Petteri Olli

Utilizing AI in customer support work

School of Technology and Innovations
Master's Thesis
Automation and Information Technology

Vaasa 2024

UNIVERSITY OF VAASA
School of Technology and Innovations
Author: Petteri Olli
Title of the Thesis: Utilizing AI in customer support work
Degree: Master of Science in Technology
Programme: Automation and Information Technology
Supervisors: Nasser Abdullah and Jouni Lampinen
Year: 2024 **Pages:** 63

ABSTRACT:
Recent advances in artificial intelligence (AI) offer new opportunities for companies to streamline operations and gain a competitive edge. This thesis explores the potential of AI to improve customer support operations within a specific unit of a case company.

The study addresses three research questions: (1) What AI solutions are currently possible for different types of requests? (2) What is the most relevant data for training the AI model? (3) What key factors should future research address to improve AI solutions?

To answer these questions, the research analyzed customer support tickets from a three-month period, categorizing them into two groups based on their resolution methods. Group 1 consisted of cases that are straightforward to resolve, typically requiring only order-specific data from the company's enterprise resource planning (ERP) system provided by SAP. Group 2 included more complex cases that required additional investigation, often involving internal documentation or collaboration with engineers.

The analysis found that tasks in Group 1, due to the dynamic nature of order-specific data, were not ideal for direct AI model training. However, an AI-powered chatbot could assist specialists by providing guidance on case resolution and navigating the SAP ERP system. In contrast, tasks in Group 2, which involve more stable data such as technical documentation, are more suitable for AI integration. AI models trained on this data could improve information retrieval efficiency.

To test these findings, a pilot implementation was conducted using a subset of Group 2 cases. The pilot showed promising results, demonstrating the potential for further AI integration. However, limitations were also identified, including challenges with data processing and handling complex queries.

This study provided a solid framework for future research and AI implementations within the case company. Moving forward, enhancing AI models through improved data preprocessing and broader training datasets is crucial. Further investment in advanced AI technologies and their integration to support ticket system can significantly improve customer support capabilities. Collaboration with IT specialists and engagement with other ongoing AI projects within the company are recommended to optimize resources and maximize benefits.

KEYWORDS: Customer support, Resolution methods, AI model training

VAASAN YLIOPISTO
Tekniikan ja innovaatiojohtamisen yksikkö
Tekijä: Petteri Olli
Tutkielman nimi: Tekoälyn hyödyntäminen asiakaspalvelu työssä
Tutkinto: Diplomi-insinööri
Oppiaine: Automaatio- ja tietotekniikka
Työn ohjaajat: Nasser Abdullah ja Jouni Lampinen
Valmistumisvuosi: 2024 **Sivumäärä:** 63

TIIVISTELMÄ:
Viimeaikaiset edistysaskeleet tekoälyssä (AI) tarjoavat uusia mahdollisuuksia yrityksille tehostaa heidän toimintojaan ja saavuttaa kilpailuetua. Tämä diplomityö tutkii tekoälyn mahdollisuuksia kehittää asiakastuen toimintaa tietyssä kohdeyrityksen yksikössä.

Työ käsittelee kolmea tutkimuskysymystä: (1) Mitkä tekoälyratkaisut ovat tällä hetkellä mahdollisia erityyppisille asiakastukipyynnöille? (2) Mikä on tärkeintä dataa tekoälymallin kouluttamisessa? (3) Mitkä keskeiset tekijät tulisi ottaa huomioon tekoälyratkaisujen parantamiseksi tulevissa tutkimuksissa?

Näihin kysymyksiin vastaamiseksi työssä analysoitiin asiakastukipyyntöjä kolmen kuukauden ajalta. Pyynnöt pystyttiin jakamaan kahteen selkeästi toisistaan eroavaan ryhmään niiden ratkaisumenetelmien perusteella.

Ryhmä 1 koostui tapauksista, jotka ovat nopeampia ratkaista ja vaativat tyypillisesti vain tilauskohtaisia tietoja yrityksen toiminnanohjausjärjestelmästä (ERP). Ryhmä 2 sisälsi monimutkaisempia tapauksia, joiden ratkaiseminen vaatii enemmän tutkimusta. Näiden ratkaisu vie enemmän aikaa, koska tietoa joudutaan etsimään erilaisista tiedostoista ja yhteistyö insinöörien kanssa saattoi olla tarpeellista.

Analyysi osoitti, että ryhmän 1 tehtävät eivät ole ihanteellisia tekoälymallin kouluttamiseen tilauskohtaisten tietojen dynaamisen luonteen vuoksi. Sen sijaan tekoäly voisi auttaa asiantuntijoita tarjoamalla ohjeita tapausten ratkaisemiseksi ja SAP ERP -järjestelmässä navigoimiseksi. Ryhmän 2 tehtävien ratkaisemiseen käytettävä data, kuten tekniset dokumentit, on vakaampaa ja soveltuu siten paremmin tekoälymallin kouluttamiseen. Näillä tiedoilla koulutetut mallit voisivat parantaa tiedonhakutehokkuutta.

Näiden havaintojen testaamiseksi suoritettiin pilottitoteutus käyttäen pientä osaa ryhmän 2 tapauksista. Toteutus antoi lupaavia tuloksia, mikä osoittaa tekoälyn potentiaalin asiakastuessa. Erilaisia rajoituksia kuitenkin havaittiin, mikä vaikeutti monimutkaisten kyselyiden käsittelyä.

Tutkimus tarjosi hyvän pohjan jatkotutkimuksille. Lisäinvestoinnit tekoälyteknologioihin ja niiden integrointi tärkeisiin asiakaspalvelu työkaluihin voivat merkittävästi parantaa asiakastuen toimintaa. Yhteistyö IT-asiantuntijoiden kanssa ja resurssien yhdistäminen käynnissä oleviin tekoälyprojekteihin yrityksen sisällä on myös suositeltavaa hyötyjen maksimoimiseksi.

AVAINSANAT: Asiakastuki, Ratkaisumenetelmät, Tekoälymallien koulutus

Contents

Figures

1 Introduction

Recent advancements in artificial intelligence (AI) have created new opportunities for businesses to improve their operations. The case company is aware of these possibilities and sees the potential of using its internal data to train advanced AI models.

This thesis focuses on the customer support team's work processes, within a specific unit in the case company. It aims to identify work processes that would benefit most from AI implementations and to determine the most useful training data for each process.

1.1 Case Company

The case company is a significant player in the industrial field specializing in robotics, automation, and electrification. They manufacture products such as industrial robots, control systems, electrical drives, and electric motors.

1.2 Purpose

Customer support specialists are responsible for helping sales units on the issues their customers have raised. Support tickets are opened to the queues based on the nature of the issue.

While solving some of the requests is straightforward, others require more effort. In these cases, specialists might need to rely on documentation to verify the feasibility of the changes. Finding the correct files can be challenging and take a lot of time, especially when the documentation is not organized well.

The primary purpose of this thesis is to develop a chatbot that uses AI to make the customer support workflow more efficient. By training the AI model with relevant documents, the chatbot could provide immediate and accurate responses to questions related to the customers' inquiries. This would make retrieval of the information faster and reduce the reliance on engineers. Overall time to solve tickets could be reduced and some of the mistakes would be avoided.

This thesis will provide theoretical suggestions on how to implement the chatbot for different types of customer support cases. Each case type will be examined to determine the best approach for the implementation. Following this, a small number of cases will be selected for pilot implementation. This will serve as a starting point for a project that is expected to continue in the future.

While the primary proposal of this thesis focuses on specific uses of AI in customer support, the case company also requested a plan for a larger-scale AI project. The last chapter will address this and show the way for future research.

1.3 Research Questions and Scope of the Thesis

To fulfill these goals, the following research questions are used.

1. **What AI solutions are currently possible for different types of requests?**
   The study uses support ticket data from Salesforce to see what kind of requests the customer support team receives and how those are solved. This data together with the author's work experience is used to determine the AI solutions.
2. **What is the most relevant data for training the AI model?**
   Explores the types of data currently used by customer support specialists and evaluates what should be used for AI implementation.
3. **What key factors should future research address to improve AI solutions?**
   Solutions are proposed for the limitations identified in the original work. These insights can be used to guide future research.

This study is limited to the customer support team's workflow. Research data is collected from three-month period, covering in total 2,875 customer requests. This research is specifically designed for the unique industry and working environment of the case company. As a result, some conclusions might not apply to other industries.

1.4 Structure

This introduction is followed by six chapters, each offering a detailed exploration of key aspects of the research topic.

- **Chapter 2** provides a detailed literature review, introducing the key concepts of artificial intelligence (AI) and its subfields. Also explores a simple transformer architecture, which is the foundation of modern large language models.

- **Chapter 3** goes through various customer requests received by the support team and explains how they are resolved.

- **Chapter 4** proposes two AI solutions, each tailored for different types of cases.

- **Chapter 5** presents the pilot implementation using Azure AI studio. Provides a starting point for future projects.

- **Chapter 6** outlines a framework for future research and introduces two other ongoing AI projects within the case company.

- **Chapter 7** concludes the study and present the results and gives suggestions for future research.

2 Theoretical background

This chapter covers the key concepts related to artificial intelligence (AI) and its subfields. It will also introduce an architecture that is the foundation of today's large language models.

2.1 Artificial Intelligence (AI)

Artificial Intelligence (AI) has gained a lot of popularity in recent years. The first obvious sign of this was the launch of ChatGPT in 2022 which made conversational AI accessible to everyday users. Since then, models have kept improving and competition in the AI sector has increased. While the models keep getting larger and smarter, they are constantly getting more use cases from various business fields.

The European Commission (2018) defines AI as "systems that display intelligent behavior by analyzing their environment and taking actions – with some degree of autonomy – to achieve specific goals." This definition covers a wide range of AI systems that can act smartly, understand their environment, and work on their own to achieve specific goals (European Commission, 2018). While the definition from the European Commission is accurate it can be viewed as too broad and general. Examining the definition from different perspectives can improve our understanding.

Russell and Norvig (2020, p. 1-5), have outlined two primary approaches for exploring AI systems and their functionalities. In the human-centric approach, the aim is to replicate human intelligence when creating an AI system. This is done, by studying human behavior and cognitive processes primarily based on psychological principles. The human-centric approach relies heavily on empirical studies in psychology.

The rationalist approach emphasizes logical decision-making to achieve goals, without human resemblance. It prioritizes intelligent behavior and rationalism as the main characteristics. This perspective integrates mathematical and engineering principles from fields like statistics and economics.

2.2 Machine learning

Machine learning is a subset of artificial intelligence. According to Muhammad and Yan (2015), It enables machines to learn from experience and improve its models over time as more training data becomes available.

Machine learning (ML) algorithm's main function is trying to analyze data, identify patterns, and determine key predictors without manual intervention. Unlike traditional database systems, which are designed for data storage and retrieval, ML algorithms can extract meaningful insights from data (Muhammad & Yan, 2015).

Muhammad and Yan (2015) further explain that the most common approaches to machine learning include supervised, unsupervised, and reinforcement learning. Each method offers pros and cons when it comes to pattern recognition and decision-making. Deciding which method to use will depend on the intended use case of the model.

2.2.1 Different Types of ML

Supervised Learning

Supervised learning, explained by Shetty et al. (2022), involves training algorithms on labeled datasets where every value is paired with a known outcome. This allows the algorithm to learn the relationships between input values and their corresponding labels. Learned patterns from the labeled datasets can then be used to make predictions on new, unlabeled datasets.

