Streamline searches in a database
Abstract

The objective of this thesis is to explore technologies and solutions and see if it is possible to make a logistical flow more efficient. The logistical flow consists of a database containing materiel for purchase or reparation. As of now, searches may either result in too many results, of which several are irrelevant, or no results at all. The search needs to be very specific to retrieve the exact item, which requires extensive knowledge about the database and its contents. Areas that will be explored include Natural Language Processing and Machine Learning techniques. To solve this, a literature study will be conducted to gain insights into existing work and possible solutions. Exploratory Data Analysis will be used to understand the patterns and limitations of the data.
Preface

First and foremost, we want to mention that this thesis has been inspiring and a great learning experience. We would like to thank Combitech for giving us this opportunity and the chance to work in their facilities. Our supervisor at Combitech, Philip Popovski, has been instrumental in helping us every step of the way, and we are truly grateful for his guidance. We also want to express our appreciation to Henrik Karlsson and Erik Margaronis for aiding in brainstorming different approaches. Lastly, we would like to thank Johan Hagelbäck, the supervisor at LNU, for his guidance, both academically and technically.
8 Conclusion
  8.1 Future work

References

A Appendix 1
1 Introduction

In modern society, organizations rely significantly on databases to store and manage enormous amounts of information. However, searching for specific data within these databases can be a time-consuming and labor-intensive process. This is especially true for companies that have multiple databases or a large volume of data. Searches in databases occur frequently, and it can be extremely frustrating when searching for an item and the correct result is not retrieved. There are several factors for this to occur, the search word may be misspelled, the formulation may be unclear or the item that is searched for is a common item but has a special characteristic that requires very specific input that is only known if the person that is making the search has a deep understanding of the database or its contents. Therefore, it is essential to find a solution that streamlines searches in databases to improve the efficiency of organizations.

1.1 Background

This thesis explores the possibilities of streamlining a logistical flow that uses text as input data. The logistical flow consists of a database containing materiel for purchase or reparation. In order to improve the logistical flow the user should with the help of artificial intelligence be able to input more human-like search queries and get the relevant and correct results back.

This has great relevance as searches in databases are something that happens countless times per day. In this specific case, it is one person who completes the search manually and then has to go through the results individually one by one.

The thesis includes fields that are important for a system that can accurately understand the needs of its end-user. Machine learning and natural language processing (NLP) are two important areas to understanding the structure, usage, and meaning of language and also enabling computers to understand and interpret human language. Both machine learning and Natural Language Processing is an area of artificial intelligence. The thesis intends to streamline database searches that require text input by combining these areas.

The company this thesis is in collaboration with is Combitech, which is a consulting company fully owned by SAAB. "With their cutting-edge expertise and experience, Combitech is one of Sweden’s biggest service provider within the defence-sphere. They assist the Swedish Armed Forces, Swedish Defence Materiel Administration (FMV) and defence industry develop systems for a safer society." [1]. The work performed in this thesis will act as a proof of concept of a bigger project the company is exploring.

1.2 Related work

A Survey of Automatic Query Expansion in Information Retrieval. The survey contains good information regarding how the Query extension work and how to evaluate it. The general information, especially the evaluation part, is a good topic since this project will also require a method to evaluate the output [2].

An article published in the National Library of Medicine discusses the issues of retrieving feedback from free-text patient experiences, and how both NLP and ML techniques can
be used to gain insights into the free text [3].

*Generating Structured Queries From Natural Language Without Reinforcement Learning.* This is a paper that uses SQLNet to generate SQL queries from a natural language input for an example from the paper is “Who is the player that wears number 42?” and makes it into an SQL query [4].

*A deep dive into deep learning approaches for text to SQL systems.* This paper discusses different deep learning techniques for the text to SQL problem. The paper encompasses the history of models and describes the development in the field. It also discusses different approaches such as grammar-based, sketch-based, and the one used for this thesis sequence to sequence. The paper also discusses potential problems with the techniques used [5].

Current problems within the seq to SQL and future directions on the study are presented in this paper. LSMTs are described and discussed in great detail. Encoder and decoder the two main parts of an LSTM approach are also described in an understandable manner. The article also discusses the importance of having qualitative training data [6].

Seq to SQL where re-enforcement learning is used is presented in this paper. A potential approach that is researched. The paper also gives good clarity into how a model works on a high level making it comprehensible without much prior knowledge. Difficulties in the different parts of the SQL query, meaning the SELECT part and the WHERE part [7].

1.3 Problem formulation

The problem investigated and to be solved is to improve the manual searching of a specific database. Currently, the specified search word and criteria need to be very specific if the resulting search shall be precise. The result may also end up with either not getting any search results at all or getting too many search results which several are irrelevant if the search is either too specific or too vague. The solution should allow diversifying the search to also include other spelling and possible synonyms. The input shall also be able to be more human-like, and less specific for the exact search criteria.

As described by the prior work this type of problem has been solved in several other ways in different contexts. The main gap this project aims to resolve is to combine, implement and evaluate the techniques mentioned in Section 1.2 and implement these to provide a proof of concept, of how searching the database can be made easier.

As for concrete output from this project it will be tested on the accuracy of the model’s ability to correctly interpret an input question and translate it into a correct SQL query. Tests will include a technical test using data unseen to the model and acceptance tests performed by users combined with interviews to evaluate how well the system performs.

1.4 Motivation

Database searches are a common occurrence in almost all industries and organizations. Human search inputs can be under or over-specified and may lead to too many or too few results. The motivation is to have the end user be able to input more human-readable and understandable text and for the output to be relevant, meaning not too many nor too few
results. The improvement over conventional database searches is that instead of having extensive knowledge of database language, the program will output what it predicts the user is looking for based on the natural language input.

From a broader perspective, databases are used in many industries. Getting an accurate search result from a database often requires extensive knowledge of the database. Creating a solution that enables a more human-friendly search would make database searches more efficient. The knowledge gained can then be used in other fields.

1.5 Milestones

Table 1.1: Milestones

<table>
<thead>
<tr>
<th>M1</th>
<th>Perform literature study</th>
</tr>
</thead>
<tbody>
<tr>
<td>M2</td>
<td>Evaluate raw data using Exploratory Data Analysis</td>
</tr>
<tr>
<td>M3</td>
<td>Find most suitable techniques based on insights gained from EDA and literature study</td>
</tr>
<tr>
<td>M4</td>
<td>Create database containing training data</td>
</tr>
<tr>
<td>M5</td>
<td>Implement text to SQL</td>
</tr>
<tr>
<td>M6</td>
<td>Final implementation</td>
</tr>
<tr>
<td>M7</td>
<td>Evaluate result</td>
</tr>
</tbody>
</table>

1.6 Scope/Limitation

This investigation will be limited to the mock data and not the real data since it is not publicly available and contains sensitive information. The implementation and testing will be proof of concept if a solution is possible. Limitations for the project will therefore be that it is only suited for this specific data set. It could be retrained on other data sets but is still limited to SQL. Since the solution will be a proof of concept, the entire content of the live database will not be used as mock data, instead a category will be chosen.

1.7 Target group

Databases are widely used in many areas to store data. Employees at organizations that maintain big databases could find themselves struggling to efficiently operate a database as the database query language known as SQL is not trivial and operating a database could require habit. A user operating a database will most likely not encounter SQL language directly, but without having knowledge of how SQL works, performing a search to retrieve the correct item(s) may be difficult.

This issue could be solved by enabling a human-friendly approach, suitable for employees of differing backgrounds and competencies, to access the data of a database without utilizing strictly structured GUI’s of varying degrees of complexity, which are commonly found today. A solution of this kind would be very appealing to many organizations that struggle to get accurate results from database searches.
At Combitech, there is a database utilized in several different logistical- and supply chain-related processes such as purchasing, inventory management, maintenance planning, etc. The problem they have is the ability to retrieve accurate results. As the database is old, the data is sub-optimal and without a uniform procedure for logging items. This makes retrieving an accurate item difficult. A solution that enables a more descriptive search text instead of precise input data would be very beneficial and time-saving.

