Empowering Recommender Systems in ITSM: A Pipeline Reference Model for AI-based Textual Data Quality Enrichment

Philipp Reinhard[1[0000-0003-3207-666X]], Mahei Manhai Li[1[0000-0002-7171-9130]], Ernestine Dickhaut[1[0000-0002-1946-4038]], Christoph Peters[1,2[0000-0001-8140-1516]] and Jan Marco Leimeister[1,2[0000-0002-1990-2894]]

[1] University of Kassel, Kassel, Germany
{philipp.reinhard, mahei.li, ernestine.dickhaut, christoph.peters, leimeister}@uni-kassel.de
[2] University of St.Gallen, St.Gallen, Switzerland
{christoph.peters,janmarco.leimeister}@unisg.ch

Abstract. AI-based recommendation systems to augment working conditions in the field of IT service management (ITSM) have attracted new attention. However, many IT support organizations possess high volumes of tickets but are confronted with low quality, to which they train the underlying models of their AI systems. In particular, support tickets are documented insufficiently due to time pressure and lack of motivation. Following design science research, we design and evaluate an analytics pipeline to address the data quality issue. The pipeline can be applied to assess and extract high-quality support tickets for subsequent model training and operation. Based on a data set of 60.000 real-life support tickets from a manufacturing company, we develop the artifact, instantiate a recommender system and achieve a higher prediction performance in comparison to naïve enrichment methods. In terms of data management literature, we contribute to the understanding of assessing textual ticket data quality. By deriving a pipeline reference model, we move towards a general approach to designing machine learning-driven data quality analytics pipelines for attached recommender systems.

Keywords: Data Quality, Artificial Intelligence, ITSM, Recommender Systems

1 Introduction

More and more enterprises are overloaded by the vast number of technical issues regarding their infrastructure and applications [1]. They rely on IT Service Management (ITSM) to deal with downtimes and system inefficiencies by performing problem-solving tasks to keep operations going [1]. IT service desk agents are responsible for solving the tickets, which are codified user issues as support requests. They are thus faced with a constantly growing number of heterogeneous support requests and at the same time have to work more efficiently, while ensuring a high level of customer satisfaction [2, 3]. Thus, research and practice turn to AI-driven ITSM (AI-ITSM), which applies machine learning and deep learning to augment the overloaded and often overworked agents [4–6]. Recently, recommendation systems based on solved IT tickets referred to as ticket recommendation systems (TRS), have gained increasing interest in research [7–9]. However, due to various reasons (e.g., time pressure or convenience) and the complexity of support services [10], support agents tend to insufficiently describe issues and summarize resolutions, which in consequence limits the capabilities of the AI-driven systems [11]. Inadvertently, data quality remains a major challenge for AI-driven cognitive IT Service Management [6] and recommendation systems in general [12, 13].

Only a few researchers have aimed at considering data quality as a key determinator of recommendation performance in the context of AI-ITSM [2, 11, 14], even though the importance of data quality for recommendation outcomes, in general, has been well-known [12, 15–17]. Currently, the literature focuses on clustering and classifying tickets (e.g., 18–22), while the role of ticket quality is largely neglected. Our work applies an approach to specify ticket documentation quality by holistically incorporating sophisticated preprocessing, clustering steps, a comprehensive set of linguistic features, as well as machine learning models. In contrast to the existing literature on data quality in recommender systems [12, 13], our research seeks to leverage existing knowledge within an organization [23] in form of solved tickets by extracting and maintaining high-quality text data to enrich TRS. Given the research motivation, we aim at answering two research questions: [RQ1]: *How can we design an analytics pipeline for considering textual ticket data quality?* [RQ2]: *How can the pipeline improve ticket recommendation systems?* We follow a design science research (DSR) approach to instantiate an analytics pipeline [24] and finally develop a pipeline reference model for data quality enrichment in AI-ITSM. Thus, we codify our design knowledge in a generally applicable reference model to make our developed knowledge accessible in new contexts.

2 Related Work

2.1 Data Quality in Recommender Systems

It is generally known that assessing data quality is important for information systems research as low data quality results in expensive data quality costs [25]. The implications for TRS are indirect quality costs such as incorrect or insufficient decision augmentation. This leads to low user satisfaction. Previous literature on recommender systems in general confirms that the quality of the underlying data has a major impact on the performance of traditional recommender systems [12, 24, 25]. The research on data management examines concepts and manifestations of data quality, although most of them do not consider unstructured text data specifically. In general, data quality is defined as a multidimensional concept [26–28]. In most methods and approaches to assess data, an individual determination of dimensions and metrics is crucial. Yet, a standardized set of data quality dimensions is not accessible due to context-specific requirements. Especially in the domain of textual data, data quality dimensions have not been widely examined. The most commonly used dimensions have been summarized by

Batini et al. [29], on whose foundation Cai and Zhu [30] consolidated a concept of data quality dimensions for big data including availability, usability, reliability, relevance, and presentation quality.