Supervised learning was divided into two main categories in the study:

- **Classification:** Categorizes data into different classes or groups. For example, deciding if an email should be marked as spam or not.
- **Regression:** Predicts a continuous values based on given inputs. Could be used to do weather forecasts or estimate stock prices.

Unsupervised Learning

Unsupervised learning algorithms use training data, where the input values have no predefined output values (Niranjani et al., 2020). Instead, algorithms are trained to group and identify the data independently (Naeem et al., 2023). By doing so, the hidden patterns and structures can be found from the datasets.

Unsupervised learning includes:

- **Clustering:** Grouping similar data points together. For example, customers with similar purchasing behaviors (Naeem et al., 2023).
- **Association:** Identifying connections or patterns between data points. For example, customers who buy product X will most likely buy product Y as well (Naeem et al., 2023).

Reinforcement Learning

Reinforcement learning uses a method called trial-and-error. Instead of training the model with labeled datasets where the desired result is predefined, the agent receives feedback for every action. According to Puiutta and Veith (2020), the designer of the task sets different reward values for each action on the model. Random actions are then tried against the dataset. As the agent continues to interact with the data, it learns from the feedback received through these rewards.

Reinforcement Learning is widely used in sectors like gaming, transportation, and robotics for its ability to adapt to dynamic environments and optimize decision-making processes in real time (Puiutta & Veith, 2020).

2.3 Deep Learning and Neural Networks

Deep learning, as defined by Deng et al. (2012), is a subset of machine learning focused on pattern recognition from raw data without the need for manual feature engineering.

Central to deep learning are Artificial Neural Networks (ANNs), which mimic the structure and function of the human brain. According to Dongare et al. (2012, p. 190), these networks are composed of interconnected neurons organized into input, hidden, and output layers. This structure is fundamental to all specialized neural network models.

Deng et al. (2012) also note that advancements in hardware, the availability of large datasets, and improvements in training techniques have led to the development of more advanced and complex neural networks. This progress has significantly impacted various fields.

2.3.1 Deep Learning Architectures

Convolutional Neural Networks; Image processing

Convolutional Neural Networks (CNNs) are specialized for recognizing patterns in images, making them useful for tasks like image classification and object detection.

CNNs work through several key parts. The input layer receives the raw pixel data of an image. The convolutional layer scans the image with filters to detect patterns and features. The pooling layer reduces the size of the data, decreasing the computational workload. Finally, the fully connected layers use the identified features to determine what the image represents (Saxena et al., 2022).

Self-Organizing Maps and Autoencoders; Data Compression

Self-organizing maps (SOMs) and Autoencoders are algorithms used for data compression. They both turn complex, high-dimensional data into simpler, lower-dimensional forms while maintaining the key features (Qu et al., 2021).

SOMs assist in visualizing and identifying patterns in complex data by converting it into a more manageable format. Qu et al. (2021) state that SOMs have two main parts: an input layer and a competition layer. The input layer receives the data, and the competition layer groups it based on feature similarities.

Autoencoders are designed to learn informative representations from the compressed data by reconstructing it. According to Bank et al. (2023), Autoencoders have three main components: an encoder, a latent feature representation, and a decoder. Encoder compresses the input data into a smaller format known as the latent feature representation. This representation then captures the key features of the data. Finally, the decoder reconstructs the data into its original form (Bank et al., 2023).

For practical applications, such as recognizing handwritten digits, an autoencoder focuses on capturing key features like the number of strokes and their orientations, rather than the exact pixel values. Michelucci (2022) explains that this approach results in a compressed yet meaningful representation of the data. As a result, the autoencoder can better group and identify the data, making it easier to understand the main features of the dataset.

Recurrent Neural Networks; Sequential data

Recurrent Neural Networks (RNNs) are designed to handle sequential data by maintaining connections that pass information from one step to the next (Salehinejad et al., 2017). Unlike traditional feedforward neural networks, which process inputs independently, RNNs capture patterns and dependencies over time by using a hidden state that acts as memory. According to Salehinejad et al. (2017), this makes them ideal for tasks where the order and context of data points are crucial.

In an RNN, each input is processed in sequence, one step at a time. As Das et al. (2023) explain, the network uses the current input along with the hidden state from the previous step to compute a new hidden state. This new hidden state influences the current output, enabling the network to dynamically understand the sequence and capture

patterns over time. This ability to retain information from previous steps allows RNNs to understand and predict sequential information accurately (Das et al., 2023).

Lipton et al. (2015) highlight that one significant challenge in training RNNs is learning long-range dependencies due to issues like vanishing and exploding gradients (see page 26). However, new designs and architectures have been introduced to help retain important information over longer sequences.

2.4 Transformer Architecture

Early natural language processing (NLP) models relied on Recurrent Neural Network architectures. However, adjusting the model's parameters correctly, especially with longer sequences posed challenges. These difficulties in the training process led to the development of improved recurrent networks like Long Short-Term Memory (LSTM) and Gated Recurrent Unit (GRU) networks. Although these models addressed some issues, they still struggled with longer sequences (Singh & Mahmood, 2021).

The breakthrough came with the Transformer architecture introduced by Vaswani et al. (2017), which overcame the limitations of RNNs. This mechanism allowed the model to process inputs in parallel without sequential dependency, significantly improving the performance.

To achieve this, transformers use self-attention layers to relate different words within the same sequence. Unlike in traditional methods, self-attention does not require the model to handle processes in fixed order. This eliminates the need for complex operations as the distance between words increases (Antipova & Horban, 2024).

To allow the model to process the inputs in parallel, layers are stacked together in a structure known as multi-head attention (Shin et al., 2022, p. 4). Additional components such as feed-forward layers, residual connections, and normalization layers are also included to capture more complex patterns and ensure stable training (Rahali & Akhloufi, 2023).

The Generative Pre-Trained Transformer (GPT) models are built using transformer architecture. This allows GPT models to perform tasks and hold conversations in ways that closely resemble human communication (Yenduri, 2023). Transformer models are the foundation for large language models used in tools like ChatGPT and MS Copilot.

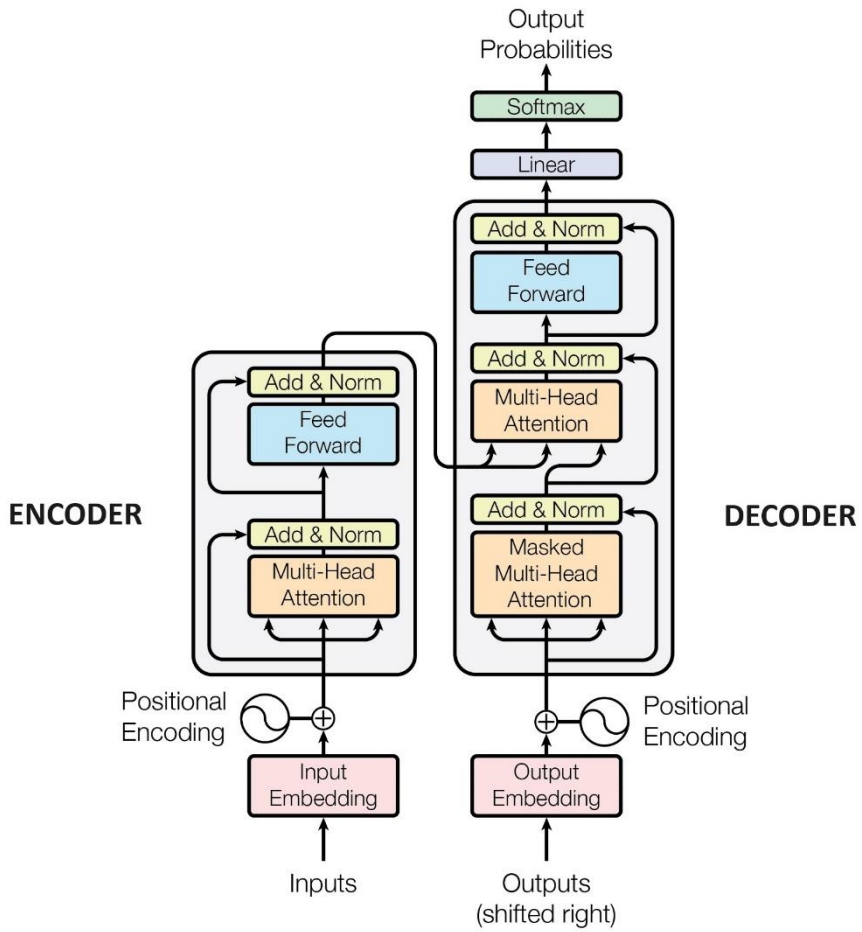

Figure 1 Transformer architecture (Vaswani et al., 2017).

2.5 Encoder-Decoder structure

The transformer model consists of two primary components: the encoder and the decoder. These components are used in sequence-to-sequence tasks such as language translation and conversational AI.

The encoder converts the input sequence into a numerical format, which is then refined to capture important details and context. The decoder then transforms this numerical data into a meaningful output (Chitty-Venkata et al., 2023, p. 4). The combination of these modules allows the Transformer to effectively manage complex tasks that require a deep understanding of context and sequential relationships.

2.5.1 Encoder

Word Embedding

Word embedding is a technique in natural language processing (NLP) that converts words into vector representations. By analyzing large amounts of text, embedding models learn to position words with similar meanings and contexts close to each other in the vector space. This helps models comprehend the relationships between words, leading to an improved ability to understand language (Madaan et al., 2024, p. 1388-1389).

To give an example, consider a sentence with the words “Gmail,” “Google,” and “Microsoft.” The resulting vector from this combination would be close to the vector of “Outlook.” This demonstrates the ability of NLP systems to understand analogies, such as “Gmail is to Google as Outlook is to Microsoft.” While this is a simplified example drawn from the study conducted by Di Gennaro et al. (2021, p. 2), the same concept applies to more complex tasks.

Positional Encoding

Similarly to the word embedding, positional encoding is used to improve the models' ability to understand sequences better. According to Li et al. (2021), positional encoding is used to capture the positional information of data, such as the order of words in a sentence or pixel coordinates in an image. In combination with content representation (word embeddings), positional encoding improves the representation of inputs.

The original Transformer model used absolute positional encoding to indicate the position of each word in a sequence. It used fixed sine and cosine functions to create unique position vectors. Rosendahl et al. (2019) state that while this approach was simple and effective, it had limitations in capturing the distances between positions.

To address these limitations, Rosendahl et al. (2019) described relative positional encoding, which focuses on the distances between words rather than their absolute positions. By using trainable distance encodings, this method allows the model to better understand and adapt to the positions in the sequence.

The study conducted by Dufter et al. (2022), highlights the significance of positional information in defining the meaning of a sentence. For example, "meeting in Teams" clearly refers to a virtual meeting on Microsoft Teams, while "Teams in meeting" has a completely different meaning. This demonstrates the critical role of positional data in accurate presentation of information.

Self-Attention Mechanism

The self-attention mechanism is a key part of modern language models. It allows the models to focus on the most relevant parts of the input data, which is crucial when handling sequential data with long-term dependencies.

Choi and Lee (2023, p. 3-4) explain how this mechanism improves the model's capability to handle complex sequences. Unlike traditional methods that process data in a strict order, self-attention allows each element in a sequence to interact with every other element in the same sequence. This approach gives the model a better ability to capture relationships between elements, regardless of how distant they are.

The core purpose of self-attention is to figure out how much attention to place on different parts of the input when generating the output. Since self-attention alone doesn't account for the order of elements, positional encoding is added to fill this gap.

