Automatic Q.A-Pair Generation for Incident Tickets Handling: An Application of NLP

Mick Lammers[1], Fons Wijnhoven[1], Faiza A. Bukhsh[1](✉), and Patrício de Alencar Silva[2]

[1] Department of Computer Science, University of Twente, 7500AE Enschede, The Netherlands
mick-lammers@hotmail.com,
{a.b.j.m.wijnhoven,f.a.bukhsh}@utwente.nl

[2] Programa de Pós-Graduação em Ciência da Computação, Universidade Federal Rural do Semi-Árido (UFERSA), Mossoró, Rio Grande do Norte, Brazil
patricio.alencar@ufersa.edu.br

Abstract. Chatbots answer customer questions by mostly manually crafted Question Answer (Q.A.)-pairs. If organizations process vast numbers of questions, manual Q.A. pair generation and maintenance become very ex-pensive and complicated. To reduce cost and increase efficiency, in this study, we propose a low threshold QA-pair generation system that can automatically identify unique problems and their solutions from a large incident ticket dataset of an I.T. Shared Service Center. The system has four components: categorical clustering for structuring the semantic meaning of ticket information, intent identification, action recommendation, and reinforcement learning. For categorical clustering, we use a Latent Semantic Indexing (LSI) algorithm, and for the intent identification, we apply the Latent Dirichlet Allocation (LDA), both Natural Language Processing techniques. The actions are cleaned and clustered and resulting Q.A. pairs are stored in a knowledge base with reinforcement learning capabilities. The system can produce Q.A. pairs from which about 55% are useful and correct. This percentage is likely to in-crease significantly with feedback in its usage stage. By this study, we contribute to a further understanding of the development of automatic service processes.

Keywords: Service request handling · Service management · Q.A. pair generation system · ICT user support management · Natural language processing

1 Introduction

I.T. Shared Service Centers are the beating heart of large organizations. They take on everything that has to do with the facilitation of I.T., like personal computers, mobile devices, workplaces, servers, applications, and VPN's. I.T. Incident management is a large part of shared service centers' responsibility [7]. As of now, incident management is performed in almost all service centers using a ticketing system. A ticketing system registers incident calls and requests for service from clients. The tickets are then either sent to persons who can act on them or persons who know most about the context of

K. Djemame et al. (Eds.): GECON 2020, LNCS 12441, pp. 15–27, 2020.
https://doi.org/10.1007/978-3-030-63058-4_2

these tickets. Especially in highly complex large-scale environments, the existing ticking systems would be most useful but are less effective because of difficulties in generating Q.A. (which stands for question-answer) pairs and high costs of maintaining Q.A. pairs manually [2, 6]. In this research, a system is designed by which the ticket data is used to create this actionable knowledge in a manner that limits the amount of manual work in QA-pair creation and maintenance using Natural Language Processing and Machine Learning. The objective of this research is to "to find an optimal design for a low-cost QA-pair generation system for a large-scale I.T. incident tickets dataset."

A state-of-the-art research is performed to identify components and techniques in QA-pair generation in Sect. 2. We provide summaries of related work and draw design conclusions for our solution in Sect. 3. Based on this literature study, we define the research gap and goals, and we build our own solution of categorizing incidents, ticket intents, and solutions in Sect. 4.

We demonstrate and test the proposed solution by the case of the SSC-ICT IT Shared Service Center of 8 Dutch ministries. SSC-ICT supports about 40,000 civil servants who almost all have a laptop and phone to be supported as well as a virtual working environment for performing their jobs. Furthermore, SSC-ICT provides services for over one thousand applications and receive around 30,000 tickets a month in ticket management system TopDesk, mainly via phone (60%), e-mails (15%), and face-to-face contact (10%). Given the highly textual nature of Q.A. pairs, natural language processing seems to be particularly useful in Q.A. pair generation. After designing this system, we evaluate its effectiveness, draw generalizable conclusions, and define the needs for further research.