The literature on recommender systems acknowledges the relevance of data quality [12, 26–28]. Researchers in that realm mainly focus on examining individual data quality dimensions. For example, Bharati and Chaudhury [26] considered data accuracy and completeness to enhance decision-making. However, most research on data quality in recommender systems focuses on analyzing the completeness dimension. Feldman et al. [25] discovered the role of incomplete data sets on a classifier and Woodall et al. [32] examined the influence of completeness on decision-support processes. Other attempts have been directed at investigating the role of completeness in terms of the number of features as well as the extent of feature values [11, 12]. By considering a multi-dimensional view of text data quality [30] and adapting the framework proposed by Heinrich et al. [11], we explore the impact of ticket data quality on TRS performance in our study.

2.2 Ticket Classification and Ticket Quality Assessment

Ticket classification is a subset of text classification, as tickets are textual data in an unstructured format. Prior research on AI-ITSM has emphasized classifying tickets according to prioritization, complexity, content, or other characteristics (e.g., [3, 29, 30]). For instance, Marcuzzo et al. [20] developed a multi-level approach for hierarchical ticket classification using BERT [31]. Revina et al. [18] classified tickets by complexity and associated effort using a set of linguistic features. Predominantly, prior research on ticket classification has shown that incorporating more complex sets of linguistic and non-linguistic features instead of naïve text classification is auspicious [10, 17]. Naïve text classification typically relies on vector representations of text such as TF-IDF, word2vec, or doc2vec. Selecting and calculating linguistic features comes with higher effort and can be more time-consuming since expert knowledge is required to determine a sufficiently sophisticated set of domain-specific features [17, 36]. In general, the use of machine learning and deep learning algorithms is being a promising tool for classifying tickets.

Assessing ticket quality is one of the subfields of ticket classification [8, 11, 32]. While there have been efforts to provide quality assessment for change request data [33], the literature on ticket quality remains limited. Baresi et al. [32] developed AC-QUA, an approach for assessing the quality of issue descriptions. Thereby they removed the need for additional communications and guided users to properly describe the incident. On the other end of the agent-customer communication, Agarwal et al. [11] introduced an automated quality assessment of unstructured resolution text in IT service systems but only considers a regression model. According to their findings, high-quality resolution text involves aspects of text layout, discourse relations (contingency and expansion), and domain vocabulary. Both approaches [11, 32] try to recommend measures for improving ticket data in real time. Analogously, Zhou et al. [8] applied character-level, entity-level, semantic-level, and attribute-level features for calculating ticket resolution quality and integrating it into a resolution recommendation system. Unlike other approaches, our study considers both issue and resolution descriptions and aims at predicting ticket quality with help of machine learning models to enrich data for TRS. Additionally, we include ticket clustering approaches such as topic modeling for identifying the helpfulness of tickets. Our proposed pipeline reference model aggregates the knowledge on how to design analytics pipelines to enrich textual data quality.

3 Research Approach

This study aims to generate prescriptive knowledge types in the form of a ticket analytics pipeline [34]. Following the DSR process of Peffers et al. [35] we aim to develop domain-specific knowledge and artifacts, which includes the steps depicted in Fig. 1. We draw on existing literature and fundamentals of text analysis and data quality and include domain experts in the conceptualization of features and the labeling of domain-specific knowledge. Our design requirements are predominantly derived from literature and interviews with 17 support agents and managers who confirm the problem space. After developing an initial instantiation of the machine learning-based scoring model with a tentative set of features and conducting a first performance evaluation, we added a second iteration to improve the data labeling quality and revise our features to develop a final analytics pipeline. By extensively evaluating the pipeline and comparing it with a naïve classification pipeline, we aim to show how ticket data quality is influencing the attached recommender system performance.

In addition to the development and evaluation of our artifact, we develop a pipeline reference model, by abstracting our knowledge, making it transferable and applicable to other contexts. Thus, we address the ongoing discussion in DSR on the lack of design knowledge reusability since design knowledge is often lost after a project ends [36]. We further address the issue of DSR contributions tending to remain isolated with little to no relation to other solutions because of little abstraction of the findings [37]. Reference models offer the possibility to store knowledge in an abstract form for further use cases and to make it accessible to other projects [38].

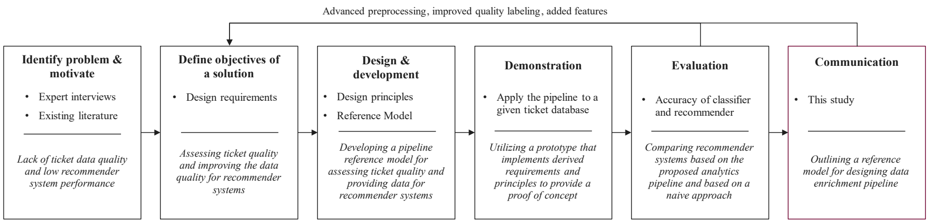

Fig. 1. Design science procedure according to Peffers et al. [35]

4 Designing a Ticket Analytics Pipeline

4.1 Problem Formulation and Objectives

Due to time constraints and a lack of motivation to create high-quality summarizations, the quality of ticket data remains a key challenge for IT support organizations [6]. Interviews with IT support agents and managers show a key goal of AI-ITSM should be to overcome inconsistent documentation, time-consuming quality assessment, inadequate quality, and inadequate knowledge base articles [2]. Intelligent quality assessment, therefore, forms the basis for improving IT support processes and augmenting the service workforce [6]. In summary, the goal of creating a knowledge repository and ensuring data quality is to improve the performance of recommender systems [12] and provide a sophisticated user experience with such TRSs. We derived the following design requirements from literature and practice (DR):

DR1: Ensure domain-specific helpfulness. The included tickets in TRS should be helpful and augment the adaptation process for a new incoming incident [39]. Helpfulness should be assured by certain quality features and meaningful problem-solution pairs. It involves domain experts in conceptualizing and developing such systems [40].