To implement self-attention, the scaled dot-product attention mechanism is used. As detailed by Choi and Lee (2023, p. 4), this mechanism calculates attention scores to determine the importance of each element in the input sequence relative to other elements. It has three main components: the query (Q), the keys (K), and the values (V). The provided image shows how these elements interact.

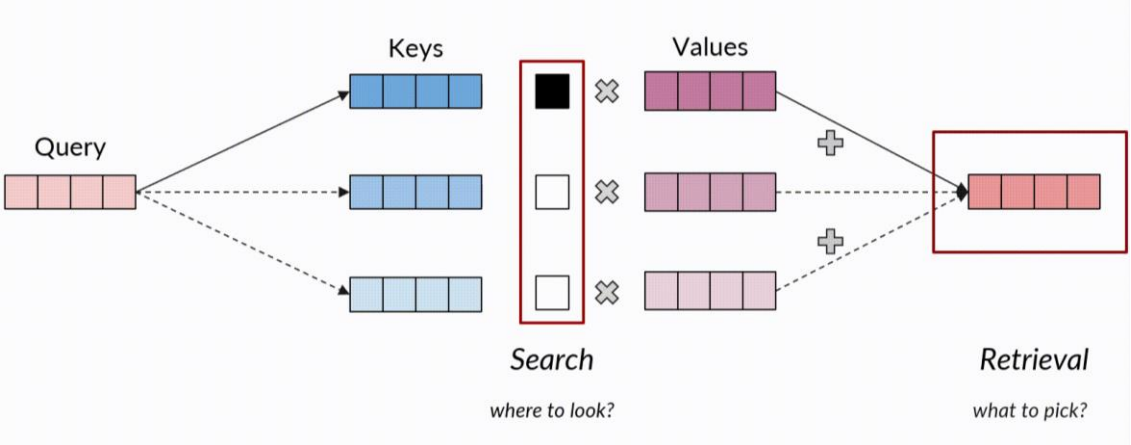

Figure 2 Scaled dot-product attention (Mittal, 2022).

Definitions

- Query (Q)**:** The information currently being processed by the model.
- Key (K): Elements compared against the query to determine their significance.
- Value (V): Final output that is selected based on the attention score.

The process starts with a search phase, where the model compares the query to each key to calculate relevance scores, known as weights. These weights serve as indicators of the importance that each key has in relation to the query. Following the computation of weights, the model enters the retrieval phase. Here, it extracts the information associated with the keys that have the highest weights. This extracted information is referred to as the value (Choi & Lee, 2023, p. 4).

To understand the self-attention mechanism better, consider an online shopping scenario presented by Lezmi and Xu (2023). When users search for desired products, they input preferences such as brand, price, and features into the search engine. The platform matches these preferences with the basic details of all available products and returns the ones that match the user's criteria. Here, the user's preferences function as the query (Q). Basic information of each product is the key (K), and the products themselves are values (V).

Just as an online shopping platform utilize the user preferences to find the most relevant products, transformer models use the attention mechanism to measure the similarity between the query and key. By focusing on the keys with the highest weights, the model can accurately identify and retrieve the most relevant values. This allows the model to understand the input data more precisely (Choi & Lee, 2023, p. 4).

Multi-Head Attention Mechanism

To improve the model's capacity to focus on different parts of the input simultaneously, the transformer uses a multi-head attention mechanism. This involves running multiple self-attention processes (heads) in parallel, each with different learned projections of the queries, keys, and values. The outputs of these parallel processes are then combined and linearly transformed to produce the final output (Shin et al., 2022).

Layer Normalization and Residual Connections

When training transformer models, it is important to have a correct balance on how much certain training data affects the model's performance. Borawar and Kaur (2023) highlight that gradients indicate the ratio of how much the weights are adjusted during training. If they become too small (vanishing gradients) the model becomes inaccurate, because the weights are not adjusted enough. Another issue is gradients becoming too big (exploding gradients). This causes the weights to be updated too much, making the training unstable.

To address these issues, two key techniques are used: residual connections and layer normalization. Nguyen et al. (2019) explain that residual connections work by adding the original input of the self-attention layer to its output. This helps the network to retain information from previous layers and simplifies the learning process.

After adding the residual connection, the combined values are normalized. Layer normalization adjusts the values, so they have a consistent scale and distribution. The normalized result is then passed on to the next layer. Together, residual connections with layer normalization maintain stability and effectiveness throughout training. (Nguyen et al., 2019).

Feedforward

Following the normalization, the feedforward network (FFN) is introduced. The FFN is designed to identify various patterns across different layers. In the lower layers, it learns simple patterns, while in the higher layers, it captures more complex ones (Pires et al., 2023).

While self-attention layers connect different parts of the input broadly, helping the model to see the big picture. The feedforward network focuses on specific details, improving the understanding of local contexts. By adjusting the way information is represented, the feedforward network makes sure the outputs are more accurate and relevant to the context (Pires et al., 2023).

2.5.2 Decoder

The decoder in the Transformer architecture shares a very similar structure as the encoder but with a few key differences. While the encoder processes the entire input sequence at once, the decoder generates the output sequence one token at a time in an auto-regressive manner (Greco & Tagarelli, 2023). This allows the decoder to use previously generated tokens to influence the next ones. This is a key feature for conversational AI.

The decoder has several components that are also found in the encoder:

- Word Embedding: Converts words into vectors that reflect their meanings.
- Positional Encoding: Adds positional details to the vectors.
- Residual Connections: Stabilizes the model by keeping track of the original values.
- Layer Normalization: Normalizes the values to keep a consistent scale.
- Feed-Forward Network: Applies non-linearity to capture complex patterns.

Masked Multi-Head Attention

The masked multi-head attention mechanism in the decoder prevents future positions in the sequence from being accessed, ensuring that predictions are made sequentially. It operates similarly to the encoder's multi-head attention but includes a masking mechanism that ensures the model only attends to earlier positions in the output sequence. This approach is crucial for maintaining the auto-regressive property of the decoder, where each token is predicted based on the previously generated tokens (Greco & Tagarelli, 2023).

Cross-Attention

Cross-attention in the decoder integrates the encoder’s output with the decoder’s context. This mechanism uses the encoder’s output as (keys and values), while the output from the decoder’s masked attention serves as the (query). By doing so, cross-attention enables the decoder to utilize the encoded information from the input sequence, which is essential for the model to generate intelligent responses (Chitty-Venkata et al., 2023).

3 Work Processes in Customer Support

Responsibilities of customer support specialists include assisting sales units with issues reported by their customers. Direct communication with customers is not part of the role. Instead, the sales units act as a link between customer support team and the customers.

Salesforce is used as the ticketing system for managing customer inquiries. When a customer has questions about their order, they reach out to their respective sales unit. If sales unit is not able to solve the issue by themselves, they will open a ticket to Salesforce, where customer support specialist can help in solving the problem.

Relevant details about the customers inquiry are inputted by the sales unit into Salesforce. This includes the nature of the problem, any helpful context, and the order number so the ticket can be connected with the correct motors. The name of the sales personnel who opened the case is also entered, alongside with resolution deadline, and the case category.

Selecting the correct category is important as it determines which queue the ticket will go to. Each Salesforce queue is managed by a specialist based on their expertise. Once the specialist has taken the case under their name, a structured approach is followed to resolve the issue. While every case type has unique resolution methods, some of the work processes are very similar.

Solving the cases begins by reading the request from Salesforce. After the issue is understood, the provided order number is copied and entered into the Enterprise Resource Planning system (ERP), provided by SAP SE. This system allows specialists to track order-related details such as technical specifications, production status, and delivery details. After the requested task is resolved, the specialist will write a message to the sales unit indicating what was done and if any issues arise. This message is automatically sent via salesforce and email, when the ticket is closed.

3.1 Research data used in this work

Research data was gathered from the customer support ticketing system Salesforce. Cases were collected from three-month time period between 1.4.2024 - 30.6.2024 and in total 2,875 requests were received. The distribution of the Support tickets in different queues can be seen in Figure 3.

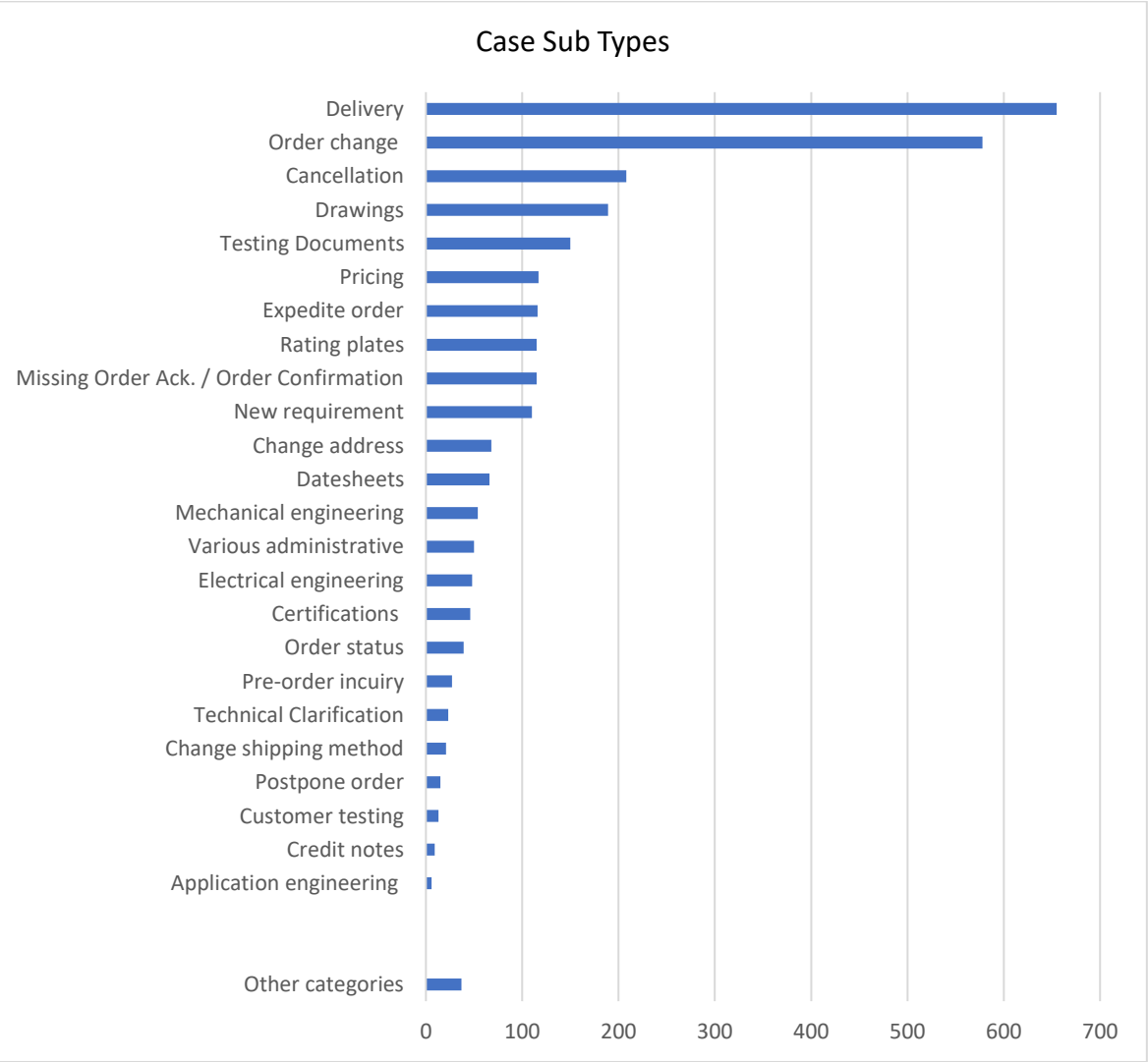

Figure 3 Case Sub Types

After organizing the Salesforce data based on case subtypes, a deeper analysis was conducted. The goal was to find a more accurate representation of the requests to understand common issues faced by customers better.

