We have chosen to call these tools and services *conversational platforms* or CP for short.

Due to the wide selection of CPs available on the market today it is not always trivial

to select the one that best suits your specific business needs [7]. What we believe also contributes to the difficulty of selecting a CP is that these services present different solutions for how to build chatbots, both in terms of the underlying technology and in terms of how this technology is visually represented in the tools graphical interface. We will go into more detail about these different technologies and how they are implemented on a high level in a selection of existing tools later in the background section of the report.

It is not only the variety of platforms and tools available for building chatbots that makes the decision of selecting one hard, it also highly depends on the use-case that the chatbot is intended to solve. But how do you know which aspects of a CP that is important?

One of the goals with this report is to give an answer to this question by presenting a set of criteria which will help companies make an informed decision about which CP to chose and start building their chatbot in. A criterion could be either a feature supported by a CP or some other CP characteristic. In order to come up with this set of criteria, we used a couple of different research approaches. First we did a literature study to get input from current knowledge and studies in the field. We then used a (hands on approach) where we looked at which features were supported by different popular CPs available on the market today. In order to validate and complement the findings from previous methods, we created a questionnaire which we sent out to experts at the company Sinch where we conducted the master thesis.

The second goal with this report is to provide a method for selecting the best suited CP based on a specific use-case that a chatbot is intended to solve. In order to use this method a company needs to consider the importance of each of the previously defined criteria in relation to the use-case of their chatbot as well as how well these criteria are supported by the CPs that they are considering. We validated the proposed method for CP selection by applying it to two popular use-cases for chatbots found on the market today: a digital commerce bot and a marketing bot. The CPs used was Chatlayer, Dialogflow and Chatfuel.

Current literature is limited on the area of building chatbots and conversational platform evaluation. Most literature focus on case studies regarding the impact of chatbots in different applications. One such a study tried to determine if a chatbot, with the personality of a famous person, could improve student-content interaction in distance education [8]. Other studies provide a method for selecting a CP based on characteristics of individual businesses, but is limited to the number of features they include [9]. Even if they were to consider adding more features in their evaluation method, it would become very time consuming and cumbersome to use because the number of comparisons required scales exponentially based on the number of features included. We wish to improve this by proposing a method that does not scale as bad. This would allow for more criteria to be considered, which we believe could provide a more accurate recommendation for selecting a CP.

Chapter 2

Background

In this section we will present background information on how chatbots work, different chatbot technologies, how they have evolved, how they are commonly used today and some predictions about chatbots in the future. We will also give an in-depth description of the different CPs that we have evaluated in the report.

2.1 How chatbots work

Simply put, a chatbot is a computer program that is designed to communicate with human users over the internet via e.g. a chat platform or a website [10]. Chatbots are developed to work independently of humans and provide responses based on either predefined scripts or machine learning applications. Responses are limited to the knowledge database that the chatbot has access to at that time. It is also the quality of the underlying software and the data it can access that generally determines the complexity and performance of a chatbot [10].

2.2 Chatbot Technologies

From previous studies we have not found any clear-cut classification for different types of chatbots. This might be because chatbots are still a relatively new technology. One attempt to classify chatbots has been into “high-level”, “mid-level” and “low-level” dialog systems [11]. High-level dialog systems take advantage of integrated artificial intelligence and are able to be aware of states in the conversation, self-improve based on algorithms and have temporal reasoning. Mid-level dialog systems possess the ability to respond to a variety of requests

about a task or domain with a limited reasoning ability. Low-level dialog systems try to mimic a person but have no extensive intelligence behind the responses it provides.

Another study classifies CPs into "WYSIWYG" and "All-purpose" platforms [9]. WYSIWYG platforms often focus on solving some specific business use-cases and are implemented on a level that does not require any high technical knowledge. All-purpose platforms provide a broader set of functionality and are targeted towards developers.

For the purpose of this study we have found the classifications of different levels of dialog systems more suitable to use as a classification for different types of CPs. We have however settled on another commonly used terminology which we found to be more descriptive. CPs which allows for development of high-level dialog systems we call NLP platforms, mid-level dialog systems we call Hybrid platforms and low-level dialog systems we call Rule-based platforms.

NLP platforms In NLP platforms the primary focus is creating NLP chatbots. The technology used for creating the conversation in NLP chatbots is almost primarily based on intents and entities, which are extracted from a users utterances and interpreted by the chatbot via an underlying NLP engine. An NLP chatbot is also often context aware [11], which means that an intent can be interpreted differently depending on where in the conversation the utterance is provided and enables the chatbot to give an context appropriate response. In some cases, NLP chatbots also have the ability to self-improve. This can be done by using the user utterances as training data for the NLP model. Much of the work required to build an NLP chatbot, thus consist of creating intents and training them with phrases. Phrases in this case is variations of what a user is expected to say. The recommended number of training phrases is anywhere from 10-20 phrases per intent, depending on the complexity of the intent [12].

Rule-based platforms In general the rule-based chatbot does not support any NLP capabilities, such as being aware of states within the conversation or come up with its own responses. The conversation in a rule-based chatbot is more linear in nature and is mainly driven by a fixed set of rules defined by the bot-creator. These rules may or may not be enforced by the chatbot with word or keyword recognition [13]. A rule-based conversation can easily be visualized with a tree-like structure. Each branch within the tree is represented by a rule and each node is represented by one or more responses. As a result it is common to see that rule-based platforms use a visual graph editor for building chatbots. In this structure it is easy to see the different paths a conversation can go. It is however hard, and often even impossible, for the user to deviate from this linear-like conversational flow since the chatbot can not provide responses that it is not programmed to handle.

Hybrid platforms A hybrid platform tries to combine the features from both NLP and rule-based platforms. With hybrid platforms it is possible to create chatbots that possesses some AI capabilities such as simple reasoning and taking advantage of natural language processing to answer user queries [14]. This is combined with the ability of some seamless scripted logic that drives the conversation. Hence, it can be seen as a hybrid between an NLP and a rule-based chatbot.

2.3 Chatbot usage

A chatbot has countless fields of applications. Ranging from simple repetitive tasks, such as a scripted FAQ, to more complex chatbots such as Apple's Siri, Google Assistant or Amazon's Alexa. In this section we will go into more detail about the history of chatbots: in which areas they are used today and some predictions for chatbots in the future.

2.3.1 Chatbots in the past

1966 As mentioned earlier, ELIZA was the very first chatbot, developed by Joseph Weizenbaum in 1966 [2]. Based on our classification of chatbots, ELIZA was a rule-based chatbot with pre-defined rules and responses for different user utterances. ELIZA was essentially a set of scripts that emulated different scenarios, the most popular being the Doctor script which simulated a Rogerian psychotherapist.

1972 The very first chatbot to pass the Turing test, developed in 1972 by Kenneth Colby, was Parry the paranoid schizophrenia bot. It was often compared to ELIZA and was sometimes described as "ELIZA with attitude" [15]. Parry provided its responses based on a complex system of assumptions, attributions and "emotional responses" triggered by shifting weights assigned to user input" [16].

1988 An early attempt at creating an artificially intelligent chatbot resulted in the chatbot Jabberwacky. It was developed by the British programmer Rollo Carpenter with the aim to simulate natural human chat in an interesting, entertaining and humorous manner [17]. Jabberwacky learns purely from conversations, it stores everything anyone has ever said and finds the most appropriate thing to say using contextual pattern matching techniques.

1995 One of the more famous chatbots is called ALICE or Alicebot, which was developed by Richard Wallace in 1995. ALICE is short for "Artificial Linguistic Internet Computer Entity" and is a natural language processing chatbot designed to engage in a conversation by reacting to human input and responding as naturally as possible [18]. ALICE is written in artificial intelligence markup language (AIML) and have won many prizes over the years, there among being a three times Loebner Prize Winner in 2000, 2001 and 2004 [18].

2010 In 2010, Apple created the voice-controlled personal assistant Siri. It was first released on the iPhone 4s and has then found its way into most of Apples product lines [19]. In comparison with previously mentioned chatbots, the purpose of Siri was to act as a personal assistant and could handle tasks such as scheduling meetings, replying to emails and checking the weather. The range of tasks that Siri can carry out is continuously increasing. Today it can set alarms, control smart home appliances, select and start playback of music and much more.

2.3.2 Chatbots today

Chatbots has evolved a lot since the time of ELIZA. They are now commonly found on websites, as digital assistants on smartphones, tablets, speakers and computers. They can be interacted with on most communication channels such as Messenger, WhatsApp, WeChat and many more. It is not only the range of interfaces where you can interact with chatbots that has increased, also the capabilities of the chatbots. Chatbots can carry out advanced task, learn from your habits and continuously self-improve.

From a chatbot developers perspective, there was conducted a study on 253 participants where they asked the question: *"If you have, or are, implementing chatbot(s), which use-case category is your focus?"* [20]. The result of this study is presented in figure 2.1 and shows that a majority of the answers were *Support and Self-Service*.

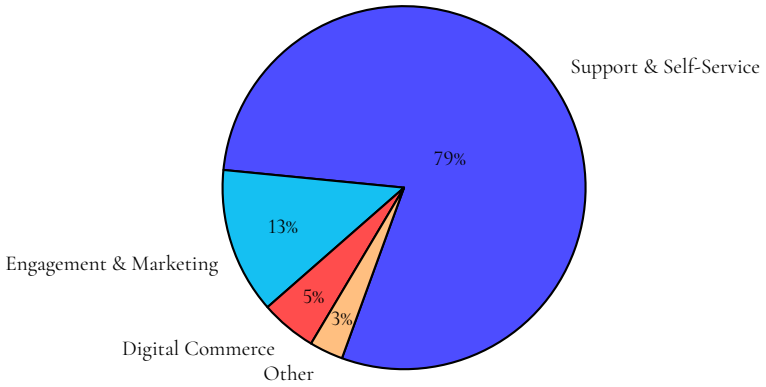


Figure 2.1: Gartner study result

Another study done by *Mindbowser* in association with *Chatbots Journal* shows that the industries, presented in bar chart 2.2, will benefit the most from the use of chatbots [21]. The study was conducted on 300+ individuals representing a variety of different industries and organisations. The result of the study shows that *E-commerce* would be the industry that would benefit the most from using chatbots, followed by the *Insurance and Healthcare* industries.

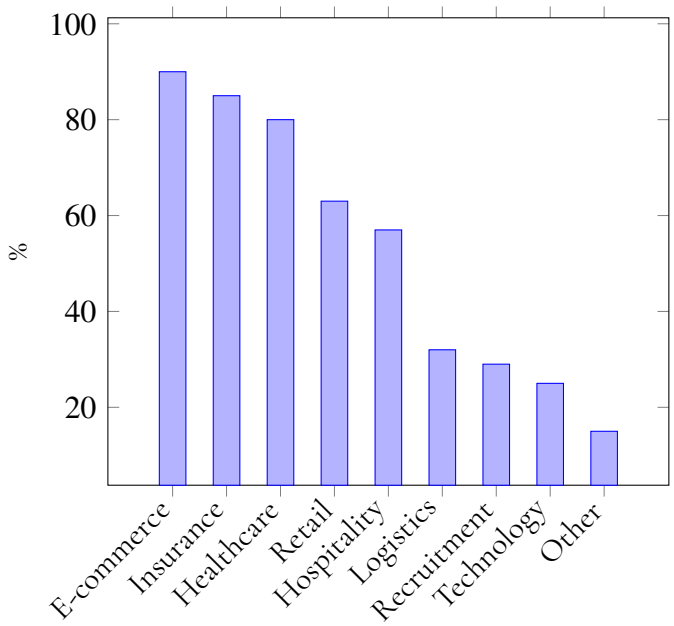


Figure 2.2: Industries that will benefit the most from chatbots

In the same study, they also asked the question: *"Which platforms(s) / Network(s) would you prefer to build chatbots for?"*. Platforms and networks refers to what we in this study call *communication channels*. The result is presented in the bar chart 2.3 and shows that the most popular channels where Facebook Messenger, which 92% of the companies asked preferred to build chatbots on, followed by the company's own website at 80% and Slack at 70%.

The study also shows which business functions that benefits the most from chatbots. The result can be found in the bar chart 2.4 and it shows that the *Customer service* function benefit

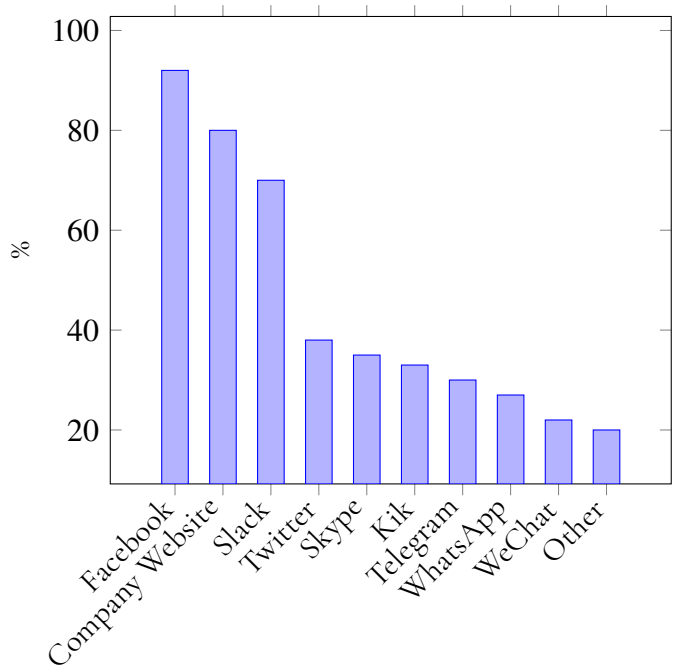


Figure 2.3: Preferred communications channels for chatbots

the most from chatbots followed by *Sales/Marketing* and *Order Processing*.

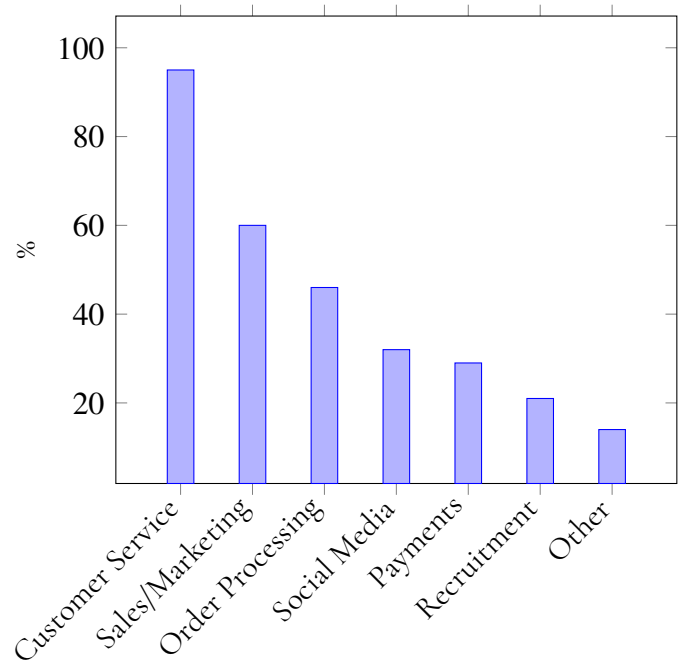


Figure 2.4: Business functions which will benefit the most from chatbots

2.3.3 Chatbots in the future

With the current development and interest of chatbots it is safe to presume that chatbots will continue to improve and play an even bigger part in the future. The general consensus among the business participating in a survey conducted by *Mindbowser* confirm this, where 96% stated that they believed that chatbots are here to stay for long and will not go away soon [21]. Another study conducted by *Tractica* predicted that the total revenue of Virtual Digital Assistants (VDAs) will reach 9 billion dollars by 2025 [22]. How the revenue share is divided into different market segments is shown in figure 2.5. The result shows that the biggest revenue share will come from *Customer Service & Marketing* followed by *Digital Commerce & Sales* and *Business Applications*.