DR2: Provide transparency on feature importance. The pipeline should provide final sets of features, that significantly determine ticket quality. Information on the impact of certain features is of special interest. By doing so, the features and the corresponding feature values can be used to support agents to produce high-quality tickets and in general offer transparency [41, 42].

DR3: Differentiate between issue and resolution description quality. The scoring model should be able to differentiate between issue and resolution to account for different quality characteristics [11, 32]. This requires thorough pre-processing and a dual labeling, feature engineering, and classification procedure.

DR4: Consider a multi-dimensional concept of data quality. Features should be a set of diverse types of criteria such as linguistic and non-linguistic features to counteract the unstructured character of ticket data [43, 44]. The set of features should be able to determine the data quality of complete ticket descriptions in terms of established data quality dimensions such as reliability, relevance, and presentation quality [45–47].

DR5: Provide an interpretable quality score. A readable, normalized quality score should indicate the usefulness of a ticket in terms of quality [11]. The quality score can be integrated into the TRS as complementary information for recommendation ranking and selection. However, primarily the score is utilized to filter a given data set.

4.2 Development and Demonstration

We combine machine learning models and feature engineering to predict ticket quality and gain transparency [42]. Analogous approaches have been performed on various text classification cases in the context of AI-ITSM [18, 20, 21]. Using the derived requirements from literature and practice, we describe our development and demonstration phase by elaborating on design principles in the following.

DP1 – Data quality conceptualization. The underlying optimization goal is to improve recommendation system performance while at the same time ensuring readability and usefulness. With reflections from the first iteration, we observed that issue and resolution descriptions possess different quality characteristics and hence as per design requirements (DR3), we differentiate between issue and resolution descriptions. Based on a workshop on ticket quality with IT support agents and quality managers, an analysis of the ticket data set, and a literature review on ticket and text analytics, we hand-selected a set of more than 30 features depending on the description type (DR4). We rely on linguistic features to address the limitation of other text representation techniques, primarily weighted words (Bag of word, TF-IDF) and word embeddings (Word2Vec, Doc2Vec) such as effortful training, incapacity of capturing word semantic similarity and limited corpus of words [18]. Additionally, linguistic features provide transparency of the characteristics of data quality (DR2).

DP2 – Data retrieval and preprocessing. The upstream steps of the pipeline comprise the retrieval and pre-processing of data [20, 30]. The pipeline first loads tickets from an ITSM platform, which are further filtered by default categories such as status and channel type. We rely on a set of 60,000 support tickets from 2021, which were provided by an international manufacturing company. The dataset was extracted from a ServiceNow environment and contains standard data fields for support tickets. We further anonymized sensitive information such as name, location, and mail [1]. Furthermore, pre-processing includes analyzing multiple ticket-related text fields including a short description, working notes, comments, and closing notes, and merging them into two fields: issue and resolution. For several linguistic features, our pipeline applies additional pre-processing steps such as the removal of links, attachments, and mail signatures, tokenization, lemmatization, and removal of special characters [48].

DP3 - Domain Knowledge Integration. To extract as many highly relevant tickets, and identify useful problem-solution matches (DR1), we apply a BERT-based topic modeling approach [31, 49]. Because topics could be redundant, we add an agglomerative hierarchical clustering approach [50] to aggregate topic clusters [51] and to automatically derive resolution clusters for later TRS training and testing (Table 1). The topic clusters, their keywords, and exemplary tickets indicate the quality and usefulness of the inherent set of similar tickets (DR1). The labeling process starts with labeling the topic clusters on a simple binary scale by two annotators. After annotating ten topics both annotators and the researchers discussed the results and aligned their approaches. We tested the inter-rater reliability and archived a substantial agreement (cohen's kappa = 0.682). In sum, we derived 863 topics and 215 resolution clusters for the given data set. Then, the annotators were instructed by the set of features and examples for ticket quality and were provided with a different set of labeling instructions for issue and resolution descriptions (DR3). For issue description quality we archived an agreement of 0.546 and for resolution description a cohen's kappa of 0.439. Participants of the labeling process were two researchers of TRS with expertise in the field of ITSM.

Table 1. Example of useful topics and a derived topic cluster

Cluster	Label	Topics
120	Power Bi license	**Topic 36:** pro, bi, power, fulfillment, licence, added, license,
		Topic 389: pro, license, bi, fulfilment, power, pbi, premium

DP4 - Extensive Feature Engineering. Above mere feature calculation, additional goals of this stage are to eliminate constant attributes, eliminate redundant features and analyze features' influence on ticket data quality [42]. Part of feature selection spans the reduction of features, which is a common challenge in statistics [52]. First, we remove all constant or quasi-constant features (e.g., sentiment score, language confidence, and spelling mistakes) by utilizing a threshold of 0.05 for variance, which we derived by experimentally applying different thresholds and evaluating the attached model performance. Next, a correlation analysis was conducted, and we removed features that correlate strongly (> 0.95) and possess less importance (e.g., stop words count, words count). Our analysis revealed that the resolution description results in more correlating features. Random Forest Classifier revealed to possess a comparatively high performance and enables analyzing the embedded feature importance (DR2). The top features according to importance calculated with help of the SHAP framework are ranked within the following Fig. 2.