The analysis showed that many similar requests were spread across different queues. Some of the original queues were also too broad to accurately describe what was asked in the cases. By placing these cases into a more detailed groups, a clearer picture of the common issues was achieved. Figure 4 shows the new case groups.

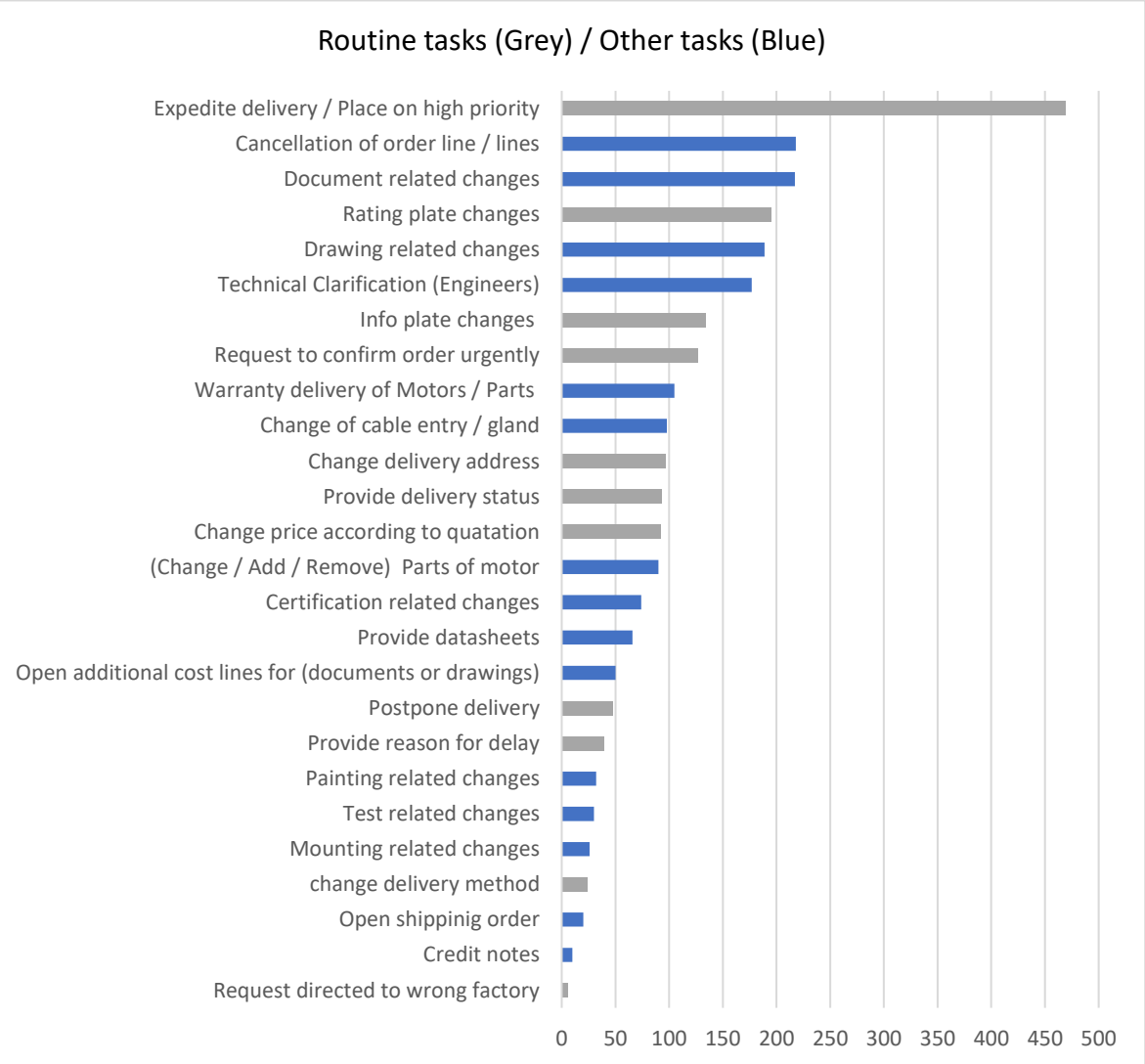

Figure 4 Requests

Cases were then divided into two groups based on their resolution methods. Cases marked with grey color are cases that can be resolved faster with a smaller amount of documentation and investigation. Cases marked with blue are cases that usually take more effort because the suitability of the change must be verified, and costs calculated. These same groups will also be used in Chapter 4 to describe AI solutions suitable for each approach.

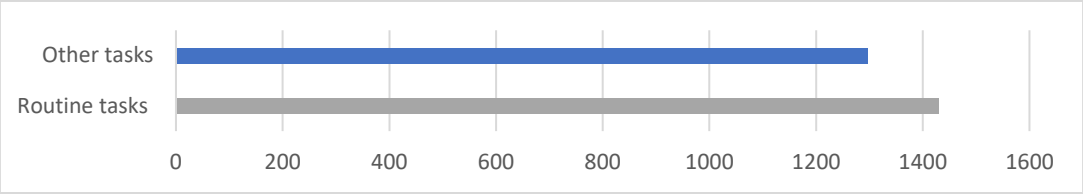

Figure 5 Case Resolution Types

3.2 Routine tasks, Group 1

Cases that follow consistent resolution patterns are categorized as routine tasks in this chapter. Resolving these tasks doesn't usually require the use of additional data sources. Because most of the needed information is found solely from Salesforce and SAP, the cases can be handled faster than other cases. In Figure 5, routine tasks are highlighted in grey.

Delivery

Delivery cases cover the largest portion of routine tasks. The most common request in these cases is to expedite the delivery dates. Whether the dates can be changed depends on when the request is made. If the production schedule is still far away, changes are easy to make. However, requests often come in late, even after the production has already started. In such cases, rescheduling is more challenging.

There are also situations where customers ask to postpone the delivery dates. This can usually be arranged, but once the production has begun, additional storage fees might be charged.

In addition to these, delivery cases may also involve requests to provide order status updates, modify delivery terms and methods, or explain reasons for order delays.

Plates

Another common category involves updating plate information. These are opened when customers either forget to include plate details or provide incorrect ones.

To make the change specialist copies the plate details provided in Salesforce and updates it into SAP. The cost for the change is also easy to determine based on the production status. Because any other validations are not needed these changes can be done rather quickly. However, if the plate information has to be changed to multiple orders the manual effort can make the change time-consuming.

Pricing

In pricing cases customers want certainties that agreed price quotations have been applied to their orders. To detect any pricing mismatches, the specialist compares the prices mentioned in the quotations to those listed in SAP. This ensures that pricing is accurate and helps avoid any possible billing errors.

3.3 Other tasks, Group 2

While routine tasks primarily relied on information from Salesforce and SAP, resolutions of cases discussed in this section might require additional data sources or collaboration with engineers. Although these cases often involve a deeper level of investigation, experienced specialists may have had similar cases previously, therefore recalling the resolution without documentation.

Technical modifications

Technical modifications require the specialist to verify that the requested changes meet the technical requirements of the motors. The process includes reviewing product details and specifications in SAP and cross-referencing those to the documentation. For more complex inquiries, specialists may need to consult engineers to verify if the changes are feasible. Technical modifications can include adjustments to how a motor is mounted, modifications to parts, changes to cable entries/glands, or painting the motors with different colors or durability classifications.

Cancellations

Handling cancellations varies in complexity based on the order's production status.
For cancellations requested before production, the case can be resolved with a straightforward change in SAP. However, if requested after the production has started it becomes more complex. In these situations, the specialist must calculate cancellation costs and issue billing statements in addition to updating the order status.

Documents

Customer requests for documentation can include product drawings, certifications, test reports, and datasheets. When handling these requests, the customer support specialist first identifies the correct Variant code associated with the requested document. This code ensures that the appropriate document is retrieved from the system.

- Product drawings (2D or 3D), provide detailed visual representations of the products. Those are used to get a better understanding of the product's design.
- Certifications confirm that products meet industry standards and regulatory requirements.
- Test reports detail the results of motor testing, verifying performance and reliability.
- Datasheets contain technical information about the product's specifications and features.

Warranty

Managing warranty shipments involves verifying details from Salesforce and working with carrier firms to ensure that replacement parts or motors are shipped according to agreements. When managing logistics, communication with the customer throughout this process is important.

4 Proposed AI solutions for Customer Support

AI has the potential to improve efficiency across various work processes in customer support. By making resources more accessible, response times can be improved. Training advanced AI models with a company's internal data can lead to significant long-term benefits. When AI models are trained with internal data, they can perform several key functions better.

This section will discuss two different approaches for utilizing AI in customer support. Each group introduced in the previous chapter will be presented with a suggested AI solution, based on their resolution methods.

4.1 Instructions for Routine Tasks

Implementing AI solutions using order-related data from SAP systems is challenging due to the dynamic nature of the data. Training large AI models with constantly changing data would be impractical. A better solution for these cases would be to train an AI model to give guidance to the specialist. Rather than accessing the data directly, the chatbot would assist users by guiding them on how to find the necessary information themselves. This approach simplifies the process and reduces the need for constant updates to the AI model. Routine tasks include requests such as:

Delivery cases

- Expedite delivery dates
- Place the order on high priority
- Confirm the order urgently
- Change delivery address
- Provide delivery status
- Postpone delivery dates
- Provide a reason for the delay
- Change delivery method

Plates

- Changes to the texts of tag plates or additional data plates
- Modifications to the values in the rating plates

Pricing

- Confirm that the price is according to the provided quotation
- Open credit notes

An approach where a chatbot gives guidance on how to resolve cases, would be helpful, especially for new employees who are still learning the work processes. Chatbot could provide consistent instructions, making resolving cases easier. It would guide users step-by-step through specific SAP transactions, helping them retrieve the needed information without directly interacting with the dynamic data. This method not only simplifies the learning process but also helps with retaining knowledge in the case company.

To make the chatbot work, customer support specialists would play an important role in the training process. This would include documenting detailed steps and guidelines for various scenarios. By sharing their knowledge and insights, the chatbot could provide accurate guidance for resolving routine tasks.

Gathering the needed information for the model would be easiest to do during the orientation period for new trainees. This way extra work would be reduced as steps of each task are repeated either way. All the existing orientation material should be analyzed before collecting new data to avoid doing the work twice.

To further refine the model, the context of the resolution steps needs to be connected with the actual customer requests. This will make the chatbot more practical as it can understand less precise and detailed inputs.

With this integration, users can input support tickets directly into the chatbot. The AI model would recognize the relevant guidelines stored earlier and provide immediate assistance, eliminating the need for users to figure out the steps themselves. Relevant information for this approach would include:

Customer Support Specialist's input

- **Resolution methods:** Provide information on the best practices to solve different requests. Connect the guidance with the proper category (Figure 4).
- **Locate information:** Document the steps involved to find needed information from SAP transactions.

Salesforce cases

- **Case description:** Contains the customer's request for what needs to be changed in the order.
- **Categories (Figure 4):** Cases must be properly categorized so the model can match requests with accurate resolution guidance.