1.8 Outline

This sub-chapter will give a brief explanation of the remainder of the thesis. Chapter 2 will give a clear picture of the problem and how it is investigated. The reader shall have a good understanding of the techniques used in the thesis after reading this chapter. Chapter 3 will give an explanation of how the problem will be approached and what methods will be used. Chapter 4 is going to present the implementation of different software that was used. Chapter 5 will present the result of the methods used in the thesis and the setup of the solution. Chapter 6 will give an analysis of the result, and what conclusions can be drawn. Chapter 7 will discuss the findings and if the solution solved the problem. Lastly, chapter 8 will present the conclusion of the entire thesis and present suggestions and opportunities for further work and research.
2 Theory

This chapter contains relevant concepts that are necessary for understanding the problem and the solution.

2.1 Databases

A database is an organized collection of data that can be accessed efficiently and retrieve information. The data stored in a database can range from items to phone numbers to credentials and more, depending on the specific purpose for which the database was created. A database is made specifically for a target audience and is adapted to their interest. For example, a business may use a database for storing employee data, or a hospital may use a database for storing patient information [8].

![Table in a database](image)

In order to define, construct and manipulate a database, a database management system (DBMS) is often necessary. A DBMS enables a user the ability to create, modify and query a database. In order to manipulate a database, meaning retrieving data or updating a table in the database, queries are used. Figure 2.1 shows a table in a database. To efficiently write queries, a tool is required. SQL (Structured Query Language) is a standardized programming language created for managing and manipulating databases which is widely used [8].

```
SELECT Credit_hours
FROM COURSE
WHERE Course_name = 'Intro to Engineering'
```

The query above would result in the number 3, as the table "COURSE" contain a column named "Credit_hours" which is 3 when selecting "Intro to Engineering". This is correct according to the table in Figure 2.1.

2.2 Natural language processing (NLP)

The subject "refers to the branch of computer science—and more specifically, the branch of Artificial Intelligence (AI)—concerned with giving computers the ability to understand
Machine learning is also a branch of Artificial Intelligence, but have its scope in learning from data. A machine learning model can learn from patterns, instead of being programmed to perform specifically pre-defined actions. A good example is an email filter, where a machine learning model can learn to detect suspicious emails that are out of the ordinary [10].

A computer does not possess the ability to make distinctions and understand human languages, since a computer only understands 0’s and 1’s. NLP can translate words into machine-understandable vectors and matrices. Machine-learning techniques can then be used to capture the meaning and context of the vectors that represent the sentences and words [11].

NLP can be used for a multitude of tasks. Some of these include speech recognition, Sentiment analysis, and Natural Language Generation. The subtopic used in this thesis is NLIDB (Natural Language Interface to Databases) the purpose of this approach is to translate the human language into SQL queries comprehensible by the database. NLIDB aims to simplify human-to-database communication by lowering the knowledge the end user needs to know about how to query the database [9].

2.3 Neural Network

Neural networks refer to machine learning techniques inspired by the human brain. It is comprised of layers of nodes, mimicking the way that biological neurons signal to each other. Neurons in a Neural Network make use of a mathematical equation in order to decide if the current neuron should be allowed to pass on information to the next neuron. This equation, without diving too deep, consists of three key components: weight, a bias, and an activation function. Weights assign importance to the inputs, and biases provide an added value to the weighted input, regulating the output of the activation function to fit the training data more accurately. The activation function determines the output of the neuron based on the weighted inputs and bias. The result is evaluated and the neuron can send information to the next neuron [12].
A neural network is trained on training data and tries to improve its accuracy over time. The neural network uses the training data to fine-tune the weight and biases of the neurons in order to achieve as good of an accuracy as possible. The structure consists of an input layer, a hidden layer, and an output layer, as illustrated in Figure 2.2. Bias symbolizes the difference between the predicted output and the true output (as defined by the training data). Bias is a tuning parameter that regulates the output of an activation function, in order to fit the training data more accurately. However, this is not enough to capture relationships between input and output. An activation function can aid a network in learning nonlinear functions and can, therefore, learn relationships better. The reason for using nonlinear functions is that they help with predictions [13] [14].

This is the "magic" that occurs in each neuron of the hidden layer(s), as seen in Figure 2.3. When training a Neural Network, the weights and biases are, as stated, adjusted to the most optimal values. This is where backpropagation is used, meaning that the difference between the correct output and the predicted output is calculated using a gradient of the loss function. Backpropagation tries to minimize this difference.

Neural networks can be applied to many use cases where you have sufficient training data. For example stock market predictions and Marketing and promotion analysis [15]. Neural
networks are good at pattern recognition and finding similarities and regularities.

2.3.1 Recurrent Neural Network (RNN)

As stated in the previous chapter, Neural Networks are good at many things, like pattern recognition and classification. However, standard Neural Networks come up short when it comes to processing sequential data, such as natural language.

A Neural Network like a Feed-Forward Neural Network takes in information at $x$, and after computations based on weights, biases, and an activation function, a $y$ output is yielded. The issue when it comes to natural language is that a feedforward neural network lacks feedback, meaning the output of the model is not fed back to itself. This is where Recurrent Neural Network comes into the picture and is illustrated in Figure 2.4 [16].

![Figure 2.4: Difference between a Recurrent Neural Network and a Feed-Forward Neural Network](image)

Figure 2.4: Difference between a Recurrent Neural Network and a Feed-Forward Neural Network

A Neural Network like a Feed-Forward Neural Network takes in information at $x$, and after computations based on weights, biases, and an activation function, a $y$ output is yielded. The issue when it comes to natural language is that a feedforward neural network lacks feedback, meaning the output of the model is not fed back to itself. This is where Recurrent Neural Network comes into the picture and is illustrated in Figure 2.4 [16].

![Figure 2.5: Overview of a Recurrent Neural Network](image)

Figure 2.5: Overview of a Recurrent Neural Network
A Recurrent Neural Network uses feedback connections between cells, instead of neurons that exist in a Neural Network, that allow them to have a memory of past inputs, which is essential for natural language sentences. This is visualized in Figure 2.5. However, even though a Recurrent Neural Network is more suitable for natural language tasks than a standard Neural Network, a Recurrent Neural Network still has some drawbacks. It cannot capture context well and can suffer from long-term dependency problems caused by the exploding/vanishing gradient problem [16].

Such dependency problems can occur when the weights and biases are adjusted through backpropagation, meaning the gradients can become too large or too small. The vanishing gradient problem refers to a gradient becoming so small that it is close to zero, which results in a trained model that will have difficulty capturing the structure and meaning of a sentence, where words that are critical in a sentence are separated by a distance. The exploding gradient problem refers to when a gradient has become too large, which results in a model that will have trouble identifying the context of a word. Some words have different meanings in different contexts [16] [17].

2.3.2 Long Short-Term Memory (LSTM)

As stated in the previous section, an ordinary Recurrent Neural Network is not good at accurately predicting the outcome of complex sentences, since they can suffer from dependency problems. One way to solve this is to use a special architecture that is designed to aid in gradient flow. LSTM is a type of RNN (recurrent neural network) that attempts to solve the exploding/vanishing gradient problem that hinders an ordinary Recurrent Neural Network. It attempts this by using a more advanced memory, that consists of input-, forget-, and an output gate, that allows the network to choose what information is important or can be discarded. Unlike an ordinary neural network, the weights and biases are the same for each cell of the LSTM units. The reasoning behind this is to let the input, forget and output gate decide what information should be used [17].

![LSTM Cell](image)

Figure 2.6: “LSTM Cell”, by Guillaume Chevalier, licensed under CC BY 4.0. [18]
As illustrated in Figure 2.6, an LSTM unit receives three inputs. Viewing top-down, the Memory (C) input represents the long-term memory in the LSTM architecture. The memory is affected by two inputs, the forget gate and the candidate memory. The forget gate decides if the previous memory state should remain the same or be changed, i.e., "forgotten". The candidate memory wants to add to the memory, which enables LSTM to be more selective with new information with the combination of the forget gate [19].