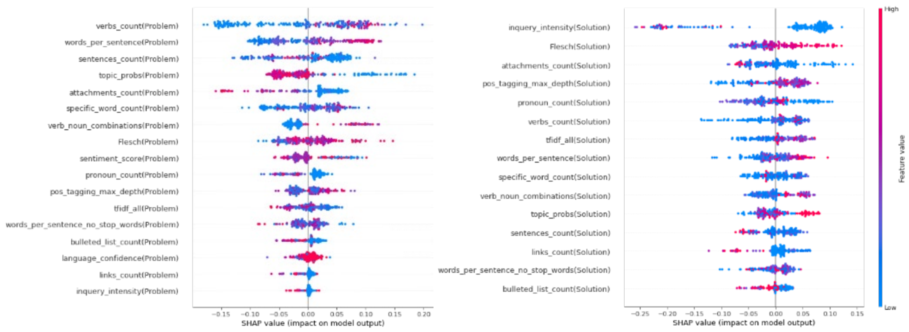

Fig. 2. Most influencing features on issue (left) and resolution (right) quality.

DP5 - Quality Scoring Model. Because deep learning-based models lack explainability and interpretability [41], we rely on certain machine learning-based models for further insights into feature importance (DR2). The analytics pipeline makes use of prior work and experiences with different ML-based classifiers in the AI-ITSM domain [18, 21]. Accordingly, common classifiers are Support Vector Machine (SVM), Random Forest Classifier (RF), Stochastic Gradient Descent Classifier (SGD), Logistic Regression (LR), and K-Nearest Neighbors (KNN) [29, 53]. Table 2 provides an overview of the applied ML-based classifiers trained on a preliminary balanced set of 160 labeled tickets differentiating between classifiers for the issue and resolution description quality (DR3). Based on the prediction we can filter the large ticket database on the highest level of quality (DR5), as shown in the evaluation part. In addition, the score can be used to influence recommendation ranking and help agents record tickets of high quality.

4.3 Pipeline Reference Model for Textual Data Quality Enrichment

Given the developed pipeline, we aggregated and abstracted our findings into a reference model for designing pipelines for data enrichment in the realm of textual data. The pipeline reference model in Fig. 3 demonstrates a generalized view of modern machine learning-based data quality assessment for AI-enabled systems. We differentiate between four key components: the data quality model, knowledge model, feature model, and scoring and operations model following Frank [54]. The data model requires the incorporation of domain-specific knowledge regarding the context-specific quality of a text. According to the literature on data quality assessment [45], developers have to define objectives, quality dimensions, and measures in a top-down manner. The resulting data quality model informs the calculation and analysis of the measures – referred to as the feature model. Domain-specific knowledge is integrated through clustering and labeling the data. By applying methods for explainability and transparency [42], organizations can get a better understanding of the underlying data quality and can optimize their prediction models accordingly. Finally, features are transferred to the scoring and operations model, where the data set is scored based on a machine learning-based classifier and then filtered to provide high-quality data to the attached recommender system. Data quality analytics pipelines should include feedback mechanisms from humans to machines and vice versa. During operations, users rate the recommended tickets regarding data quality, while the feature model provides recommendations to the user for improving an incoming ticket. The last two elements are out of the scope of this study.

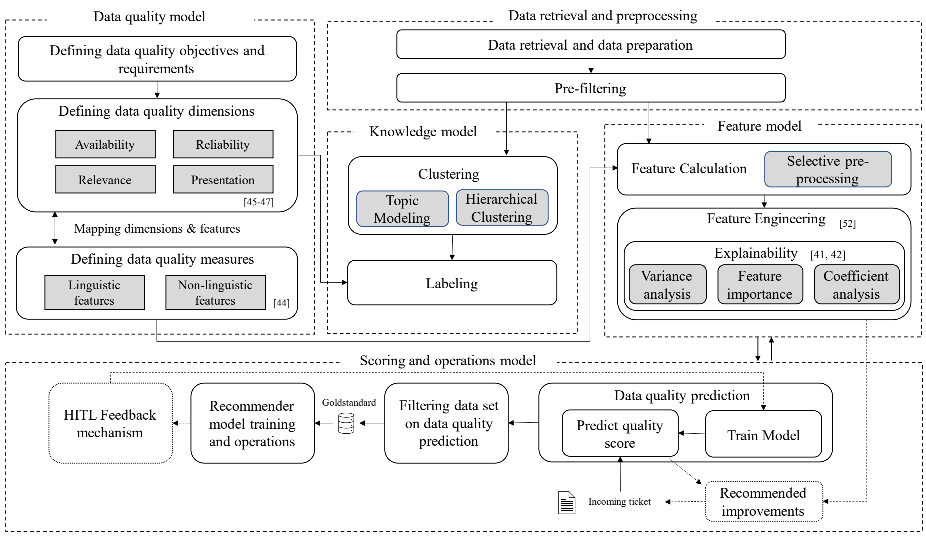

Fig. 3. Pipeline reference model for textual data quality enrichment.