4.2 Information Retrieval for Other Tasks

When resolving technical modifications, it often requires the specialist to navigate a variety of resources. Unlike order-related data, this information is more stable. Because the data does not require frequent updates, it is more suitable for AI integration. Typical requests that can utilize this documentation include.

Technical Adjustments

- Modifications to cable entry or gland sizes
- Alterations to components in the motor
- Changes in the mounting position of the motor
- Clarifications regarding technical specifications
- Adjustments to painting systems or surface colors

Documentation Provision

- Providing product drawings (2D or 3D) for detailed visual representations of the products.
- Supplying certifications to confirm compliance with industry standards and regulations
- Issuing test reports to verify the performance and reliability of the motors.
- Delivering datasheets containing data about motor's performance and features

Order Cancellations

- Cancel the order and issue an invoice for any incurred expenses.

Warranty

- Coordinate with couriers to ship replacement parts or motors as per warranty agreements.

To resolve these tasks variety of different documentation can be utilized. One of the most frequently used resources is the product catalogs of each motor type. Catalogs contain detailed information about the products the company offers, including mechanical specifications, performance metrics, and IP ratings. It also features images and diagrams to help visualize the product. Because the case company is specialized in highly customizable products, the catalog also gives the options for different modifications.

Another important data source for specialists is the variant code descriptions. They work as identifiers used to specify unique configurations and features of a product. These codes simplify the process of detailing modifications, ensuring precise customization of the products. Each of these codes have a documentation that describes what they are used for.

In addition to the documentation that specifies features related to the motors, excel files also hold vital information. They have information about costs regarding modifications and cancellations. These files also have formulas that are used to evaluate costs and delivery impacts in different situations. Using those makes resolving cases easier and human error in calculations can be reduced.

Specialists need to retrieve information from these sources often, when doing technical modifications. Because the documentation is poorly organized, locating the correct files can be challenging and time-consuming.

When files are not organized well, it can cause the specialists to skip checking the documents altogether and rely solely on their memory and experience. This may lead to mistakes and addressing those later can be difficult. It is also common to consult engineers with technical questions, that the specialist could have found from the documentation themselves. While it is a good practice to validate the changes from engineers, simpler modifications should not be their priority.

To make the retrieval of this information easier and faster, an AI model could be trained with this data. By providing the customer inquiry and product details, the user can ask the chatbot how to resolve the issue. Response from the chatbot might give the user a straight answer to the problem or lead them to the correct documentation. Relevant documentation for training the model would include.

Product Catalog Details

- **Product specifications:** Detailed information on mechanical features and performance metrics of specific motor types and sizes.
- **Modification options:** A comprehensive list of customization possibilities.

Variant Code Documentation

- **Code Explanations:** Clear descriptions of each variant code and their impact on product configuration, enabling accurate customization guidance.

Excel Files

- **Cost evaluation formulas:** Formulas for calculating costs associated with modifications and cancellations.
- **Delivery impact formulas:** Formulas to evaluate delivery delays in different situations.

In addition to this, the same process as in the routine tasks should be done with the salesforce tickets. By gathering enough customer request data from salesforce and categorizing those the model could understand the context better.

The following chapter will showcase a pilot implementation involving a small subset of order change tasks. If successful, the approach can be expanded to cover the broader range of customer requests mentioned in this chapter.

5 Pilot Implementation

For this thesis, the goal was to develop a pilot version of an AI model to demonstrate the potential of using Azure AI Studio for larger-scale implementations in the future. Implemented chatbot would assist specialists to solve cases related to cable entry and gland changes.

From the 2,875 cases received by the customer support division in three months, 98 involved cable entry and gland changes, representing 3.4% of the total cases.

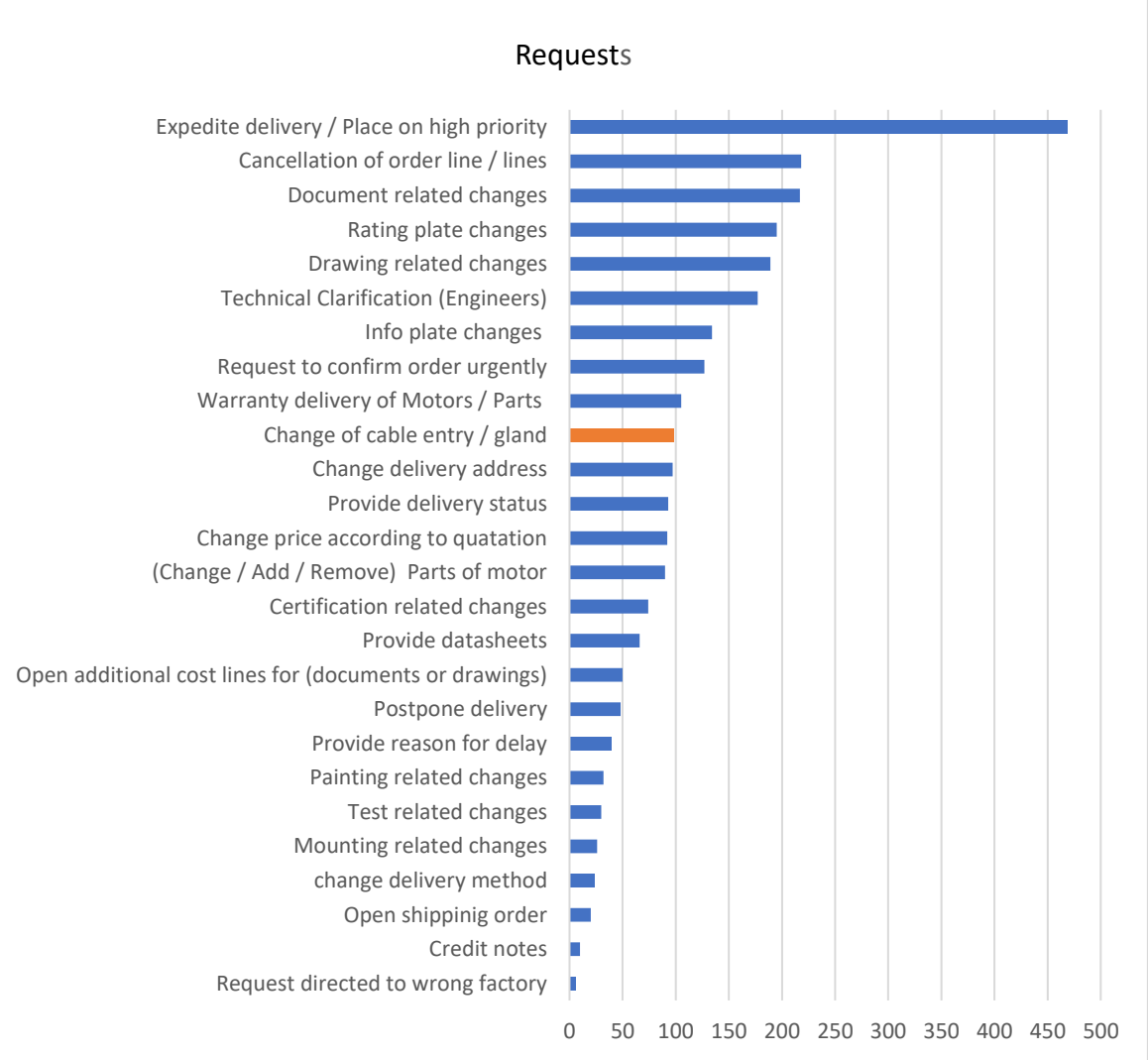

Figure 6 Scope of Pilot Implementation

5.1 Purpose

The case company manufactures motors that include terminal boxes for cable connections. These boxes have flanges where cable entry holes are drilled. Sizes and quantities of entry holes vary based on the product type and frame size. It is common for customers to request custom cable entry combinations when standard solutions do not meet their needs.

Changes are straightforward when the required entries are smaller than the standard ones and when the quantity of the entries does not increase. However, when the sizes or quantity of cable entries increase, feasibility needs to be verified. To make sure the solution is valid, specialists must search information from documentation. In some instances, engineers may need to be consulted to verify, if certain entry holes can be fitted in the flanges of particular motors.

This work process was selected for pilot implementation because the data for model training is stable and will not change in the near future. Although all the necessary information is available, it has not been easily accessible for specialists to utilize.

By making the data easier to access, the workflow of specialists can be made faster and more efficient. Engineers will also save valuable time when they don’t have to validate cable entry combinations.

5.2 Data Selection

Every product type has a catalog that contains all the necessary technical data of the motors. All of these catalogs were used as training data for the model.

AI models typically perform more effectively when working with text-based information that has a clear context. However, when the catalog data was uploaded to the model, it became clear that the model was not able to fully utilize this data. This challenge occurred because the technical data primarily consisted of numerical values, presented in data tables. When the chatbot was prompted to retrieve specific values from these tables, the results were inconsistent and unreliable.

Because the training data in this implementation is not well-structured and has no predefined outputs, unsupervised learning algorithms are used. This approach enables the model to identify relevant patterns and understand the data better (Niranjani et al., 2020; Naeem et al., 2023).

Standard Cable Entry Sizes

To help the model find these patterns, information regarding cable entries was gathered from all 15 product catalogs. The extracted data was placed into a single file named "Standard Cable Entry Sizes". This file includes details on four aspects: motor type, frame size, flange type, and standard cable entry sizes.

The data was organized and categorized by motor types and sizes, with corresponding flange types and standard sizes presented in a clearly formatted table. The purpose of this was to make the information more accessible and understandable for the AI model, which previously struggled with data tables. Figure 7 shows the information of one motor as an example. “Standard cable entries” file has tables for 15 motor types.

3GBL: Totally enclosed fan-cooled synchronous reluctance motor with cast iron frame		
Size	**Flange type**	**Terminal box standard cable entries**
160	B	2xM40
180	B	2xM40
200	C	2xM63
225	C	2xM63
250	C	2xM63
280	C	2xM63
315SM	D	2xM63
315ML	D	2xM63
315LKA	E	2xM75
315LKC	E	2xM75

Figure 7 Standard cable entries

Maximum cable entry size combinations per flange type

While the "Standard Cable Entry Sizes" file helped the model to read standard cable entry solutions, it didn't address scenarios where customers require custom cable entry sizes. To find the maximum cable entry sizes, a detailed investigation was carried out.

Mechanical engineers were first consulted to give their insights on what cable entry solutions are possible. With their help, old drawings from the internal database were found. These drawings had information about cable entries that were designed for each flange type in the past. In addition to this, previous modifications from Salesforce were reviewed to get a better understanding of possible solutions.

With this investigation, enough information was gathered to determine what custom cable entries are possible for each flange type. This information is documented in the file titled "Maximum Cable Entry Size Combinations per Flange Type." While this file only mentions the maximum sizes, it can be assumed that smaller sizes can be fitted if the larger ones are possible.

Flange Type	2 Holes	3 Holes	4 Holes	5 Holes	6 Holes
B-flange	2xM40	1xM40 + 1xM32 + 1xM20	2xM32 + 2xM20	5xM20	
		2xM32 + 1xM25			
		2xM40 + 1xM16			
C-flange	2xM63	2xM50 + 1xM20	1xM63 + 3xM20	1xM50 + 2xM32 + 2xM20	4xM25 + 2xM20
	2xM75		2xM63 + 2xM25	2xM50 + 1xM25 + 2xM20	
			2xM75 + 2xM25		
D-flange	2xM90	3xM75	2xM75 + 2xM20	4xM50 + 1xM32	4xM50 + 2xM25
			4xM63		
			3xM75 + 1xM25		
E-flange	2xM90	3xM75	4xM75	4xM63 + 1xM32	4xM63 + 2xM25

Figure 8 Custom cable entries

This file includes the maximum cable entry combinations for each flange type. Gathered information allows the model to determine custom cable entry solutions by cross-referencing with the flange types provided in the "Standard Cable Entry Sizes" file. With this new data, the model was provided with easier access to both standard and custom cable entry information.