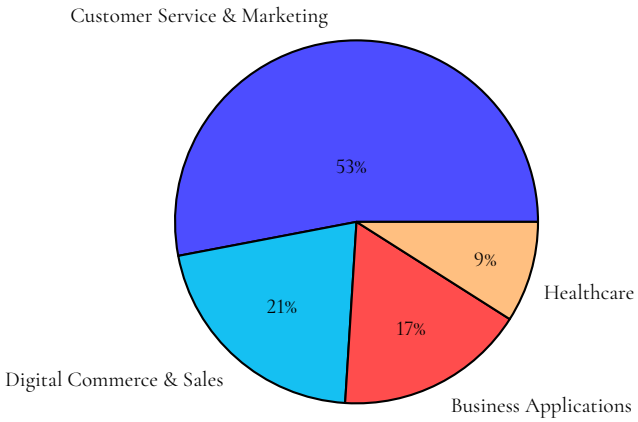


Figure 2.5: Tractica study result

2.4 In-depth look at three CPs

There is a large selection of CPs to choose from on the market today [7]. The common aspect of all these platforms is that they aim to provide an interface that will make the process of building chatbots easier in comparison to implementing chatbots programatically. There is a large spectrum of companies offering these services, from tech giant such as Google, IBM, Microsoft and Facebook which all have 1000+ employees, to small start-ups with 1-10 employees, such as Botsify, Flaotbot and FlowXO [23], [24]. In this study we looked at three different CPs. *Dialogflow* which primarily focus on the development of AI chatbots, *Chatfuel* which main focus is on rule-based chatbots and *Chatlayer* which allows for the development of a hybrid chatbot.

2.4.1 Dialogflow

Dialogflow is one of the markets leading conversational platforms developed by Google. The chatbots developed in Dialogflow are heavily driven by AI and NLP to both understand incoming utterances and to reply to them. The latter could be parsed FAQs or articles with

manual intent recognition editing. Dialogflow functions for both voice-based and text-based conversations with different combinations of these as input and output. Dialogflow supports several communication channels. There among: Genesys Cloud, Voximplant, Facebook Messenger, Slack and more.

A chatbot created with Dialogflow is dependent on intents and the contexts in which they are defined. It is not possible to configure a linear step-by-step conversation, but instead the conversation is built by specifying and configuring intents. With the help of contexts the user utterances are routed to the correct part of the conversation. In Dialogflow, a conversation is structured as a set of intents and sub-intents that are presented in the form of drop-down menus.

Dialogflow provides several ways of analyzing a chatbot's performance and user inputs. They also provide analytic capabilities, where intent and sentiment analysis may be performed. The analytics enables a way to see where users drops off and then make improvements to the chatbot accordingly. With sentiment analysis users' opinions and attitudes (positive, neutral, negative) are analyzed and intents may be built upon these results. However, this feature is currently only available for enterprise level accounts [25].

For bots that only support text-based messaging, Dialogflow has support for up to 20 languages which is a continuously growing number. They also provide client libraries in several popular programming languages which allows for custom programming and scripting of the bot.

2.4.2 Chatlayer

Chatlayer is one of the leading AI bot platforms in which you can build both chatbots and voicebots on. One of their main selling points is their extensive language support where you can define a bot once and make it available in 100+ languages [26]. Chatlayer has taken a "no-code" approach when it comes to building chatbots. This allows users to create and manage chatbots without any advanced IT skills. In addition, much like Dialogflow, Chatlayer provide a set of analytic features where user interactions with the chatbot can be analysed in more detail. Chatlayer also support a large amount of integration options as well as support for some of the most commonly used communication channels, such as Facebook Messenger, SMS, web-chat and voice.

A bot conversation in Chatlayer can be visually represented in two ways. Either through a configuration UI where all messages and actions are listed in a table format or through a graphical interface where the conversation is represented in a tree structure. Each message and action is a node within the tree and where these nodes can be connected via edges in a way that makes it easier for the user of the tool to get a structural overview of the conversation.

Even though Chatlayer is heavily focused on AI bots and classifies itself as a platform for developing AI bots [26], they still provide functionality for configuring rule-based logic. This is why we have chosen to classify it as a hybrid CP.

2.4.3 Chatfuel

Chatfuel is heavily focused on the development of marketing and customer engagement chatbots. They have gathered a large community over the years as one of the big actors in providing highly integrated chatbots for Facebook Messenger. Last found figures (March, 2018) showed that Chatfuel had 300 000 live Messenger bots [27]. Even though Messenger is their only supported channel, it is possible to integrate the chatbot as a plugin on any website.

Just as Chatlayer, Chatfuel is targeted towards business users and does not require any higher technological knowledge to use. One way they achieve this is by introducing the concept of conversational blocks. A block is a subset of the conversation and is used to structure the chatbot's content, aiming to provide structural aid when designing the conversation [28]. The blocks can be grouped together and interconnected - creating a conversational flow.

Chatfuel does not provide much in terms of raw AI or NLP support with the exception for simple intent routing. They do however allow you to integrate with Dialogflow and Janis.ai which is a framework for building Dialogflow-enhanced Chatfuel bots. Due to this limited raw AI/NLP support and focus on more scripted logic, we classify Chatfuel as a rule-based CP.

Chapter 3

Research Approach

3.1 Criteria breakdown

To come up with the criteria for building chatbots, we used several research approaches. Firstly, we conducted a literature review where we investigated what previous studies and gray literature (GL) concluded to be viable or required characteristics for a CP. Gray literature refers to content that can be written as blog posts, articles, web-pages and business research material. Then we took a more "hands on approach" with a selection of existing CPs where we tried them out and built chatbots ourselves. The results from these methods were then summarized into a list of criteria, where each criteria was assigned a priority value and a motivation. This list was then used and sent out as a survey to a selected number of experts in the area to gather additional input about their importance.

Finally, the combined result was analyzed to establish a final list of criteria (see Section 5.1) to be used in our proposed CP selection method.

3.1.1 Literature review

The goal with the literature review was to find out what previous studies and grey literature had found to be important criteria for building chatbots. The scientific contributions was found by searching on Google Scholar [29] and LUBsearch [30] where the following search terms was used; *"chatbots"*, *"chatbot/conversational platform"*, *"chatbot tool"*, *"building chatbots"* or *"chatbot characteristics/feature"*. To complement the search of scientific contributions, we used the method of backwards snowballing which is done by *"following the references"* [31]. The approach used for including a paper in the review was performed in the following steps.

1. Firstly, only papers written in English or Swedish was considered.

2. Title, abstract was read in order to determine the initial relevance of the paper. Whether or not the paper concerned the subject of chatbot development or conversational platforms.

3. If deemed relevant from step 1, the conclusion and to some extent the result was read to determine if the paper actually showed on any useful results for our purposes, e.g. if the study showed any results or discussion around requirements/features/characteristics of chatbots or conversational platforms?

4. If the paper fulfilled the third inclusion criteria it was included in the review and the whole paper was read. Backward snowballing was also performed on these papers where we went through the same steps.

In total we found 5 scientific contributions that passed our inclusion criteria, which can be found in [32], [13], [9], [33], [34].

There was a greater supply of gray literature which focused on the subject of chatbot development and requirements of conversational platforms. This allowed us to use more specific search terms when searching for GL. The primary search engine used for this was Google. Search terms used were: *chatbot/conversational platform features/characteristics*, *chatbot features/characteristics*, *chatbot/conversational platform comparison* and *Top chatbot features*. The approach used for including a GL in the review was performed in the following steps.

1. Title, meta description was read in order to determine the initial relevance of the search result. If it specifically focus on features and/or characteristics for building a chatbot.

2. If deemed relevant from step 1, the content of the page was read to determine of it also provided any criteria for selecting a conversational platform.

3. If the GL fulfilled both previous criterias it was included in the review.

The gray literature that was included can be found in [35], [36], [37], [38], [39].

The data found in the scientific contributions and grey literature we looked at were not always easily comparable in a systematic way, which is why we decided to go with a semi-systematic review approach as defined by Snyder [40]. In short, a semi-systematic review seeks to identify and understand all potentially relevant research traditions that have implications for the studied topic and to synthesize these using meta-narratives instead of by measuring effect size.

In order for us to come up with relevant conclusions from the review we created a simple system for how to summarize the result of each contribution. For each contribution we took notes of any mentions of CP features or chatbot characteristics along with any potential priority or motivation for it. We then did an analysis of these notes where we appointed a priority value for each feature and characteristic. The priority value was based on a combination of how many times the feature was mentioned and how well we thought it was motivated

in the literature. The priority values used is described in Table 3.1 and the result of the review is composed in a table in the form of Table 3.2.

The result of the literature review is presented as a prioritized list in Section 5.1.2 in the result part of the report.

Priority	Value
Low	1
Medium	2
High	3

Table 3.1: Priority value mapping

Feature/characteristic	Priority	Motivation
name	value	motivation behind its importance

Table 3.2: Priority ranking

3.1.2 Hands on with existing tools

Another method we used for identifying criteria for building chatbots was a hands on method, where we looked at and evaluated different popular CPs on the market today, more specifically Dialogflow, Chatlayer and Chatfuel. The purpose of this method was to get input on which features existing tools support and promote, as well as which features we found to be essential when building chatbots.

In order to do a fair and structured evaluation of the CPs we defined two benchmark cases (BC). A BC is a use-case that a chatbot is intended to solve with the purpose of acting as a benchmark for our evaluation of the CPs. One of the BCs was developed to represent digital commerce, while the other was developed to represent marketing. Both areas were found to be the most common use-cases for chatbots in Section 2.3. In Appendix A both tables A.1 and A.2, for both BCs, are presented as high level descriptions.

All features and characteristics that we came across whilst building the chatbots were added to a list, including features that where not strictly required for the development of those specific chatbots. To come up with relevant and comparable conclusions, a similar approach was used as in Section 3.1.1, where we appointed a priority value based on the motivation for the feature, the number of platforms that supported it and if we used it when we built the chatbots. The priority values used are described in Table 3.1 and the result of this analysis is compiled in a table based on Table 3.2. The resulting table can be found in Table 5.2 in the result chapter of this report.

3.1.3 Consulting experts

To validate and gain additional input on the results from the literature review and hands on experience we created a qualitative questionnaire which is suggested to be a fruitful method for information studies [41]. The questions is based on a combined list of the features and characteristics found from the results of the literature review and hands on experience. In the questionnaire, we asked the respondents to appoint a priority value for each one of these features and characteristics. The priority values used in the questionnaire were the same as those used in the literature review and hands on method, see Table 3.1. In addition to the priority value, respondents could give a short motivation for the values they appointed. In the questionnaire we also included a field where we asked for additional features or characteristics that the respondents felt were missing from the list, for which we also asked for a priority value and a motivation. We created the questionnaire using Google forms and a copy of it can be found in Appendix B.

We selected the respondents based on their current role and previous experience in the field of chatbot technology. The respondents found were considered to be experts, thus possessing high domain knowledge, in the fields of chatbots and development of conversational platforms. As of writing this thesis the current roles of the respondents where; one product manager, one team lead, one product owner and one senior developer. All the respondents are currently working within a team that is developing a conversational platform and both the team lead and senior developer have extensive previous experience in this field.

The answers from the questionnaire were summarized in a table in the form of Table 3.2. The priority value for a feature was calculated as the means of each respondents individual priority value for that feature and the motivation is presented as a list of each respondents motivation behind the value they appointed, if any. The summarized result of the questionnaire is presented in Table 5.3 in the result chapter of the report.

3.1.4 Feature analysis

From the previous methods we combined, compared and analyzed the results to create one definitive list of criteria for building chatbots. For each of the features and characteristics we did a side-by-side comparison where we compared the results from the *Literature review*, the *Hands on method* and the *Expert consulting*. This comparison was possible since we used the same priority ranking across all methods. The determining factors for if a feature or characteristic should be included or excluded from the final list of criteria was calculated using the following formula:

$$f_{pv} = \frac{l_{pv} + h_{pv} + 2 \cdot q_{pv}}{4}, \tag{3.1}$$

where f_{pv} is the final priority value, l_{pv} is the priority value from the literature review, h_{pv} is the priority value from the hands on method and q_{pv} is the priority value from the expert consulting. We valued the input from the expert consulting two times higher than the literature review and the hands on experience since those priority values were more heavily influenced by our experience and interpretation.

To procure a list with the most important criteria we excluded any feature or characteristic with an average priority value below 2.0. In order to make the list more readable and usable for the CP selection method we categorized the criteria into related main -and sub-criteria. In addition to this list of criteria we also present a description for each criterion. This result can be found in Section 5.1 in the result chapter of the report.

3.2 CP Selection Method

The problem of selecting a CP can be classified as a multi-criteria decision (MCD) problem, where the criteria to consider are different CP features and characteristics and the decision problem is to select the CP that best support these from a chosen set of alternatives.

There are a number of different methods that aim to solve this kind of problem, one of the most popular being Saatys Analytical Hierarchy Process (AHP) method [42]. Previous studies have compared different MCD methods and found that AHP is well suited for strategy and planning problems [43], [44] as well as for selection of software [45]. We believe that the problem of selecting a CP applies to these areas.

In short, AHP works by doing pair-wise comparisons between different criteria and assigning weights to how well these criteria are met by the different alternatives. However, the regular AHP method has a big drawback and that is that the number of comparisons between criteria that is required grows exponentially based on the number of criteria used. One method that is derived from AHP is called AHP-express. It tries to solve this issue by structuring the criteria into a hierarchy of main-criteria and sub-criteria which greatly reduces the number of comparisons needed [46]. Since we are considering a large number of criteria, AHP-express seems to be a suitable method for doing the comparison and coming up with an recommendation.

The result we wish to present is how AHP-express can be used for the purpose of selecting a CP and describe, step by step, how to use it with the resulting list of criteria for building a chatbot found in Section 3.1.4 Feature Analysis. We call this proposed method *"CP Selection Method"* and it is presented in Chapter 4.

3.3 Evaluation of CP Selection Method

We aim to validate the CP selection method by applying it on the two previously mentioned BCs, specified in more detail in Appendix A. The different CP alternatives we will compare are Dialogflow, Chatlayer and Chatfuel. We perform the evaluation by following the steps stated in Chapter 4 in order to determine how easy the method is to use, which alternative that scores highest for each of our BCs and whether or not this result reflects our own experience of implementing the BCs during our hands on experience with the tools.

Chapter 4

CP Selection Method

This chapter presents a suggested method to be used for selecting the best suited CP, based on a company's needs and the intended use-case of the chatbot they plan to develop. The method consists of a number of steps that needs to be performed. Each step is described in detail below, along with any potential supporting documentation which might be required to perform the step.

Step 1: Select CP alternatives

Firstly, select the different CPs that should be compared. For the purpose of this method we refer to the the different CPs as alternatives.