4.4 Evaluation

We tested the pipeline on a dataset mentioned in section 4.2. To confirm that the designed artifact, the pipeline reference model, generates a better assessment of data quality and subsequently improves the performance of the attached TRS, this study relies on a two-leveled technical evaluation (Fig. 4) [55].

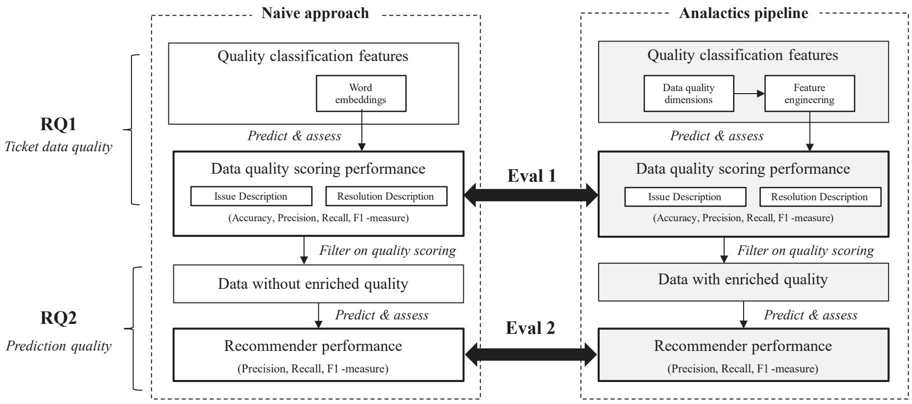

Fig. 4. Evaluation framework for the designed analytics pipeline

First evaluation (Eval 1)– Comparing data quality scoring performance.

Within our first evaluation phase, we compare the performance of the scoring model of the ticket analytics pipeline with a naïve text classifier and thereby answer our first research question. The simple classifier applies a TF-IDF-based text classifier [44]. Table 2 indicates that the developed pipeline can improve the performance of text quality assessment for both issue and resolution. However, the impact is more prevalent in the case of resolution descriptions.

Table 2. Eval 1- Overall results of the first evaluation of the data quality scoring model [1].

Type	Cl. [2]	Naïve approach				Ticket analytics pipeline			
		A	R	P	F1	A	R	P	F1
Issue Description	SVM	0.794	0.481	0.410	0.442	0.762	0.637	0.662	0.646
	RF	0.810	0.490	0.411	0.447	0.786	0.688	0.702	***0.694***
	SDG	0.825	0.608	0.680	***0.626***	0.357	0.544	0.557	0.354
	LR	0.825	0.500	0.413	0.452	0.524	0.584	0.562	0.506
	KNN	0.746	0.524	0.528	0.525	0.690	0.591	0.585	0.587
Resolution Description	SVM	0.825	0.500	0.412	0.452	0.537	0.536	0.535	0.532
	RF	0.825	0.535	0.668	***0.528***	0.732	0.728	0.725	***0.726***
	SDG	0.777	0.506	0.513	0.498	0.390	0.471	0.200	0.281
	LR	0.825	0.500	0.412	0.452	0.610	0.529	0.800	0.431
	KNN	0.746	0.523	0.527	0.524	0.732	0.719	0.724	0.721

[1] A= Accuracy; R=Recall; P=Precision; F1 = F1-Score
[2] Cl. = Classifier for predicting a quality score for a given support ticket

Interestingly, the naïve approach achieves higher accuracy results. However, as the goal is to identify high-quality data, precision, and recall are more important criteria, as the ability to identify true positives is more important for the underlying problem and the aim of extracting high-quality data. We argue that it is more crucial for an automated ticket recommender system, which is based on a large dataset, to miss appropriate recommendations than to make inappropriate ones. Our pipeline outperforms the simple pipeline in terms of recall, precision, and F1 score, leading to more reliable prediction results. In the case of issue description, the ticket pipeline achieves an F1-score of 0.69 by utilizing an RFC while the naïve approach shows a score of 0.63 applying SDG. For resolution description the results are more impressive: The pipeline shows a 0.2 higher F1-score by comparing the Random Forest Classifier. Additionally, classifiers based on word embeddings cannot provide insights into what determines ticket quality and how data quality can be improved in the long term. Our scoring model not only outperforms these approaches but ensures transparency and interpretability [41, 42].

Second Evaluation (Eval 2) – Comparing recommender system performance.
Eval 2 includes comparing recommendation systems using the prior described simple word embedding-based classifier and the here-designed enhanced analytics pipeline. In contrast to Eval 1, we do test how better data quality through filtering enables providing better recommendations. We utilize prediction performance (mean average precision, mean average recall, f-score) as the by far most established quality metric in the recommendation systems literature [56]. Accuracy, as used in Eval 1, can not be utilized for evaluating recommender systems because they output multiple options and thus are different from classification problems. From clustering topics of tickets, we derived problem-solution-pairs to evaluate the TRS. We evaluate a TF-IDF- and cosine similarity-based recommender system, that compares a given query with the set of solved tickets and proposes three tickets as possible resolutions. Given the best classifier of analytics pipeline and naïve classification (Eval 1), we filtered the data set on issue and solution description quality. Then we initiated the recommender systems on 205 problem-solution pairs. The results (Table 3) demonstrate that a prior data enrichment can slightly improve TRS performance by 0.5 points in terms of mean average precision (MAP) and can reach a total mean average recall of 0.75 and a total F-Score of 0.73. Despite showing lower accuracy values for quality scoring, the analytics pipeline outperforms a naïve method, which confirms the relevance of precision and recall.