2 State-of-the-Art

QA pairs have a question and an answer. In incident management, the question is often referred to as "intent." The intent is the user's intent for creating the ticket. The answers are called actions, resolutions, or just answers. Previous studies that describe the development of Q.A. pair generating systems are described below. These studies were found using the literature research methodology of [15]. In total, 200 articles are found using forward and backward snowballing. After inclusion/exclusion criteria (for details see [9]), we have selected 60 most relevant articles. In the following, we will highlight only a few.

The study found in [5] designed a cognitive support system for a specific client with 450 factories operating in 190 countries. For extracting the intents, they used a combination of n-gram and Lingo techniques [11], as well as field experts to manually identify intents. Another very well-known system used by [1] describes a cognitive system developed by researchers from IBM for a service desk. The knowledge extraction processes applied is divided into three steps: problem diagnosis, root cause analysis, and resolution recommendation. A similar study found in [12] designed a system to automatically analyze natural language text in network trouble tickets. Their case is a large cloud provider of whom they analyze over 10,000 tickets. An overview of the different steps and knowledge discovery techniques mentioned in well-known studies is given in Table 1.

Table 1. Q.A.-pair identification steps and techniques from the literature.

Author	Q.A.-pair identification steps					Identification techniques		
	Category clustering	Root-cause analysis	Intent identification	Resolution finding	Reinforcement learning	Unsupervised		Supervised
						POS patterns	Topic modeling	Classifier
N. Berente et al. (2019)		●		●			●	●
Suman et al. (2018)			●				●	
P. Dhoolia et al. (2017)		●	●	●	●	●		
S. Agarwal et al. (2017)		●	●	●				●
Vlasov et al. (2017)		●	●	●				●
Mani et al. (2014)	●		●	●		●		
Jan et al. (2014)			●				●	
Potharaju & Nitarotaru (2013)			●	●		●		

3 Design Principles of Q.A.-Pair Generation

All the articles discussed use an intent identification process as well as a resolution recommendation process (except for [8] who focus on intent identification techniques only). Reinforcement learning and root cause analysis are used only in a small number of articles. Root cause analysis is used where the datasets are smaller in contrast to reinforcement learning that is more valuable with larger numbers of tickets and potential feedback mechanisms.

The largest dataset used in the described articles has 80,000 tickets, less than half of the number of tickets of this research. Consequently, the datasets from the articles have fewer categories, and they identify relatively few problems, 130 at the most, then the expected 1,000 problems from SSC-ICT.

This, along with tests that showed that clustering techniques on the complete corpus showed inconsistent clustering results. Moreover, good results on using a Latent Semantic Analysis (LSA) based method for grouping tickets based on subjects provide the foundation to add a component to the pipeline, which we call categorical clustering. In this step, we first group the tickets in large categories. After that, we apply for each category a unique iteration of the intent identification component.

Furthermore, we decided not to implement Root Cause Analysis in this iteration of the system due to a lack of resources. A methodological overview of the steps followed is provided in Fig. 1.

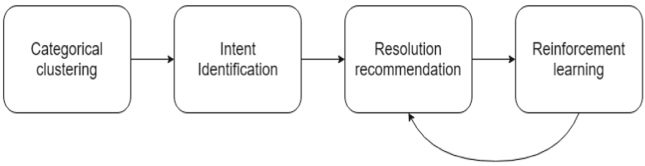

Fig. 1. Four steps of a Q.A.-pair generation system

In **step one**, the tickets need to be ordered on categories, because detecting intents right away leads to very inconsistent and noisy clusters. For identifying categories, keyword based-clusters (supervised) and word-embedding based clustering (unsupervised) are mentioned in the literature [3]. The downside to keyword-based categorization is that unimportant words like operations or adjectives may also be identified as clusters. Therefore, Categorization using word-embeddings, Latent Semantic Analysis (LSA), is the best method for this process, as it benefits from the single keyword categories, and it excludes low-informative words automatically.

Step two involves intent identification or problem identification by which specific problems are identified from tickets. This can be done by a supervised learning methodology in which intents are identified beforehand, and new tickets are classified based on one of these intents or in an unsupervised way in which topics are created using either POS patterns in tickets or from topical word embeddings. Supervised intent identification is most effective in a rule-based environment. Unsupervised methodologies for intent identification are either word embeddings (LDA/LSA) or patterns in word or POS forms.