5.3 Azure AI

After all the required training data was gathered, Azure AI Studio was used to build and train the model. Azure AI Studio provides tools and frameworks for creating custom AI solutions designed for particular requirements (Microsoft Azure, 2023). This platform allows the integration of pre-trained models, which can be fine-tuned with internal data (Brown et al., 2020).

AI model in this chapter follows the same principles as self-attention mechanisms. When the user types their question to the chatbot, their input serves as the query (Q). The training data used for the model acts as keys (K), and each key is given weights that describe the relevance in relation to the query. The model then compares the query and key values, using the scaled dot-product attention mechanism. If the model is trained accurately, the output values (V) can give useful answers to the user (Choi & Lee, 2023, p. 4; Lezmi & Xu, 2023).

The following files were uploaded into the Azure AI model:

- Catalogs for 15 motor types
- Standard Cable Entry Sizes
- Maximum Cable Entry Size Combinations per Flange Type
- The variant code description +554 - flanges modified as per the request.

Before the files were uploaded to the model, they were indexed with the Azure AI Search service. This indexing process organized the data, improving the model's ability to comprehend information. With the necessary data indexed, the next step was to upload it into the model and test its performance with different queries. Figure 9 shows how the chatbot was fine-tuned to understand the prompts better.

You are an AI assistant that helps customer support specialist to solve support tickets. Required actions:

- Make sure the prompt has all the needed information. Motor type, motor size and the requested cable entry is mandatory.
- Check “Standard cable entries.pdf” and see if the cable entry size match with Terminal box standard cable entries of the motor type and size.
- If match is not found, read the flange type from “Standard cable entries.pdf”
- Custom cable entry sizes can be found from file “Maximum cable entry size combinations per flange type.pdf”
- Give the user detailed response on the feasibility of the change and link the used sources.

Figure 9 Fine-Tuning

5.4 Testing

Chatbot’s ability to handle queries was tested through several interactions. This section summarizes the model's performance with different prompts.

Standard cable entries

Responses were accurate for simple inquiries regarding standard cable entries. The chatbot was also able to provide references so the user could verify the information.

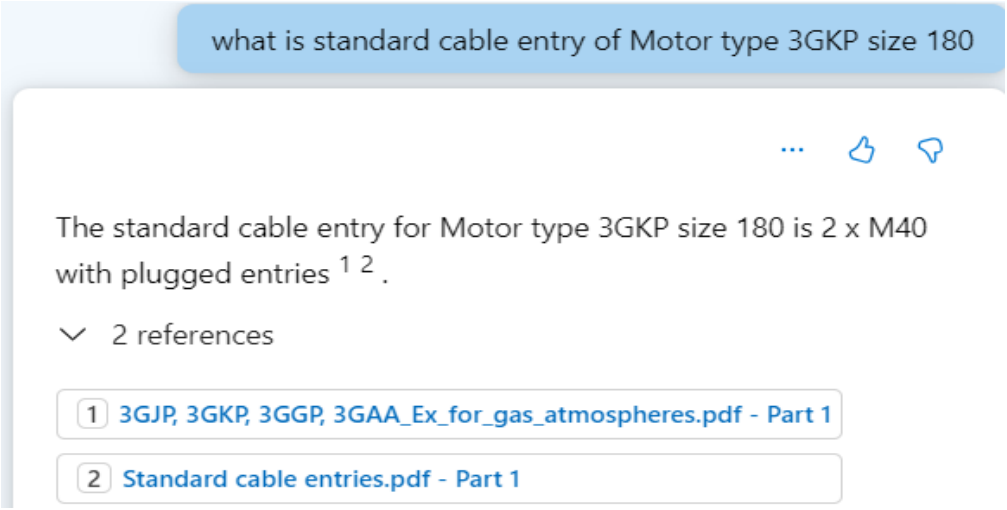

Figure 10 Test 1

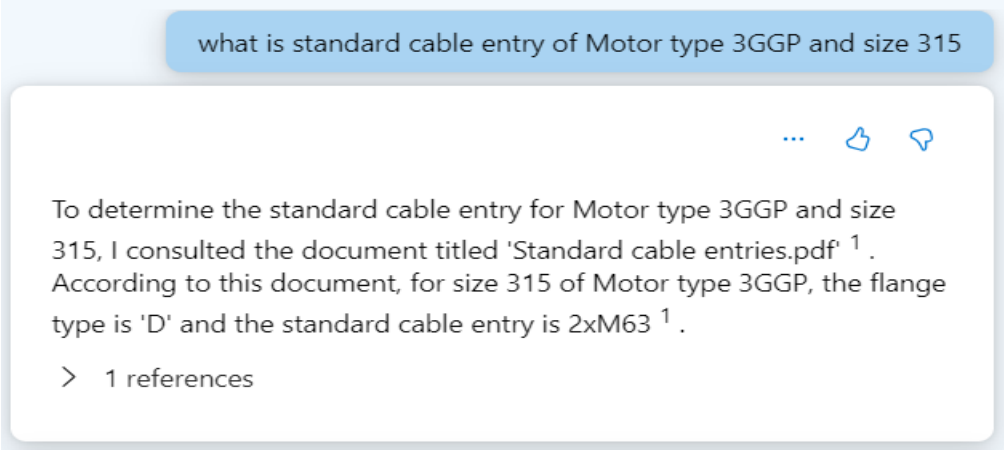

Figure 11 Test 2

Custom cable entry solution

Conversation in the figure 11 was continued with a question about the feasibility of a custom cable entry solution. The model was able to correctly read the flange type from the documentation and gather information about custom cable entry combinations

However, the response was incorrect. Reason for this was most likely the format how the information was presented in the file. Better preprocessing of the source data could fix this issue.

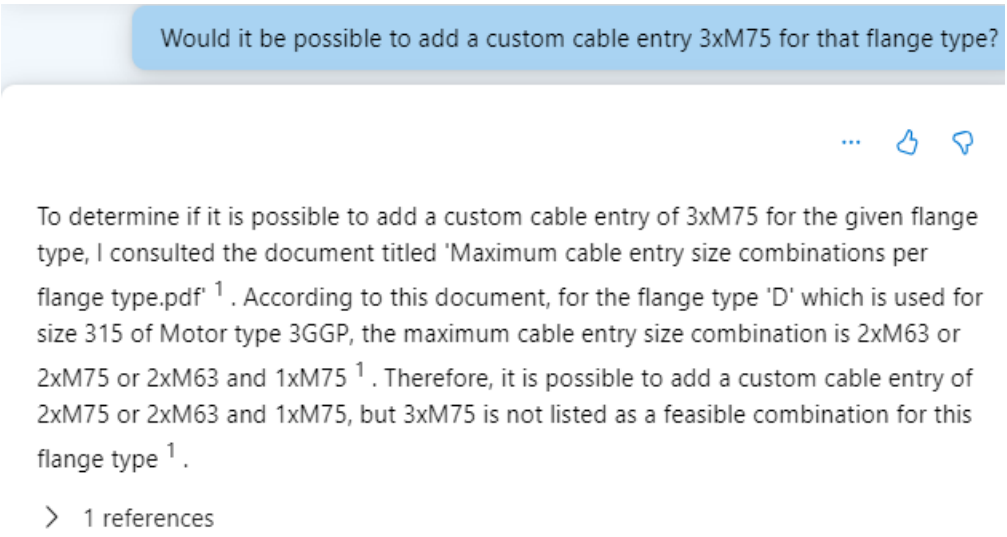

Figure 12 Test 3

Salesforce request

In the last test contents of the support tickets were directly entered into the chatbot. Responses were not useful, because the prompt did not follow the fine-tuned guidelines. While the chatbot should have given a follow-up question asking for more detail like motor type and size, it gave irrelevant information from catalog data.

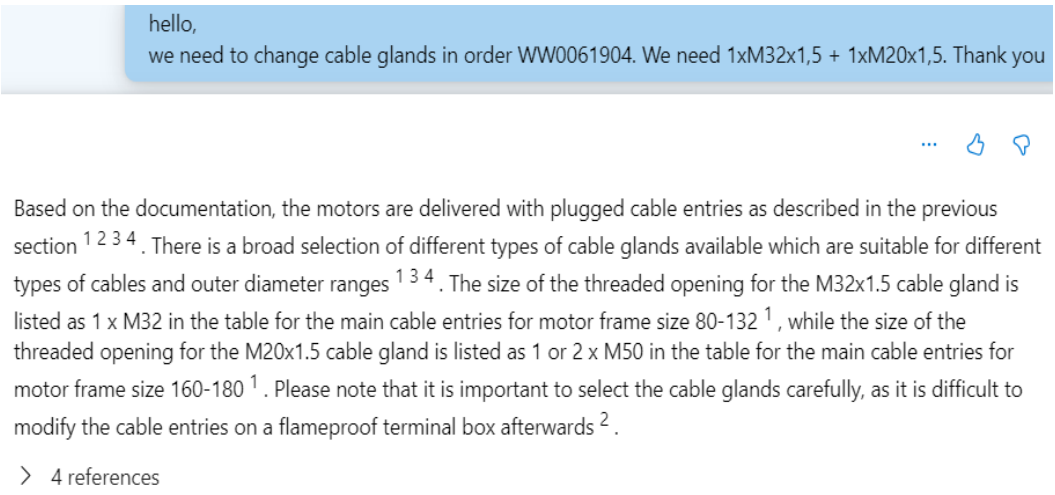

Figure 13 Test 4

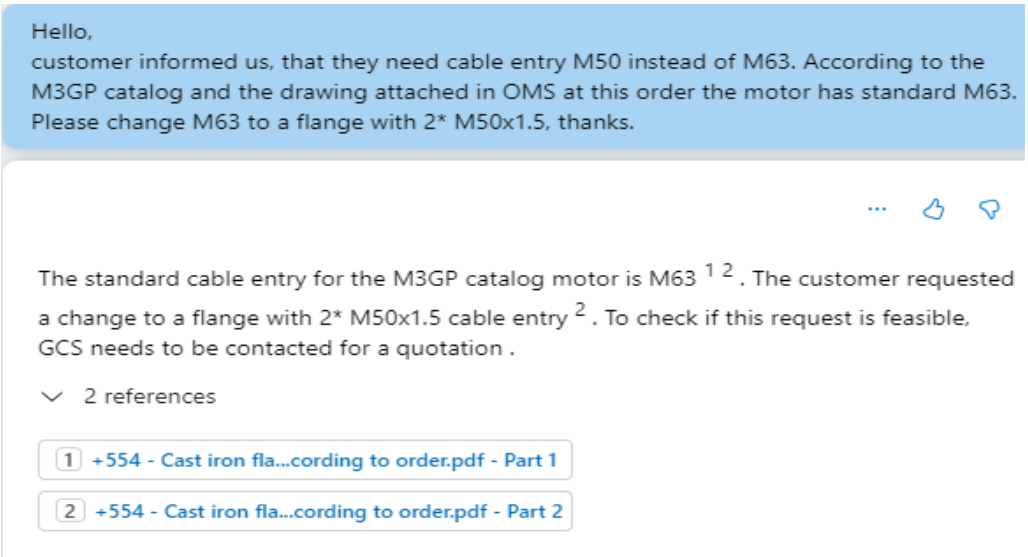

Figure 14 Test 5

5.5 Results

Developing a chatbot is a complex task that requires significant expertise and time, particularly when it comes to preprocessing and indexing the source data. The main goal of this project was to test Azure AI Studio's ability to assist customer support specialists.

