Toward a knowledge management method for training customer support AI agents

Edgars Dzenuska[1*], Peteris Rudzajs[2]

[1] *Visma Labs SIA, Riga, Latvia,* [2] *Pearl Latvija SIA, Riga, Latvia*

Abstract

This paper summarizes preliminary findings on a knowledge management method to train generative AI agents for customer support in software companies. Despite advances allowing AI deployment with minimal technical skills, companies struggle with documenting and maintaining suitable knowledge bases. A survey of 20 software firms found that 75% face challenges in training AI with domain-specific knowledge. Through literature review and industry analysis, this research develops guidelines for creating and managing knowledge articles as training data. We report initial results from a ten-article pilot in a European software company, where our method improved answer quality, as measured by BERTScore F1, Cosine similarity and human-rated correctness of answers.

Keywords

knowledge management, generative AI, customer support, retrieval augmented generation

1. Introduction

As users of software, we expect the supplier – a software company - to provide a fast and competent customer support. Depending on the company, the customer support team offers troubleshooting, answering queries, and guidance on using the software. Thus, the support we receive can have profound impact on our experience, business results or personal wellbeing. Providing this essential service creates substantial cost and operational challenges for the companies. Generative AI models present an opportunity to lower the costs and increase efficiency. Currently there is a wide choice of no-code and low-code solutions in the market, enabling relatively easy implementation and use of generative AI components. According to estimate of Boston Consulting Group, "the technology, once implemented at scale, could increase productivity by 30% to 50%". [1] However, generative AI models don't come with knowledge about the specific software or services, and need to be trained to understand the particular business domain. We surveyed customer support leaders in 20 software companies that have either completed or are currently implementing AI agents in their support organisations. On the question *"What challenges did you face during implementation?"*, 75% of leaders responded *"Training the generative AI agent with knowledge about our software and services"*, 50% indicated that insufficient quality of the AI agent's responses delayed or complicated the implementation. Most common issues were partial responses (90%) and misleading answers or "hallucinations" (80%). Most rewrote (90%) or re-structured (80%) the knowledge articles to fix the response quality issues.

The current scientific literature lists specific methods, requirements and challenges related to training generative AI agents. These range from building a custom GPT model or fine-tuning an existing model (requiring specific training data and financial investments) to using retrieval augmented generation (requiring the right content at the right time and quality). However, the current scientific literature does not provide enough detail about knowledge management, necessary

BIR-WS 2025: BIR 2025 Workshops and Doctoral Consortium, 24th International Conference on Perspectives in Business Informatics Research (BIR 2025), September 17-19, 2025, Riga, Latvia.

* [1*] Corresponding author.

edgars.dzenuska@visma.com (E. Dzenuska); peteris.rudzajs@pearlgroup.no (P. Rudzajs)

0009-0005-6126-8048 (E. Dzenuska); 0000-0002-7377-5996 (P. Rudzajs)

to create a knowledge corpus for training generative AI agents, enabling them to provide correct and useful answers to customer support queries. Also, it does not address additional challenges of knowledge management in software companies, such as frequent knowledge changes due to agile software development life cycle. Based on these findings, this paper aims to design a new method for managing knowledge. The proposed method offers a practical way for implementing generative AI agents in customer support with a higher probability of success – in particular customer satisfaction and organisation efficiency. The authors carried out an initial experimental test of the method by comparing the quality of answers generated by a generative AI agent before and after applying the method in a software company.

Section 2 examines the main use cases and technologies of generative AI agents in customer support and the methods of training these agents on domain knowledge. **Section 3** outlines the work related to knowledge management methods relevant for training generative AI agents. In **Section 4**, a new method is proposed for capturing the needed knowledge to ensure high quality of the generated responses. **Section 5** describes the initial results of the experimental test, with the conclusions and areas for further research outlined in **Section 6**. **Appendix** contains the requirements for writing knowledge articles.

2. Background

AI agents can assist companies with customer service in several ways. In this paper, the focus is on the tasks of 1st Level support - the team at the forefront of receiving customer support requests. In addition to registering, classifying and handling incidents, 1st level support also processes service requests and keeps users informed about incidents' status. Service requests in most cases are minor (standard) changes (e.g. requests to change a password) or requests for information. [2] Generative AI agents can add significant value, especially when responding to repeat and low-complexity requests for information (prompts), received as non-formal and freely structured textual input that instructs the model (used by the AI agent) to provide an answer. [3]

Based on the conducted literature research, the three most often mentioned types of AI agents for customer support are text chatbots, voice assistants, and recommenders of solutions [4]. These can be implemented via no- or low-code platforms leveraging pre-trained large language models (LLMs) for natural language processing (NLP). However, to respond effectively to customer queries, these models require domain-specific training. Companies can customize models by building agent-specific layers on pre-trained LLMs, such as training on business data to generate relevant responses [5], [6]. Without proper training, AI agents may misinterpret queries or hallucinate. Two main methods to adapt LLMs to specific domains are fine-tuning [5] and Retrieval Augmented Generation (RAG) models [7].

RAG offers advantages over fine-tuning, including lower implementation effort, widespread use in commercial solutions, and the ability to cite specific documents in responses. Challenges include document retrieval accuracy and quality [8], but RAG can produce more factual and diverse responses, reducing hallucinations [7],[9]. Therefore, RAG is recommended for integrating domain knowledge via vector databases containing embeddings of company knowledge sources, which must be accurate, current, and well-structured for effective response generation. Proper knowledge management is essential for high-quality AI responses, as discussed in the following section.

3. Related works

Lineberry 2019 [10] discusses the Knowledge Centered Service (KCS) which distinguishes two loops in the organization’s knowledge base: Solve and Evolve. The article stipulates that an indication of a “healthy” knowledge content is its usage, however, does not provide detailed instructions on how to create the knowledge content, nor on how to use it for training generative AI agents.