Step three identifies resolutions or action recommendations (i.e. the A in Q.A.) from resolution texts. In this process, action fields are cleaned from source-related or e-mail related noise. Furthermore, hot sentences are extracted, and duplicate actions are removed. The sorting and providing of these actions are improved by step four.

Step four involves the process of increasing the accuracy of the system based on client feedback. Client feedback will act as being the assessor on the accuracy of the action recommendation of the system. This assessment can then be used to classify the action as relevant or irrelevant to the intent, based on which new intents can be solved better. Relevant examples of feedback mechanisms are the number of clicks on a specific action, a like/dislike option, or search history.

4 Design of Q.A.-Pair Generation System

4.1 Ticket Data Description

For our case, the ticket data includes a dataset from the start of February 2018 till the 31st of December 2018. This is a dataset of 340,000 tickets with 40+ attributes. We focused on all first-line tickets, and with this step, we exclude 40,000 tickets. Then, we chose to include only incidents, requests for service, and requests for information. Other ticket types were mainly computer-generated tickets and, therefore, not of interest to this research. This results in a final dataset of 210,000 tickets. The selected tickets have the attributes listed in Table 2.

The 'short description' (containing intent information) and the 'action' fields are the main sources for Q.A. pair generation. The request field appeared too inconsistent for use. We keep the request field, the category, and subcategory fields out of this research scope because these categories are not problem-focused.

4.2 Categorical Clustering

The column with the "short description" along with their ticket id's, is exported from the excel dataset and converted to the XML-format. This is a file of 450,000 lines. For the categorical clustering, three techniques are attempted based on outcomes of the state-of-the-art research: LDA, POS Patterns, and Lingo3G clustering. LDA did not show good results. The resulting clusters are overlapping. POS patterns were also not effective. The POS patterns were too specific and did not capture the global category. Lingo3G, however, worked very well on the dataset. After having tweaked with the attribute settings, amongst other things promoting short (one-word) labels and increasing the expected number of clusters, a process-based ticket cluster overview appeared (vide Fig. 2). For Lingo3G, we used the custom parameters on top of the standard parameters as shown in Table 3.

Lingo3G applies a custom version of LSA (Latent Semantic Analysis) using Term Frequency – Inverse Document Frequency (TF-IDF) word embeddings on a text corpus and then applying Singular Value Decomposition (SVD) for dimensionality reduction. Its algorithm consists of preprocessing, frequent phrase extraction, cluster label induction, and cluster content discovery steps.

Table 2. Attributes used for Q.A.-pair generation

Data field	Description
Ticket id	A unique id for each ticket, automatically generated
Short description	A summary of the ticket problem, written by the service desk operator
Request	The full description of the ticket, in case of an e-mail, the full e-mail is displayed here. In other cases, it is like a short description
Action	A summary of the suggested action steps by the operator
Type of ticket	Type of customer request, such as a request for service, internal management notification, request for information, security incident, SCOM (a monitoring system), complaint
Category	The highest level of Categorization: User-bound services, Applications, Premise-bound services, Housing & hosting, Security
Subcategory	Each of the main categories has at least five subcategories. In total there are 42 subcategories. 50% of the tickets are covered by three subcategories: location specific services, housing and hosting services, and security services
Practitioners group	This is the division that solved the ticket, 85% of the ticket has the service desk as practitioner group, the other tickets are solved by about 300 different small groups
Entry type	The means by which the customer contacted the service desk: telephone, e-mail, physical service desk, portal, website, manually

Table 3. Custom parameters for Lingo3G application

Parameter	Description
Minimum cluster size: 0.0010%	Lowers the threshold for minimal cluster size
Cluster count base: 20	Increases the number of resulting clusters
gMaximum hierarchy depth: 1	Limits optional clustering depth to 1 layer
Phrase-DF cut-off scaling: 0.20	Decreases the length of labels
Word-DF cut-off scaling: 0.00	Further limits the length of labels to 1 word
Maximum top-level clustering passes: 8	Increases the computational effort
Default clustering language: Dutch	Change NLP language to Dutch

The preprocessing step removes stop words from an external list that is created by a field expert. This also identifies synonyms and label name. Because the input consists of only one sentence, we skip the frequent phrase extraction process. Lingo3G generates 138 clusters from the ticket data. With the largest being 10% of the whole ticket corpus and the smallest 0.05%. The ten largest clusters accumulate to 65% of the ticket corpus, 15% is part of the other 107 clusters, 20% is not categorizable.