The hidden state (H) represents the short-term memory of LSTM. This gate is affected by an input X, the previous hidden state (H), and the long-term memory (C). The hidden state is important since it highlights important information at the current step [19].

The input x is a sequence. A concrete example would be a natural language sentence, "My dog is cute". The first step would be "My", and would be used to update the memory (C) and hidden state (H). The next step will be "dog", and the memory and hidden state will be updated based on the importance of the word, where the importance is defined depending on the context and the previous history of the sequence [19].

2.4 *seq-to-seq*

`seq-to-seq` stands for sequence to sequence. This concept encompasses the previously explained techniques and is a solution for sequence-related tasks, that solve the problems that have been brought up previously. The model takes a given sequence input and returns another sequence as output. The benefit of a `seq-to-seq` model is that the input sequences does not need to match the number of words of the output sequences, which is essential for this problem, as languages in general do not always equal the same number of words when translated. For instance, during the process of translating between two languages, there exist no predefined regulations or guidelines pertaining to how a sentence should be translated. This is made possible by using an encoder and decoder, which are used in `seq-to-seq`. The reason for the encoder-decoder model is due to the input and output are fundamentally different. An understandable analogy for this is the game Dictionary (One player gets a word, draws a figure of it and the 2nd player is to guess what it is.) [20] [21].

![Figure 2.7: Overview of Pictionary Game / encoder-decoder](image)
As seen in Figure 2.7, the encoder (1st player) interprets the word into a vector (picture). The Decoder (2nd player) then decodes the vector (picture) and concludes the best answer. As explained in a game like this there are no set rules to succeed, similar to language translation where there are no set rules for how to translate. This is why both the encoder and decoder are needed.

The encoder consists of a stack of LSTM cells where each one is responsible for a single word of an input sequence. The final state of the encoder aims to encapsulate the information in the input sequence which results in a context vector, in order for the decoder to make accurate predictions. The decoder also contains a stack of LSTM cells, where each cell is responsible for a single word of the target sequence and gets the state from the previous unit and context vector, and produces its own state along with output. The final output is then predicted using a probability vector to determine the final output [22].

The encoder and decoder are trained in the training phase, with the goal of minimizing the difference between the predicted output and the true output. This way of training a model, by comparing the predicted output to the true output is called teacher forcing. This means that in order to train a seq-to-seq model you need training data that includes an input and an expected output. The quality and quantity of data are important as the more example and nuances that are captured the better the model will understand and in turn, lead to better accuracy.

When the model has been finalized and trained, new previously unseen data is used as the input and is in the same way converted into a numerical representation that the encoder encodes, then predicts the most likely output based on the training. The output is then decoded by the decoder and a predicted text sequence is returned.

2.5 TensorFlow

TensorFlow is an open-source framework for all kinds of predictive analytics like machine learning and deep learning. It is developed by Google. Tensor flow market itself as providing "easy model building", robust ML production anywhere, and powerful experimentation for research. This thesis uses TensorFlow for its built-in algorithms and relative ease of use [23].
3 Method

In this section of the thesis, we will delve into the approach used to investigate the problem and propose a potential solution. To ensure that the research findings are credible, specific methods used will be presented.

3.1 Research Project

A design science will be the research method, which includes six steps; Identification of the problem, defining solution objectives, the design and development of the solution, demonstration of solution effectiveness on the problem, evaluation, and communication, meaning presenting research findings [24]. This approach requires an understanding of the problem before a design of potential solutions can take place. After the problems are understood, prior work is built upon to solve the problems (called artifacts in design science). In order to get a better understanding of the problem, a literature study will be used to gather information about relevant work and technical areas. An Exploratory Data Analysis (EDA) will be used to gather insights about the data. This method is frequently used by data scientists to analyze data sets and find important characteristics [25]. Insights gained from both the literature study and the EDA are essential for designing and constructing a prototype/artifact. The solution will be demonstrated using a technical test and a user test, in order to see how the solution performs. The tests are further explained in Section 5.1. The results will be evaluated to see if the solution solved the problem which can be seen in Section 6. The research findings after analysis and evaluation will be presented in Section 7 and Section 8.

3.2 Method

As stated in the previous section, a design science will be the thesis’s main research method. In this section, the data-gathering methods of the design science will be described: what they are and how they were conducted. Section 3.2.1 will give an introduction to a literature study and how it was used to gain knowledge from both previous work and relevant techniques. Section 3.2.2 will give an introduction to an Exploratory Data Analysis, how it was conducted, and what insights were gained.

3.2.1 Literature Study

The literature study is a method for gathering information from related work. It is performed to get a grasp of the related work along with a better understanding of the problem at hand. It is performed by reviewing and reading articles and research papers from credible sources. A literature study starts broad and then interesting subjects are selected and researched in more detail. In order to get an understanding of the article’s relevance the introduction along with the abstract is read.

To find a suitable approach for this thesis problem, several potential approaches are to be gathered from the literature study and reviewed in more detail. The combined expertise of both the supervisor of the university and experts from Combitech is used to discard certain approaches and identify viable ones. The candidate approaches require a deeper understanding before a final approach can be chosen. The selection of a final approach considers several factors, including related work, prerequisites, and available data.
Specific search queries are used in order to get relevant articles on different topics. The majority of searches are conducted on Google Scholar. Google Scholar provides academic papers, articles, and books from a wide range of sources by academic publishers, professional societies, online repositories, and universities. Since all work is scientific research it is generally of a higher validity compared to other sources and has to different extents been reviewed.

The literature study was divided into different parts. The first part is more general to gain insight and potential approaches, whereas the following parts were more specific on the approaches. A selection of search strings used was: "NLP python", "seq to seq", "Text to SQL", and "text clustering for searching". A huge number of results were found, and Google Scholar presents the results based on relevance. The result of the searches can be seen in Table 3.2.

<table>
<thead>
<tr>
<th>Date</th>
<th>Search strings</th>
<th>Search engine</th>
<th>Results</th>
</tr>
</thead>
<tbody>
<tr>
<td>2023-04</td>
<td>NLP python</td>
<td>Google Scholar</td>
<td>77 600</td>
</tr>
<tr>
<td>2023-04</td>
<td>seq to seq</td>
<td>Google Scholar</td>
<td>3 520 000</td>
</tr>
<tr>
<td>2023-04</td>
<td>Text to sql</td>
<td>Google Scholar</td>
<td>320 000</td>
</tr>
<tr>
<td>2023-04</td>
<td>text clustering for searching</td>
<td>Google Scholar</td>
<td>1 320 000</td>
</tr>
</tbody>
</table>

When acquiring a potential approach, specific search queries are utilized to gain a better technical understanding of the approach. Google Scholar is used to obtain results ranked by relevance, which are then scanned to find articles pertaining to the desired methods. Various search strings are used, such as "RNN with LSTM," "RNN LSTM NLP," and "RNN LSTM Python.". Resulting searches can be seen in Table 3.3

<table>
<thead>
<tr>
<th>Date</th>
<th>Search strings</th>
<th>Search engine</th>
<th>Results</th>
</tr>
</thead>
<tbody>
<tr>
<td>2023-04</td>
<td>RNN with LSTM</td>
<td>Google Scholar</td>
<td>190 000</td>
</tr>
<tr>
<td>2023-04</td>
<td>rnn lstm nlp</td>
<td>Google Scholar</td>
<td>62 800</td>
</tr>
<tr>
<td>2023-04</td>
<td>rnn lstm python</td>
<td>Google Scholar</td>
<td>49 600</td>
</tr>
</tbody>
</table>

To efficiently sort out irrelevant search results, reading the title and abstract provides key information regarding the relevance of the articles to the problem. The combination of keywords also gives insight into the relevance of the article. Employing these strategies allows for efficient filtering of articles relevant to the problem.