Lou et al. 2021 [11] mention several methods of managing knowledge (referring to chatbots in the specific article), indicating that corpus-based chatbots are best for applications that need large knowledge bases – however not addressing the specificity of the modern generative AI agents.

Referring to deployment of AI agents using a RAG-based system, O'Leary 2024 [12] suggests that experts of a company (knowledge management resources) choose information that should be used to train the model. The author provides an example of PWC knowledge base, without providing instructions on how to capture knowledge to increase probability of a successful application with RAG-based systems.

Ngai et al. 2021 [13] discuss integrating chatbots with a knowledge base to allow them to search it and use the data to generate a personalized response. The authors propose a knowledge base design framework for customer knowledge management strategy and practices of a company. The authors note that the sources that they reviewed do not investigate the design of the knowledge base sufficiently and don't substantiate it with theoretical basis. The challenge of continuously updating the knowledge base is also not addressed sufficiently.

According to Wilde 2011 [14], three types of knowledge can be distinguished: 1) knowledge about the customer, 2) knowledge from the customer, 3) knowledge for the customer. According to Ngai et al. 2021 [13], each of these types can be further divided into unverified reference information (such as information from external sources) and confirmed knowledge (verified by an expert in the company). The article does not delve into specifics about how the knowledge of these types should be captured to make sure they are suitable for training generative AI agents.

Dagkoulis et al. 2022 [15] discuss the implementation of a chatbot using Chatbot Development Platforms (CDP's), their architecture containing a search knowledge service which retrieves info from documents, web content and other knowledge management tools. The article does not provide any requirements towards the structure or quality of the information that the chatbots would use.

Guimaraes et al 2024 [3] recognize several limitations of the current LLMs (such as difficulties with mathematical reasoning and hallucinations) and urge to discuss new methods for the integration of commonsense inference in LLMs that go beyond just increase of the number of parameters and the training data. Whilst the article emphasizes the importance of these new methods to develop intelligent systems based on pre-trained language models, it does not provide more specific suggestions.

O'Leary 2023 [16] recognizes that LLMs have knowledge gaps which the LLMs themselves don't know about, and that LLMs can't access enterprise knowledge management systems and internal knowledge, with no solutions offered.

Suppliers of generative AI solutions focus on making their solutions easy to implement, user-friendly, and compatible with text-based sources of various formats and languages, as well as recommending what content to select for training, however with the prerequisite that a suitable knowledge corpus with the domain knowledge is available.

To summarize, whilst the scientific literature points out several factors to be mindful of when managing the knowledge and training the AI agent (such as impact of knowledge source distribution and knowledge transmission across the organization, AI agent integration with knowledge base, and others), the described methods don't provide enough specificity and instructions for companies to implement a solid business process for the given use case.

4. Proposed knowledge management method

The proposed method consists of requirements and guidelines for creating a knowledge base that is suitable for training of RAG-based AI agents in customer support. The proposal is based on the insights gained from the scientific literature (such as [14] and [17]) as well as best practices within the software industry (such as instructions from Salesforce [18], Zendesk [19] or Intercom [20] – vendors of customer support platforms).

The two key components of the method are:

1. Guidelines for identifying relevant knowledge in the company.
2. Requirements for how the knowledge should be captured and stored.

Figure 1 shows the key contribution of the method - creating or improving the external knowledge base used by RAG-based AI agents. A detailed process of how a knowledge base is created in a given software company is not outlined since knowledge management systems, roles, and AI agents differ significantly across companies, and it is not within the scope of this article to detail the knowledge management process.

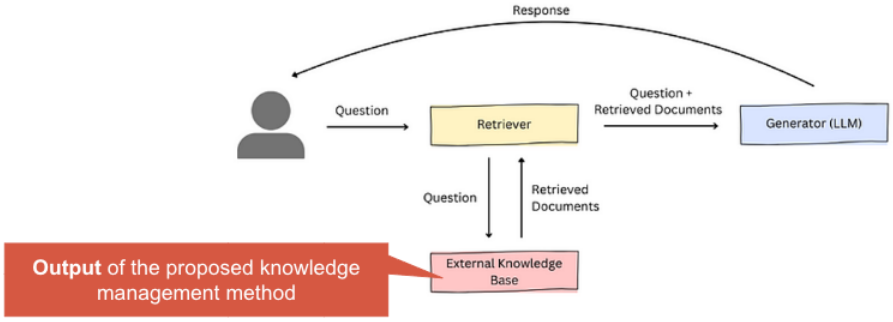


Figure 1. RAG mechanism with the proposed method (adapted from [8]).

4.1. Identifying relevant knowledge in the company

The company increases probability of a successful knowledge management process by ensuring the following elements are in place:

- A centralized knowledge management role or team to coordinate the process. This role or team defines the requirements and standards for the process, provides guidance within the organization, and oversees that the process runs smoothly. It is common for the role or team to be part of the customer support function which keeps the accountability where the most value from the process is perceived. As per Wilde 2011 [14], the role or team needs to have executive sponsorship to mobilize the stakeholders who are essential for the process to work but may not be part of the customer support function.
- Identify questions that customers ask frequently. A useful heuristics to perform this analysis is to 1) identify frequent questions in the historical support, whereby mostly it is sufficient to look at the requests received within the last 12 months which would account for seasonality of requests based on the business cycle, as well as the recent software releases; 2) identify knowledge needs customers mention in their feedback (such as Net Promoter Score (NPS) and Customer Effort Score (CES) surveys within the software, logs of phone conversations and emails as well as customer satisfaction (CSAT) surveys). Examples of such questions are *"How do I import data into product x from product y?"*, *"How can I reset my password?"*, *"Where can I see my payslip?"*, *"How can I create a new report?"*, and so on.
- Identify information needed to continuously answer the frequently asked questions (FAQ). The objective is to understand the triggers of the FAQs that arise from the changes in the business in order to identify and capture knowledge that answers the potential questions arising from these changes as soon as they occur: 1) software changes that impact the customer workflow; 2) disruptions of IT services that drive incident requests from customers, such as an unplanned interruption to an IT service or a reduction of its quality; 3) business model changes, such as the product offering, pricing, terms and conditions; 4) changes in professional services (consultancy, support, training, etc.) and their delivery; 5) non-observance of specific software usage best practice guidelines, such as a need to follow specific government regulations; 6) non-observance of customer-specifics, such as if software or services for a specific customer (type) require a different workflow.