Evaluation
We evaluated the categories manually on hierarchical independence, synonymity, and informativity. Synonymity is about whether there is another synonymous cluster or not, for instance, in the case of SSC-ICT, the clusters "mail" and "e-mail." Hierarchical independence is about whether a cluster should be subjective to another cluster. For instance, the tickets in the cluster "paper" are 95% of the time regarding "printer" and should, therefore, be part of the "printer" cluster. Further Categorization would then happen in the next component, intent identification. Relevance is about whether a category describes an actual single subject. For instance, "defect" is not a category because it does not provide any information about the subject apart from that it is probably defective. In the evaluation process, 21 clusters were removed, and the tickets were reallocated. This resulted in 117 final categories. A visualization of the weighted clusters is provided in Fig. 2.

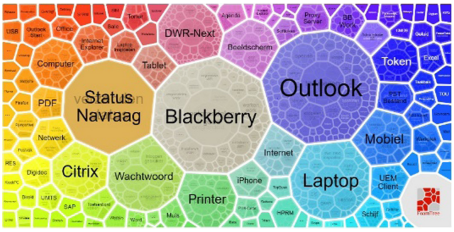

Fig. 2. High-level cluster results from Lingo3G

4.3 Intent-Level Clustering

For the intent-level clustering process, we tried two techniques: POS patterns and LDA. The POS patterns technique is used in most reviewed articles. Due to the high expectations of LDA in text clustering (in research but also in online communities and data science companies that we had contact with) and also the high scores of the technique in the article of [8], we decided to attempt this technique.

External Evaluation
We evaluated the techniques on the coverage, the number of total intents, and the clustering quality. For the clustering quality, we chose for the external evaluation of samples using a golden test set. We chose this method because we believe the internal evaluation is not practical due to the subjectiveness of the clusters.

We create this golden test set by manually clustering a sample of 1,000 tickets for 3 categories by 3 independent people, of which the best classification was chosen each time. We give a quantitative score to the degree to which the technique's results comply with the golden test set. The score given is the proportion of tickets in a cluster that is both in the golden test set as well as in the technique's resulting cluster. A score of 0.50 for a technique thus means that on average, each ticket's parent cluster contains 50% the same tickets as the parent cluster of the golden test set.

Option 1: POS patterns
For the identification of unique problems, we applied POS Patterns to the "Korte omschrijving" (Dutch phrase for 'short description') text. From the related works, it was clear that this was the go-to method to extract intents for short text and high variety corpus. We use the combination of operation-entity POS patterns, as suggested by [5]. The operations are verbs. The entities are nouns and adjectives.

For preprocessing, the first stopwords are removed using an online freely available stopword-list. Labels of the categories in which the tickets are classified are removed as well, to avoid redundant intent labels. Next, we tag the remaining words on 'Part of Speech.' If a verb is detected, the system combines the nearest nouns or adjectives with them to form a two-word phrase. If no verb is identified, the system uses the remaining words as an intent. We found that in most cases, there existed no verb in sentences. We show the results in Table 4. The total amount of tickets that the system converts to intents is about 110,000; this is slightly more than 50% of the categorized tickets.

Option 2: LDA
For this experiment, we used the complete dataset of the outlook cluster, which comprises about 15,000 tickets. For preprocessing, we lemmatized the dataset, and we used the same dutch stopword list that we used for the POS patterns. We use these files as input for training the LDA model. For determining the number of topics, we have used well known methodology, namely the perplexity score, of the clustering results. However, this methodology recommends using a maximum of 30 topics, which we find small, and the results also show very general topics. We then choose to go for 100 topics.

As summarized in Table 4, the main difference between the techniques lies in the number of tickets covered by the algorithms. LDA covered 100% of the dataset tickets and POS only 36%. The reason for the low score of the POS pattern technique is the high exclusion of words that are not part of the set POS patterns. Many short descriptions do not have a verb, which is the main ingredient of POS patterns. For this reason, we use LDA for the intent identification process.