### 3.2.2 Exploratory Data Analysis

Exploratory data analysis is used to gain insight into the data. To discover patterns, find anomalies, and test hypotheses. Oftentimes when data is first seen, it is easy for a user to make assumptions, data analysis is also used to ensure the assumptions are correct or to find misconceptions. It can also be used to gain insight into linguistics and to summarize the main characteristics of the data. It is common to use statistical graphics to
visualize the data and make it easier to draw conclusions. This can then at a later stage be used to draw conclusions on what methods are suitable. For example, some algorithms are better at interpreting small data sets with many unique words whilst others may want big amounts of data that are more coherent. This, in turn, means that in most cases it is integral to conduct a data analysis to find strengths and weaknesses of the data.

The application of Exploratory data analysis for this thesis is used to gain a general insight into how the data is constructed and research what benefits and drawbacks there are to how the data is constructed. This knowledge is then to be used when choosing an approach.

The insight gained from the EDA aided in constructing a final viable approach. The EDA also assisted in eliminating previously thought-out approaches which were considered from earlier work before the analysis was conducted.

3.3 Reliability and Validity

Reliability is referred to as consistency, meaning, that if others attempt to solve this problem using the methods in this thesis, the same results can be accomplished. Due to the nature of machine learning and neural networks, the model will not be fully replicable but similar results will be achievable. The code base will not be publicly available after this thesis is published. Furthermore, training models are reliant on training data. If different training data is used the outcome will prove different from this thesis. Summarized, without the code base and the data sets used in this thesis, the same results can not be accomplished.

Results and conclusions must be validated to ensure that the results can be trusted for further work and development. A possible validity concern is the construction of the training data. Since it will be constructed, bias may come into effect and the training data may resemble the test cases. This may lead to the model not capturing nuances and working more favorably for the test cases. This will be validated by having multiple people test the solution to see how the solution performs on real user input. The people testing the solution will also receive a questionnaire. This will allow for a better understanding of the people’s experience and aid in the analysis of the solution.

3.4 Ethical considerations

It is important to note that the data in this project is strictly mock data and has been modified to exclude sensitive information. It is still of priority to keep it confidential as it should not be shared with third parties. This is due to the data not being publicly available and therefore should be kept secret as per request by Combitech. This is important to this thesis and all material will be reviewed by Combitech before publication. The identities of the people will be kept confidential, but their relevant technical backgrounds will be presented to further validate the feedback.
4 Implementation

This section will give an overview and explanation of how the solution is implemented. Subsection 4.1 will explain how the model was trained and what steps are necessary to take in order to train a seq-to-seq model. These steps include what the training data consist of, and how it will be produced. The training data is then converted to vector representations that a machine can understand. The last step of the training phase explains how the model is trained using these vectors. The remaining Subsections 4.2 and 4.3, will individually showcase how the database is used and the role of a Graphical User Interface.

4.1 Training Phase

This Subsection will provide an overview and explanation of the training phase. This is a critical aspect of implementing a well-performing seq-to-seq model for natural language translation. The section will include what is considered good data, how the training data was created, vector representations of the natural language words and how the model is trained using an encoder and decoder using a special Recurrent Neural Network architecture called Long Short-Term Memory.

Figure 4.8: Overview of the training phase

Figure 4.8 illustrate the flow of the training phase. In order to train a model to make predictions on natural language, the model needs to be fed numerical representations of the natural language text. The seq-to-seq model will adjust its inner workings to produce good accuracy. The model can be saved for future use.
4.1.1 Training data

The quality of data is very important for machine learning in general. For data to be considered optimal, certain criteria need to be met:

- **Volume** - The volume of the data is an important criterion for training a model accurately.
- **Relevant** - The data should be relevant to the problem to be solved.
- **Clean** - Absence of outliers, inconsistencies, and other unwanted noise.

When training a machine learning model, the quality of the data is crucial, since many factors can impact its accuracy. A larger volume of data allows for a more complex model and more accurate predictions. The significance of quality is usually greater than that of quantity since irrelevant or noisy data can hurt model performance. However, these criteria are not sufficient for Neural Networks, such as Recurrent Neural Networks and Long Short-Term Memory.

- **Sequential** - A certain order exists in the data.
- **Length** - The data has different shapes and sizes, learn the model to be adaptive.
- **Balance** - Good distribution of categories.

The volume, relevance, and purity of the data are also important for Recurrent Neural Networks such as Long Short-Term Memory.

The purpose of the training data is to train a model to make accurate predictions on unseen data. This model will be trained to translate the English language to SQL queries. In order to train a model to be able to predict this, training data that consist of both English sentences and a part of the corresponding SQL queries are required.

With the insight gained from the Exploratory Data Analysis, combined with the expertise of the supervisor at Combitech, training data is produced by taking key features from the mock data, constructing a human-like sentence around that data, and corresponding semi-complete SQL queries for each question. The natural language questions are not manually constructed, they are instead generated using a rule-based system to ensure a wide diversity of formulated questions. It is possible to receive questions with incorrect grammatical structure since the approach used is to produce quantity instead of quality sentences. The format and pattern of the English sentences, as well as the keywords and columns, are important when creating the training data. These questions attempt to resemble how the average user performs searches in the live database. The data in the live database contains a large number of categories, but for the purpose of this thesis, the most relevant is selected to create the mock data. The English sentences need to include words from these categories since the training sentences are to produce an SQL query for retrieving the wanted items in the database. The corresponding SQL queries for each English sentence are created manually by analyzing the sentence and writing the corresponding query. The query does not consist of a complete SQL query and excludes the "SELECT * FROM" since the initial phrase is precisely the same for each query.

```
ITEM_NAME LIKE '%hexagon%' AND THREAD_DIAMETER LIKE 'Z'
```
The text above is an example of a query that the model is trained on. The corresponding natural language question could look like the following:

- Searching after any is CAP SCREW hexagon 6 is diameter.
- Tell screw of the type is CAP SCREW hexagon mm nominal diameter 3.5.
- Need an screw is SCREW hexagon also millimeters 16 is nominal diameter.

The purpose is to teach the model to identify patterns in questions that seek the same answer. The training data consist of different complexity and variance in combinations of columns/features in sentences, among the 10 columns of the raw data. Regarding the data that is used, a decision is made for each column/feature if a name from the raw data can be used, or if a special character is required to be used as a placeholder. Some of the columns/features in the raw data have a good distribution of words, and others have poor distribution. For example, a column consisting of 100 rows and having 4 unique words is a good candidate for training a model on, but a column with 100 rows consisting of unique numbers is a bad candidate because the model would need to train on all these unique numbers excessively. The amount of training data would be unnecessarily large. To conclude, this means that columns containing too much variance are instead trained with a placeholder where the search word or number is replaced as part of the post-processing of the predicted sentence.

The importance of correct training data is crucial for a well-performing seq-to-seq model. If the model is trained on low-quality data, the model will not be able to predict correct and accurate results, meaning the sentences and queries has to be correct.

The English sentences will be used as input data, and the SQL queries will be used as target data. This means that the model learns to recognize words and patterns to identify the correct SQL query, i.e., the model understands what the user is looking for. This will be explained further in Section 4.1.3. However, before the training data can be used for training the model, it needs to be processed into something computers can understand; numbers.

4.1.2 Pre-processing

The process of data manipulation and transformation is crucial, as it guarantees data usability and compatibility with the model. This thesis focuses on natural language, a form of communication that machines are unable to comprehend, thus it needs conversion into a machine-readable format for the model to understand.

In order for the data to be comprehensible by the machine, the data is vectorized, meaning that each distinct word is represented as a numerical vector. For this thesis, one-hot vectors are used, which results in each unique word getting a unique index in a vector that represents that word.