Note: Due to the wide selection of CPs on the market, it could be advantageous to do an initial screening of which platforms to include in the decision. Criteria for the screening could for example be: pricing, supported channels and security. These are aspects that we have chosen to exclude as criteria for our proposed method since they often, by themselves, are hard requirements. We go into more detail about these hard requirements (see Section 6.1.1) in the discussion chapter of the report.

Step 2: Evaluate sub-criteria for alternatives

For the second step we provide a list of main criteria and related sub-criteria found in Chapter 5 in Figure 5.1. An in depth description for all main -and sub-criteria can be found in Section 5.1.1.

For each of the sub-criteria create a small table in the form of Table 4.2 and appoint a comparison value for how well this sub-criteria is met by each of the alternatives. The comparison values are derived from Table 4.1 where the platform that best meets the criterion

should be assigned the value 1 and used as a baseline for the values appointed to the other platforms. Intermediate values between those shown in the table can also be considered. For the purpose of evaluating sub-criteria for alternatives, the AHP-scale is interpreted as how well a criterion is supported by an alternative in comparison to each of the other alternatives. E.g. if a feature is best supported in $\boldsymbol{CP}_1$, this CP gets assigned the value of 1. If $\boldsymbol{CP}_2$ supports the same feature equally well as $\boldsymbol{CP}_1$ then $\boldsymbol{CP}_2$ also gets assigned the value 1. However, if $\boldsymbol{CP}_2$ has moderate support for the feature in comparison to $\boldsymbol{CP}_1$ then it gets assigned the value 3, etc. The priority values can then be calculated with the following means formula:

$$pr_j = \frac{1}{a_j \cdot \sum_n \frac{1}{a_n}} \tag{4.1}$$

where j is the alternative for which one wishes to calculate the priority, a_j is the comparison value of alternative j and pr_j is the priority of alternative j against the criterion considered. The result of this step is a vector $\boldsymbol{PASC}_i$ containing the priority values of a sub-criteria for each platform evaluated. The priority vectors $\boldsymbol{PASC}_i$ should then be combined into a matrix $\boldsymbol{PASC}$, in the form of Table 4.3, that will be used for further calculations in a later step.

Intensity of importance	Definition
1	Equal importance
3	Moderate importance
5	Strong importance
7	Very strong importance
9	Extreme importance

Table 4.1: AHP Scale

$Sub-criteria_m$	CP_1	CP_2	$\cdots$	CP_n
Baseline CP	a_1	a_2	$\cdots$	a_n
1/a				
Priority/$\boldsymbol{PASC}_i$	pr_1	pr_2	$\cdots$	pr_n

Table 4.2: Priority calculation of alternatives for sub-criteria (SC)

Step 3: Prioritize sub-criteria in main criteria

This step repeats the activities performed in the previous step, but with the purpose of determining the priority of a sub-criterion within a main criterion. Begin with creating a table on the form of Table 4.4 for each of the main criteria found in Figure 5.1. Then appoint comparison values to the sub-criteria from the AHP-scale (Table 4.1) where the sub-criterion considered most important for the specific use-case should be appointed the value 1 and used as a baseline for the values appointed to the remaining sub-criteria. Priority values can then be calculated using the same formula as in the previous step (Equation 4.1). The result from this is a number of priority vectors $\boldsymbol{PSC}_i$ (Table 4.4), and when combined it results in a matrix $\boldsymbol{PSC}$, see Table 4.5, where each row represents the criteria and each column represents the sub-criteria. Each intersection in $\boldsymbol{PSC}$ where a sub-criteria was not relevant for a main criteria should (for further calculation purposes) be assigned the value 0.

PASC				
Sub-criteria	CP_1	CP_2	$\cdots$	CP_n
Intent routing				
Entity extraction				
Languages				
Context				
Dynamic content				
Custom logic				
Access account information				
OMNI-channel				
Rich content				
Translations				
Visual builder				
Chat simulator				
Auto-configuration				
Conversation history				
Visualization of data				
Webhooks				
External services				
Human handover				
Custom scripting				
Low technical depth				

Table 4.3: Priorities of the alternatives for each sub-criterion (***PASC*** matrix)

$Main - criteria_m$	$Sub - criteria_1$	$Sub - criteria_2$	$\cdots$	$Sub - criteria_n$
Baseline sub-criteria				
1/a				
Priority				

Table 4.4: Template table for prioritizing sub-criterias within a main criteria (PSC_i)

Main criteria / *Sub-criteria*	**NLP capabilities**	**Variable support**	**Messaging**	**Usability**	**Analytics**	**Integrations**	**Custom scripting**	**Low technical depth**
Intent routing		0	0	0	0	0	0	0
Entity extraction		0	0	0	0	0	0	0
Languages		0	0	0	0	0	0	0
Context		0	0	0	0	0	0	0
Dynamic content	0		0	0	0	0	0	0

Continued on next page

Table 4.5 – *Continued from previous page*

Main criteria / *Sub-criteria*	**NLP capabilities**	**Variable support**	**Messaging**	**Usability**	**Analytics**	**Integrations**	**Custom scripting**	**Low technical depth**
Custom logic	0		0	0	0	0	0	0
Access account information	0		0	0	0	0	0	0
OMNI-channel	0	0		0	0	0	0	0
Rich content	0	0		0	0	0	0	0
Translations	0	0		0	0	0	0	0
Visual builder	0	0	0		0	0	0	0
Chat simulator	0	0	0		0	0	0	0
Auto-configuration	0	0	0		0	0	0	0
Conversation history	0	0	0	0		0	0	0
Visualization of data	0	0	0	0		0	0	0
Webhooks	0	0	0	0	0		0	0
External services	0	0	0	0	0		0	0
Human handover	0	0	0	0	0		0	0
Custom scripting	0	0	0	0	0	0		0
Low technical depth	0	0	0	0	0	0	0	

Table 4.5: Transposed ***PSC*** matrix

Step 4: Calculate priority of alternatives for main criteria

The matrices ***PSC*** and ***PASC***, which were calculated in the two previous steps, should then be multiplied using Equation 4.2 resulting in the matrix ***PAC*** shown in Table 4.6 containing the priorities of each platform ***a*** for the main criteria.

$$PAC = PSC \cdot PASC \tag{4.2}$$

CP / *Main criteria*	CP_1	CP_2	$\cdots$	CP_n
NLP capabilities				
Variable support				
Messaging				
Usability				
Analytics				
Integrations				
Custom scripting				
Low technical depth				

Table 4.6: Template table of the ***PAC*** matrix

Step 5: Prioritize main criteria

In this step the main criteria is evaluated against each other. Appoint comparison values from Table 4.1 to each main criterion where the most important main criteria for the use-case should be appointed the value 1 and serve as a baseline for appointing values to the other main criteria. The means Equation 4.1 should then again be used to calculate the priority values, resulting in the priority vector ***PC*** shown in Table 4.7.

	NLP capabilities	**Variable support**	**Messaging**	**Usability**	**Analytics**	**Integrations**	**Custom scripting**	**Auto-scaling**	**Low technical depth**
Baseline main criteria									
1/a									
PC									

Table 4.7: Template table for prioritizing main criteria for a use-case

Step 6: Calculate final score

Finally, using the formula 4.3 perform a multiplication of the vector ***PC*** and matrix ***PAC*** in order to get the priority vector for the alternatives (***PA***). This vector gives the final score on

which CP that is best suited for a specific use-case.

$$PA = PC \cdot PAC \tag{4.3}$$

CP	CP_1	CP_2	$\cdots$	CP_n
Final score				

Table 4.8: The final scores for the n CP alternatives

Chapter 5

Result

In this chapter we present our findings and results gathered from the different research approaches used in the thesis. The result is divided into two sections; *Criteria breakdown* and *Evaluation of CP Selection Method*.

5.1 Criteria breakdown

The criteria breakdown consists of the results found from the literature review, hands on experience, expert consulting and feature analysis. These results are mainly presented in a table format accompanied with descriptions where ever needed.*

5.1.1 Description of features and characteristics

The following are brief descriptions of all the features and characteristics we came up with from our research. Separate results for each research method are presented in the following sections.

NLP capabilities From our research we have found the following four NLP features to be the most essential for building chatbots. These features include, intent routing, entity extraction, the languages supported by the NLP engine and context.

- **Intent routing** There are two common ways for a chatbot to interpret user utterances. Traditionally this logic is achieved through rule-based conditions, e.g. chatbot expects

*The feature and characteristics in the tables of this section are hyperlinked to their descriptions in the Section 5.1.1 Description of features and characteristics

an utterance to be *yes* or *no* and provide a specific response if the next user utterance matches either case. An alternative for rule-based logic is intent routing, which as the name suggest tries to extract the intent from the user utterance using Natural Language Processing (NLP) and provide responses based on which intent got matched. Defining and training these intents can be a time-consuming process which includes specifying a number of similar phrases used as training data NLP engine. As an example phrase: "I want to book a ticket from Stockholm to Malmö", the intent could be "ticket booking".

- **Entity extraction** Closely related to Intent routing is Entity extraction. In comparison, an intent is the overall meaning of a sentence where as an entity is a specific parameter within an utterance. As an example phrase: "I want to book a ticket from Stockholm to Malmö", both "Stockholm" and "Malmö" could be considered entities where Stockholm is the "source" entity and Malmö is the "destination" entity. Entities can be used for a number of different things when building a chatbot. It can for example be used for customizing responses or performing different kinds of logic.

- **Languages** We have defined the term *languages* in the context of NLP capability as languages supported by the NLP engine. The extent to which languages are supported by the NLP engine is measured both in terms of the number of languages it supports, as well as to what extent the languages are supported.

- **Context** Some CP allows for the configuration of context, which basically refers to the current state of the conversation. What the context enables is the possibility to use the same intent in several parts of a conversation. It gives the chatbot a context in which to interpret the users response and provide an appropriate response to the user.

Variable support A variable is a common term used in programming where it can be used for referring to a number, string or many other things. In the context of building a chatbot it basically serves the same purpose. If supported by the CP, allowing the developer of the chatbot to use it both as a means to store relevant information such as entities, choices the user have made or metadata about the user etc. as well as create dynamic content and perform custom logic.

- **Dynamic content** Dynamic content refers to the the possibility to use variables as a means to create dynamic bot responses. It is a powerful feature allowing for personalized responses or responses customized by the choices the user have done during the conversation.

- **Custom logic** There are a variety of ways that CPs support configuring *custom logic*, using variables, within a conversation. This feature refers to the possibility to perform conditional statements, e.g. if variable a equals 5 then do this or respond with that.

- **Access account information** Access to account information such as user name and e-mail is a feature that usually is enabled by the communication channel and implemented by a CP. Such information may be used in personalized messages, i.e. "Hi there {name}! I'm your favourite bot.".

Messaging Our research has shown the following messaging features to be the most important when building a chatbot. Messaging features are related to messages sent from the chatbot and their formatting.

- **OMNI-channel** A CP that supports OMNI-channel allows for the builder of a chatbot to specify a message once across several channels. I.e. the same conversation may be held in Messenger and WhatsApp, without the need to specify the message in both channels.

- **Rich content** Rich content is the ability for the CP to send messages that contain different type of media formats, i.e. audio or video. This usually enrich the conversation that the bot is having with its users.

- **Translations** This feature allows for translation of messages into different languages and also a method for detecting what language to use depending on the users preference.

Usability Here we collect features that is within the subject of user experience when building a chatbot and that our research has shown to be important for doing so.

- **Visual builder** A visual builder provides a visual representation of the conversational flow. A conversational flow can be represented in many ways, i.e. a tree or stack structure, or even as just text. However, we found the tree structure to aid the most during implementation of the chatbots.

- **Chat simulator** A chat simulator is integrated in the tool and allowing to test out and simulate the conversation with the chatbot. Some chat simulators also contain a debugger that makes it easier to detect bugs within the conversation.

- **Auto-configuration** Auto-configuration makes the set-up process of different features and settings faster and more convenient. This could be beneficial depending on if the auto-configuration is in line with what the user want. E.g. Chatfuel offers a very streamlined and highly automated way of integrating with Messenger, but with the drawback of only supporting Messenger. Whereas for the case of Dialogflow and Chatfuel, they require more manual configuration in order to integrate with different channels, but on the other hand also supporting several channels.

Analytics From our research we have concluded that *Conversation history* and *Visualization of data* are two of the most important features for analyzing and monitoring a chatbots performance. These features are very helpful when evaluating the chatbots performance and making improvements over time.

- **Conversation history** Conversation history is rather self explanatory. It is a feature making it possible to go back and view previous conversations between a user and the chatbot. In addition to being useful for analytics and performance evaluation it can in some case be used as training data for the NLP model.

- **Visualization of data** There is a large amount of data that can be gathered from chatbot conversations. The data itself is only useful if it is presented well. The visualization of the data can be more or less advanced, with some CP presenting the user response rate as a tree graph clearly showing where users drops of in the conversation.

Integrations The following features allow for different types of integration between a CP and other services. These integrations vastly expand the functionality natively supported by the CP itself. We have grouped different types of integrations into: *Webhooks*, *External services* and *Human handover*.

- **Webhooks** There are two ways that a CP can support *Webhooks*. Either by allowing an external source to trigger a conversation or affect the conversation when an event occurs. The other option is the possibility to trigger a webhook when an event happens within the conversation. E.g. if a user responds with a selection of an item for a shopping bot, one might want to update a shopping cart managed by an external system.
- **External services** External services is a broad categorisation of all external services that a CP supports or are integrated with. It includes services such as Google Sheets, Zapier, CRM systems, databases for storing message logs or analytics and much more. The external services available might vary a lot between different CPs and as a result the potential benefits with them might also vary.
- **Human handover** However good a chatbot is considered to be, chances are that sooner or later the chatbot will encounter a user which it cannot help. A common solution for this is transferring the conversation to a human agent and is referred to as *Human handover*. This feature makes it possible to configure when in a conversation a human agent is required to take over the conversation from the chatbot, thus making sure that the user leaves with a positive experience.

Custom scripting We consider custom scripting to be a more advanced feature that complements the chatbot building experience with the possibility to use programming. The extent to which programming can be used may vary from CP to CP. Some common use-cases where custom scripting is regularly used are to handle custom logic, create dynamic content or set and manipulate variables.

Low technical depth This characteristic examines the technical skills required to use a CP. There are many aspects that contributes to the technical depth of a CP. If it for example is required to use custom scripting or more advanced integrations in order to build a chatbot within a CP, then it would be considered to have a high technical depth. For companies lacking technically skilled resources, this criteria might be especially important to consider.

Additional criteria The following criteria have been found to be essential for selecting a CP and can by themselves be a deal-breaker for if a CP is suitable for a company or not. These are hence not included in our final list of criteria to be used with the selection method.

- **Price** The cost of using a CP, whether it being a subscription fee, price per message or any other pricing option.
- **Supported channels** The communication channels which the CP is integrated with, such as Facebook Messenger, WhatsApp, SMS etc.
- **Security** How secure is the CP. Which measures have been taken to secure data, access data and data backup? Is the CP GDPR compliant? Is it possible to self-host the CP or is only available through SaaS?

Excluded features and characteristics Some of the features and characteristics that were found through the literature review and hands on experience with CPs were excluded from the final list of criteria due to their limited importance. The following are short descriptions of each of these.