4.4 Resolution Recommendation

For the resolution recommendation process, we combine the tickets in the clusters with their respective actions. Using the ratio of verbs as well as numbers in a sentence, we successfully removed all e-mail related noise like signature and salutation as well as TopDesk similar noise consisting of the name of the operator and timestamp. Next, we remove empty action fields and combine double actions; this increases the weight rate that we match to these actions. A domain expert manually labeled 2,000 actions as solutions to intents. 30% of the tickets appear to contain useful actions. The smallest intents of the system contain at least 20 tickets. So even the smallest intents have, on average 6 useful actions. It then depends on the reinforcement learning component to recommend these useful actions first.

4.5 Reinforcement Learning

We developed an interface for a user to type in a short description of any incident upon which the system will identify the corresponding cluster and provide previously

Table 4. Number of tickets automatically converted to intents

Method	POS patterns	LDA
Total tickets	210.000	210.000
Threshold	10	10
Coverage	36%	100%
# of intents External evaluation	1490	1500
Large (Outlook)	0.4063	0.4106
Medium (Excel)	0.4200	0.4844
Small (P-direkt)	0.3062	0.2403
Average	0.3775	0.3784

applied actions for the incident. The user can then leave feedback for the action that was most suitable to his incident using a like-button. This feedback is used automatically to improve the sorting of actions using reinforcement learning. Further potential improvements are identifying intent variations, identifying flaws in the intent disambiguation process, learning new intents, and learning new mappings between words and intents.

4.6 The Architecture of the Q.A. Pair Generator

Figure 3 depicts the complete process of training a Q.A. pair system and recommending actions to customer input. For training the system, the categorical clustering and intent identification are used. First, the categories are determined using LSA indexing. Then, the tickets are appointed to one of around 100 categories (for the SSC-ICT dataset). After that, the intents are identified.

The QA pair generator preprocesses the short descriptions of the tickets and the complete corpus of brief descriptions for a category transformed into a TF-IDF corpus, in which the preprocessed short descriptions are the documents Once we trained the model, the tickets are given a dominant topic, which is the intent. The system than grabs the action fields for each of the tickets of each intent excludes doubles and actions that are remarkably similar using the Levenshtein distance, and thus produces a list of actions for each intent. When the customer has chosen an intent that he or she thinks fits best, the Q.A. pair can produce a resolution from a matching action list. The list is sorted based on the feedback of customers as well as on a score that is provided by a deep learning classifier that can distinguish useless and useful actions.

5 Discussion

As summarized in Table 5, the success rate of the intent identification process is, on average, around 55%. The success rate is calculated by subtracting the number of tickets in “non-informative clusters” from both the “total number of tickets” and the “number of tickets clustered correctly” and then dividing the “tickets clustered correctly” by the

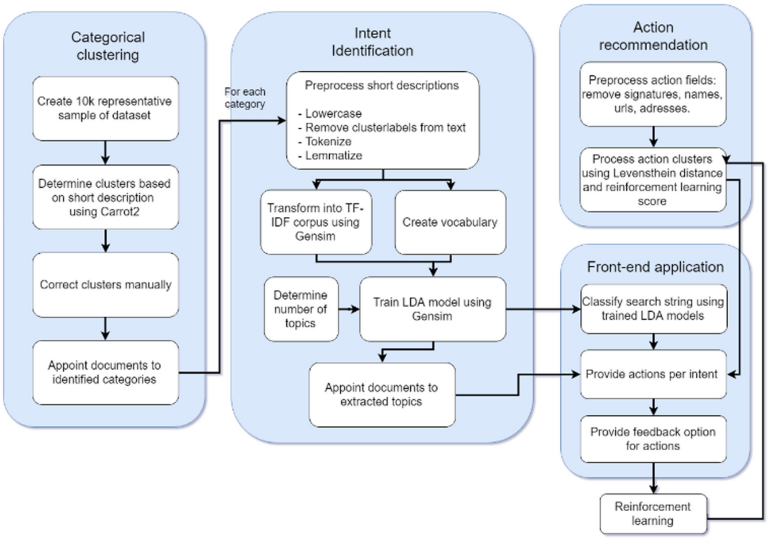

Fig. 3. Q.A. pair generator: process view

"total number of tickets." This score means that, on average, the system can identify a correct intent for a ticket 55% of the time. Furthermore, we conclude that between 10 and 20% of the tickets that are part of a category are described too vaguely to extract any meaning out of them. On top of the 12% of the category clustering component (88% success rate), we say that between 20 and 30% of all tickets are described too vaguely by the operators.