When using an entire dataset, each unique word is represented in one vector. The vector’s size is equal to the number of unique words in the dataset. A word is then represented in this vector, among all unique words. For the following example only one sentence is used, "I have a cat". A one-hot vector would be of length 4 where each position corresponds to the word in that position. For example [0,1,0,0] equals "have". This is the fundamentals
of how one hot vector work. The size of the vector is equal to the number of unique words.

In the pre-processing stage, dictionaries are constructed. They contain one-hot vectors and the corresponding sentence. The purpose of the dictionaries is to be able to translate the one-hot vectors back into words and also words back into vectors. The purpose is to help in the testing phase, by having a convenient method of achieving the translation.

When the words are represented as vectors they can be used to train the model. Likewise, unseen data such as input is also pre-processed in a similar way for the model to understand and draw conclusions. The pre-processing is an integral part of the machine to understand the input. It also serves to simplify handling that data at later stages.

### 4.1.3 Training model

A machine learning model needs to be trained in order to eventually give predictions. This process is iterative and may, depending on data size and other parameters take a substantial amount of time. Parameters, therefore, need to be set to get a good balance of model accuracy and time spent. Batch size sets the number of training samples to work through before the model parameters are updated. Epochs are the number of times the algorithm will work through the entire training data. latent dimensionality is the dimensionality of the neural networks. The following parameters were chosen after testing several combinations.

- Batch size - 15
- epochs - 60
- latent dimensionality - 100

There are no set rules to follow in order to ensure satisfying parameters, but generally, a large number of epochs and a great latent dimensional along with a small batch size. Although it is important to keep in mind the importance of small batch sizes, many epochs and a large latent dimensional lead to long training times.

The training data set contains 9600 different questions and their corresponding SQL answers. There are roughly 300 different questions for each corresponding SQL output meaning the model is trained to answer a fixed set of unique semi-completed SQL queries. The final completion of the SQL queries is performed as part of a post-processing step to include all needed characteristics.

Python libraries Keras and TensorFlow are machine learning libraries that implement the mathematics of ML models, meaning that a developer can focus on high-level details of a Neural Network instead of low-level tasks, such as math. Before training commences the encoder and decoder are set up. They consist of a series of Long Short-Term Memory cells, defined by TensorFlow. The encoder inputs consist of both input one-hot vectors and an encoder state that is received from the previous cell. The decoder receives target one-hot vectors, a state from the previous cell, and a context vector from the encoder which is a representation of the words of sequences that the encoder received. The decoder’s final output is a probability vector to predict the final output based on the input. One-hot vectors used in the encoder are vector representations of the input data, while
one-hot vectors for the decoder are vector representations of the target data.

The optimizer is an algorithm that modifies a neural network’s weights and biases to reduce the difference between expected and actual results. The optimization algorithm known as the "rmsprop" optimizer is based on a modified form of gradient descent, and will be used for this model.

The encoder and decoder input (one-hot vectors) are also sent in along with the other parameters discussed (batch size, epochs). The training data is split up to reserve 10% of it for validation. Using a validation set is important since it allows to see how the model is performing on unseen data. The model is then trained with the set parameters. In essence, the training changes the inner states of the cells in the LSTM in order to get the best accuracy.

Once training is completed, final versions of the encoder and decoder are saved. This means saving their structure and inner states. The saving process is a tensor flow function that saves the model as a .h5 file that can then later be loaded. Once saved they may at any time be loaded and given new unseen input data and will in turn output and prediction of what the translation is.

### 4.2 Testing

A way of testing the model’s performance is important to see how it makes predictions on both seen and unseen data. This procedure is similar to the training phase but differs in some ways. In training, the encoder and decoder are refined by comparing the predicted output to the correct output, but in testing, the encoder and decoder are used to make predictions in real-time, without their inner states being adjusted for a more accurate prediction.

The input consists of input text that is entered by a user, and similar to the training data, the user input text also needs to be converted into vectors that the encoder and decoder can use to make predictions. The training data is constructed to cover much diversity, but for the scope of this thesis, every possible formulation can’t be covered. A user’s input may also contain irrelevant information and words that have no meaning for the end goal of the search. These words may be "Hello Ai", "Please", or "thank you". Including these words and phrases when creating a vector could confuse the model and result in incorrect output. Therefore, the testing includes a check to see if each word in the input text is previously known among the trained words. As the training data contains high diversity, important information is rarely excluded and does not impact predictions.

---

![Figure 4.9: Preprocessing sentence](image)

Figure 4.9: Preprocessing sentence
As seen in Figure 4.9, a sentence containing words without important relevance for the search is indicated. The final sentence is stripped from these words, allowing for vectors that the encoder and decoder can utilize for an accurate prediction of what the user’s intent is. There is no other check on the user’s input, meaning all input is allowed. This means that the user needs to input a correct request. When a user performs a test, the user will receive a protocol for item characteristics that exist in the database and that the user can formulate a question with. The protocol will be presented in Section 5.1. If a user attempts to request an item that does not exist in the database, the model will still make a prediction. The predicted SQL query will be used to retrieve items from the database, and if the database search result in zero results, the user will be prompted to rewrite the request.

4.3 Database

A local database is created to serve as a test environment. This is in order to demonstrate the SQL queries generated by the model. The reasoning behind the creation of a local database is twofold, first and foremost the data is not publicly available and can not be accessed without permission. Due to that reason, the data should not be uploaded to an online database. Secondly, since this thesis serves as a proof of concept, a local database is more configurable and more convenient to manage to serve the needs of this thesis. As previously mentioned the data is not publicly available. Therefore the data in the local database had been altered and excluded sensitive parts, but overall bears resemblance to the original database. For the setup of the database “sqlite3”, a Python library was used. It serves as a local easy-to-use database. sqlite3 allows you to create a local database and execute SQL queries.

4.4 GUI

A graphical user interface (GUI) is a type of interface that allows users to interact with digital devices through graphical elements such as icons and buttons. The purpose of this thesis is not to develop an advanced GUI but rather to serve as a proof of concept of the viability of translation from natural language to SQL queries. In this regard, "PySimpleGUI" was utilized as a tool to rapidly create a GUI for the purpose of demonstration.

The primary motivation for creating the GUI was to simplify the testing process and enable inexperienced users to perform testing more easily, by not having to use a command line interface. This kind of GUI also aims to eliminate complex searches, as database GUIs today may be confusing and complicated to use. Additionally, the use of a GUI allows for the output to be presented more clearly and in an easily interpretable format.

The testing procedure explained in Subsection 4.2 is utilized in the background of the GUI. The user input is required to be processed into vectors that can be used for predictions. Each word of the user input is used to map vectors for the encoder input data, which is used for generating a prediction on what the user is looking for. The encoder input data is used for creating a target sequence to be used for the decoder to make predictions on each word.

Before presenting the results in the GUI, post-processing is performed. Since the beginning of the SQL query is always the same, "SELECT * FROM" is added after the model has predicted what should fill the WHERE part of the query. If the model predicted any
of the predefined placeholders as explained in Subsection 4.1.1, the GUI also fills the placeholders with the relevant number before the query and result are presented to the user.

Figure 4.10: Decoder prediction

Figure 4.10 is showing how a user input consisting of the sentence "show me all hexagon head screw with thread diameter 6" is processed by the model. Each word or step is individually processed and predicted. The predicted query is post-processed for possible placeholders.
5 Experimental Setup and Results

5.1 Setup

With a trained model, a demo can be set up. The demo will consist of a GUI, and the
local database containing the raw data. The person testing the database will receive a
protocol for using the demo and is instructed to retrieve an item. The protocol can be seen
in Appendix 1.

Figure 5.11: Overview of the GUI

Figure 5.11 shows the created GUI that will be presented to the user testing the demo.
The GUI consists of a user input field, and three buttons, one for entering the input,
one for clearing the previous search bar and result, and one button for exiting the demo.
If the search result from a user input is translated correctly, and if the item(s) exist in
the database, the result from the database will be presented in the GUI, below the user
input field and the buttons. In the case of input that cannot be correctly translated to the
corresponding query, a popup will appear, prompting the user to change their search. The
popup can also be triggered by attempting to retrieve an item that does not exist in the
database.