- **Pre-defined intents** Do the CP offer pre-defined intents, such as "Greeting intent" etc.
- **Versioning** Versioning can serve many purposes. Firstly, it can allow to both save a current version of the chatbot and revert back to older versions if need be. Secondly, it can assist with keeping track of different released versions of the chatbot. In some cases it can also make it possible to release several different versions of a chatbot.
- **Conversation templates** Do the CP offer conversational templates, which basically is "ready to use" chatbots that can be used directly or be modified to suit the purpose of a company.
- **Message per channel** The possibility to customize a bot message per communication channel basis.
- **View active chats** With this feature the chatbot admin may monitor any active conversations between a user and the chatbot allowing them, in real time, to detect if the user gets stuck or experience any other problems with the chatbot.
- **A/B-testing** The possibility to perform A/B-testing of the chatbot.
- **Message preview** The ability to get a preview of a message as it would be rendered on the intended communication channel.
- **Import/export conversations** Is there a possibility to export and save a conversation, and likewise import an existing conversation into the CP.
- **Pre-defined entities** Just as pre-intents, do the CP offer pre-defined entities, such as country or city entities.
- **Sentiment analysis** The chatbots ability to extract the sentiment from a user input, e.g. if they are happy, angry, annoyed etc.
- **Re-engage users** A feature which allows to set a timer for re-engaging a user to the conversation. E.g. sending out a reminder message.

- **SmallTalk** In addition to messages created and defined by the CP user, if enabled and supported, SmallTalk allow the chatbot to provide additional responses to some generic messages and questions without the need to specify additional messages, intents and rules.
- **Auto-scaling** Auto-scaling refers to a chatbots ability to handle a varying amount of traffic (number of concurrent users per time unit).

5.1.2 Literature review

The semi-systematic approach used for the literature review yielded a priority list shown in Table 5.1, where each row represents a CP feature or characteristic, a priority value (from Table 3.1) as well as a motivation behind the appointed value.

Feature / characteristic	Priority	Motivation
Visualization of data	3	Referenced in one article [9] and several GL [36], [37], [38], [39]. Generally stated as important or essential for a CP
Entity extraction	3	Referenced in several articles [33], [9], [34]. Sometimes stated as essential
External services	3	Referenced in several articles [9], [34] and one GL [37]. Stated as essential in several areas for the construction of chatbots
Intent routing	3	Referenced in all articles [32], [47], [33], [9], [34] and GL [36], [37], [38], [35], [39]. Also stated as essential in some articles
Languages	3	Referenced in one article [34] and one GL [35]. Stated as a core feature
OMNI-channel	3	Referenced in one article [9] and several GL [36], [38], [35], [39]
Visual builder	3	Referenced in different ways in all articles [32], [47], [33], [9], [34] and two GL [39], [36]. Often said to be important for non-tech users
Webhooks	3	Referenced in several articles [34], [9] and one GL [37], Stated as essential for large enterprises
Auto-configuration	2	Referenced in one article [9] and one GL [35]. Stated as important for non-tech users
Context	2	Referenced in several articles [32], [34], [9] and GL [36], [35]. Stated as important for intent routing
Custom logic	2	Referenced in article [34], and several GL [39], [37]
Dynamic content	2	Referenced in several articles [32], [9] and GL [36]
Low technical depth	2	Referenced in several articles [34], [9] and one GL [39]
Sentiment analysis	2	Referenced in several GL [36], [38], [35]
Translations	2	Referenced in several GL [36], [37]
A/B testing	1	Only referenced in one GL [36]

Continued on next page

Table 5.1 – *Continued from previous page*

Feature / characteristic	Priority	Motivation
Access account information	1	Only referenced in one GL [37]
Auto-scaling	1	Referenced in one GL [38]
Chat simulator	1	Only referenced in one GL [39]
Conversation history	1	Only referenced in one GL [36]
Conversation templates	1	Only referenced in one article [9] and one GL [39]
Import/Export conversations	1	Only referenced in one GL [37]
Message per channel	1	Only referenced in one GL [39]
Pre-defined entities	1	Only referenced in one article [34]
Pre-defined intents	1	Only referenced in one article [34]
Re-engage users	1	Only referenced in one article [9]
Versioning	1	Only referenced in one GL [37]

Table 5.1: Result from the literature review

When conducting the literature review we noticed a lack of a common terminology in literature regarding chatbots and conversational platforms. Many of the found features and characteristics were written and defined using different words across the literature but referred to the same feature or characteristic. The opposite could also be true, where the same terminology was used but where the authors did not refer to the same thing. An example of this is the term *variable*. Variables was used to describe the data points used in *Entity Extraction*. It could however also be used to describe variables used for custom scripting or defining dynamic messages. When compiling the result we either selected the term that was most used or the term that we found to be most descriptive. The definitions and descriptions of each can be found under the Section *Feature Analysis*.

5.1.3 Hands on with existing tools

Here we present the result found from our hands on experience. The CPs we looked at were Dialogflow, Chatlayer and Chatfuel. The result is presented in Table 5.2, where each row represents a characteristic or feature, a priority value as well as a motivation behind the appointed value.

Feature / characteristic	Priority	Motivation
Chat simulator	3	Supported by all CPs. Was a very helpful tool during the development, both in order to test the chatbot and in detecting bugs.
Conversation history	3	Supported by all CPs. Presented as conversational log.

Continued on next page

Table 5.2 – *Continued from previous page*

Feature / characteristic	Priority	Motivation
Custom logic	3	Supported by all CPs. Very important in order to make the chatbot "smart" and adapting responses to user input.
Dynamic content	3	Supported by all CPs. Dynamic messages were essential for the development of the digital commerce bot where a message content could change based on user selection.
Human handover	3	We did not try this feature out, but it was both well supported and promoted by each CP we looked at.
External services	3	To varying extent supported by all CPs. Essential feature in order to connect to different channels, do human handover, custom logging, connect to CRMs etc.
Intent routing	3	Supported by all CPs, even Chatfuel which primarily focuses on the development of rule-based chatbots.
Low technical depth	3	We found this to be an important aspect of all CPs. We believe this aids in providing a view on how technical a CP is to use.
Rich content	3	Rich content was well supported by all platforms for the channels they supported.
Translations	3	We see that there is great value in the possibility to make a bot multi-lingual for businesses that operates in different countries. This feature was to some extent supported by each CPs.
Visualization of data	3	We found that all CPs did a good job of presenting most of the relevant data.
Webhooks	3	Supported by all CPs. It would almost be impossible to do dynamic content without it.
A/B-testing	2	Naively supported by Chatfuel, Supported by Chatlayer and Dialogflow as well but more cumbersome to use.
Auto-configuration	2	To some extent supported by each platform, but did not provide much benefits for us when implementing our BCs.
Conversation templates	2	Supported by Chatfuel and Dialog flow. Did not provide much benefit for our implementation, other than acting as examples to look at.
Custom scripting	2	Supported by Dialogflow and Chatlayer. It require programming skills to use but allowed us to do more advanced logic and customization's.
Entity extraction	2	Supported by Chatlayer and Dialogflow. Was necessary for developing the Marketing BC in Dialogflow.
Import/export conversations	2	Only supported by Dialogflow but did come in useful for development for collaboration and for importing external example conversation to look at.

Continued on next page

Table 5.2 – *Continued from previous page*

Feature / characteristic	Priority	Motivation
Languages	2	Supported by both Dialogflow and Chatlayer. Essential for building multi-lingual bots.
Message per channel	2	Chatfuel only supports one channel which inherently makes it support this. Dialogflow have the best support for this where each message easily can be configured on a per channel basis. It is possible to achieve in Chatlayer but is inconvenient.
OMNI-channel	2	Only supported by Dialogflow, but currently being developed for Chatlayer. Not a crucial feature for our implementations since we only implemented chatbots on one channel (Messenger).
Pre-defined intents	2	Only supported in Dialogflow, but saved a lot of development time.
Versioning	2	The platforms have all taken different approaches when it comes to how they support versioning, where we only found the approaches used by Dialogflow and Chatlayer somewhat useful.
Visual Builder	2	Only supported by Chatlayer. We found this feature very helpful. It both made the development easier and faster.
Context	2	Supported by Chatlayer and Dialogflow. We found it to be a required feature for building chatbots in Dialogflow.
Message preview	1	Partly supported in each CP, but very limited in Chatlayer and Dialogflow where a message is not truly visualized as it would be on a live channel.
Pre-defined entities	1	We have not found any benefits with this feature during our development.
Re-engage users	1	Only supported by Chatfuel.
View active chats	1	Only supported by Chatfuel. Can be seen as a quick filter for conversation logs. Might aid human to take over the chat if they spot a problem.
Sentiment analysis	1	Only supported by Dialogflow.
SmallTalk	1	Only supported by Dialogflow. A nice to have feature which makes the bot more "human-like", but not necessary.
Access account information	1	Supported by all CPs, but we found it to be more of a "nice to have" feature than essential for building a chatbot.

Table 5.2: Result from the hands on with existing tools

5.1.4 Consulting experts

The questionnaire used for the expert consulting yielded the following result presented in Table 5.3. The table contains priority values which the respondents appointed to each feature and characteristic as well as the average score deducted from these values. For some of the characteristics the respondents also provided motivations behind the values they appointed. These motivations can be found in Tables B.2, B.3, B.4 and B.5 in Appendix B.

Feature / Characteristic	Resp. 1	Resp. 2	Resp. 3	Resp. 4	Avg. Mean
Dynamic content	3	3	3	3	3
External services	3	3	3	3	3
Visual builder	3	3	3	3	3
Auto-scaling	2	3	3	3	2.75
Chat simulator	3	3	2	3	2.75
Custom logic	3	2	3	3	2.75
Custom scripting	3	2	3	3	2.75
Intent routing	3	3	2	3	2.75
Access account information	3	2	2	3	2.5
Auto-configuration	1	3	3	3	2.5
Entity extraction	3	2	2	3	2.5
Human handover	2	2	3	3	2.5
Rich content	3	3	2	2	2.5
View active chats	2	2	3	3	2.5
Conversation history	3	2	2	2	2.25
Context	2	2	3	2	2.25
OMNI-channel	2	3	1	3	2.25
Pre-defined intents	1	2	3	3	2.25
Versioning	3	2	1	3	2.25
Webhooks	2	2	3	2	2.25
Conversation templates	2	2	2	2	2
Low technical depth	2	2	2	2	2
Message per channel	1	2	3	2	2
Message preview	1	2	2	3	2
Visualisation of data	2	2	1	3	2
A/B-testing	2	2	1	2	1.75
Languages	1	2	1	3	1.75
Pre-defined entities	2	2	2	1	1.75
Translations	1	2	1	3	1.75
Re-engage users	1	2	1	2	1.5
SmallTalk	1	2	1	2	1.5
Import/export conversations	2	1	1	1	1.25
Sentiment analysis	1	1	1	2	1.25

Table 5.3: Results from the questionnaire

5.1.5 Feature analysis

Here we present our result from the feature analysis. Table 5.4 summarizes the priority values found from the literature review, hands on experience and expert consulting together with an average mean, 3.1, calculation. In this result we also show how we categorized the features and characteristics into main-criteria and sub-criteria which is presented in figure 5.1. A main-criterion is a group of features or characteristics that all relates or contribute to a related topic. Features and characteristics that relates to a main-criterion is referred to as sub-criteria. We did not find any suitable grouping for *Custom scripting* and *Low technical depth* which is why these are categorized as individual main-criteria. Criteria that made it in to this list was only features and characteristics with a mean priority value of 2.0 or higher. This table is also complemented with a list of all these criteria where they are described in more detail.

Feature / characteristic	Hands-on	Literature review	Expert Consulting	Avg. Mean
External services	3	3	3	3.00
Intent routing	3	3	2.75	2.88
Dynamic content	3	2	3	2.75
Visual builder	2	3	3	2.75
Custom logic	3	2	2.75	2.63
Webhooks	3	3	2.25	2.63
Entity extraction	2	3	2.5	2.50
Chat simulator	3	1	2.75	2.38
OMNI-channel	2	3	2.25	2.38
Auto-configuration	2	2	2.5	2.25
Human handover	3	1	2.5	2.25
Rich content	3	1	2.5	2.25
Visualization of data	3	2	2	2.25
Low technical depth	3	2	2	2.25
Context	2	2	2.33	2.17
Conversation history	3	1	2.25	2.13
Custom scripting	2	1	2.75	2.13
Languages	2	3	1.75	2.13
Translations	3	2	1.75	2.13
Access account information	2	1	2.5	2.00
Auto-scaling	1	1	2.75	1.88
Pre-defined intents	2	1	2.25	1.88
Versioning	2	1	2.25	1.88
Conversation templates	2	1	2	1.75
Message per channel	2	1	2	1.75
View active chats	1	1	2.5	1.75
A/B-testing	2	1	1.75	1.63
Message preview	1	1	2	1.50

Continued on next page

Table 5.4 – *Continued from previous page*

Feature / characteristic	Hands-on	Literature review	Expert Consulting	Avg. Mean
Import/export conversations	2	1	1.25	1.38
Pre-defined entities	1	1	1.75	1.38
Sentiment analysis	1	2	1.25	1.38
Re-engage users	1	1	1.5	1.25
SmallTalk	1	1	1.5	1.25

Table 5.4: Means calculation of the priority values gathered from the literature review, hands on experience and expert consulting.

Figure 5.1: The final criteria divided into main -and sub-criteria.

In general the results found and priority values assigned from each research approach matches up quite well. The most important features and characteristics are roughly the same. There are however a few features and characteristics that stands out and is worth taking an extra look at. The first one being *auto-scaling* which was found to be of low importance from both the literature review and hands on experience, but was rated very high from the consulted experts. It could be argued that it is such an important feature, just based on experts input, that it would be valuable to consider. In order to do an fair evaluation for each feature and characteristic we kept to the final priority value from the means calculation, found in table 5.4, and is why *auto-scaling* was not included in the final list of criteria. Another result

that stands out is *rich content* which scored high for both hands on experience and from the expert consulting but was not mentioned at all in any literature we looked at. The reason for it not being mentioned in literature we believe is down to two reasons. Firstly, it is something that can be assumed to be supported if a specific communication channel is supported. E.g. if the CP offers the possibility to build chatbots in Facebook Messenger, then it would also support creating all messages that Messenger supports (which includes rich content). The second reason for it not being included we believe is because much of the literature is focused on text-based chatbots, which by its very nature does not include *rich content*.

5.2 Evaluation of CP Selection Method

In this section we present the results from our evaluation of the CP selection method. The goal with the evaluation was to determine the quality of the results from using the method as well as finding out how feasible it is to use as a selection method for conversational platforms. We tested the method on our two benchmark cases in order to determine which of Chatfuel, Dialogflow and Chatlayer that is best suited CP for each case respectively. We then present how well these results reflect our own experience from implementing the BCs in each tool.

We tested the method by following each step from Chapter 4. For these tests, step 1 and 2 were common for both use-cases since we used the same conversational platforms for both evaluations. Step 1 and 2 is first presented together, followed by a separate presentation of the result of step 3 through 6 for both evaluations.

Step 1: Select CP alternatives

The alternatives we have chosen to compare is the conversational platforms Chatfuel, Dialogflow and Chatlayer. The alternatives were chosen so that we had a conversational platform that represented each of the chatbot technologies found in section 2.2 Chatbot Technologies.

Step 2: Evaluate sub-criteria for alternatives

In the second step we appointed a value for each sub-criteria of how well it is supported by a CP alternative, where we used the final list of criteria from figure 5.1. The values that we set were based on our experience from using the conversational platforms previously in the study.