For all tickets clustered well, about 30% contains a useful action. Looking at the intents, which are almost always larger than 10 tickets and often larger than 100 tickets, the chance that intent has at least one user action is large. Furthermore, if this does not appear to be the case, the action could always be added manually by an operator. So once enough feedback is received from users, the right actions are filtered from the less informative actions, and the system will be able to recommend a useful action to an intent most of the time.

Table 5. Intent classification success for three identified categories

Category	"Outlook"	"Excel"	"P-direkt"
Total number of tickets	13341	721	286
Tickets clustered correctly	8034	436	220
Number of tickets in non-informative clusters	1323	167	89
Success rate	55.8%	48.6%	66.5%

6 Conclusion and Further Research

From research on comparable State of the Art systems, we identified the following components for a Q.A. pair generation system: Intent Identification, action recommendation, and Reinforcement Learning. We added to this the element of Categorization due to the large dataset and wide variety of tickets of SSC-ICT. We also identified specific relevant techniques for Q.A. pair generation. For Categorization, we identified LSI, LDA, and POS patterns. To avoid the risk of overfitting, we created three golden cluster sets of three different sizes and different types of categories for our evaluation method. Furthermore, we determined that the number of tickets covered, along with a threshold for intent-size, was relevant for evaluation. For the action recommendation component, we decided that the percentage of unique and useful actions proposed is a good measure. However, this is meant for future use of the system, thus not evaluated in this research, in contrary to the other two components. The reinforcement learning component also requires feedback to be able to be evaluated. Furthermore, its results can be seen in increased results for the other three components rather than having its own measure.

The main subjects that this research puts forward which are not extensively researched are that of categorization clustering, the use of Topic Modelling (LDA), our clustering quality evaluation method, and the reinforcement learning for improving the intent identification component. We believe that decreasing the number of expected topics by one hundredfold by first applying categorization clustering is the reason why LDA could be utilized as successful as we did for a dataset that is as big as ours. However, as of 2017, GuidedLDA has been discovered, a method to seed keywords in LDA topics, steering the algorithm in a preferred direction to identify topics around. GuidedLDA and its potential have, however, barely been researched yet. We are curious to see how far this steering can go, especially in combination with applying reinforcement learning. Its potential seems unlimited, reaching towards topic databases in which topics instead of lexical keywords are stored, with hundreds of weighted terms per topic.

The deep Categorization that this research suggests using categorization clustering as well as intent identification makes advanced business intelligence possible. The complete system, including the result recommendation and reinforcement learning component, has multiple use cases as well. Frequently Asked Questions (FAQ) could be easily identified and updated; the system could be used as a knowledge base for operators or be made available for all customers. Furthermore, this Q.A. pair system is the first necessary step towards building a chatbot.

This paper provides a low-cost and quick set-up method for being able to categorize the largest ticket datasets on problem-level. Topic modeling shows to be able to handle inconsistent and short ticket summaries well, in contrast to Part of Speech modeling, which depends on the accuracy of POS taggers and the presence of known verbs and nouns. The methodology in this paper is an excellent way to kickstart your use of Artificial Intelligence and see quick results. Furthermore, it provides and is designed with great opportunities for further enhancement of the system using reinforcement learning based on user feedback.

In this paper, we have presented the workflow of the solution (vide Fig. 3), POS, and LSA are traditional for natural language processing. As a future work direction, a one-state solution could be implemented using a deep neural network extracting the

intent automatically and matching it intelligently to the desired action when training data is available, although we need to find if the proposed solution is not expensive.

