Automated Quality Assessment of Incident Tickets for Smart Service Continuity

Luciano Baresi[1], Giovanni Quattrocchi[1(✉)], Damian Andrew Tamburri[2,3], and Willem-Jan Van Den Heuvel[3,4]

[1] Politecnico di Milano, Milan, Italy
{luciano.baresi,giovanni.quattrocchi}@polimi.it
[2] Eindhoven University of Technology, Eindhoven, Netherlands
d.a.tamburri@tue.nl
[3] Jheronimus Academy of Data Science, Eindhoven, Netherlands
W.J.A.M.vdnHeuvel@tue.nl
[4] Tilburg University, Tilburg, Netherlands

Abstract. Customer management operations, such as Incident Management (IM), are traditionally performed manually often resulting in time consuming and error-prone activities. Artificial Intelligence (AI) software systems and connected information management can help handle the discontinuities in critical business tasks. AI Incident Management (AIIM) becomes therefore a set of practices and tools to resolve incidents by means of AI-enabled organizational processes and methodologies. The software automation of AIIM could reduce unplanned interruptions of service and let customers resume their work as quick as possible.

While several techniques were presented in the literature to automatically identify the problems described in incident tickets by customers, this paper focuses on the qualitative analysis of the provided descriptions and on using such analysis within the context of an AI-enabled business organizational process. When an incident ticket does not describe properly the problem, the analyst must ask the customer for additional details which could require several long-lasting interactions. This paper overviews *ACQUA*, an AIIM approach that uses machine-learning to automatically assess the quality of ticket descriptions with the goals of removing the need of additional communications and guiding the customers to properly describe the incident.

Keywords: Incident Management · Service continuity · Digital transformation · Artificial intelligence · Natural Language Processing

1 Introduction

Modern companies more and more require data-driven corporate services as drivers for better quality and for saving money: the more data companies can

E. Kafeza et al. (Eds.): ICSOC 2020, LNCS 12571, pp. 492–499, 2020.
https://doi.org/10.1007/978-3-030-65310-1_35

collect, the more "observable" they become. Stakeholders can then exploit these data to carry out dedicated analyses and react in a more appropriate and timely way, with positive effects on the organizational performance of the company. Successful companies, therefore, are those that harness the benefits of automation, data analytics, and connected advanced human-machine interfaces [16].

In this context, service management—and specifically service *incident* management (IM)—is the set operations and processes that manages customer services during their utilization, e.g., through the integration of tools and best-practices [8]. One of the key aspects of IM is to provide *service continuity* [10], that is, the capability of preventing, predicting, and managing service incidents with the goal of maintaining the desired quality of service (QoS) during and after unexpected events. These practices do not only aim to keep users engaged and satisfied of the services they use.

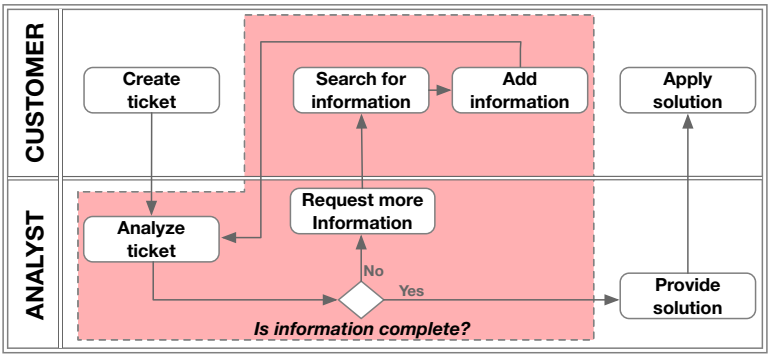

Fig. 1. Customer-Analyst interactions when no automation is in place.

If no automation is in place, IM requires that customers and analysts interact through a workflow similar to the one presented in Fig. 1. Customers describe incidents through (*service incident*) tickets while analysts manually inspect them and provide a solution. Several research efforts have already tried to automate this process [6] but mostly concentrate on the semantic analysis of incident descriptions. These works assume that customer inputs are always sufficiently detailed to perform an analysis, while this in practice could not be true. Users may not interact with the failed component directly and their description of the incident could well be partial or unclear [7]. Consequently, analysts typically interact with customers for extra inputs (as shown in the colored area of Fig. 1), but this slows down ticket resolution—and hence, proper service operations—dramatically.

What is more, although IM approaches have been studied since the seventies [19], the resolution of incident tickets is still mainly done manually by analysts, strongly based on their experience and on interactions with customers [17]. This means that this task is one of the most time consuming and fallible activities [9,21].