One example result for the main-criteria *NLP Capabilities* is presented as a table in Table 5.5 where the priorities for our CP alternatives for each sub-criteria have been calculated. Tables for all main-criteria and corresponding sub-criteria can be found in the appendix C, the resulting ***PASC*** matrix can be found in Table 5.6[†].

[†]The ***PASC*** matrix has been transposed for readability purposes in the report but will not be used as such in calculations in later steps.

Approach for assigning priority values

We assigned the priority values (PV) for the sub-criteria for each platform based on our experience from using and implementing the two use cases (digital commerce and marketing chatbot) in each platform during the hands on experimentation with the platforms. For reference see Section 3.1.2. For the complete UML representations of the use cases, see Appendix A. The following are a few of the narratives we implemented during the hands on experimentation. These were used to evaluate the following five criteria: *Rich content*, *Webhooks*, *Custom scripting*, *Custom logic* and *Entity extraction*. The motivation behind the PV we assigned for the these criteria only serve as examples behind how we assigned the PV. However, the same approach was also used, if possible, for the other sub-criteria as well.

Digital commerce narrative examples

1. **Bot:** Which kind of shoes are you interested in? *Shows categories of different women shoes, represented as carousels containing images, text and a quick reply button*

   User: I'm interested in sports shoes. *Selects a category with text message or selection button*

   Bot: *Shows different shoes within the chosen category*

2. **Bot:** What size in shoes are you?

   User: *Alternative answers:*
   1. I'm size 42.
   2. 42.
   3. I prefer 42 in shoe size.

   Bot: Great, we have the size 42 in stock!

3. **Bot:** Would you like to buy some more or are you happy with your selection? *This message is triggered by an mocked external system via webhooks*

   User: I'd like to continue shopping. *Triggers a webhook, adding a previously selected shoe to a mocked shopping cart*

Marketing narrative examples

4. **Bot:** Please describe the problem you encountered.

   User: I did not like the bot being so nice

5. **Bot:** Oweee, we're half-way through! In this section you rank the questions from 1 to 5. *Shows selection options 1 to 5*
   1 - Strongly disagree
   5 - Strongly agree

   Bot: It was clear for me how to proceed in each step of the conversation. + *Also sends out quick reply buttons for values 1 through 5*

   User: *Responds to bot by pressing quick reply option 5*

Rich content

From the implementation of narratives 1 and 5, we found that all three platforms supported the same kind of rich content for messages. This was validated by testing to send a carousel message, containing shoes categories, in narrative 1 and also suggested replies in narrative 5. Facebook messenger was the platform used for this test. The messages looked identical on the three chatbots created with the different platforms. Since we found that all platforms performed equally well for this criteria we assigned the PV **1** to each platform.

Webhooks

By implementing and testing narrative 3, we found that the support for webhooks differed between the platforms. From narrative 3 (first message) we tried to initialize the bot conversation using an external system via an API provided by the platforms. This was only possible in Chatlayer and Dialogflow. It was however possible to update a mocked shopping cart using webhooks in each platform. This was validated by specifying that a webhook should be triggered in narrative 3 after the user replied with *I'd like to continue shopping.*. For this criterion, Dialogflow and Chatlayer got appointed the PV of **1** due to them exposing an API for triggering different events. Since Chatfuel did not support this it got appointed the PV **3**.

Custom scripting

Both Chatlayer and Dialogflow allows for custom scripting. This was tested by implementing the first bot message in narrative 4, using only the custom scripting feature. It was possible to create the same intended message in both platforms. However, we noticed that Chatlayer was more restricted in the scripting environment than Dialogflow. Hence, Dialogflow got the edge in the prioritization of this criterion and was appointed the PV of **1**. Chatlayer got appointed the PV of **3** and since Chatfuel did not support this feature at all, it got a PV of **9**.

Entity extraction

We found that it was possible to extract entities from user messages in both Chatlayer and Dialogflow. This was tested by implementing narrative 2, where we tried to extract the shoe size from the user message. The support for entity extraction is implemented very similarly in both platforms. Dialogflow got the edge for this prioritization due to it also including a wide range of pre-defined entities. Because of this we gave Dialogflow a priority value of **1** and Chatlayer **3** Chatfuel did not support this feature at all and hence got assigned the priority value **9**.

NLP Capabilities

	Chatfuel	**Dialogflow**	**Chatlayer**
Intent routing			
Chatlayer	5	3	1
1/a	0.2	0.3333	1

Continued on next page

Table 5.5 – *Continued from previous page*

	Chatfuel	Dialogflow	Chatlayer
Priority	0.1304	0.2174	0.6522
Entity extraction			
Dialogflow	9	1	3
1/a	0.1111	1	0.3333
Priority	0.0769	0.6923	0.2308
Languages			
Chatlayer	9	3	1
1/a	0.1111	0.3333	1
Priority	0.0769	0.2308	0.6923
Context			
Chatlayer	9	3	1
1/a	0.1111	0.3333	1
Priority	0.0769	0.2308	0.6923

Table 5.5: Priorities for the NLP main-criterion

PASC

Sub-criteria	Chatfuel	Dialogflow	Chatlayer
Intent routing	0.1304	0.2174	0.6522
Entity extraction	0.0769	0.6923	0.2308
Languages	0.0769	0.2308	0.6923
Context	0.0769	0.2308	0.6923
Dynamic content	0.2174	0.1304	0.6522
Custom logic	0.1304	0.2174	0.6522
Access account information	0.6000	0.2000	0.2000
OMNI-channel	0.0886	0.7975	0.1139
Rich content	0.3333	0.3333	0.3333
Translations	0.1304	0.2174	0.6522
Visual builder	0.2308	0.0769	0.6923
Chat simulator	0.2174	0.1304	0.6522
Auto-configuration	0.6000	0.2000	0.2000
Conversation history	0.2000	0.2000	0.6000
Visualization of data	0.1304	0.2174	0.6522
Webhooks	0.1429	0.4286	0.4286
External services	0.1304	0.6522	0.2174
Human handover	0.7	0.1	0.2
Custom scripting	0.0769	0.6923	0.2308
Low technical depth	0.7627	0.0847	0.1525

Table 5.6: ***PASC*** table for the CP alternatives

5.2.1 Digital Commerce

The steps 3 through 6 need to be performed separately for each use-cases. In this section we present the results from performing these steps for the *Digital commerce* BC.

Step 3: Prioritize sub-criteria in main-criteria

The resulting prioritization of sub-criteria for each main-criteria is presented in Tables 5.7 and 5.8, where the resulting ***PSC*** matrix is presented in Table 5.9.

NLP capabilities	**Intent routing**	**Entity extraction**	**Languages**	**Context**
Intent routing	1	3	7	5
1/a	1	0.3333	0.1429	0.2
Priority	0.5966	0.1989	0.0852	0.1193

Variable support	**Dynamic content**	**Custom logic**	**Access account information**
Dynamic content	1	3	5
1/a	1	0.3333	0.2
Priority	0.6522	0.2174	0.1304

Messaging	**OMNI-channel**	**Rich content**	**Translations**
Rich content	3	1	5
1/a	0.3333	1	0.2
Priority	0.2174	0.6522	0.1304

Usability	**Visual builder**	**Chat simulator**	**Auto-configuration**
Visual builder	1	3	5
1/a	1	0.3333	0.2
Priority	0.6522	0.2174	0.134

Analytics	**Conversation history**	**Visualization of data**
Visualization of data	3	1
1/a	0.3333	1
Priority	0.25	0.75

Integrations	**Webhooks**	**External services**	**Human handover**
Webhooks	1	5	3
1/a	1	0.2	0.3333
Priority	0.6522	0.1304	0.2174

Table 5.7: Priority of sub-criteria within main-criteria for the digital commerce BC

	Priority
Custom scripting	1
Low technical depth	1

Table 5.8: Priority of the standalone main-criteria for the digital commerce BC

Main-criteria / Sub-criteria	NLP capabilities	Variable support	Messaging	Usability	Analytics	Integrations	Custom scripting	Low technical depth
Intent routing	0.5966	0	0	0	0	0	0	0
Entity extraction	0.1989	0	0	0	0	0	0	0
Languages	0.0852	0	0	0	0	0	0	0
Context	0.1193	0	0	0	0	0	0	0
Dynamic content	0	0.6522	0	0	0	0	0	0
Custom logic	0	0.2174	0	0	0	0	0	0
Access account information	0	0.1304	0	0	0	0	0	0
OMNI-channel	0	0	0.2174	0	0	0	0	0
Rich content	0	0	0.6522	0	0	0	0	0
Translations	0	0	0.1304	0	0	0	0	0
Visual builder	0	0	0	0.6522	0	0	0	0
Chat simulator	0	0	0	0.2174	0	0	0	0
Auto-configuration	0	0	0	0.1304	0	0	0	0
Conversation history	0	0	0	0	0.25	0	0	0
Visualization of data	0	0	0	0	0.75	0	0	0
Webhooks	0	0	0	0	0	0.6522	0	0
External services	0	0	0	0	0	0.1304	0	0
Human handover	0	0	0	0	0	0.2174	0	0
Custom scripting	0	0	0	0	0	0	1	0
Low technical depth	0	0	0	0	0	0	0	1

Table 5.9: The transposed ***PSC*** matrix of sub-criteria priorities in a main-criteria for the digital commerce use case

Step 4: Calculate priority of alternatives for main-criteria

CP / *Main-criteria*	**Chatfuel**	**Dialogflow**	**Chatlayer**
NLP capabilities	0.1088	0.3146	0.5766
Variable support	0.2484	0.1584	0.5932
Messaging	0.2537	0.4191	0.3272
Usability	0.276	0.1046	0.6194
Analytics	0.1478	0.213	0.6391
Integrations	0.252	0.3929	0.3551
Custom scripting	0.0769	0.6923	0.2308
Low technical depth	0.7627	0.0847	0.1525

Table 5.10: ***PAC*** for to the digital commerce BC calculated using ***PSC*** · ***PASC***

Step 5: Prioritize main-criteria

	NLP capabilities	**Variable support**	**Messaging**	**Usability**	**Analytics**	**Integrations**	**Custom scripting**	**Low technical depth**
Integrations	8	3	2	4	6	1	5	7
1/a	0.125	0.3333	0.5	0.25	0.2	1	0.2	0.1429
PC	0.046	0.1226	0.184	0.092	0.0613	0.3679	0.0736	0.0526

Table 5.11: ***PSC*** for the marketing BC

Step 6: Calculate final score

CP	Chatfuel	Dialogflow	Chatlayer
PA	0.255	0.3337	0.4113

Table 5.12: ***PA*** for the digital commerce BC calculated using ***PC*** · ***PAC***

The highest scoring CP for the *Digital commerce* BC is Chatlayer followed by Dialogflow.

5.2.2 Marketing

The steps 3 through 6 needs to be performed separately for each use-cases. In this section we present the results from performing these steps for the *Marketing* BC.

Step 3: Prioritize sub-criteria in main-criteria

The resulting prioritization of sub-criteria for each main-criteria is presented in Tables 5.13 and 5.14, where the resulting ***PSC*** matrix is presented in Table 5.15.

NLP capabilities	Intent routing	Entity extraction	Languages	Context
Intent routing	1	5	5	5
1/a	1	0.2	0.2	0.2
Priority	0.6250	0.1250	0.1250	0.1250

Variables support	Dynamic content	Custom logic	Access account information
Custom logic	9	1	3
1/a	0.1111	1	0.2
Priority	0.0847	0.7627	0.2542

Messaging	OMNI-channel	Rich content	Translations
OMNI-channel	1	3	7
1/a	1	0.3333	0.1429
Priority	0.6774	0.2258	0.0968

Usability	Visual builder	Chat simulator	Auto-configuration
Visual builder	1	3	7
1/a	1	0.3333	0.1429
Priority	0.6774	0.2258	0.0968

Analytics	Conversation history	Visualization of data
Visualization of data	3	1
1/a	0.3333	1
Priority	0.25	0.75

Integrations	Webhooks	External services	Human handover
Human handover	3	5	1
1/a	0.3333	0.2	1
Priority	0.2174	0.1304	0.6522

Table 5.13: Priority of sub-criteria within a main-criteria for the marketing BC

	Priority
Custom scripting	1
Low technical depth	1

Table 5.14: Priority of the standalone main-criteria for the marketing BC

Main-criteria / *Sub-criteria*	**NLP capabilities**	**Variable support**	**Messaging**	**Usability**	**Analytics**	**Integrations**	**Custom scripting**	**Low technical depth**
Intent routing	0.625	0	0	0	0	0	0	0
Entity extraction	0.125	0	0	0	0	0	0	0
Languages	0.125	0	0	0	0	0	0	0
Context	0.125	0	0	0	0	0	0	0
Dynamic content	0	0.0847	0	0	0	0	0	0
Custom logic	0	0.7627	0	0	0	0	0	0
Access account information	0	0.2542	0	0	0	0	0	0
OMNI-channel	0	0	0.6774	0	0	0	0	0
Rich content	0	0	0.2258	0	0	0	0	0
Translations	0	0	0.0968	0	0	0	0	0
Visual builder	0	0	0	0.6774	0	0	0	0
Chat simulator	0	0	0	0.2258	0	0	0	0
Auto-configuration	0	0	0	0.0968	0	0	0	0
Conversation history	0	0	0	0	0.25	0	0	0
Visualization of data	0	0	0	0	0.75	0	0	0
Webhooks	0	0	0	0	0	0.2174	0	0
External services	0	0	0	0	0	0.1304	0	0
Human handover	0	0	0	0	0	0.6522	0	0
Custom scripting	0	0	0	0	0	0	1	0
Low technical depth	0	0	0	0	0	0	0	1

Table 5.15: The transposed ***PSC*** matrix of sub-criteria priorities in a main-criteria for the marketing BC

Step 4: Calculate priority alternatives for main-criteria

CP / *Main-criteria*	Chatfuel	Dialogflow	Chatlayer
NLP capabilities	0.1104	0.2801	0.6095
Variables support	0.2704	0.2277	0.6035
Messaging	0.1479	0.6365	0.2156
Usability	0.2635	0.1009	0.6356
Analytics	0.1478	0.213	0.6391
Integrations	0.4734	0.2633	0.2633
Custom scripting	0.0769	0.6923	0.2308
Low technical depth	0.7627	0.0847	0.1525

Table 5.16: ***PAC*** for to the marketing BC calculated using ***PSC*** · ***PASC***

Step 5: Prioritize main-criteria

	NLP capabilities	Variable support	Messaging	Usability	Analytics	Integrations	Custom scripting	Low technical depth
Integrations	7	5	4	3	1	6	8	2
1/a	0.1429	0.2	0.25	0.3333	1	0.1667	0.125	0.5
PC	0.0526	0.0736	0.092	0.1226	0.3679	0.0613	0.046	0.184

Table 5.17: ***PSC*** for the marketing BC

Step 6: Alternative scores

CP	Chatfuel	Dialogflow	Chatlayer
PA	0.2989	0.2444	0.4642

Table 5.18: ***PA*** for the marketing BC calculated using ***PC*** · ***PAC***

The highest scoring CP for the *Marketing* BC is Chatlayer followed by Chatfuel.