The performance of the model could have been better if the training data was preprocessed before indexing. However, doing so would have extended the scope of the thesis too much. Therefore, the data was only collected to assist the model in finding information more easily.

Limitations regarding the quality of training data became apparent during testing. The chatbot struggled to provide accurate and reliable responses, as its performance was highly dependent on how well the questions were drafted.

The data indexing was managed by Azure AI Search Service, which only supported keyword and semantic searches. While these methods can be useful, they have limitations. Keyword search looks for exact word matches, and semantic search tries to understand the general meaning of phrases. With unstructured training data, these methods often fail to capture the full context of queries, resulting in less accurate results.

Better option would have been vector indexing, but due to technical issues and it was not supported. Vector indexing converts data into numerical representations that capture deeper relationships from the data. This is similar to word embedding, where words with similar meanings were positioned close to each other in the vector space (Madaan et al., 2024).

For future projects it is recommended to partner up with the team responsible for Tool 1 (Introduced in Chapter 6.4.1). This would be beneficial as they have experience in data preparation. In addition to better preprocessing of the data, including vector indexing and the support for GPT-4 model are crucial aspects to overcome the limitations identified in this implementation.

Despite these challenges, the chatbot demonstrated the potential of using Azure AI Studio to assist specialists in managing cable entry and gland changes.

6 Future possibilities

While the previous chapters focused on specific, uses of AI in customer support, this section brings out possibilities for a future larger-scale implementation. An approach like this would require more resources and collaboration with specialized IT consultants. If carried out, implementation would not only make resolving tickets faster for the customer support team, but also reduce the overall cases that the sales unit needs to open.

6.1 Goal

The goal of the proposed AI implementation is to create an advanced chatbot integrated into Salesforce. It would provide support to the sales unit and customer support team.

When a customer has a request regarding their order, they contact their representative sales unit. The sales unit then opens a Salesforce ticket to forward the request to a customer support specialist. In many cases, these requests are simple and follow set of rules when resolving them.

The issue with this workflow is that specialists need to spend time, solving problems that could have been addressed without opening a case. Opening tickets is also time-consuming for the employees in the sales unit.

A chatbot that helps the sales unit to analyze the customer inquiries, could reduce the number of routine cases opened for the customer support team. By training the AI model with past cases and other relevant data, it could identify patterns and offer potential solutions automatically by reading the customer's request. Chatbot would also offer follow-up questions to reach more accurate case resolutions.

This pre-screening process could resolve simple, repetitive issues within the sales unit, reducing the number of cases that need to be opened to queues handled by the customer support team. If the chatbot's responses do not provide a satisfactory solution or if the inquiry requires more detailed analysis, the sales unit could then be guided to open a normal support ticket.

The customer support team would then utilize the same chatbot to solve the cases from the Salesforce queues. While the use case would be slightly different for each team, the model could be fine-tuned accordingly. The sales Unit should be provided with a model that focuses on handling the simpler inquiries, while the customer support team would have the chatbot focus on finding solutions to more complex issues. By customizing the AI model for each team, common issues could be addressed more efficiently.

For the chatbot to deliver these benefits, the model must be trained on a large set of high-quality data. The next section will detail the most relevant data for the training process.

6.2 Training Data

Research data for this thesis consisted of Salesforce cases from a three-month period. Future work should expand this scope to include all cases ever received by the customer support team, prioritizing more recent cases. The initially processed dataset could be used as a reference for the AI model to identify the most relevant data points from the remaining cases. The key elements of the Salesforce data are:

- **Subject:** Order number, Short title describing the change
- **Description:** Detailed explanation of what changes are needed
- **Case resolution description:** Comment from a specialist stating what was done for the order

Information gathered from salesforce should be then paired with quality notification data. This will give information about whether or not the requested changes were actually done and how they impacted the order.

Quality notifications are opened as a part of the order change process. They work as an assignment to production workers to make the changes that were updated to the order. Every notification goes through an acceptance process where engineers and other employees verify the feasibility of the change. If approved, the request is sent to the production line and the modification is made. Relevant information in Quality notifications includes.

Notification feed:

- Updates on the progress of the change
- Possible validation errors
- Detailed information about proposed order changes

Description:

- Production status at the time of the modification
- Tells what variant codes are used in the change

For a more precise AI model, it's important to connect motor-specific details from SAP with the quality notification data. This way chatbot’s responses can consider the limitations of certain motor types. Relevant details include the motor type and size, as well as the variant codes associated with each motor. By combining Salesforce and quality notification data with the motor details, the AI model can gain an understanding of the entire order change process, from the initial request to the final resolution.

6.3 Combining Resources with Ongoing AI Projects

When a company starts a new, expensive project, it's crucial to ensure that ongoing projects do not overlap, so the work is not done twice. Currently, the case company has at least two other ongoing AI projects. After interviewing the people involved, an overview of these projects was created to ensure proper coordination and effective use of resources.

6.3.1 Tool 1

In 2020, the case company launched a project to develop a chatbot capable of being trained with internal data. The time was not right for the initial launch and the project was restarted in the fall of 2023. As of June 2024, (Tool 1) is primarily used by the warranty and technical teams in one of the case company's departments, with plans for future expansion to other departments.

Features

(Tool 1) was built using Blazor full stack and is hosted on Azure's virtual machines. Several features are offered to improve user experience. It provides example questions to help users get started and allows them to choose between GPT-3.5-turbo and GPT-4 AI models, depending on whether they want faster or more accurate responses. The chatbot also saves conversation history, assists with follow-up questions, and includes a search function that allows for more precise searches.

Training data

Data for the chatbot has been collected from the case company's library database, prioritizing documents based on their usage frequency. Once sufficient data is gathered from the library, attention will shift to other important databases.

Data used for (Tool 1) involves mainly PDF files and some Excel spreadsheets. Accurate indexing is crucial for the model to understand the content. Training the AI model

includes entering metadata into Excel spreadsheets, linking this metadata to corresponding PDF documents, splitting PDF documents into pages and chunks, and converting the processed data into JSON objects that can be hosted on Azure virtual machines.

Testing and feedback

Since the spring of 2024, a four-person team has been testing (Tool 1), identifying bugs, and suggesting new features. Each chatbot response includes a feedback button to evaluate its usefulness.

The project has received positive feedback, with many employees saying that they primarily use (Tool 1) before searching the answers elsewhere.

6.3.2 Tool 2

Another ongoing project helps the sales unit match customer specifications with the right motor configurations. This tool reduces the manual effort and improves the accuracy of product selection.

Features

(Tool 2) assists sales professionals by automating the extraction and organization of motor specifications from customer-provided PDF files. It extracts relevant specifications, and based on this data it can help on finding the correct motor for this customer.

If a customer forgets to provide some of the relevant data, (Tool 2) will identify the missing information, simplifying the process of following up with the customer.

The AI tool operates using in-house coded algorithms to gather and process information for immediate use and to improve future performance. As the model handles more specifications, it becomes more accurate in its recommendations.

(Tool 2) also offers intelligent suggestions, such as adding a specific variant code to better match customer requirements. Specialists can choose to use or ignore these suggestions based on their judgment.

To summarize the final specifications, the tool leverages GPT-4 alongside its in-house algorithms. It scans the entire PDF and other data to produce a clean summary. This task runs in the background, allowing specialists to continue working without interruptions. After the summary is ready, the professional double-checks the summary to make sure there are no mistakes.

Possibilities in the future, Feedback

At the moment, the sales unit asks questions from customer support team via Salesforce. As (Tool 2) improves, the sales unit can ask some of the technical questions directly from it, reducing the workload of customer support specialists.

(Tool 2) can also serve as a discussion forum for questions about different specification options and their feasibility. This integration ensures that questions are connected to the context of the motor specifications. These discussions can also train the model, enabling it to provide answers without needing help from other specialists if the same questions are asked repeatedly. If (Tool 2) doesn't know the answer, it can provide a link to the correct Salesforce queue, where the case is solved with the methods mentioned earlier in this thesis. The development of (Tool 2) can help reduce the volume of repetitive queries in customer support queues.

7 Conclusions

This chapter presents the conclusions drawn from the study. It will summarize the research objectives, key findings, limitations, and give recommendations for future research.

7.1 Results

The following three research questions were presented:

RQ1. What AI solutions are currently possible for different types of requests?
RQ2. What is the most relevant data for training the AI model?
RQ3. What key factors should future research address to improve AI solutions?

To answer these questions, data was collected from a Salesforce ticketing system over a three-month period, covering 2,875 customer inquiries. The goal was to gain a better understanding of the types of requests customers make regarding their orders.

After organizing the Salesforce data based on case subtypes, a deeper analysis was conducted. The goal was to find a more accurate representation of the requests to understand the common issues faced by customers better.

The analysis showed that many similar requests were spread across different queues. Some of the original queues were also too broad to accurately describe what was asked in the cases. By placing these cases into a more detailed groups, a clearer picture of the common issues were achieved.

Cases were then divided into two groups based on their resolution methods. This categorization guided the identification of suitable AI implementations for each type of case.

For routine cases, the necessary data was mostly available in the SAP ERP system. Due to the order-specific nature of this data, it was not practical to use it directly in AI model training. Instead, the study concluded that implementing a chatbot that provides instructions, on how to resolve cases would be better. By giving step-by-step instructions how to find information from specific SAP transactions, the AI model is not required to directly interact with the dynamic data.

In contrast, resolving technical cases requires the specialist to retrieve information from other sources to verify the feasibility of certain modifications. Data found from these technical documents is well-suited for training an AI model because it is stable, unlike the order-related data. By training the model using this information, a chatbot can provide direct answers to specialists regarding order changes.

To make the study more practical a small subset of customer requests were chosen for a pilot implementation. This was carried out using the Azure AI platform with the goal of developing an AI model capable of answering simple questions related to changes in cable entries or glands. While the implementation did not meet all expectations, it provided a solid starting point for future work. Findings offered a clear picture of how AI could support the Customer Support team and help improve their efficiency in the future.

The second research question focused on identifying the most relevant data for training the AI model. Through a detailed analysis of Salesforce cases and leveraging the author's work experience, a list of useful data sources was collected.

- **Product catalog:** Technical details of motors and customization options.
- **Variant Codes:** Descriptions of the possible modifications
- **Excel Files:** Formulas for cost and delivery impacts
- **Specialist Input:** Guides for resolution methods and SAP navigation.
- **Salesforce Cases:** Support ticket details.

7.2 Limitations

The research gave promising results, but several limitations must be acknowledged. The study's scope was limited to cases gathered from a three-month period and every request was manually processed to gather meaningful information. Manual processing of the data may have introduced some inaccuracy in the findings. Additionally, the data only reflected customer requests without tracking the actual changes made to motors. Therefore, it cannot be determined how many cases were only inquiries and how many actually resulted in modification.

After moving to the pilot implementation, several challenges were identified, particularly in the preprocessing and indexing of the training data. These issues impacted the ability to generate accurate chatbot responses.