- Identify knowledge possessors. Establish who in the company (functions or individual roles) possesses the knowledge identified as needed to answer FAQs. The possessor ideally is the person(s) who will be the first to notice the changes.
- Identify who captures the knowledge. Persons who have experience of working in customer support related roles have a better understanding of how to write the content so that majority of the customers would be able to understand it, given the different levels of customer technical competence and experience. Examples of such roles are software implementation consultants or support agents who work with customers daily, if they have the necessary level of skills in technical writing.

4.2. Capturing the knowledge

The main output of the proposed knowledge management process is a knowledge article - a document that contains a solution to a particular problem [10] of the user of the software product, or any of its products or services. A key objective of the method is to ensure a knowledge article is created and maintained as the "one source of truth" for answering a particular support query, regardless of who uses it or through what channel. The same knowledge article should be used by the support agents as a reference material, by the customers in an online knowledge base, and for retrieval by the AI agent. With this approach, the company minimizes the risk that customers get conflicting messages, as well as time and resources spent on capturing the knowledge.

The authors defined 30 requirements that provide specific instructions on how to create a knowledge base, consisting of knowledge articles, that would achieve the above objective. The requirements are split into 4 distinct categories: 1) Authoring & Content, 2) Metadata, 3) Structure & Formatting, 4) Storage & Maintenance. For detailed requirements see A. Appendix. Following the defined requirements allows the company to also engineer very effective prompts for the generative AI agent. For example, the AI agent may be enabled to retrieve the most up to date information for the right geography, actuality and product domain by referencing the metadata parameters (language, date and domain keywords). A company should evaluate if it needs to include any additional parameters in the knowledge article according to its business domain (such as identifiers of roles on the customer side that the article is meant for, of the country or region, and so forth).

4.3. Evaluating quality and updating existing knowledge articles

With some exceptions, companies already have domain knowledge documented in some shape or form that can speed up creation of the knowledge articles for the generative AI agent. LLM platforms (such as OpenAI ChatGPT 4o1 or Anthropic Claude 3.5 Sonnet) can evaluate the conformity of the existing knowledge articles with the requirements, based on a prompt that contains the requirements included in the A. Appendix that refer to the Authoring & Content, Metadata, and Structure & Formatting, and not its environment or other specifics that the LLM can't infer (such as if the article has been updated). LLM platforms are also able to generate a new knowledge article based on the foundation of the existing one and the requirements. However, as of the time of writing such an approach is risky, especially if the existing knowledge article contains pictures or videos which the LLM may not interpret accurately, and if the information it contains is incomplete. Considering the risks, the article must undergo scrutiny of a qualified human before exposing to the customers and/or an AI agent.

4.4. Information systems for storing the knowledge articles

As stated by Hosseingholizadeh in 2014 [21], it's the stage of storage, embodiment and updating of acquired or created knowledge in organization memory.

Creating a record of customer support requests in a customer service platform is a common practice to keep track of service requests, product incidents, their related problems, as well as to maintain and improve the quality of service delivery. Thus, it is likely that a company that wants to

implement a customer facing AI chatbot or a recommender of solutions based on incoming customer support requests, uses a customer service platform (such as Zendesk, Salesforce Service Cloud, and others [22]). Therefore, to ensure that the format of the knowledge article is suitable for uploading into customer service platforms (RQ15 in A. Appendix), the company should first explore the viability of using their existing customer service platform for authoring, storing and maintaining of the knowledge articles. An added benefit of this approach is maintaining the existing IT systems landscape (and potentially not increasing operating costs). Despite this obvious benefit, the company should evaluate the concrete customer service platform for complying with the requirements outlined in the A. Appendix before deciding.

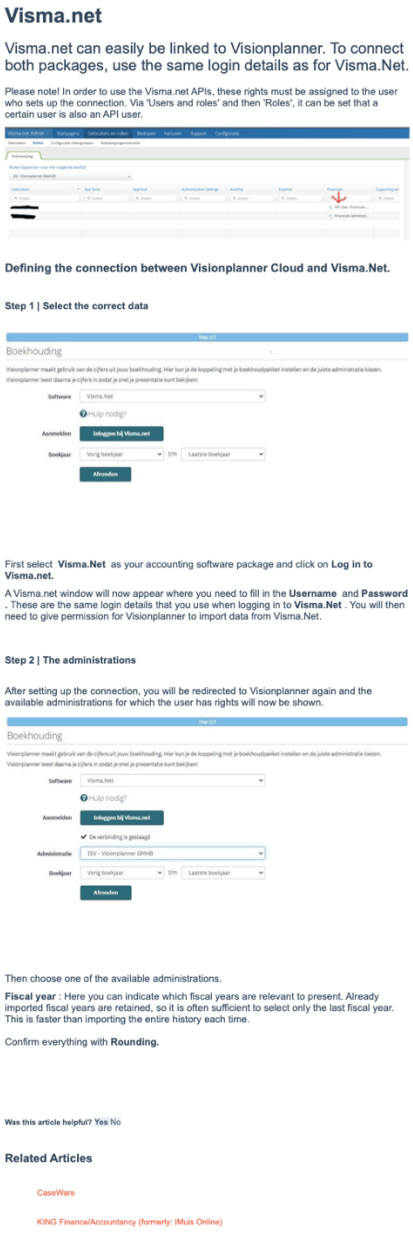
5. Experimental test

The objective of the initial experimental test is to evaluate applicability of the proposed method. This is achieved through verifying the quality of the responses generated by a generative AI agent improves after adjusting the articles according to the requirements in A. Appendix. Details on the implemented agent are provided in Section 5.1. The quality is measured using BERTscore [23] and Cosine similarity [24] metrics, as well as human-evaluated correctness. The ongoing application of the method in a company was not evaluated – how a company would continuously train AI agents and update the knowledge articles, which is an area for further research. The test includes these steps:

1. Select 10 knowledge articles in an actual software company that are currently used in customer support, providing a manageable yet diverse sample representing various customer support topics, and sufficient data to observe patterns and draw meaningful conclusions while keeping the scope feasible for in-depth analysis.
2. Adjust these 10 knowledge articles following the requirements in categories "Authoring & Content" and "Metadata" in A. Appendix (carried out by the knowledge manager of the company).
3. Define 2 questions that customers may ask about the content included in each of the current knowledge articles (20 questions in total), and the expected, ideal answers on the 20 questions (defined by the knowledge manager of the company). Expected answers provide a reference point for evaluating the answers generated by the AI agent, using the evaluation methods described in steps 5 and 6.
4. Generate answers with a generative AI agent, gaining 20 question and answer pairs with the original knowledge articles, and 20 – with the adjusted knowledge articles.
5. Calculate BERTscore. It is a well-established method that computes token-level similarity using contextual embeddings, providing an evaluation of semantic similarity in terms of precision (candidate token match to the reference), recall (reference token match to the candidate) and F1 score (combination of precision and recall). It provides a similarity score of -1 to 1 for tokens in the reference sentence (expected answer) with tokens in the candidate sentence (generated answer).
6. Calculate Cosine Similarity. While BERTScore is particularly effective at detecting paraphrases, in customer support paraphrasing can lead to undesired results and misunderstandings. Therefore, to more directly compare the reference text and the generated response, similarity score for each text fragment will be calculated using the cosine similarity evaluation method. The employed method uses embeddings model 'all-MiniLM- L6-v2' - a sentence transformer model designed to generate dense vector representations (embeddings) of sentences or short paragraphs (in our case the expected answer and the generated answer) in high-dimensional space and measures cosine distance between them on a scale of -1 (perfect mismatch) to 1 (perfect match). Sudhi et al 2024 [25] inspired the use of this method.

7. Compare the responses generated using the current and adjusted articles to the expected answers. For BERTScore, only F1 will be considered as it is a combination of recall and precision, and provides sufficient information for the given purpose.
8. Manually evaluate the generated responses, following the approach of Afzal et al 2024 [26], grading responses in four categories (Readability, Relevance,Truthfulness, Usability) on Likert scale. [27]
9. Summarise the findings to evaluate the proposed knowledge management method.

Figure 2 shows an example of a simple knowledge article, responding to a specific user's question, such as "How can I import data into Visionplanner platform from Visma.net platform?"



Visma.net

Visma.net can easily be linked to Visionplanner. To connect both packages, use the same login details as for Visma.Net.

Please note! In order to use the Visma.net APIs, these rights must be assigned to the user who sets up the connection. Via 'Users and roles' and then 'Roles', it can be set that a certain user is also an API user.

[redacted]

Defining the connection between Visionplanner Cloud and Visma.Net.

Step 1 | Select the correct data

Boekhouding

Software Visma.Net

Hulp nodig?

Aanmelden Inloggen bij Visma.net

Boekjaar Vorig boekjaar t/m Laatste boekjaar

Afronden

First select **Visma.Net** as your accounting software package and click on **Log in to Visma.net.**

A Visma.net window will now appear where you need to fill in the **Username** and **Password** . These are the same login details that you use when logging in to **Visma.Net** . You will then need to give permission for Visionplanner to import data from Visma.Net.

Step 2 | The administrations

After setting up the connection, you will be redirected to Visionplanner again and the available administrations for which the user has rights will now be shown.

Boekhouding

Software Visma.Net

Hulp nodig?

Aanmelden Ingelogd bij Visma.net

De verbinding is gelaagd

Administratie [redacted]

Boekjaar Vorig boekjaar t/m Laatste boekjaar

Afronden

Then choose one of the available administrations.

Fiscal year : Here you can indicate which fiscal years are relevant to present. Already imported fiscal years are retained, so it is often sufficient to select only the last fiscal year. This is faster than importing the entire history each time.

Confirm everything with **Rounding.**

Was this article helpful? Yes No

Related Articles

CaseWare

KING Finance/Accountancy (formerly: IMuis Online)

a)

Linking V isma.net to Visionplanner

You can link Visma.net to an administration in Visionplanner. This way you import the data from your accounting package into Visionplanner. Visma.net is your import source. In this article you can read how to link Visma.net. Follow the instructions:

- Instruction: Linking Visma.net and Visionplanner

Introduction

What you need

To complete the instructions in this article, you will need:

- the user role **administrator** or **office worker** ;
- **the Core** module ;
- and **no** subscription required.

Before you continue

- Use your Visma.net login details to link the two systems.
- Make sure that the user in Visma.net is granted the rights to use APIs for the connection with Visionplanner. Are you unable to set these rights? Contact Visma.net support .
- You can use the following components in Visionplanner if you connect Visma.net as an import source:
  - General ledger account balances
  - Cost centers
  - Accounts Receivable
  - Creditors
  - Transactions
  - Documents transactions
  - Documents outstanding invoices

Instruction: Connecting Visma.net

1. In the **Administration Monitor, select** the administration for which you want to create the link.
2. In the top bar, click **Import & Export > Import Settings** .
3. Click on **Financial: No Import Source** .
4. Under **Software,** select **Visma.net** .
5. Click on **Log in to Visma.net** .
6. Log in with your Visma.net username and password.
7. Select the administration you want to import.
8. **Under Financial Year,** select the financial years you want to import.
   - Want to import faster? Choose a small number of fiscal years. You don't have to re-select the fiscal years you've already imported. Those fiscal years will be saved.
9. Click **Save** .
10. Click on **Import and Export > Start this import** behind the name of the link to import the data into Visionplanner.