Table 3. Eval 2 – Ticket recommender system performance evaluation

Type	Precision	Recall	F-Score
Naïve pipeline	0.667	0.713	0.689
Analytics pipeline	0.717	0.747	0.732

5 Discussion, Limitations, and Implications

Our artifact provides innovative insights into how to address the challenge of data quality and TRS performance [2, 6]. The analytics pipeline presented as a novel solution for AI-ITSM [57] in this paper is contributing to the emerging stream in data management literature to investigate machine learning techniques for auditing and curating data quality. [58]. Furthermore, we add to the research on data quality in recommendation systems [12], by revealing how the quality of ticket data affects the overall performance of recommendations. With our pipeline principles and reference model, we provide a general method and instantiation guidelines for future ticket recommender systems and answer the first research question. As a result, a comprehensive set of linguistic features and a scoring model for the ticket documentation was derived and domain-specific knowledge was incorporated through topic modeling, clustering, and labeling. The second research question was answered by evaluating an attached recommender system. Since our artifact is not based on predefined rules, it enables customizability, scalability, and effortless maintenance of data enrichment in practice. Contrary to non-transparent word embedding methods or deep learning models, the extensive feature engineering approach guarantees transparency and explainability of the ticket quality. With our paper, we show how DSR can be successfully applied to machine learning projects. Machine learning artifacts are rarely developed using DSR. However, combining a technical and a design perspective is useful to structure and guide the design project.

In addition to our designed DSR artifact and the application of DSR according to Peffers et al. [35], we codify our knowledge in a conceptual model. Reference models enable domain knowledge to be generalized and thus made transferable. Due to the generalizability, the findings are made usable for several use cases. Therefore, reference models enable what is an important criterion in the codification of design knowledge, namely the generalizability of the knowledge to be able to apply and reuse it in new situations. Being an abstract representation of domain knowledge, reference models codify prescriptive and descriptive design knowledge and facilitate the reusability of design knowledge [59].

Despite the previously mentioned contributions and implications, our research comes with certain limitations. First of all, the performance results do not improve significantly and should be improved. Further research could extend the feature selection process by more advanced feature selection techniques and consult even more relevance-related features (e.g., likes, star ratings, comments, click-rate, etc.) as well as domain knowledge after introducing an improved recommender system in IT service organizations. New approaches of ensemble methods and hybrid models of word embeddings, linguistic features, and especially large language models remain unconsidered. In addition, an evaluation with users should be conducted to validate the TRS performance in terms of usefulness. Additionally, within further research individual analysis of features and their impact on recommender performance could be conducted to improve data quality sustainability within IT support organizations and pave the way for highly performant AI-ITSM systems. Another limitation comes with the restriction to one case company. Testing the pipeline and applying the pipeline reference model to another set of IT support tickets could strengthen the here-stated results and implications. Simultaneously, further research could examine different types of machine learning mechanisms and reveal how the pipeline impacts the performance of other recommendation models above a basic TF-IDF-based similarity model.

References

1. Swain AK, Garza VR (2022) Key Factors in Achieving Service Level Agreements (SLA) for Information Technology (IT) Incident Resolution. Inf Syst Front:1–16. doi:10.1007/s10796-022-10266-5
2. Schmidt S, Li M, Peters C (2022) Requirements for an IT Support System based on Hybrid Intelligence Proceedings of the Annual Hawaii International Conference on System Sciences. Hawaii International Conference on System Sciences
3. Paramesh SP, Ramya C, Shreedhara KS (2018) Classifying the Unstructured IT Service Desk Tickets Using Ensemble of Classifiers 3rd International Conference on Computational Systems and Information Technology for Sustainable Solutions (CSITSS). IEEE
4. Al-Hawari F, Barham H (2021) A machine learning based help desk system for IT service management. Journal of King Saud University - Computer and Information Sciences 33(6):702–718. doi:10.1016/j.jksuci.2019.04.001
5. Fuchs S, Drieschner C, Wittges H (2022) Proceedings of the 55th Hawaii International Conference on System Sciences (HICSS). University of Hawai'i at Manoa Hamilton Library, Honolulu, Hawai
6. Meng FJ, Xu J, Zhang X, Yang L, Chen P, Wang Y, Liu X, Ayachitula N, Murthy K, Shwartz L, Galambos G, Su Z, Zheng J (2018) Opportunities and Challenges Towards Cognitive IT Service Management in Real World IEEE Symposium on Service-Oriented System Engineering (SOSE). IEEE
7. Ali Zaidi SS, Fraz MM, Shahzad M, Khan S (2022) A multiapproach generalized framework for automated solution suggestion of support tickets. Int J of Intelligent Sys 37(6):3654–3681. doi:10.1002/int.22701
8. Zhou W, Xue W, Baral R, Wang Q, Zeng C, Li T, Grabarnik GY (2017) Star: A system for ticket analysis and resolution Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, S 2181–2190
9. Schmidt SL, Li MM, Weigel S, Peters C (2021) Knowledge is Power: Provide Your IT-Support with Domain-Specific High-Quality Solution Material. Springer, Cham, S 209–222
10. Peters C, Blohm I, Leimeister JM (2015) Anatomy of Successful Business Models for Complex Services: Insights from the Telemedicine Field. Journal of Management Information Systems 32(3):75–104. doi:10.1080/07421222.2015.1095034
11. Agarwal S, Sridhara G, Dasgupta G (2016) Automated Quality Assessment of Unstructured Resolution Text in IT Service Systems International Conference on Service-Oriented Computing, S 221–235
12. Heinrich B, Hopf M, Lohninger D, Schiller A, Szubartowicz M (2021) Data quality in recommender systems: the impact of completeness of item content data on prediction accuracy of recommender systems. Electron Markets 31(2):389–409. doi:10.1007/s12525-019-00366-7
13. Heinrich B, Hopf M, Lohninger D, Schiller A, Szubartowicz M (2022) Something's Missing? A Procedure for Extending Item Content Data Sets in the