Figure 5.12: Overview of the test phase

The flow of the demo can be seen in Figure 5.12. The user input will be cleaned and pro-
cessed into machine-understandable vector representations of the sentence, which then
will be fed into the model. The model will output the predicted SQL query, which will be
used on the database to retrieve the item(s). The result of the search will be presented to
the user in the GUI.

The two tables, Table 5.4 and Table 5.5, presented below are part of the user test, they represent the required user input. The content of the tables is structured to allow a quality analysis of the result. This structure applies to the technical test as well as the user test. The results from the user test will be more subjective than the technical test, since a person may have different knowledge and prejudice in operating a database and formulating searches.

The aforementioned tables provide the user with a comprehensive overview of searchable criteria. For instance, in Table 5.4, the user can conduct searches based on the type/name in one search, while in another search, they can search for the thread length. These tables are presented to the user during the test to serve as a guiding reference for the available search criteria. This gives the user an indication of what information is possible to search for, otherwise, a user could search for the color of a screw, which is not stored in the database and therefore cannot be retrieved through an SQL query.

The numbers and examples following this statement serve as examples of known values that exist in the database. It is important to note that it is possible to search for a unit or name that does not exist. While this will generate the correct SQL query, it may not yield any results since the requested item may not exist in the database.

Table 5.4: Tests of using one search criteria

<table>
<thead>
<tr>
<th><strong>type/name</strong></th>
<th>either socket or hexagon</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>thread diameter</strong></td>
<td>either; 5, 10, 80, 16, 4, 3, 6, or 12</td>
</tr>
<tr>
<td>optional <strong>thread diameter</strong></td>
<td></td>
</tr>
<tr>
<td><strong>thread length</strong></td>
<td>either; 16, 8.5, 6, 7.5, 4, 20, 30, or 28</td>
</tr>
<tr>
<td>optional <strong>thread length</strong></td>
<td></td>
</tr>
<tr>
<td><strong>fastener length</strong></td>
<td>either; 10, 12, 6, 30, 25, 80, 30, or 100</td>
</tr>
<tr>
<td>optional <strong>fastener length</strong></td>
<td></td>
</tr>
<tr>
<td><strong>width between flats</strong></td>
<td>either; 2, 5, 6, 8, 115, 36, 13, 18, 16, or 24</td>
</tr>
<tr>
<td>optional <strong>width between flats</strong></td>
<td></td>
</tr>
<tr>
<td><strong>material</strong></td>
<td>either; wood, plastic, steel, stainless steel, or titanium</td>
</tr>
<tr>
<td>optional <strong>material</strong></td>
<td></td>
</tr>
<tr>
<td><strong>strength grade</strong></td>
<td>either; 8.8, 10.9, or 12.9</td>
</tr>
<tr>
<td>optional <strong>strength grade</strong></td>
<td></td>
</tr>
</tbody>
</table>

Table 5.5 operates in a similar fashion to Table 5.4 but provides instructions on searching for two parameters simultaneously. Table 5.5 consist of examples of search criteria that are allowed to combine when formulating a question. This allows for a precise search that can output fewer search results that are more accurate from the database compared to the questions allowed in Table 5.5. The goal of these combinations is, beyond a more accurate search result, to evaluate the model’s performance on more complex user input requests.
Table 5.5: Tests of using two search criteria

<p>| | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>optional type/name</td>
<td>optional thread diameter</td>
<td></td>
</tr>
<tr>
<td>optional type/name</td>
<td>optional material</td>
<td></td>
</tr>
<tr>
<td>optional type/name</td>
<td>optional strength grade</td>
<td></td>
</tr>
<tr>
<td>fastener length</td>
<td>optional material</td>
<td></td>
</tr>
<tr>
<td>fastener length</td>
<td>optional strength grade</td>
<td></td>
</tr>
<tr>
<td>optional thread diameter</td>
<td>optional material</td>
<td></td>
</tr>
<tr>
<td>optional thread diameter</td>
<td>optional strength grade</td>
<td></td>
</tr>
</tbody>
</table>

Each row in the tables represents a test. The model has been trained on data for each of the item keywords with different ways of asking for an item using that keyword. The model will be tested by searching for an item keyword along with the attributes available in the database. The test also allows for a stress test, meaning asking for a keyword with an optional attribute. The first table is constructed for simpler queries and the second is constructed for more complex ones.

After a user completes the test, the user will receive a form containing questions to be answered. The goal of the form is to give the authors of the thesis a better picture of the overall state of the developed solution. These questions will include the general experience, what the users think are the strengths and weaknesses, and other general opinions of the solution. The questionnaire can be seen in Appendix 1.

5.2 Results

This section presents the results of the solution. The model’s performance is evaluated by feeding the model natural questions aimed at retrieving an item by targeting specific columns with different complexities and combinations. This will be measured both on a test data set, meaning data unseen to the model but has a similar structure to the natural language questions that the model has been trained on, and by having people with different backgrounds and habits of using a database perform tests using the demo without any restraints on how to write a question. The people participating in the test submit feedback through a form, which is summarized and presented.

5.2.1 Model

During training some statistical parameters are kept track of, accuracy as well as the loss for both the training set and validation set. The accuracy tracks the overall similarity of the model’s predictions on the training data. The loss parameter indicates the average difference between the model predictions and the target outputs on the training data. Lower loss values indicate better alignment between predictions and targets. Table 5.6 shows the parameters for the last epoch.
Table 5.6: Model score

<table>
<thead>
<tr>
<th>Time</th>
<th>Loss</th>
<th>Accuracy</th>
<th>Validation loss</th>
<th>Validation Accuracy</th>
</tr>
</thead>
<tbody>
<tr>
<td>13</td>
<td>0.2344</td>
<td>0.4059</td>
<td>0.6911</td>
<td>0.4032</td>
</tr>
</tbody>
</table>

Result of training the model with the hyperparameters specified in chapter 4.1.3. Training time is on average 13 seconds for each epoch and a total of 60 epochs leading to a total training time of around 13 minutes.

5.2.2 Test set

Testing of the complete model is conducted by having the model make predictions on a data set containing observations unseen to the model. In machine learning, this is called a test set and is used after the model has been trained and fine-tuned using the training and validation set. The test data is constructed in the same way as the data used for training the model. Examples of input sentences are:

- Show me a THREAD DIAMETER is 30 millimeter
- Do display an LENGTH OF THREAD 96.25 milimeter
- What are an BUILT IN is STEEL
- I need a THREAD DIAMETER is 5 milimeter and also BUILT BY is PLASTIC
- Search after a DIAMETER is 14 STRENGTH CLASSIFICATION is 8.8 megapascal

The test conducted on the test data provides 100% accurate results. This test serves to prove that the model is able to predict all possible outcomes. However, this does not mean that the model, in reality, is 100% on every possible formulation. The test data is constructed in a similar way as the training data, which both follow the same pattern. A user, however, who doesn’t have any idea of how the model is trained and what kind of pattern on a question the model expects, will formulate a question which they see fit. The result from the user test is presented in Chapter 5.2.3.

5.2.3 User result

Four people with different backgrounds and habits of operating a database performed this test. Each person has been chosen based on their habit of using the live database. This is considered valuable insight since every user formulates their request in different ways.

<table>
<thead>
<tr>
<th>Person 1</th>
<th>Person 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>Education: Civil engineer</td>
<td>Education: Engineer</td>
</tr>
<tr>
<td>Role: Software developer</td>
<td>Role: Junior material data specialist</td>
</tr>
<tr>
<td>Habit of computer systems: High</td>
<td>Habit of computer systems: Medium</td>
</tr>
<tr>
<td>Habit of live database: Undefined</td>
<td>Habit of live database: Medium</td>
</tr>
</tbody>
</table>
Person 3
Education: Vocation education
Role: Junior material data specialist
Habit of computer systems: Medium
Habit of live database: Low

Person 4
Education: Nontechnical education
Role: Technology informant
Habit of computer systems: Low
Habit of live database: High

A user is given 30 minutes to conduct the test, the user is instructed to perform searches that consisted of different complexities and diversity among the categories that exist in the data. There are no restraints on how the user formulates a question to retrieve items, meaning the user may formulate their search for an item however they like. The users will receive a protocol containing the tables, presented in Section 5.1 and can also be seen in Appendix 1.