Good to know

Now that you have linked Visma.net to the administration, you can continue with linking the general ledger accounts, as described here: linking the general ledger accounts .

Read more

- Linking the general ledger accounts .
- Linking periods in the linking screen
- Emptying a fiscal year

Metadata
Author: Belinda Nagtegaal
Created on: 16.10.2024
Domain keywords: Visma.net, Linking, Visionplanner, Administration monitor
Language: Dutch

b)

Figure 2. Example of a knowledge article a) before and b) after adjustments.

5.1. Technical setup of the generative AI agent

To ensure identical conditions for evaluation of the results obtained with the original and the improved knowledge articles, a generative AI agent was deployed using OpenAI Playground "assistants=v2" (a cloud-based out-of-the-box solution for deployment of simple AI agents), "File

Search" feature (allowing to define the set of files that the AI agent uses for RAG) and the model "gpt-4-turbo" [28]. After obtaining the generated answers from the AI agent, the answers were compared with the expected answers, using an evaluation model built in Google Colab [29]. All the knowledge articles used by the AI agent and the questions and answers were in Dutch language. BERTscore evaluation used Python library "bert_score", method "score". Cosine Similarity evaluation used Python library "sklearn.metrics.pairwise", method "cosine_similarity". Knowledge articles were uploaded as HTML files and stored in the vector database, provided as part of the OpenAI Playground.

The generative AI agent was given free-text instructions for response generation, describing the role of the AI agent (customer support assistant) and the context and geography of the audience, enforcing usage of the header structure in the document and quoting the content, if possible, dictating the expected layout of the response (step-by-step guides as lists), and limiting the length of the answer. The exact same instructions and setup was used to generate the answers based on both the original and adjusted knowledge articles. It was recognized that the free-text instructions can have a significant impact on how the responses are generated, however comparison of the results is possible due to a consistent test environment, even if the response generation could be altered or improved, given different instructions.

5.2. Test results

The quality of answers obtained from the knowledge articles before adjustments are shown in Figure 3. In the chart a), the bars F1-before and F1-after indicate the BERTscore F1 similarity before and after the knowledge article was adjusted. Respectively, the b) chart indicates Cosine similarity of the text fragments.

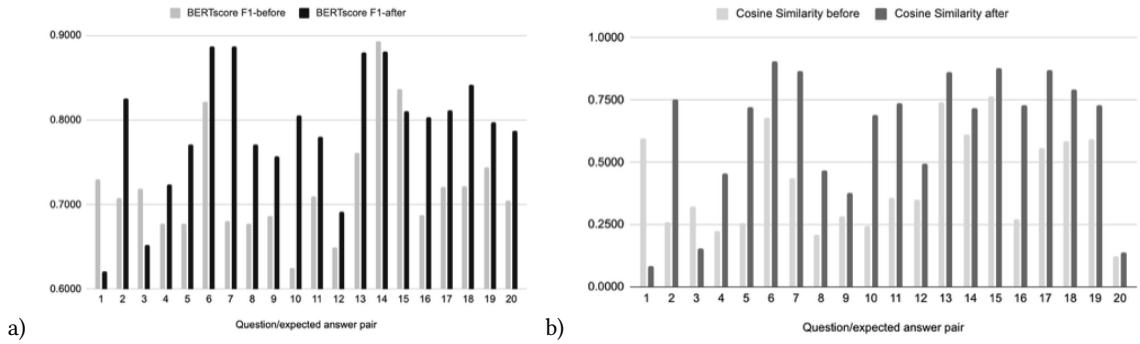


Figure 3. Similarity of the text fragments before (light colour) and after the adjustments following the method, measured through a) BERT score and b) Cosine Similarity.

The average BERTscore of the responses before adjustments is 0.7215, and the average Cosine Similarity is 0.4232. Figure 4 summarizes the findings by showing the difference between the BERTscore F1 and Cosine Similarity scores before and after the adjustments. A positive value indicates quality improvement, and negative – vice-versa – indicates the responses after the adjustment got worse.

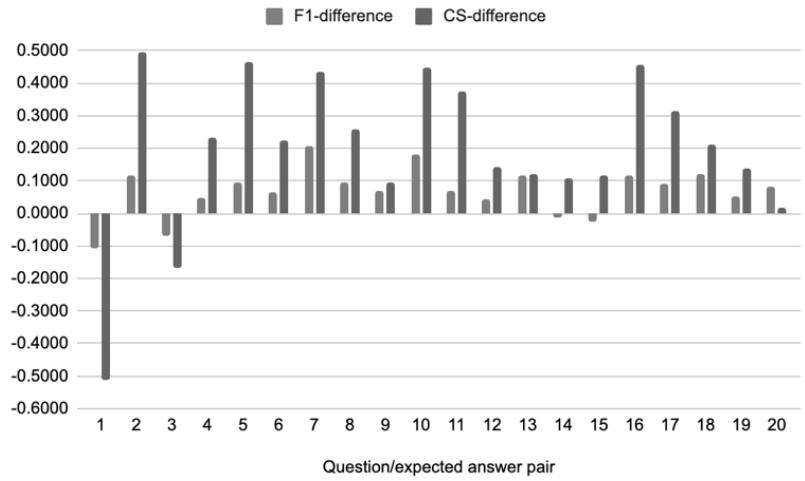


Figure 4. Overall difference of the response quality based on articles before after the adjustments following the method, measured through F1 and Cosine Similarity.

The manual evaluation of the answers before and after adjustments yielded the results shown in Figure 5, whereby any values below or above 0 indicate a deterioration or improvement of the answer in the given category. As is visible, overall, the quality improved, whereby i) readability increased by 0.4 points, ii) relevance increased by 0.65 points, iii) truthfulness increased by 0.9 points, and iv) usability increased by 1.25 points. These results validate the findings using BERTscore and Cosine similarity metrics. For example, the answers to questions 1 and 3 emerge as being worse after the adjustments in all 3 evaluation methods, whereas answers to questions 2 and 11 are significantly better after the adjustments.