Context of Recommender Systems. Inf Syst Front 24(1):267–286. doi:10.1007/s10796-020-10071-y

14. Wang Q, Zhou W, Zeng C, Li T, Shwartz L, Grabarnik GY (2017) Constructing the Knowledge Base for Cognitive IT Service Management IEEE International Conference on Services Computing (SCC). IEEE
15. Konstan JA, Riedl J (2012) Recommender systems: from algorithms to user experience. User Model User-Adap Inter 22(1-2):101–123. doi:10.1007/s11257-011-9112-x
16. Picault J, Ribière M, Bonnefoy D, Mercer K (2011) How to Get the Recommender Out of the Lab? In: Ricci F, Rokach L, Shapira B, Kantor PB (Hrsg) Recommender Systems Handbook. Scholars Portal, Boston, MA, S 333–365
17. Sar Shalom O, Berkovsky S, Ronen R, Ziklik E, Amihood A (2015) Data Quality Matters in Recommender Systems. In: Werthner H, Zanker M, Golbeck J, Semeraro G (Hrsg) Proceedings of the 9th ACM Conference on Recommender Systems. ACM, New York, NY, S 257–260
18. Revina A, Buza K, Meister VG (2020) IT Ticket Classification: The Simpler, the Better. IEEE Access 8:193380–193395. doi:10.1109/access.2020.3032840
19. Koehler J, Fux E, Herzog FA, Lötscher D, Waelti K, Imoberdorf R, Budke D Towards Intelligent Process Support for Customer Service Desks: Extracting Problem Descriptions from Noisy and Multi-lingual Texts, S 36–52
20. Marcuzzo M, Zangari A, Schiavinato M, Giudice L, Gasparetto A, Albarelli A (2022) A multi-level approach for hierarchical Ticket Classification Proceedings of the Eighth Workshop on Noisy User-generated Text (W-NUT 2022), S 201–214
21. Zicari P, Folino G, Guarascio M, Pontieri L (2021) Discovering accurate deep learning based predictive models for automatic customer support ticket classification Proceedings of the 36th Annual ACM Symposium on Applied Computing. ACM, New York, NY, USA
22. Agarwal S, Aggarwal V, Akula AR, Dasgupta GB, Sridhara G (2017) Automatic problem extraction and analysis from unstructured text in IT tickets. IBM J. Res. & Dev. 61(1):4:41-4:52. doi:10.1147/jrd.2016.2629318
23. Li MM, Peters C, Leimeister JM (2017) Designing a Peer-Based Support System to Support Shakedown. International Conference on Information Systems (ICIS), Seoul, South Korea
24. Zschech P (2022) Beyond descriptive taxonomies in data analytics: a systematic evaluation approach for data-driven method pipelines. Inf Syst E-Bus Manage:1–35. doi:10.1007/s10257-022-00577-0
25. Batini C, Barone D, Mastrella M, Maurino A, Ruffini C (2007) A Framework And A Methodology For Data Quality Assessment And Monitoring. ICIQ:333–346
26. Bharati P, Chaudhury A (2004) An empirical investigation of decision-making satisfaction in web-based decision support systems. Decision Support Systems 37(2):187–197. doi:10.1016/S0167-9236(03)00006-X