The implementation relied on Azure Search Service for data indexing. However, due to restricted resources, the system could only use keyword and semantic search methods. While these techniques are good with well-structured data, they were less effective for the data used in this project. A more suitable approach would have been vector indexing, which could have handled the data more effectively. This method would have likely improved the accuracy of the chatbot’s responses by capturing the relationships within the training data better.
Another challenge was the use of the GPT-3.5 model. While it performed reasonably well, the adoption of GPT-4 could have produced more accurate responses.

Additionally, the scope of the pilot was limited to a single use case. The model was fine-tuned for this specific scenario, which means expanding it to handle multiple cases may introduce additional challenges. Broader use cases will likely require further adjustments and fine-tuning of the model.

7.3 Future recommendations

To address the third research question on what factors future AI projects should consider, several key areas were identified.

1. **Expanding the Dataset:** The initial study used Salesforce data from a three-month period. Future AI models would benefit from a broader dataset including all cases ever received. Cases that were manually processed for this thesis could be used to train the AI model to automate the processing. This dataset should be then linked to quality notifications that were opened when doing the particular change. This way the model will get better understanding on what modifications were actually made. To further improve the model's accuracy motor-specific details from SAP should be linked with each quality notification.

2. **Fine-Tuning AI Models:** Fine-tuning AI models for specific teams such as the sales unit and customer support is crucial when expanding the scope of the implementations. Customizing the model for the sales team can refine its responses to determine if customer requests can be fulfilled. This pre-screening of simple and repeating issues could reduce the number of cases that need to be escalated to the customer support team. Meanwhile, the chatbot can be tailored to handle more complex issues for the customer support team. This approach enhances overall effectiveness by ensuring that the model provides appropriate levels of support based on the specific needs of each team.

3. **Upgrading AI Capabilities** Upgrading from GPT-3.5 to GPT-4 is recommended to improve the model's ability to handle a broader range of complex queries. To address the limitations of keyword and semantic indexing, adopting vector indexing is important. This will significantly improve the chatbot's ability to handle complex queries and improve response quality, even with unstructured data handled in this thesis.

4. **Coordination with IT Specialists and Ongoing Projects** Effective coordination with other ongoing AI projects within the case company is crucial for optimizing resources. For technical challenges like data preprocessing and indexing, it's advised to consult IT specialists to get more reliable results.

This thesis has shown that AI has great potential to enhance customer support workflows. Despite some limitations, the research provides a strong foundation for future AI projects by offering practical recommendations. Applying the mentioned recommendations can help the case company utilize AI in customer support work more efficiently.

References

Antipova, K., & Horban, H. (2024). POSITIONAL ENCODING FOR TRANSFORMERS. Publishing House “Baltija Publishing”. https://doi.org/10.30525/978-9934-26-436-8-1

Bank, D., Koenigstein, N., & Giryes, R. (2023). Autoencoders. Machine Learning for Data Science Handbook: Data Mining and Knowledge Discovery Handbook, 353-374. https://doi.org/10.1007/978-3-031-24628-9_16

Borawar, L., & Kaur, R. (2023). ResNet: Solving vanishing gradient in deep networks. Proceedings of International Conference on Recent Trends in Computing: ICRTC 2022. Singapore: Springer Nature Singapore. https://doi.org/10.1007/978-981-19-8825-7_21

Brown, T. B., Mann, B., Ryder, N., Subbiah, M., Kaplan, J., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., Agarwal, S., Herbert-Voss, A., Krueger, G., Henighan, T., Child, R., Ramesh, A., Ziegler, D. M., Wu, J., Winter, C., Hesse, C., Chen, M., Sigler, E., Litwin, M., Gray, S., Chess, B., Clark, J., Berner, C., McCandlish, S., Radford, A., Sutskever, I., & Amodei, D. (2020). Language models are few-shot learners. https://doi.org/10.48550/arXiv.2005.14165

Choi, S. R., & Lee, M. (2023). Transformer architecture and attention mechanisms in genome data analysis: a comprehensive review. Biology, 12(7), 1033. https://doi.org/10.3390/biology12071033

Chitty-Venkata, K. T., Mittal, S., Emani, M., Vishwanath, V., & Somani, A. K. (2023). A survey of techniques for optimizing transformer inference. Journal of Systems Architecture, 102990. https://doi.org/10.1016/j.sysarc.2023.102990

Das, S., Tariq, A., Santos, T., Kantareddy, S. S., & Banerjee, I. (2023). Recurrent neural networks (RNNs): architectures, training tricks, and introduction to influential research. Machine Learning for Brain Disorders, 117-138. https://doi.org/10.1007/978-1-0716-3195-9_4

Deng, L. (2012). Three classes of deep learning architectures and their applications: a tutorial survey. APSIPA Transactions on Signal and Information Processing,

57, 58. https://www.microsoft.com/en-us/research/wp-content/uploads/2016/02/Transactions-APSIPA.pdf

Di Gennaro, G., Buonanno, A., & Palmieri, F. A. (2021). Considerations about learning Word2Vec. The Journal of Supercomputing, 1-16. https://doi.org/10.1007/s11227-021-03743-2

Dongare, A. D., Kharde, R. R., & Kachare, A. D. (2012). Introduction to artificial neural network. International Journal of Engineering and Innovative Technology (IJEIT), 2(1), 189-194. https://citeseerx.ist.psu.edu/document?repid=rep1&type=pdf&doi=04d0b6952a4f0c7203577afc9476c2fcab2cba06

Dufter, P., Schmitt, M., & Schütze, H. (2022). Position information in transformers: An overview. Computational Linguistics, 48(3), 733-763. https://doi.org/10.1162/coli_a_00445

European Commission. (2018). A Definition of AI: Main Capabilities and Scientific Disciplines. High-Level Expert Group on Artificial Intelligence, Directorate-General for Communication. https://ec.europa.eu/futurium/en/system/files/ged/ai_hleg_definition_of_ai_18_december_1.pdf

Greco, C. M., & Tagarelli, A. (2023). Bringing order into the realm of Transformer-based language models for artificial intelligence and law. Artificial Intelligence and Law, 1-148. https://doi.org/10.1007/s10506-023-09374-7

Lezmi, E., & Xu, J. (2023). Time series forecasting with transformer models and application to asset management. Available at SSRN 4375798. https://doi.org/10.2139/ssrn.4375798

Li, Y., Si, S., Li, G., Hsieh, C. J., & Bengio, S. (2021). Learnable fourier features for multi-dimensional spatial positional encoding. Advances in Neural Information Processing Systems, 34, 15816-15829. https://doi.org/10.48550/arXiv.2106.02795

Lipton, Z. C., Berkowitz, J., & Elkan, C. (2015). A Critical Review of Recurrent Neural Networks for Sequence Learning. https://doi.org/10.48550/arXiv.1506.00019

Madaan, N., Chaudhury, P., Kumar, N., & Bedathur, S. (2024). TransDrift: Modeling Word-Embedding Drift using Transformer. Companion Proceedings of the ACM on Web Conference 2024, 1388-1389. https://doi.org/10.1145/3589335.3651894

Michelucci, U. (2022). An introduction to autoencoders. https://doi.org/10.48550/arXiv.2201.03898

Microsoft Azure. (2023). Azure AI Documentation. Microsoft Azure. https://azure.microsoft.com/en-us/services/machine-learning/

Mittal, S. (2022). Compositional Attention: Disentangling Search and Retrieval. Mila - Quebec AI Institute. https://mila.quebec/en/article/compositional-attention-disentangling-search-and-retrieval

Muhammad, I., & Yan, Z. (2015). SUPERVISED MACHINE LEARNING APPROACHES: A SURVEY. ICTACT Journal on Soft Computing, 5(3), 946-952. https://doi.org/10.21917/ijsc.2015.0133

Naeem, S., Ali, A., Anam, S., & Ahmed, M. M. (2023). An unsupervised machine learning algorithms: Comprehensive review. International Journal of Computing and Digital Systems. https://doi.org/10.12785/ijcds/130172

Niranjani, V., Duraisamy, P., Priyadharshan, M., & Gayathri, B. (2020). Advancements in Machine Learning Techniques for Optimizing Cognitive Radio Networks: A Comprehensive Review. https://doi.org/10.53759/acims/978-9914-9946-9-8_20

Nguyen, T. Q., & Salazar, J. (2019). Transformers without tears: Improving the normalization of self-attention. https://doi.org/10.48550/arXiv.1910.05895

Pires, T. P., Lopes, A. V., Assogba, Y., & Setiawan, H. (2023). One wide feedforward is all you need. https://doi.org/10.48550/arXiv.2309.01826

Puiutta, E., & Veith, E. M. (2020). Explainable reinforcement learning: A survey. International Cross-Domain Conference for Machine Learning and Knowledge Extraction. Cham: Springer International Publishing. https://doi.org/10.48550/arXiv.2005.06247

Qu, X., Yang, L., Guo, K., Ma, L., Sun, M., Ke, M., & Li, M. (2021). A survey on the development of self-organizing maps for unsupervised intrusion detection. Mobile Networks and Applications, 26, 808-829. https://doi.org/10.1007/s11036-019-01353-0

Rahali, A., & Akhloufi, M. A. (2023). End-to-end transformer-based models in textual-based NLP. AI, 4(1), 54-110. https://doi.org/10.3390/ai4010004

Rosendahl, J., Tran, V. A. K., Wang, W., & Ney, H. (2019). Analysis of positional encodings for neural machine translation. Proceedings of the 16th International Conference on Spoken Language Translation. https://aclanthology.org/2019.iwslt-1.20.pdf

Russell, S., & Norvig, P. (2020). Artificial Intelligence: A Modern Approach (4th edition). https://dl.ebooksworld.ir/books/Artificial.Intelligence.A.Modern.Approach.4th.Edition.Peter.Norvig.%20Stuart.Russell.Pearson.9780134610993.EBooksWorld.ir.pdf

Salehinejad, H., Sankar, S., Barfett, J., Colak, E., & Valaee, S. (2017). Recent advances in recurrent neural networks. https://doi.org/10.48550/arXiv.1801.01078

Saxena, A. (2022). An introduction to convolutional neural networks. International Journal of Research in Applied Science and Engineering Technology, 10(12), 943-947. https://doi.org/10.22214/ijraset.2022.47789

Shetty, S. H., Shetty, S., Singh, C., & Rao, A. (2022). Supervised machine learning: algorithms and applications. Fundamentals and methods of machine and deep learning: algorithms, tools and applications, 1-16. https://doi.org/10.1002/9781119821908.ch1

Shin, A., Ishii, M., & Narihira, T. (2022). Perspectives and prospects on transformer architecture for cross-modal tasks with language and vision. International Journal of Computer Vision, 130(2), 435-454. https://doi.org/10.48550/arXiv.2103.04037

Singh, S., & Mahmood, A. (2021). The NLP cookbook: modern recipes for transformer based deep learning architectures. https://doi.org/10.1109/ACCESS.2021.3077350

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., & Polosukhin, I. (2017). Attention is all you need. In Proc. NeurIPS. https://doi.org/10.48550/arXiv.1706.03762

Yenduri, G., Ramalingam, M., Selvi, G. C., Supriya, Y., Srivastava, G., Maddikunta, P. K. R., Raj, G. D., Jhaveri, R. H., Prabadevi, B., Wang, W., Vasilakos, A. V., & Gadekallu, T. R. (2024). Generative pre-trained transformer: A comprehensive review on enabling technologies, potential applications, emerging challenges, and future directions. https://doi.org/10.1109/ACCESS.2024.3389497