Figure 5.13: Result of user test on Table 5.4

Figure 5.13 is presenting the summarized result of four different people making searches for items in Table 5.4. This result is from the table which is constructed for questions containing one keyword/column.

Figure 5.14: Result of user test on table 5.5

Figure 5.14 is presenting the summarized result of four different people making searches for items in Table 5.5. This result is from the table which is constructed for questions with a combination of two keywords/columns.
The goal of presenting the protocol to a user is to test the model on each possible column combination but with natural language questions the user believes should suffice. The user has, as stated, no restraint on how to formulate a question. The two tables consist of the total number of predictions and the number of successful predictions. Each column pair, the number of predictions and successful predictions, represents each corresponding column in the database that the user is trying to use in their natural language question to retrieve an item.

Figure 5.15: Result of user test

Figure 5.15 is presenting the summarized result of the four different persons performing the test. The tables present each person’s result, the total number of predictions, and their number of successful searches. Table 5.7 presents the rate of successfully predicted questions for each person.

<table>
<thead>
<tr>
<th>Table 5.7: Success percentage of each person</th>
</tr>
</thead>
<tbody>
<tr>
<td>Success rate</td>
</tr>
<tr>
<td>------------------</td>
</tr>
<tr>
<td></td>
</tr>
</tbody>
</table>

The result in Figure 5.15 also includes predictions on questions that were not included in the test protocol. Each user was encouraged to write their own questions without limitations of complexity and number of combinations. The total success rate for the persons performing the test resulted in 52.7%. This is calculated through a total of 184 predictions and 98 successful predictions.

5.2.4 User feedback

After a completed test, each user has the opportunity to submit their feedback of the solution. The questions present in the form are presented here. The answers of each person
are summarized and presented below.

**Q:** How did you experience the solution, in general?
**A:** The solution works to some extent, it is not 100% accurate on every search. A user made a note of the importance of the need for correct search criteria, otherwise, the result was poor. Some searches result in too many database articles, more than wanted. The goal of free text as input is appreciated and a good feature, the search does not require specific search criteria compared to a regular database. The GUI was well-designed and easy to use.

**Q:** How did you experience the accuracy of the search?
**A:** The accuracy was not optimal. The users all found that with higher complexity and combinations of the questions, the more the model struggled and the accuracy declined. The model seemed to perform well on simpler questions. Definitely room for improvement, sometimes the columns were interchanged between what was requested and what the predicted SQL query resulted in.

**Q:** What were your expectations of the solution?
**A:** Some of the users had no expectations at all, and some were expecting a text-to-SQL tool. Overall they did not expect the solution to predict at 100% accuracy, but they agreed that it was an impressive tool.

**Q:** Did the generated result match the result you were expecting?
**A:** The users agreed that the model performed better at simpler questions compared to complex ones. Some of the users discovered that poorly formulated questions resulted in the model predicting the wrong columns.

**Q:** What do you believe the solution’s weaknesses are?
**A:** The weakness of the model is its accuracy. It cannot, in the state it is in, be relied on in real work. The model had a habit of predicting the wrong column regardless of some questions. Some of the users with medium to high habits of operating the live database expressed that the requirement for correct formulations still was an issue in this solution, which is a problem in the live database. However, they pointed out that the problem can’t be compared as the issue in the real database requires an understanding of the database and its content, and this solution requires an understanding of how the model is expecting a question to be formulated.

**Q:** What do you believe the solution’s strengths are?
**A:** The ability to type in text without having to worry about irrelevant information and be less specific in searches is an improvement, compared to other databases. Some of the users also stated that the number of database articles that were retrieved was an improvement compared to other databases. Overall, the users experienced the solution to be fast, clear and concise, and an appreciation of having to be less specific in searches.

**Q:** Other comments or opinions?
**A:** The overall opinion was to improve the accuracy of the model. A user should be able to write more diversely and flexibly. The users with medium to high habits of the live database found the solution interesting and can see the potential of a more sophisticated tool. All of the users agreed that artificial intelligence is the future and organizations
should embrace improvements and solutions AI can offer.
6 Analysis

In this Chapter, the results will be analyzed. The test results from Chapter 5.2.2 will be analyzed and explained. The user results and feedback from Chapters 5.2.3 and 5.2.4 are discussed and analyzed in order to draw insight from the results.

The result presented in Chapter 5.2.2 test set proved to be 100% accurate. This is an interesting observation as it shows that the model responds well to unseen data with a similar structure and pattern of the sentence as the training data. A perfect test set score could point towards such a fact. The most important aspect observed in the testing on the test data is that all of the possible SQL outputs can be predicted and the model does not exclude any of the possibilities. Meaning that all trained columns of the database can be predicted and translated into an SQL query. The possible search criteria can be seen in Chapter 5.1.1.

The user results give more insight into how well this implementation works in a production environment. The results indicate that the model works on input provided by people with no prior knowledge of how the model is trained, meaning the user does not have any restraints on how they may formulate a question. It is important to note that the user results are less accurate. This is in part due to the users having no knowledge of how the model works, there is a learning curve in order to know what the database contains. Due to the user’s questions being influenced by their experience, some of the user inputs were invalid, meaning the user asked for things the model and database do not contain. For example, the total length of the screw, when such fact is not stored in the database, therefore, no accurate SQL query can be constructed and no correct response can be given. It can also be observed that at times the model can not predict a valid input even though all necessary information is provided, such cases are interesting to study in order to further improve the model. The model can predict an SQL query even if it will return zero results as long as all relevant input is given. For example, if the user is looking for a screw with a thread length of 100 mm and no such screw exists, the correct SQL query is still predicted.

The feedback from the people participating in the test is interesting. Some of the people had no idea of what to expect and some knew what was to be expected. The overall experience was good, the solution is impressive even though time and resources were limited throughout the development of the solution. However, the feedback is stating that the solution is not performing at a reliable level, meaning the search result are not always 100% accurate. A search requires a specific sentence formulation with correct search criteria, to avoid poor search results. The number of resulting database articles is not always optimal, and more than wanted is sometimes presented. These problems all have a common denominator, the training data. These problems are also brought up in the previous text snippet, regarding the analysis of the user result. The model is trained on data in an attempt to mimic the way a user is formulating a question to retrieve an item. The gap between the test result and the user result is because of how the test data is structured and how a user believes is a good way to write a question to retrieve an item. The training data consist of structured sentences for retrieving an item, while a user, based on the user result, is blunt when inputting a request and wants the model to output the correct result based on little context. As the model is trained on a sentence pattern, it struggles to predict the correct SQL query on fewer words. This is an issue with insufficient training data. The authors are convinced this issue can be resolved with more training data. An approach to
creating new training data to fix this issue is to send out forms to people with different backgrounds and analyze how they operate a database and how they wish a search would be made easier compared to database searches today. However, there are still issues with the user result when a person is attempting to make a formulated sentence for an item but fails. The model sometimes fails to predict the correct columns and keywords. There are several reasons for this to occur, insufficient training data, uniquely formulated sentences, using a column or keyword that does not exist or using words that the model has not been trained on.