5.2.3 Analysis

The purpose of the validation of the CP selection method was to determine if it is usable for the purpose of giving recommendations on the most suitable conversational platform for the needs of a specific company. By applying the method on the two previously defined BCs we found that it in general provided a result that was in line with our expectations from previously implementing chatbots, based on the BCs, in each CP. When testing the method for the Digital Commerce BC, Chatlayer was found to be the most suitable conversation platform to use, closely followed by Dialogflow. This result was much in line with our expectations since Chatlayer in general was superior in both the required set of features and the way they were implemented.

Chatlayer did also score the highest for the Marketing BC, followed by Chatfuel. The most important criteria for this BC we found to be *Analytics*, *Low technical depth*, *Usability* and *Messaging*. Chatlayer scored very high in both analytics and usability, moderate in Messaging and low in Low technical depth. In general we are not surprised by this results, even if we thought that Chatfuel might score higher. The reason for this being that the Marketing BC is a simple pre-defined conversation, which does not require any higher technical skills to understand or to develop. From our personal experience, developing the BC using Chatfuel, we found it to be very well suited for this kind of use-case. However, we also found the BC to be easily implemented in Chatlayer, which more or less has all of the functionality that Chatfuel offers and more. In this case the recommendation provided by the method might not be enough for a decision to be made, but should be complemented with additional analysis and evaluation of the hard requirements such as pricing, supported communication channels and security.

We believe that one of the main reasons behind Chatlayer scoring highest for both BCs is because that it is a hybrid conversational platform which supports features found in both rule-based and AI platforms. Chatfuel and Dialogflow which are rule-based and AI-based respectively, is more focused on a single type of chatbot development and as a result lacks support for some features. It is not only the fact that Chatlayer support a big variety of features that makes it score so high, it is also because the features, for the most part, is well implemented in comparison to Dialogflow and Chatfuel.

It is important to note that the scores and final recommendation provided by the method will vary depending on who is using the method. Companies have different priorities and will hence also assign different comparison values for each criterion even if the use-case for the chatbot might be in the same area. Additionally, the recommendation will also vary depending on how the comparison values are assigned for how well the criteria are met by the different CPs. This prioritization will always be subjective to the one using the method. The method does not aim to give a definite result on which CP that is the best, but rather help companies making an informed decision about which platforms could suit their needs the best.

Chapter 6

Discussion

Through a literature review, hands on experience and by consulting experts we came up with a taxonomy of 20 features and characteristics offered by different conversational platforms. We categorized these into a set of criteria for a company to consider when they want to develop a chatbot in a conversational platform. When comparing the results in 5.1.5 Feature Analysis with Braun and Matthes [32] study, where requirements are defined at a rather high-level and focused on functional requirements, we believe that our list of criteria is more suited to use for a business for selection of a platform. The study done by Kostelník, Pisařovic, Muroň, Dařena and Procházka [9] provided a list of eight criteria for companies to consider when deciding on a CP. This low number of criteria made their proposed selection method relatively easy to use, but left out a number of important aspects of chatbot development. This limitation was mentioned in their article and is one of the things we tried to improve upon in this study. Two of the criteria stated in their list was *Price* and *Delivery channels support*, which we left out from our list of criteria and instead include in a list of hard requirements that is described in more detail under the Section 6.1.1.1 Price and Section 6.1.1.2 Channel support later in the report. In short, we believe these criteria by themselves will make or break if a CP is suitable for a company to use. Based on the results from these studies, as well as a study made by Canonico and De Russis [34] which presents a list of 13 characteristics for chatbot development, we believe that the taxonomy presented in this study covers a broader set of requirements for chatbot development as well as being verified by experts in the field.

In comparison with the selection method presented by Kostelník et al. [9], the selection method we suggest works similarly by doing a pair-wise comparison of criteria. The difference is that our suggested method is more suited for conducting a comparison of larger number of criteria, whereas the method proposed by Kostelník et al. becomes unfeasible to use when more than 8-10 criteria needs to be considered.

We believe that there are a number of benefits with considering larger number of criteria.

A more obvious one is that the decision will be more grounded and adapted to the needs of a company. A less obvious benefit is that it makes the company understand, discuss and define their chatbot needs, which can be especially helpful if it is the first time that the company plans to develop a chatbot.

6.1 Limitations

6.1.1 Hard requirements

We define hard requirements as features or characteristics of a CP that we believe by themselves make or break if a platform is suitable for the development of a chatbot for specific use-cases or a company needs. We have thus chosen to exclude hard requirements are hence not included in our method for selecting a CP but could instead be considered beforehand in order to determine which CP to consider or exclude from consideration.

6.1.1.1 Price

There is a number of reasons to why we have chosen to exclude the price of using the CP in our proposed method for selecting a CP. The first and foremost reason being that there are big differences when it comes to both the pricing models that different CP providers offers, as well as the actual price they charge for using the service. Some CP providers offers a subscription model where you pay a monthly fee for using the platform and depending on the amount of traffic (number of messages per month) or the features you require, this fee might be higher or lower. Chatfuel uses a model where they offer three different subscription levels. They have a free version which offers basic functionality, but also sends out a branded message in the beginning of each conversation, a pro version (15$/month) which removes the branding and unlocks some more advanced features. They also offer a premium version which includes a full set of features and VIP support. The price for the premium version is set per customer basis. Another common pricing model is a price per message model. This model is e.g. used by Dialogflow where you pay a small fee for each message that is being sent from the chatbot. Dialogflow offers different account levels where each higher level unlocks more features and increase the price per message. For the standard version a limited amount of messages is free. For the Essential version the fee is 0.002$/message and for Enterprise level the price per message is 0.004$. In order to do a fair comparison on pricing one would have to do a proper investment calculation which is outside of the scope of this thesis.

6.1.1.2 Channel support

Another requirement that could make or break if CP should be considered is which communications channels it supports. This is perhaps the first thing a company should consider before starting to decide and compare different CP solutions. If the specific channel(s) where a company is interested in publishing their chatbot on is not supported by a platform, then that platform should most likely be removed from consideration.

It is however worth mentioning that if there is support for the specific channel(s) that is essential for the company and the CP at the same time supports other channels, we believe it could be considered to be an additional feature to add to the list of criteria. This could be valuable if there are plans to expand the chatbot across multiple channels in the future.

6.1.1.3 Security

Security is vital for all systems, especially when it comes to storing and processing sensitive data. This is a common occurrence in a chatbot setting where there is no guarantee for what information a user might provide in a message. The CPs we have looked at in this study have all had some level of security implemented. They all state that they are GDPR compliant, and in Chatlayer it is possible to mark variables as sensitive data in order to not save it. The security requirements for different companies might vary greatly and it is very hard to assess the level of security provided by a CP without having direct contact with the CP provider. This is outside the scope of this thesis but it is a factor that we believe that one should consider before settling for a CP.

6.1.2 CP selection method

Testing the method on some common business use-cases we found that the method provided results which was well in line with our own experience from using the conversational platforms. We have however identified a few potential problems that needs to be considered. The first one being that is uses a relatively large number of main-criteria. The issue with this it that for each additional criteria the effect of the priority values diminishes. This is in theory not a problem with the method. In practice however, it makes it especially important to assign a priority value that truly reflects the importance of a criterion for the specific use-case and that it is proportional to the priority values assigned to each other criterion. With this in mind, we still believe that the added value of considering a larger number of criteria out-weights the extra effort required in using the method.

Another drawback with the method is that it is time consuming, both to understand the method and all its steps, but also performing the prioritization and calculations in each step. However, we believe that any company that are investing or planning to invest in chatbots and are looking for a CP could greatly benefit from spend extra time and effort into the selection process. Some extra time spent on investigation could potentially save a lot of time and resources in the long run since most of the aspects of developing a chatbot have been considered and development of the chatbot will from start be done using the most suited platform. We found that using Excel with our proposed CP selection method worked well and sped up the use of the method. The reason being due to all of the results being in a table format and most of the involved calculations where matrix calculations. We would hence also recommend Excel or any equivalent program for any other user of this method.

6.2 Future work

An obvious limitation with this study is the lack of practical application of the CP decision method on real world use-cases in a live environment. A suggested future work would thus be to apply this method on several company specific use-cases and evaluate how well the method performs, both in terms of the usability of the method and the scores and recommendation the method provides. Additionally, more conversational platforms could be used and evaluated, either with the use-cases specified by this study or on other use-cases to further validate the method. The CP decision method we propose involves a number of steps, data collection and calculations. One proposed future work would thus be to develop a tool which assist with performing these steps and automate the calculations. We believe this would both simply the usage of the method and also help it gain more traction.

Chapter 7

Conclusions

The goal with the master thesis was to investigate and introduce a set criteria for companies to consider before developing a chatbot. Additionally, we wanted to propose a method for selecting a conversation platform based on the needs of a specific company. Through a literature review, hands on experience and by consulting experts we arrived at a list of criteria to be analyzed. After further analysis we arrived at a list of 20 criteria, categorized into 8 main-criteria and 20 sub-criteria. However, three criteria (*Price*, *Channels* and *Security*) we believe are out of scope for this study due to their high impact on CP decisions which is further discussed in Section 6.1.1.

To arrive at a selection method for conversational platforms, we looked at previous research in the field of multi-criteria decision methods (MCDM). The goal was to find a method that was feasible to use for comparison on a large number of criteria. Previous research showed that *AHP* could be used as an MCDM for selecting chatbot platforms [9]. However, it was not suitable to use for comparing a large number of criteria. Additional research showed *AHP-express* as a viable selection method for a larger number of criteria [46]. We validated the CP selection method by applying it on two use-cases (Digital commerce and Marketing) which we found to be the two most commonly used areas of application for chatbots today and in the near future, see Section *2.3.3 Chatbots in the future*. The conversational platforms compared when testing the CP selection method were Dialogflow, Chatlayer and Chatfuel. Chatlayer was found to be the most suitable CP to use for both use-cases which matched our expectations from using each platform to implement these use-cases ourselves.

In conclusion, we hope that the feature and characteristic breakdown as well as the CP selection method provided in this thesis will help companies identify their chatbot needs and make an informed decision on selecting the most appropriate CP in which to develop it.

References

[1] A. M. Turing, "Computing machinery and intelligence", en, *Mind*, vol. LIX, no. 236, pp. 433–460, Oct. 1950. DOI: 10.1093/mind/LIX.236.433. [Online]. Available: https://academic.oup.com/mind/article/LIX/236/433/986238.

[2] J. Weizenbaum, "Eliza - a computer program for the study of natural language communication between man and machine", *Commun. ACM*, 1966. DOI: 10.1145/365153.365168.

[3] Tomberry, *Clippy*, 2010. [Online]. Available: https://knowyourmeme.com/memes/clippy.

[4] R. Dale, "The return of the chatbots", en, *Natural Language Engineering*, vol. 22, no. 5, pp. 811–817, Sep. 2016. DOI: 10.1017/S1351324916000243. [Online]. Available: https://www.cambridge.org/core/journals/natural-language-engineering/article/return-of-the-chatbots/0ACB73CB66134BFCA8C1D55D20BE6392/core-reader.

[5] W. Meisel, "Building digital assistants and bots: A vendor guide and market analysis", en, Tech. Rep., p. 400. [Online]. Available: https://www.aitrends.com/research-bots/.

[6] "Chatbot market growth predicted at over 30 percent till 2024: Global market insights, inc", *GLOBAL-MARKET-INSIGHT*, Feb. 2020. [Online]. Available: http://ludwig.lub.lu.se/login?url=https://search.ebscohost.com/login.aspx?direct=true&db=bwh&AN=202002250500PR.NEWS.USPR.NY28002&site=eds-live&scope=site.

[7] Anonymous, *Top 175 chatbot platforms of 2020: In-depth guide
*, 2020. [Online]. Available: https://aimultiple.com/chatbot-platform.

[8] B. Heller, M. Procter, D. Mah, L. Jewell, and B. Cheung, "Freudbot: An investigation of chatbot technology in distance education", Jan. 2005.

[9] P. Kostelník, I. Pisařovic, M. Muroň, F. Dařena, and D. Procházka, "Chatbots for enterprises: Outlook", en, *Acta Universitatis Agriculturae et Silviculturae Mendelianae Brunensis*, vol. 67, no. 6, pp. 1541–1550, 2019. [Online]. Available: https://ideas.repec.org/a/mup/actaun/actaun_2019067061541.html.

[10] *How do chatbots work? an overview of the architecture of a chatbot*, en-US, May 2019. [Online]. Available: https://bigdata-madesimple.com/how-do-chatbots-work-an-overview-of-the-architecture-of-a-chatbot/.

[11] "An evaluation of the chat and knowledge delivery components of a low-level dialog system:the az-alice experiment", en, *Decision Support Systems*, vol. 42, no. 4, pp. 2236–2246, 2007. DOI: 10.1016/j.dss.2006.07.001. [Online]. Available: https://www-sciencedirect-com.ludwig.lub.lu.se/science/article/pii/S016792360600114X.

[12] *Training phrases | dialogflow documentation*, en. [Online]. Available: https://cloud.google.com/dialogflow/docs/intents-training-phrases.

[13] J. Singh, M. H. Joesph, and K. B. A. Jabbar, "Rule-based chabot for student enquiries", English, *Journal of Physics: Conference Series*, vol. 1228, 2019. DOI: 10.1088/1742-6596/1228/1/012060. [Online]. Available: http://dx.doi.org/10.1088/1742-6596/1228/1/012060.

[14] A. Flycht-eriksson and A. Jönsson, "Dialogue and domain knowledge management in dialogue systems", 2000. DOI: 10.3115/1117736.1117750.

[15] M. A. Boden, "Mind as machine: A history of cognitive science two-volume set", 2006, p. 370. DOI: 10.5860/choice.44-6202.

[16] K. Kuligowska, *Chatbot parry, kenneth mark colby | virtual assistant parry | virtual agent parry | chat bot parry | conversational agent parry | (9706)*, en. [Online]. Available: https://www.chatbots.org/chatbot/parry/.

[17] *About the jabberwacky ai*. [Online]. Available: http://www.jabberwacky.com/j2about.

[18] *A closer look at chatbot alice*, May 2017. [Online]. Available: https://blog.ubisend.com/discover-chatbots/chatbot-alice.

[19] *What is siri and how does siri work?*, en-gb, Sep. 2019. [Online]. Available: https://www.pocket-lint.com/apps/news/apple/112346-what-is-siri-apple-s-personal-voice-assistant-explained.

[20] M. Revang, B. Elliot, B. Manusama, and A. Mullen, "Market guide for virtual customer assistants", en, Gartner, Tech. Rep., Jul. 2019. [Online]. Available: https://www.gartner.com/en/documents/3947357/market-guide-for-virtual-customer-assistants.

[21] *Chatbot market survey- 2017 : Mindbowser info solutions*, en-US, 2017. [Online]. Available: https://mindbowser.com/chatbot-market-survey-2017/.

[22] M. Beccue and A. Kaul, "Executive summary: Virtual digital assistants for enterprise applications", Omdia, Tech. Rep., 2019, pp. 1–2. [Online]. Available: https://tractica.omdia.com/research/virtual-digital-assistants/.

[23] *Top 60 chatbot companies of 2020: In-depth guide*, 2020. [Online]. Available: https://research.aimultiple.com/chatbot-companies/.

[24] *Chatbot platforms*. [Online]. Available: https://www.trustradius.com/chatbot.

[25] *Detect intent with sentiment analysis | dialogflow documentation*, en. [Online]. Available: https://cloud.google.com/dialogflow/docs/how/sentiment.