27. Feldman M, Even A, Parmet Y (2018) A methodology for quantifying the effect of missing data on decision quality in classification problems. Communications in Statistics-Theory and Methods 47(11):2643–2663
28. Woodall P, Borek A, Gao J, Oberhofer MA, Koronios A (2014) An Investigation of How Data Quality is Affected by Dataset Size in the Context of Big Data Analytics ICIQ
29. Zicari P, Folino G, Guarascio M, Pontieri L (2022) Combining deep ensemble learning and explanation for intelligent ticket management. Expert Systems with Applications 206:117815. doi:10.1016/j.eswa.2022.117815
30. Rizun N, Revina A, Meister VG (2021) Assessing business process complexity based on textual data: Evidence from ITIL IT ticket processing. BPMJ 27(7):1966–1998. doi:10.1108/BPMJ-04-2021-0217
31. Devlin J, Chang M-W, Lee K, Toutanova K (2019) BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. arXiv
32. Baresi L, Quattrocchi G, Tamburri DA, van Heuvel W-J den (2020) Automated quality assessment of incident tickets for smart service continuity International Conference on Service-Oriented Computing, S 492–499
33. Cavalcanti YC, Da Mota Silveira Neto PA, Machado IdC, Vale TF, Almeida ES de, Meira, Silvio Romero de Lemos (2014) Challenges and opportunities for software change request repositories: a systematic mapping study. Journal of Software: Evolution and Process 26(7):620–653
34. Sonnenberg C, vom Brocke J (2012) Evaluations in the Science of the Artificial – Reconsidering the Build-Evaluate Pattern in Design Science Research. Springer, Berlin, Heidelberg, S 381–397
35. Peffers K, Tuunanen T, Rothenberger MA, Chatterjee S (2007) A Design Science Research Methodology for Information Systems Research. Journal of Management Information Systems 24(3):45–77. doi:10.2753/MIS0742-1222240302
36. Brendel AB, Lembcke T-B, Muntermann J, Kolbe LM (2021) Toward replication study types for design science research. Journal of Information Technology 36(3):198–215. doi:10.1177/02683962211006429
37. Chandra Kruse L, Nickerson JV (2018) Portraying Design Essence
38. Legner C, Pentek T, Otto B (2020) Accumulating Design Knowledge with Reference Models: Insights from 12 Years' Research into Data Management. JAIS 21(3):735–770. doi:10.17705/1jais.00618
39. Das A (2003) Knowledge and Productivity in Technical Support Work. Management Science 49(4):416–431. doi:10.1287/mnsc.49.4.416.14419
40. Elshan E, Ebel PA, Söllner M, Leimeister JM (2022) Leveraging Low Code Development of Smart Personal Assistants: An Integrated Design Approach with the SPADE Method. Journal of Management Information Systems (JMIS)
41. Wambsganß T, Engel C (2021) Using Deep Learning for Extracting User-Generated Knowledge from Web Communities
42. Zacharias J, Zahn M von, Chen J, Hinz O (2022) Designing a feature selection method based on explainable artificial intelligence. Electron Markets:1–26
43. Pitler E, Nenkova A (2008) Revisiting readability: A unified framework for predicting text quality. In: Lapata M, Tou NG H (Hrsg) Proceedings of the 2008

conference on empirical methods in natural language processing, Honolulu, Hawai, S 186–195
44. Landolt S, Wambsganss T, Söllner M (2021) A Taxonomy for Deep Learning in Natural Language Processing. Hawaii International Conference on System Sciences, Hawaii
45. Cai L, Zhu Y (2015) The Challenges of Data Quality and Data Quality Assessment in the Big Data Era. CODATA 14(0):2. doi:10.5334/dsj-2015-002
46. Pipino LL, Lee YW, Wang RY (2002) Data quality assessment. Commun. ACM 45(4):211–218. doi:10.1145/505248.506010
47. Batini C, Cappiello C, Francalanci C, Maurino A (2009) Methodologies for data quality assessment and improvement. ACM Comput. Surv. 41(3):1–52. doi:10.1145/1541880.1541883
48. Subbarao MV, Venkatarao K, Suresh C (2022) Automation of Incident Response and IT Ticket Management by ML and NLP Mechanisms. Journal of Theoretical and Applied Information Technology 100(12):3945–3955
49. Vayansky I, Kumar SA (2020) A review of topic modeling methods. Information Systems 94:101582. doi:10.1016/j.is.2020.101582
50. Bouguettaya A, Yu Q, Liu X, Zhou X, Song A (2015) Efficient agglomerative hierarchical clustering. Expert Systems with Applications 42(5):2785–2797. doi:10.1016/j.eswa.2014.09.054
51. Lee HJ, Lee M, Lee H, Cruz RA (2021) Mining service quality feedback from social media: A computational analytics method. Government Information Quarterly 38(2):101571
52. Liu JY, Zhong W, Li R (2015) A selective overview of feature screening for ultrahigh-dimensional data. Science China Mathematics 58(10):2033–2054. doi:10.1007/s11425-015-5062-9
53. Oliveira DF, Nogueira AS, Brito MA (2022) Performance Comparison of Machine Learning Algorithms in Classifying Information Technologies Incident Tickets. AI 3(3):601–622. doi:10.3390/ai3030035
54. Frank U (2007) Evaluation of reference models Reference modeling for business systems analysis. IGI Global, S 118–140
55. Venable J, Pries-Heje J, Baskerville R (2016) FEDS: a Framework for Evaluation in Design Science Research. Eur J Inf Syst 25(1):77–89. doi:10.1057/ejis.2014.36
56. Shani G, Gunawardana A (2011) Evaluating Recommendation Systems Recommender Systems Handbook. Springer, Boston, MA, S 257–297
57. Gregor S, Hevner AR (2013) Positioning and Presenting Design Science Research for Maximum Impact. MISQ 37(2):337–355. doi:10.25300/misq/2013/37.2.01
58. Chua C, Indulska M, Lukyanenko R, Maass W, Storey VC (2022) MISQ Research Curation on Data Management. MISQ Research Curations:1–12
59. Schermann M, Böhmann T, Krcmar H (2009) Explicating design theories with conceptual models: Towards a theoretical role of reference models Wissenschaftstheorie und gestaltungsorientierte Wirtschaftsinformatik. Springer, S 175–194