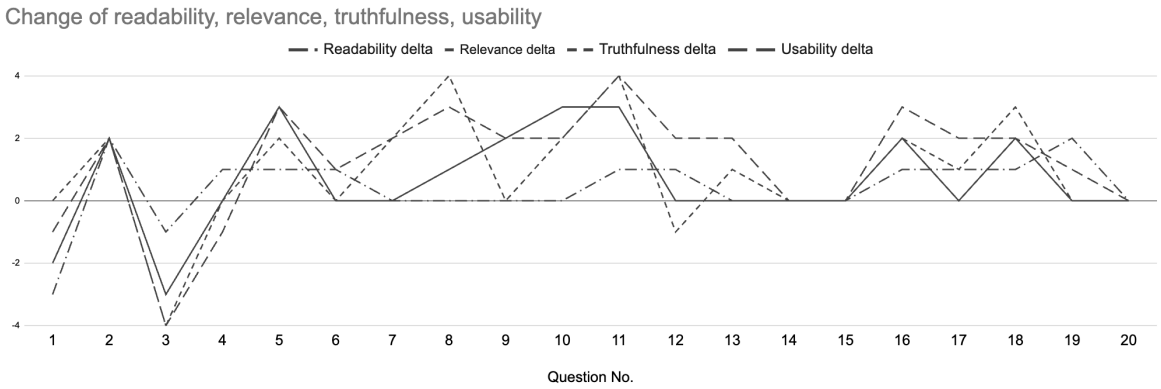


Figure 5. Human-perceived difference of the response quality based on articles before after the adjustments following the method.

As is visible from Figure 5, in 16 out of 20 questions, the adjustments of the knowledge articles resulted in a higher similarity of the answer with what the knowledge manager was expecting. The average BERTscore of the responses after adjustments is 0.7892 (+0.0677 or 9.38% improvement), and the average Cosine Similarity is 0.6218 (+0.1986 or 46.93% improvement). Put in another way, Cosine Similarity of the generated answers indicated that the quality had improved of the answers based on 8 out of 10 knowledge articles (80%), and 18 out of 20 questions (90%).

It was observed that in specific cases (e.g., question/expected answer pair 20) BERTscore F1 value is significantly higher than cosine similarity (0.7049 and 0.1215 respectively). After reviewing the samples, the explanation is that BERTscore compares embeddings of chunks created from the text it compares, and the similarity of individual words causes the score to be quite high in both answers.

However, Cosine Similarity evaluates the embeddings of the whole text fragment and more accurately evaluates the similarity of the meaning of the generated answer to the expected one.

The deterioration of both BERTscore F1 and Cosine Similarity for question-expected answer pairs 1 and 3 is notable. It is due to that the headers in the adjusted knowledge article did not contain the information that would allow to identify the right instructions in the article, and the adjusted article did not contain the information that was included in the expected (reference) answer. These edge cases emphasise the importance of the description section (RQ3, see Appendix) and including the terms that describe parts of the software solution or the company's business model, relevant to the solution.

6. Discussion and conclusion

A new knowledge management method has been proposed in this paper that software companies can use to create a knowledge base for customer support that is able to serve both the customers directly as well as an external non-parametric memory for RAG-based generative AI agents. Initial testing of the proposed method was performed using a generative AI agent built with OpenAI Playground and evaluating correctness of 20 responses generated based on 10 knowledge articles before and after the adjustments to match the requirements of the method.

Based on the evaluation, the average BERTscore across the 20 questions of the generated answers improved by 9.38%, and the average Cosine Similarity improved by 46.93%. The quality improvements were observed for answers generated based on 8 of 10 knowledge articles (80%), and 18 of 20 questions (90%). The results indicate a strong positive impact of the proposed method on the quality of responses generated. The human evaluation confirms the positive effect in terms of readability, relevance, truthfulness and usability of the answers.

The proposed method has a high practical potential as it can help companies across the world to implement generative AI agents in customer support with a higher probability of success – in particular customer satisfaction and organisation efficiency. In addition, it provides a theoretical background for further research and development of the method to address specifics of other industries, business models, and use cases (i.e., not only in customer support).

The authors acknowledge that the method relies on a manual oversight for creating and updating knowledge articles, however the rapid development of generative AI solutions allows using LLM to rapidly evaluate existing articles against the requirements, and generate new, better knowledge articles based on the most recent and accurate information sources for the given topic. This is a topic for further research.

It is acknowledged that the results can vary significantly between different companies that would implement the proposed method. The characteristics of the software and services, sources of knowledge in the company, skills of the technical writers, the configuration of the AI agent are but a few aspects that can significantly influence the results of the proposed method in another company. Conducted research and evaluation revealed further areas for research. Evaluation with a wider dataset would add more objectivity and surface additional important factors that determine success of the AI agent implementation. During the evaluation, it became obvious that the instructions sent to the LLM within the prompt could significantly change the way responses were generated. Researching, engineering and testing various prompts that leverage the knowledge article metadata and parameters could yield new, creative methods of understanding the customer context and maintaining a multilingual knowledge base with knowledge articles going back in time and addressing multiple customer segments. Graph-based RAG systems emerge as a new way [30] of capturing and retrieving knowledge for AI agents. Graphs can not only indicate sources of data, information and knowledge, but also interconnect them in the way that allows an AI agent to retrieve information with a better "understanding" of context. Exploration of this area and the state-of-the-art technology is recommended to even further improve the quality of AI agent responses. Considering that companies often work across borders and serve customers in multiple languages, it may be necessary to ensure that the terminology in the customer query is maintained correctly

when retrieving embeddings of knowledge articles from the vector database. Therefore further research also could cover the creation of a terminology dictionary for a RAG-based AI agent to find the right term given a specific language pair. The initial experimental test conducted by the authors of this paper did not evaluate the ongoing application of the method in a company – specifically continuous training of AI agents and updating the knowledge articles. This is a potential area for further research, given the fact that knowledge articles and thereby the external non-parametric memory of a RAG-based generative AI agent can get outdated quickly. For example, duplication and versioning, or full replacement of outdated knowledge articles are two approaches that could be researched for applicability and feasibility in a company environment.