[26] *Bot platform*, en-GB. [Online]. Available: https://chatlayer.ai/bot-platform/.

[27] *Chatbot development with chatfuel platform - rootinfosol*, en, Mar. 2018. [Online]. Available: https://www.rootinfosol.com/chatbot-development-chatfuel-platform.

[28] Edgar, *Elements and blocks*, 2020. [Online]. Available: https://docs.chatfuel.com/en/articles/1601883-elements-blocks.

[29] *Google scholar*, English, 2004. [Online]. Available: https://scholar.google.com/.

[30] *Lubsearch: Lund university libraries*. [Online]. Available: https://eds.b.ebscohost.com/eds/search/basic?vid=0&sid=d43c8e95-24bd-4366-bd75-1786fec4d672%40pdc-v-sessmgr01.

[31] S. Jalali and C. Wohlin, "Systematic literature studies: Database searches vs. backward snowballing", *Proceedings of the 2012 ACM-IEEE International Symposium on Empirical Software Engineering and Measurement, Empirical Software Engineering and Measurement (ESEM), 2012 ACM-IEEE International Symposium on*, pp. 29–38, Sep. 2012. DOI: 10.1145/2372251.2372257.

[32] D. Braun and F. Matthes, *Towards a Framework for Classifying Chatbots*. 2019. DOI: 10.5220/0007772704960501.

[33] K. Mabunda and A. Ade-Ibijola, "Pathbot: An intelligent chatbot for guiding visitors and locating venues", *2019 6th International Conference on Soft Computing and Machine Intelligence (ISCMI)*, pp. 160–168, Nov. 2019. DOI: 10.1109/ISCMI47871.2019.9004411.

[34] M. Canonico and L. D. Russis, "A comparison and critique of natural language understanding tools", *Cloud Computing*, vol. 2018, p. 120, 2018.

[35] *7 characteristics of a great chatbot*. [Online]. Available: https://www.bluewolf.com/bluewolf-now/7-characteristics-great-chatbot.

[36] *How to select the best chatbot platforms for your business in 2020 - reviews, features, pricing, comparison*, English, Oct. 2019. [Online]. Available: https://www.predictiveanalyticstoday.com/what-is-chatbot-platform/.

[37] *Conversational ai platform: 10 features to look for - dzone ai*, en, Feb. 2018. [Online]. Available: https://dzone.com/articles/conversational-ai-platform-10-features-to-look-for.

[38] S. Gaurav, *12 point guide to pick the best chatbot platform*, Sep. 2017. [Online]. Available: https://www.linkedin.com/pulse/12-point-guide-pick-best-chatbot-platform-gaurav-sarien.

[39] M. Techlabs, *7 key factors to consider before choosing a chatbot platform*, en, Dec. 2018. [Online]. Available: https://chatbotslife.com/7-key-factors-to-consider-before-choosing-a-chatbot-platform-dfc4c8f3a3fa.

[40] U. of Georgia College of Business Administration, "Journal of business research", en, vol. 104, pp. 333–339, Nov. 2019. [Online]. Available: http://www.sciencedirect.com/science/journal/01482963.

[41] J. R. E. H. Charlotte, "Qualitative questionnaires as a method for information studies research", en, vol. 22, no. 1, May 2017. [Online]. Available: http://informationr.net/ir/22-1/colis/colis1639.html.

[42] G. Odu, "Weighting methods for multi-criteria decision making technique", *Journal of Applied Sciences and Environmental Management*, vol. 23, p. 1449, Sep. 2019. DOI: 10.4314/jasem.v23i8.7.

[43] I. Palcic, "Analytical hierarchy process as a tool for selecting and evaluating projects", *International Journal of Simulation Modelling*, vol. 8, pp. 16–26, 2009. DOI: 10.2507/IJSIMM08(1)2.112.

[44] M. Velasquez and P. Hester, "An analysis of multi-criteria decision making methods", *International Journal of Operations Research*, vol. 10, pp. 56–66, 2013.

[45] M. Hokey, "Selection of software: The analytic hierarchy process", *International Journal of Physical Distribution and Logistics Management*, vol. 22, no. 1, pp. 42–52, 1992, 29. DOI: 10.1108/09600039210010388. [Online]. Available: https://doi.org/10.1108/09600039210010388.

[46] J. E. Leal, "Ahp-express: A simplified version of the analytical hierarchy process method", English, *MethodsX*, vol. 7, p. 100 748, 2020. DOI: 10.1016/j.mex.2019.11.021. [Online]. Available: http://dx.doi.org/10.1016/j.mex.2019.11.021.

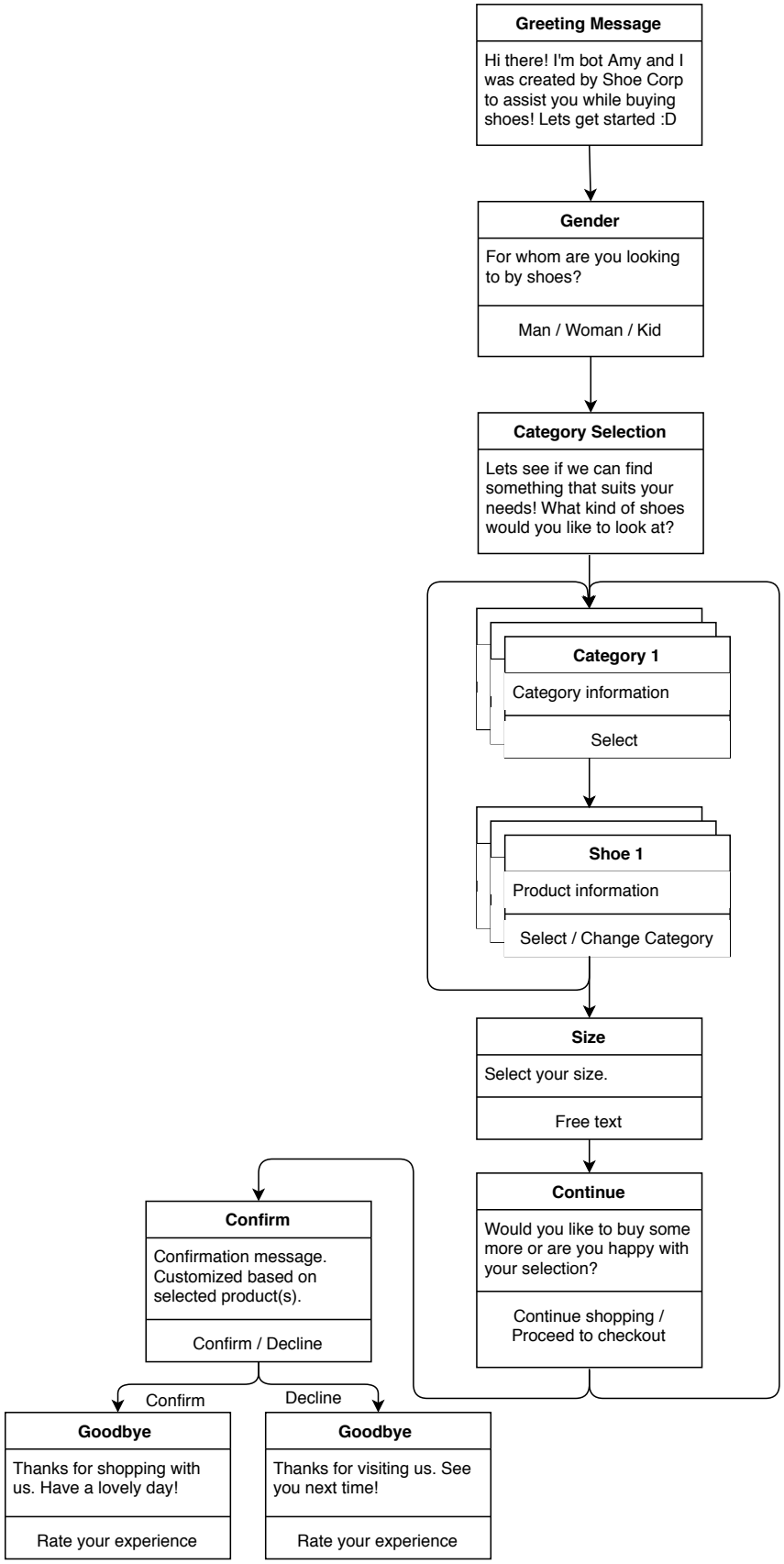
[47] "Rule-based chabot for student enquiries", *Journal of Physics: Conference Series*, vol. 1228, no. 1, p. 1, Jun. 2019.

Appendices

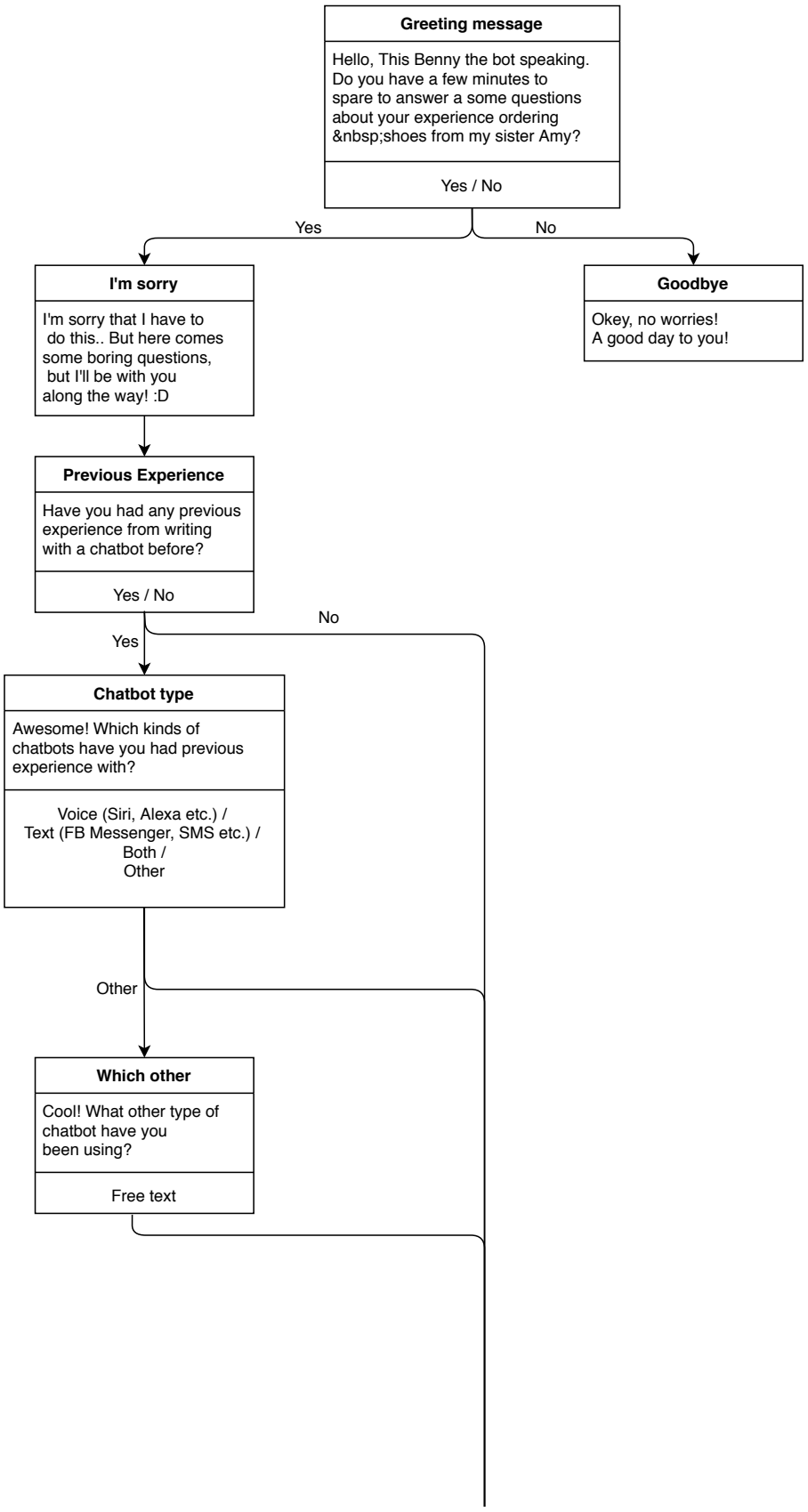
Appendix A

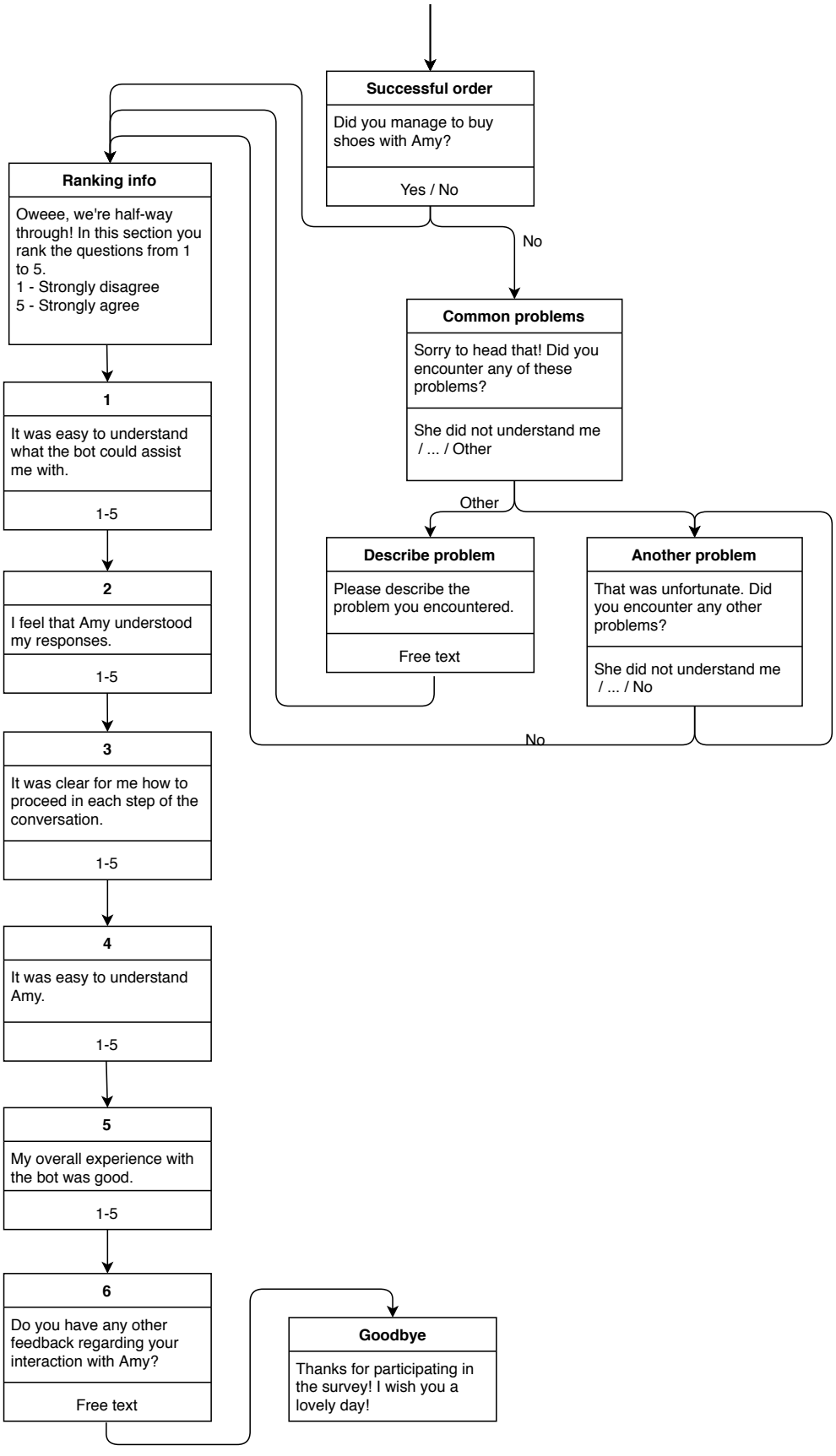
Benchmark Case

A.1 UML Digital Commerce



A.2 UML Marketing




Successful order
Did you manage to buy shoes with Amy?
Yes / No
No
Common problems
Sorry to head that! Did you encounter any of these problems?
She did not understand me / ... / Other
Other
Describe problem
Please describe the problem you encountered.
Free text
Another problem
That was unfortunate. Did you encounter any other problems?
She did not understand me / ... / No
No
Ranking info
Oweee, we're half-way through! In this section you rank the questions from 1 to 5.
1 - Strongly disagree
5 - Strongly agree
1
It was easy to understand what the bot could assist me with.
1-5
2
I feel that Amy understood my responses.
1-5
3
It was clear for me how to proceed in each step of the conversation.
1-5
4
It was easy to understand Amy.
1-5
5
My overall experience with the bot was good.
1-5
6
Do you have any other feedback regarding your interaction with Amy?
Free text
Goodbye
Thanks for participating in the survey! I wish you a lovely day!