In conclusion from the test set, all questions can be predicted. This is an important insight as shows the model’s capabilities. The user results show that the model has a harder time accurately predicting the more diverse input given by real people. The user feedback states that a tool like this solution is interesting and something that is useful. On the other hand complaints on this model’s accuracy is the main complaint. Further work is needed before an implementation is ready.
7 Discussion

With a trained model and a test setup, the results are both surprising and interesting. The model responded well to input data that was in a similar structure as the training data. The model is not perfect and is sensitive to the structure of sentences, but can output correct SQL queries and differentiate between keywords and columns. The model does suffer from poor accuracy with input that the model has not been trained on, which results in wrong keywords or columns. The model is trained in a certain way of asking questions when in reality, different users have different ways of asking for items. Some users are less verbose when typing a request than others which leads to worse performance. The authors believe that the solution for this lies in future work and more training data covering more possible ways of asking for an item.

The objective of this thesis is to explore technologies and solutions to see if it is possible to make a logistical flow more efficient. It can be determined from the results and user feedback that a solution like the one provided is useful if further developed. The implementation and model are a proof of concept, the model’s limitations are noticed by the users and can be observed in the feedback. This means that in order for an implementation to be viable in a production environment more development is needed.

Due to the implementation conducted in this thesis being specific to Combitech requirements, results can not be directly compared to related work and/or other models. Most of the solutions found in published articles are trained on given, well-known data sets. The solution produced for this thesis is not intended for such cases and thus a comparison cannot be made. There are other implementations of natural language to SQL but there are not enough similarities to draw any conclusions on what method is best.

When the implementation took form, the goal was to be able to handle questions containing one search criteria and expand the capacity ongoing. The greatest roadblock was time, training a model with new training data requires adjustment to parameters, and more data equals longer training time. The supervisor at Combitech was satisfied with the model being able to handle questions containing two search criteria, which was considered a good result for the proof of concept. Expanding the capacity to three search criteria was experimented with, but due to limited training data and time, it was concluded to be beyond the scope of the thesis. A decision was made to limit searches to two search criteria. This was the best option in order to keep the overall accuracy and performance of the model good.

Besides the scope and time of the thesis, a possible reason for not successfully implementing support for being able to handle a question with three search criteria may be because of LSTM’s limitations. LSTM is a good solution for sequence-to-sequence tasks, but it is not an approach modern and sophisticated machine translation models use. These more advanced architectures will be explained as potential approaches in a chapter covering future work. However, this type of sequence-to-sequence model using LSTM can still be enough and be a viable solution for a problem this thesis attempts to solve. The solution requires more development if this would be used in a live production environment.
8 Conclusion

Based on the results of this thesis, it can be determined that a solution of this nature is beneficial and something that the company deems to be useful. It is important to note that the solution is a proof of concept to explore how an implementation using artificial intelligence can solve the problem. From the user tests, it can be determined that the implementation has some flaws, most predominantly the accuracy. This is expected as it is the first iteration of the model. Further development of the model will be explained in Chapter 8.1. From the user test results, it can be determined that a solution such as the one provided has the potential to operate in a production environment.

The implementation is tailored towards Combitech’s needs and it would not be possible to directly implement it in other organizations. The model itself has no limitations to where it could not be implemented elsewhere, the main requirement to implement this solution for another database would be the raw data. The problems that arise when users search in databases are not limited to Combitech and, therefore, a similar solution would most certainly be interesting for other industries, companies, and society overall.

It is always possible to further improve the results of the thesis, but as outlined, this serves as a proof of concept. The model has flaws that make it hard to implement more complicated questions, as new training data is needed for each possible formulation. To conclude and summarize this thesis, the problem formulation has been accomplished to some extent; the solution allows for a diverse search, meaning there are multiple ways of asking for a specific item, unique spelling, and synonyms are allowed as long as the formulation and structure are deemed sufficient by the model, and the search input is allowed to be human-like. The accuracy does suffer with input from a target audience, which can be solved by a more sophisticated trained model.

8.1 Future work

The main point that could benefit from future work is the model itself. Currently, the model needs to be trained on all possible formulations. The goal of a more refined model would be to only have the model trained on each possible column of the database and then be able to combine them to make a SQL prediction on several columns. As it currently stands a question containing two search terms needs to be trained separately this leads to a limitation of how complex the search inputs can be.

Furthermore, it might be beneficial to investigate having the model remember prior searches. The benefit would be that the user could then ask for some specifics and then in a new input add additions to the search based on the results. For example "I am looking for all screws with thread diameter 12", when the user then gets presented with the result it could be possible to in a new search input ask for "Ok show me only the ones made of steel". In conclusion, the model would remember and be able to make additions this would allow the user to have a natural conversation with the model to further improve the ease of use.

A future solution would not necessarily have to be a seq-2-seq model consisting of LSTM. New technologies arise constantly, and Natural Language Processing is no exception. LSTM and Recurrent Neural Networks are actually considered to be old news today and have been replaced by attention- and transformer models. These are more sophisticated NLP models and should be considered if a solution is to be further developed [26].
References


**Test & Evaluering**

Testprotokoll

Du som användare ska söka i en databas. Databasen är en materiell databas som innehåller flera poster med olika sorts skruvar, dessa skruvar har olika tekniska egenskaper (varje teknisk egenskap motsvaras av en engelsk term).

Sökverktyget är baserad på en AI-modell, som ska kunna tolka dina fritextfrågor till korrekt söksträngar (i SQL). Verktygsspråket är engelska.

Anga din sökinput på valfritt sätt, modellen ska kunna hantera ett vardagligt mänskligt språk. Rådgör med testledaren vid frågor.

Anga en sökinput (sök efter skruvar med/av) för följande tekniska egenskaper;

<table>
<thead>
<tr>
<th>Term</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>type/name</td>
<td>Anginen socket eller hexagon</td>
</tr>
<tr>
<td>thread diameter</td>
<td>Antingen; 5, 10, 80, 16, 4, 3, 6 eller 12 mm</td>
</tr>
<tr>
<td>thread diameter (i hela)</td>
<td>mm</td>
</tr>
<tr>
<td>thread length</td>
<td>Antingen; 16, 8.5, 6, 7.5, 4, 20, 30, eller 28 mm</td>
</tr>
<tr>
<td>thread length (i hela)</td>
<td>mm</td>
</tr>
<tr>
<td>fastener length</td>
<td>Antingen; 10, 12, 6, 30, 25, 80, 30, eller 100 mm</td>
</tr>
<tr>
<td>fastener length (i hela)</td>
<td>mm</td>
</tr>
<tr>
<td>width between flats</td>
<td>Antingen; 2, 5, 6, 8, 115, 36, 13, 18, 16 eller 24 mm</td>
</tr>
<tr>
<td>width between flats (i hela)</td>
<td>mm</td>
</tr>
<tr>
<td>material</td>
<td>Antingen; skruv, plast, titan</td>
</tr>
<tr>
<td>valfri material</td>
<td></td>
</tr>
<tr>
<td>strength grade</td>
<td>Antingen 8.8, 10.9 eller 12.9</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Term</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>valfri type/name</td>
<td>Valfri thread diameter</td>
</tr>
<tr>
<td>valfri type/name</td>
<td>Valfri material</td>
</tr>
<tr>
<td>valfri type/name</td>
<td>Valfri strength grade</td>
</tr>
<tr>
<td>valfri fastener length</td>
<td>Valfri material</td>
</tr>
<tr>
<td>valfri fastener length</td>
<td>Valfri strength grade</td>
</tr>
<tr>
<td>valfri thread diameter</td>
<td>Valfri material</td>
</tr>
<tr>
<td>valfri thread diameter</td>
<td>Valfri strength grade</td>
</tr>
</tbody>
</table>

(valfri) OCH/ELLER type/name, thread diameter, thread length, fastener length, width between flats, material, strength grade

Figure 1.16: Test protocol
Test & Evaluering

Utvärdering

Hur upplever du sökverktyget, generellt?

Hur upplever du sökningarnas träffsäkerhet?

Vad var dina förväntningar på sökverktyget?

Motswarar genererad sökoutput det du förväntade dig?

Vad anser du är modellens svagheter?

Vad anser du är modellens styrkor?

Hur hade du önskat att verktyget fungerade annorlunda?

Övriga kommentarer/synpunkter/åsikter?

Figure 1.17: Questionnaire