Declaration on Generative AI

The authors have not employed any Generative AI tools.

References

[1] S. Clark, N. Ramachandran, S. Sokolova, V. Bamberger, "How generative AI is already transforming customer service," https://www.bcg.com/publications/2023/how-generative-ai-transforms-customer-service.

[2] "ITIL roles and responsibilities," 2024, Accessed: Oct. 27, 2024. [Online]. Available: https://wiki.en.it-processmaps.com/index.php/ITIL_Roles#ITIL_roles_and_boards_-_Service_Operation.

[3] N. Guimarães, R. Campos, and A. Jorge, "Pre-trained language models: What do they know?," *Wiley Interdiscip Rev Data Min Knowl Discov*, vol. 14, no. 1, 2024, doi: 10.1002/widm.1518.

[4] E. F. Ohata, C. L. C. Mattos, S. L. Gomes, E. D. S. Reboucas, and P. A. L. Rego, "A text classification methodology to assist a large technical support system," *IEEE Access*, vol. 10, 2022, doi: 10.1109/ACCESS.2022.3213033.

[5] J. Yun, J. E. Sohn, and S. Kyeong, "Fine-tuning pretrained language models to enhance dialogue summarization in customer service centers," in *4th ACM International Conference on AI in Finance*, New York, NY, USA: ACM, Nov. 2023, pp. 365–373. doi: 10.1145/3604237.3626838.

[6] A. Beheshti *et al.*, "ProcessGPT: Transforming business process management with generative artificial intelligence," in *Proceedings - 2023 IEEE International Conference on Web Services, ICWS 2023*, 2023. doi: 10.1109/ICWS60048.2023.00099.

[7] U. Kamath, K. Keenan, G. Somers, and S. Sorenson, "Retrieval-augmented generation," in *Large language models: A deep dive*, Cham: Springer Nature Switzerland, 2024, pp. 275–313. doi: 10.1007/978-3-031-65647-7_7.

[8] "Understanding retrieval pitfalls: Challenges faced by retrieval augmented generation (RAG) models." Accessed: May 28, 2024. [Online]. Available: https://medium.com/@researchgraph/understanding-retrieval-pitfalls-challenges-faced-by-retrieval-augmented-generation-rag-models-5bcc28a03842.

[9] P. Lewis *et al.*, "Retrieval-augmented generation for knowledge-intensive NLP tasks," in *Advances in Neural Information Processing Systems*, 2020.

[10] R. Lineberry, "Solve and evolve: Practical applications for knowledge-centered service," in *Proceedings ACM SIGUCCS User Services Conference*, 2019. doi: 10.1145/3347709.3347793.

[11] B. Luo, R. Y. K. Lau, C. Li, and Y. W. Si, "A critical review of state-of-the-art chatbot designs and applications," 2022. doi: 10.1002/widm.1434.

[12] D. E. O'Leary, "The rise and design of enterprise large language models," *IEEE Intell Syst*, vol. 39, no. 1, 2024, doi: 10.1109/MIS.2023.3345591.

[13] E. W. T. Ngai, M. C. M. Lee, M. Luo, P. S. L. Chan, and T. Liang, "An intelligent knowledge-based chatbot for customer service," *Electron Commer Res Appl*, vol. 50, 2021, doi: 10.1016/j.elerap.2021.101098.

[14] S. Wilde, Customer knowledge management. 2011. doi: 10.1007/978-3-642-16475-0.

[15] I. Dagkoulis and L. Moussiades, “A comparative evaluation of chatbot development platforms,” in *ACM International Conference Proceeding Series*, 2022. doi: 10.1145/3575879.3576012.
[16] D. E. O’Leary, “Enterprise large language models: Knowledge characteristics, risks, and organizational activities,” 2023. doi: 10.1002/isaf.1541.
[17] S. Wood and R. J. Howlett, “A web-based customer support knowledge base system,” in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 2008. doi: 10.1007/978-3-540-85563-7_47.
[18] Salesforce: How you can write a good knowledge base article, 2025. URL: https://www.salesforce.com/service/knowledge-base/article/.
[19] Zendesk: Getting started with self-service – Part 4: Writing your knowledge base articles, 2025. URL: https://support.zendesk.com/hc/en-us/articles/4408887322522-Getting-started-with-self-service.
[20] Intercom: How to write great help articles, 2025. URL: https://www.intercom.com/help/en/articles/56645-how-to-write-great-help-articles.
[21] R. Hosseingholizadeh, “Managing the knowledge lifecycle: An integrated knowledge management process model,” in *Proceedings of the 4th International Conference on Computer and Knowledge Engineering, ICCKE 2014*, 2014. doi: 10.1109/ICCKE.2014.6993467.
[22] Gartner, “CRM customer engagement center (CEC) reviews and ratings,” 2024, Accessed: Nov. 23, 2024. [Online]. Available: https://www.gartner.com/reviews/market/crm-customer-engagement-center.
[23] T. Zhang, V. Kishore, F. Wu, K. Q. Weinberger, and Y. Artzi, “BERTScore: Evaluating text generation with BERT,” in *8th International Conference on Learning Representations, ICLR 2020*, 2020.
[24] “Semantic similarity with sentence embeddings.” Accessed: Oct. 20, 2024. [Online]. Available: https://fastdatascience.com/natural-language-processing/semantic-similarity-with-sentence-embeddings/.
[25] V. Sudhi, S. R. Bhat, M. Rudat, and R. Teucher, “RAG-Ex: A generic framework for explaining retrieval augmented generation,” in *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval*, New York, NY, USA: ACM, Jul. 2024, pp. 2776–2780. doi: 10.1145/3626772.3657660.
[26] A. Afzal, A. Kowsik, R. Fani, and F. Matthes, “Towards optimizing and evaluating a retrieval augmented QA chatbot using LLMs with human-in-the-loop,” in *Proceedings of the Fifth Workshop on Data Science with Human-in-the-Loop (DaSH 2024)*, Stroudsburg, PA, USA: Association for Computational Linguistics, 2024, pp. 4–16. doi: 10.18653/v1/2024.dash-1.2.
[27] Encyclopedia Britannica, “Likert scale.” Accessed: Sep. 28, 2024. [Online]. Available: https://www.britannica.com/topic/Likert-Scale.
[28] OpenAI Assistant Playground, 2024, Accessed: Nov. 23, 2024. [Online]. Available: https://platform.openai.com/playground.
[29] Google Colab, Accessed: Oct. 20, 2024. [Online]. Available: https://colab.research.google.com/.
[30] Z. Xu *et al.*, “Retrieval-augmented generation with knowledge graphs for customer service question answering,” in *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval*, New York, NY, USA: ACM, Jul. 2024, pp. 2905–2909. doi: 10.1145/3626772.3661370.