Digital Commerce bot	Customer buys shoes
Actor	Customer
Use Case Overview	The customer is interested in seeing what shoes the store offers with the intent to buy a pair.
Subject Area	Digital Commerce
Actors(s)	Chatbot and Customer
Trigger	Customer initiates conversation with chatbot
Precondition 1	Customer is a registered user on Facebook Messenger

Table A.1: High level description of the Digital Commerce bot

Marketing bot	Customer responds to follow up survey
Actor	Customer
Use Case Overview	The chatbot wishes to ask a user follow up questions to know how another chatbot performs.
Subject Area	Marketing
Actors(s)	Customer and Chatbot
Trigger	Customer initiates conversation by writing to the chatbot
Precondition 1	Customer has been in contact with another chatbot

Table A.2: High level description of the Marketing bot

Appendix B

Consulting Experts: Questionnaire

Here you can find the questionnaire that was sent out to in the 3.1.3 consulting experts method.

The respondent had to rank each feature from 1 to 3, where 1 was the lowest and 3 was the highest.

The following text was presented for all features: *Please rank the feature based on its importance for building chatbots. The ranking scale goes from 1 to 3, where 1 is low priority, 2 is medium priority and 3 is high priority.*.

The features and/or characteristics that were ranked can be found in table B.1.

Characteristic	Description
AB-testing	I value the possibility to AB-test different bot conversations
Access account information	The possibility to access metadata from the account used to enrich and personalize messages. E.g. name and gender from Facebook Messenger.
Auto-configuration	I value auto-configuration of different features and/or integrations. I.e. Easy setup of human handover or easy setup of channel to use
Auto-scaling	Automatically scales the bot to handle everything from a few active sessions to thousands of active session.
Chat simulator	I value the possibility to continuously test the chatbot with a chat simulator
Context	Intents can be defined in a context. Which means that a specific intent will only be evaluated when an associated context is active in the conversation. Enables the use of the same intent in more than one place in a conversation.
Conversation logs	I value the possibility to access old conversations

Continued on next page

Table B.1 – *Continued from previous page*

Characteristic	Description
Conversation Templates	I value the the availability of using existing conversational templates
Custom Logic	I value the possibility to create more advanced logic than the UI itself allows for
Custom Scripting	I value the possibility to have the ability to create more advanced logic in my flow.
Dynamic content	I value the possibility to add dynamic content to the bot messages
Entities extraction	I consider the possibility of extracting entities from user messages as important
External services	I value the possibility to be able to integrate to a variety of different external services
Human handover	I value the feature of handing over the conversation from the bot tho a human agent
Import/export of flows	I value the possibility to import/export of a conversation
Intent routing	I value the possibility to route user responses based on intents
Languages	I benefit from that the chatbot can understand messages in different languages
Message per channel	I want to have full control over how messages gets rendered on different channels
Message Preview	Its important for me to be able to inspect how a message look on a specific channel
OMNI-channel	It's beneficial for me to be able to define a message once and have it automatically converted into several messaging formats
Pre-defined Entities	Having a set of pre-defined entities improves my bot building experience
Pre-defined Intents	Having a set of pre-defined intents improves my bot building experience
Re-engage users	I value a feature that allows the bot to automatically re-engage with a user after a specified amount of time
Rich content	I want the ability to send richer messages such as rich cards, carousels etc
See active chats	I value the possibility to access and monitor active chats
Sentiment Analysis	I value the ability to make decisions based on a users mood when writing to the bot.
SmallTalk	I value the possibility to easily make my bot more human like
Translations	I value the possibility to easily translate bot messages into several languages.
Variables	It's important for me to be able to set and use variables to create dynamic content or in custom logic
Versioning	I consider it to be important to be able to save versions of the conversation that I'm building
Visual Builder	I value the ability to see a visual representation of the bot conversation that I'm building

Continued on next page

Table B.1 – *Continued from previous page*

Characteristic	Description
Visualisation of data	I place a big value in how analytical data is visualised
Webhooks	I value the possibility to be able to use webhooks when building chatbots.
Low technical depht	The technical depth of the conversational platform is important for building chatbots.

Table B.1: The characteristics and descriptions for the experts

For each feature the respondent could motivate their priority and in the end the respondent was presented with an open question which was: *Do you feel there's any important features that should be included? Please write the feature, priority and motivation..*

In the tables B.2, B.3, B.4 and B.5 the motivations behind the priorities for a characteristic for each respondent are presented, note however that characteristics that that did not have their priority motivated has been removed.

Characteristic	Respondent 1
AB-testing	Nice to have, but not critical
Context	This can be achieved with variables and jumps if supported. Some kind of mechanism to allow different target nodes based on intent + state is required.
External services	Without this you can only create very basic NLP driven bots that have no user specific knowledge.
Low technical depth	More important that the tool support everything that you would like to do and is flexible. You can always learn something complicated, but you can not use a non-existing feature.
Pre-defined entities	There are a few predefined entities that are very useful, but for the most part no.
Pre-defined intents	Predefined intent packages rarely matches what I would want. It's better to have a way to easily bulk import intents so I can define my own
Sentiment analysis	Nice in theory but rarely used as far as I have seen
SmallTalk	Predefined SmallTalk is usually very basic and generic. I want control over all intents
Webhooks	Depends on the API. In many cases the API is post conv id + next input and the result is the output and the next state. In that case webhooks aren't really essential.

Table B.2: Motivations behind priorities for respondent 1

Characteristic	Respondent 2
AB-testing	Never tried that in practice but sounds useful
Access account information	Nice to have, but for the sake of OMNI-channel this info can be collected by the bot itself too
Auto-scaling	Since bot implementations are not very portable and you'll never know where you end up in terms of # of users, I'd say this is very important
Chat simulator	It makes the process of building a bot much smoother
Conversation templates	Useful especially for multi-lingual bots
Custom logic	Feels like the same question as custom scripting?
Custom scripting	The ability to do custom scripting is important but I would also like to avoid it as long as possible
Context	Powerful, but can also result in logic that is hard to follow I think
Dynamic content	Feels like the same question as integrations?
External services	Bots usually need to be dynamic to provide any real value, just as websites
Translations	Translating individual messages doesn't add a lot of value since the whole conversation flow and number of messages are different depending on the language in all but the simplest cases.

Table B.3: Motivations behind priorities for respondent 2

Characteristic	Respondent 3
Context	Intents can be very generic, without context they might be misleading
External services	Critical for making the bot interaction relevant / personalized for the user; critical for the bot to fit in the existing landscape of applications that a business has

Table B.4: Motivations behind priorities for respondent 3

Characteristic	Respondent 4
AB-testing	nice to have
Chat simulator	very important
Custom scripting	very important, because this will always happen, soon or later in the design

Dynamic content	As soon as you want to do a real-time solution, this is needed
Entities extraction	for good integrations, this is necessary
External services	Chat bot is never a standalone service, it needs to communicate with other internal services
Human handover	mandatory
Import/export of flows	not really important
Intent routing	NLP need this to make the logic simpler to visualize
Message per channel	I want to be able to visual how the message look and perhaps modify the message to better fit all channels
Rich content	nice to have
Translations	Mandatory feature to have
Webhooks	many services can filter their notifications with Webhooks. I think this is good to have for specific use-cases but probably not mandatory

Table B.5: Motivations behind priorities for respondent 4

Appendix C

Priority of Alternatives for each Sub-Criteria

C.1 Variable support

	Chatfuel	Dialogflow	Chatlayer
Dynamic content			
Chatlayer	3	5	1
1/a	0.3333	0.2	1
Priority	0.2174	0.1304	0.6522
Custom logic			
Chatlayer	5	3	1
1/a	0.2	0.3333	1
Priority	0.1304	0.2174	0.6522
Access account information			
Chatfuel	1	3	3
1/a	1	0.3333	0.3333
Priority	1	0	0

Table C.1: Priorities for the *Variable* criteria

C.2 Messaging

	Chatfuel	Dialogflow	Chatlayer
OMNI-channel			
Dialogflow	9	1	7
1/a	0.1111	1	0.1429

Priority	0.0886	0.7975	0.1139
Message per channel			
Dialogflow	5	1	3
1/a	0.2	1	0.3333
Priority	0.1304	0.6522	0.2174
Rich content			
Dialogflow	1	1	1
1/a	1	1	1
Priority	0.3333	0.3333	0.3333
Translations			
Chatlayer	5	3	1
1/a	0.2000	0.3333	1
Priority	0.1304	0.2174	0.6522

Table C.2: Priorities for the *Messaging* criteria

C.3 Usability

	Chatfuel	**Dialogflow**	**Chatlayer**
Visual builder			
Chatlayer	3	9	1
1/a	0.3333	0.1111	1
Priority	0.2308	0.0769	0.6923
Chat simulator			
Chatlayer	3	5	1
1/a	0.3333	0.2	1
Priority	0.2174	0.1304	0.6522
Versioning			
Chatfuel	1	3	5
1/a	1	0.3333	0.2000
Priority	0.6522	0.2174	0.1304
Auto-configuration			
Dialogflow	1	3	3
1/a	1	0.3333	0.3333
Priority	0.6000	0.2000	0.2000

Table C.3: Priorities for the *Usability* criteria

C.4 Analytics

	Chatfuel	**Dialogflow**	**Chatlayer**
Conversation history			
Chatlayer	3	3	1

1/a	0.3333	0.3333	1
Priority	0.2000	0.2000	0.6000
Visualization of data			
Chatlayer	5	3	1
1/a	0.2	0.3333	1
Priority	0.1304	0.2174	0.6522
View active chats			
Chatfuel	1	3	3
1/a	1	0.3333	0.3333
Priority	0.6	0.2	0.2

Table C.4: Priorities for the *Analytics* criteria

C.5 Integrations

	Chatfuel	Dialogflow	Chatlayer
Webhooks			
Dialogflow	3	1	1
1/a	0.3333	1	1
Priority	0.1429	0.4286	0.4286
External services			
Dialogflow	5	1	3
1/a	0.2	1	0.3333
Priority	0.1304	0.6522	0.2174
Human handover			
Chatfuel	1	5	3
1/a	1	0.2	0.3333
Priority	0.6522	0.1304	0.2174

Table C.5: Priorities for the *Integrations* criteria

C.6 Standalone

	Chatfuel	Dialogflow	Chatlayer
Custom scripting			
Dialogflow	9	1	3
1/a	0.1111	1	0.3333
Priority	0.0769	0.6923	0.2308
Auto-scaling			
Dialogflow	5	1	3
1/a	0.2000	1	0.3333
Priority	0.1304	0.6522	0.2174
Low technical depth			

Chatfuel	1	9	5
1/a	1	0.1111	0.2000
Priority	0.7627	0.0847	0.1525

Table C.6: Priorities for the *Standalone* criteria

EXAMENSARBETE Building chatbots; a tool study
STUDENTER Amar Vrbanjac, Carl Hartzell
HANDLEDARE Emelie Engström (LTH)
EXAMINATOR Pierre Nugues (LTH)

Bygg chatbotar; en verktygsstudie

POPULÄRVETENSKAPLIG SAMMANFATTNING **Amar Vrbanjac, Carl Hartzell**

Vad ska man tänka på när man vill utveckla en chatbot och hur kan man på ett effektivt sätt välja vilket verktyg som man ska bygga chatboten med? Med denna uppsats försöker vi besvara dessa frågor hjälp av en taxonomi av chatbotskriterier samt en urvalsmetod, testad på två användarfall.

Chatbotar har ökat drastiskt i popularitet de senaste åren. Idag är det nästan svårt att hitta en kundinriktad hemsida utan att det dyker upp ett välkomnande meddelande från en glad chatbot. Att bygga dessa chatbotar är inte alltid trivialt. Det finns en uppsjö av termer och funktioner som relaterar till att förstå grunderna i chatbotsutveckling. Dessa termer och funktioner kallar vi för chatbotskriterier. Vår uppsats försöker besvara följande frågor: Vilka av dessa kriterier är de viktigaste för företag att förstå och ta ställning till? Hur ska ett företag gå tillväga för att effektivt välja den konversationsplattform som är bäst lämpad för deras behov? En konversationsplattform är ett grafiskt onlineverktyg i vilket man utvecklar chatbotar.

Utvecklingen av en chatbot samt valet av verktyg i vilket man utvecklar den kan vara en stor investering för ett företag. Är man då inte expert på området, kan det vara svårt att veta vart man ska börja samt vilket verktyg man ska välja. Genom vår studie har vi kommit fram till de kriterier som är viktigast för företag att ta ställning till och vidare analysera. Detta hoppas vi kommer hjälpa företag att få en bättre förståelse av vad som krävs för att bygga chatbotar. Det kommer också hjälpa dem att från början välja rätt verktyg för deras behov vilket kan spara dem både tid och pengar.

De kriterier vi kommit fram till går att använda i flera syften samt anpassa efter behov. Vi har kommit fram till 20 kriterier som företag bör tänka på och utvärdera innan de väljer konversationsplattform. Vi har också kommit fram till en urvalsmetod som syftar till att underlätta jämförelsen och valet av olika konversationsplattformar. De kriterier vi kommit fram till kan användas direkt med vår föreslagna urvalsmetod. De kan även användas som bas i andra urvalsmetoder om så önskas. Kriterierna ger också en stabil grund för diskussioner och analyser hos företag som vill bygga en chatbot i en existerande plattform eller helt enkelt utveckla en egen plattform eller chatbot från grunden.

För att komma fram till de 20 kriterierna för chatbotsutveckling genomförde vi en litteraturstudie, testade bygga chatbotar i olika populära konversationsplattformar samt konsulterade experter på företaget där vi genomförde uppsatsen. Urvalsmetoden validerades genom att applicera den på två användarfall, en e-shop chatbot och en marknadsförings chatbot.