Another possible future research direction could be to incorporate modern word-embedding techniques like *cbow* or *skip-gram* and could be character-based for dealing with spelling mistakes. This would boost the accuracy of the Q.A pair solution. Moreover, incorporating reinforcement learning for the neural network seems more natural as for the existing solution, but will it be cost-effective or not is still an open question.

References

1. Agarwal, S., Aggarwal, V., Akula, A.R., Dasgupta, G.B., Sridhara, G.: Automatic problem extraction and analysis from unstructured text in I.T. tickets. IBM J. Res. Dev. **61**(1), 4:41–4:52 (2017). https://doi.org/10.1147/JRD.2016.2629318
2. Bensoussan, A., Mookerjee, R., Mookerjee, V., Yue, W.T.: Maintaining diagnostic knowledge-based systems: a control-theoretic approach. Manag. Sci. (2008). https://doi.org/10.1287/mnsc.1080.0908
3. Berry, M.W., Kogan, J.: Text Mining: Applications and Theory. Wiley (2010). https://doi.org/10.1002/9780470689646
4. Berente, N., Seidel, S., Safadi, H.: Research commentary—data-driven computationally intensive theory development. Inf. Syst. Res. **30**(1), 50–64 (2019)
5. Dhoolia, P., et al.: A cognitive system for business and technical support: a case study (2017). https://doi.org/10.1147/JRD.2016.2631398
6. Grosan, C., Abraham, A.: Rule-based expert systems. In: Grosan, C., Abraham, A. (eds.) Intelligent Systems. ISRL, vol. 17, pp. 149–185. Springer, Heidelberg (2011). https://doi.org/10.1007/978-3-642-21004-4_7
7. Iden, J., Eikebrokk, T.R.: Using the ITIL process reference model for realizing I.T. governance: an empirical investigation. Inf. Syst. Manag. **31**(1), 37–58 (2014). https://doi.org/10.1080/10580530.2014.854089
8. Jan, E., Chen, K., Ide, T.: A probabilistic concept annotation for I.T. service desk tickets. In: Proceedings of the 7th International Workshop on Exploiting Semantic Annotations in Information Retrieval - ESAIR 2014, pp. 21–23 (2014). https://doi.org/10.1145/2663712.2666193
9. Lammers, M.: A QA-pair generation system for the incident tickets of a public ICT Shared Service Center. Mater thesis, University of Twete (2019). http://essay.utwente.nl/77562/
10. Mani, S., Sankaranarayanan, K., Sinha, V.S., Devanbu, P.: Panning requirement nuggets in stream of software maintenance tickets. In: Proceedings of the 22nd ACM SIGSOFT International Symposium on Foundations of Software Engineering - FSE 2014, pp. 678–688 (2014). https://doi.org/10.1145/2635868.2635897
11. Osiński, S., Stefanowski, J., Weiss, D.: Lingo: search results clustering algorithm based on singular value decomposition. In: Kłopotek, M.A., Wierzchoń, S.T., Trojanowski, K. (eds.) Intelligent Information Processing and Web Mining. AINSC, vol. 25, pp. 359–368. Springer, Heidelberg (2004). https://doi.org/10.1007/978-3-540-39985-8_37
12. Potharaju, R., Nita-Rotaru, C.: Juggling the jigsaw : towards automated problem inference from network trouble tickets. In: NSDI, pp. 127–141 (2013)
13. Roy, S., Malladi, V.V., Gangwar, A., Dharmaraj, R.: A NMF-based learning of topics and clusters for IT maintenance tickets aided by heuristic. In: Mendling, J., Mouratidis, H. (eds.) CAiSE 2018. LNBIP, vol. 317, pp. 209–217. Springer, Cham (2018). https://doi.org/10.1007/978-3-319-92901-9_18

14. Vlasov, V., Chebotareva, V., Rakhimov, M., Kruglikov, S.: AI user support system for SAP ERP (2017)
15. Wolfswinkel, J.F., Furtmueller, E., Wilderom, C.P.M.: Using grounded theory as a method for rigorously reviewing literature. Eur. J. Inf. Syst. **22**(1), 45–55 (2013). https://doi.org/10.1057/ejis.2011.51