A. Appendix - Requirements for writing knowledge articles

No.	Requirement
RQ1	A single KA should describe a single problem, question, or use case, clearly reflected by the title.
RQ2	For guidance related to software usage, use action words in the title, such as “How to...”, “Using ...”, “Setting Up ...”, etc. For company-related information, use terms like "Pricing of...", "Customer support opening hours...", etc. A title of a KA should be unique across the knowledge base.
RQ3	The KA should include a description section, clarifying its value proposition in a concise manner. For example, "Problem: Brief description of the problem to be solved and the typical reasons why it occurs."
RQ4	The KA should use direct language and avoid ambiguity (e.g., instead of suggesting, "You may need to update your software," directly state, "Update your software to the latest version for optimal performance.")
RQ5	The content in the KAs should be self-contained and complete, using full sentences and making each paragraph complete and easily understandable on its own. If the KA includes answers to multiple frequently asked questions, write the answers so that they are self-contained, and avoid simple "Yes.", "No. " or links as an answer to the question.
RQ6	Ensure the KA is tailored to the competence of the user. Avoid words that users may not be familiar with in the given context, or technical jargon.
RQ7	The content in the KA should not repeat the same information or using more words than necessary to convey the meaning.
RQ8	The KA should contain the most up to date information about the topic.
RQ9	If the user needs to know information described in a separate KA to fully understand the solution, the article should include cross-reference as a link to the other article, and not a copy of the information. Minimise number of such links and clearly explain their inclusion (e.g., "In order to understand ... , take a look at <this article>")
RQ10	If the KA contains pictures or videos to supplement the text, the visuals should have clear descriptions, compliant with accessibility standards.
RQ11	In case of a change in the given topic, the KA should be updated as soon as possible (ideally same business day) by overwriting the original content and updating the metadata parameters "Updated:" and "Updated by:".
RQ12	The KA should contain keywords to recognise metadata parameters: "Language:", "Created on: ", "Created by:", "Version:", "Updated on: ", "Updated by:", "Access:", "Domains". These parameters should be in the respective language of the KA, included in normal text or small text at the beginning or end of the KA, depending on the authoring platform.
RQ13	Include domain in the metadata section "Domains" to locate a KA describing a specific software product, feature, or service - the smallest item that can change based on the company's business processes.
RQ14	Use domain keywords in sections of the KA that refer to the particular domain, especially within the relevant headings.

No.	Requirement
RQ15	The KA must be written in a flexible, structured format (e.g., Markdown or HTML) that supports metadata, media embedding (pictures, videos, tables), and accessibility standards, while also allowing for responsive design and SEO optimization. The format must be suitable for direct publication to web platforms, usage by AI platforms, or uploading into customer service platforms (e.g., Zendesk, Salesforce) as a repository without reformatting or change of structure
RQ16	The KA should be structured in individual sections using multi-level headings to indicate titles, subtitles, and subsections and separate them from the normal text. Title - H1 (or equivalent to # in Markdown). Second highest level heading - H2 (or equivalent to ## in Markdown), used for titles of the main sections of the KA, such as "Description", "Troubleshooting", "Summary", a.o. Third highest level heading - H3 (equivalent to ### in Markdown), used for subtitles within the main sections, such as "Step 1: Do this...", "Keep in mind", and similar.
RQ17	The KA should not contain duplicate section titles (headings).
RQ18	If the KA contains guidance that guides the user through sequential steps to achieve a result, format the specific steps as number or bullet lists.
RQ19	The KA should be readable by humans without a need of converting to another document or using additional tools, except a web browser.
RQ20	The KA must be stored in a cloud storage solution, accessible remotely.
RQ21	The cloud storage must offer secure, real-time collaboration, scalable infrastructure, and integration capabilities for future technologies.
RQ22	The cloud storage must comply with security and privacy regulations.
RQ23	The cloud storage should load an article within 2 seconds.
RQ24	The uptime of the cloud storage should be 99.9%.
RQ25	The cloud storage must provide data backup and disaster recovery capabilities in line with the defined recovery time and point objectives.
RQ26	The cloud storage should allow referencing the KA as a persistent URL in the metadata when used as a source by the generative AI agent.
RQ27	The cloud storage must provide analytics, allowing to see usage of KAs and user rating, based on the chosen user feedback method.
RQ28	The cloud storage must allow filtering the KAs by as many metadata parameters (RQ12) as possible, and as a minimum by domain keywords.
RQ29	The cloud storage must be compatible with version control systems and include access controls to support role-based visibility.
RQ30	The cloud storage must be compatible with import/export standards and REST API to ensure easy integration.