To address the problem, the paper presents *ACQUA* (*Automatic tiCket Quality Assessment*), an approach based on Machine Learning (ML) that aims to reduce—and eventually eliminate—the need of many customer-analyst iterations. *ACQUA* automatically evaluates the quality of incident descriptions and notifies the customer in case additional details are required. *ACQUA* is part of a novel IM family of approaches and techniques that we call AIIM—and that fuses Artificial-Intelligence (AI) with practices from Incident Management.

ACQUA consists of three main activities: i) *feature engineering*, that is, the extraction of meaningful characteristics and metrics from an initial dataset of incident tickets ii) *service ticket modeling*, that is, the creation of different models from the extracted features that are able to evaluate the quality of new, unseen, service tickets, and iii) *service operations validation*, that is, the selection of one of the produced models based on their validated performance over available data as part of conventional Machine-Learning operations.

We plan to evaluate *ACQUA* through comparison with three state-of-the-art approaches: (1) BLEU [18] (BilinguaL Evaluation Understudy) (2) ROUGE [13] (Recall-Oriented Understudy for Gisting Evaluation), and (3) a baseline that uses a simple heuristic for computing the quality of incident tickets. On the one hand, the two reference approaches exploit well-known metrics used in the field of Natural Language Processing (NLP) to evaluate text quality; on the other hand, the baseline approach offers an optimistic take at the problem. To do that we will utilize a real-life industrial implementation and experimentation conducted on a real dataset provided by a large banking corporation (from now on called BANK) in The Netherlands We consider *ACQUA* as a valid first step in the direction of more autonomous large-scale AIIM and connected service governance operations.

The paper is organized as follows. Section 2 discusses some significant related work. Section 3 illustrates the research questions, and methodology used to build *ACQUA*. Section 4 describes the details of *ACQUA* and Sect. 5 concludes the paper.

2 Related Work

Given that downtime causes monetary loss [3], Incident Management became a key activity for businesses, and several works in the literature were presented in order to enhance Service Continuity [7,14]

Shao et al. [20] propose a prioritization algorithm to rank the relevance and severity of tickets according to their descriptions. This way more significant tickets are handled by analysts before the others and service continuity is improved. *ACQUA* and this work are complementary. Our approach can be used as a preliminary step to analyze the quality of the ticket and, when users are able to provide enough details, the ticket can be ranked and processed accordingly.

Gupta et al. [6] analyze the input requests made by analysts to customers to understand how they impact the user experience. When calculating resolution time, the time waiting for user inputs is not counted. Hence, they distinguish

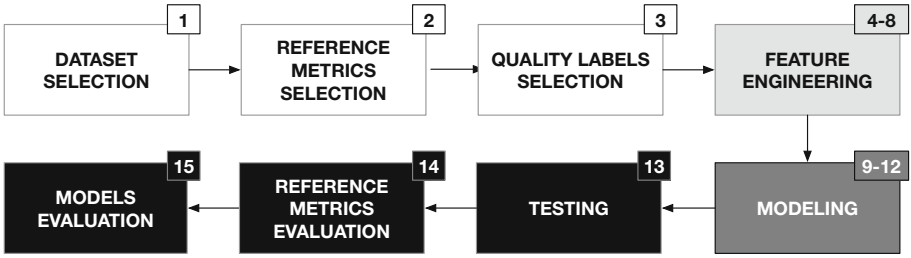

Fig. 2. The *ACQUA* methodology, an overview.eps

between two types of input requests: real and tactical. Real requests are sent to effectively seek for useful additional details, while tactical ones are merely raised to stop the downtime counting. Therefore, they created a system to automatically detect tactical input requests using algorithm TF-IDF [2] for the decision process and Principal Component Analysis [22] to reduce the dimensions composing the feature space. This work does not validate the quality of the user ticket as *ACQUA* does but the working efficiency of analysts during the resolution phase.

3 The *ACQUA* Methodology

This paper addresses the problem of evaluating the quality of service incident tickets in order to speed-up their resolution. Indeed, the research effort concentrates on *how the quality of input text can be measured in the context of Incident Management*. With this goal we present *ACQUA*, a 15-step methodology—tailored from the Cross-Industry Standard Process for Data-Mining (CRISP-DM) [4]—shown in Fig. 2.

The figure illustrates a concrete overview of the *ACQUA* AIIM methodology in a simple box-and-line notation. *ACQUA* is based on machine learning and employs different types of features and classifiers in order to predict the quality of incident tickets. *ACQUA* is composed of four main types of actions. First, preliminary actions (steps 1–3) are depicted in white boxes and are explained in the rest of this section. Second, in feature engineering tasks (steps 4–8, shown in the light gray box) different types of features are extracted and combined in meaningful datasets. Third, in modeling phase (steps 9–12, shown in the dark gray boxes) different classifiers are trained using selected features. Finally, black boxes represent the evaluation actions (steps 13–15) which will reported in our future work.

In order to properly assess *ACQUA* we formulated the following research questions.

RQ1 How do existing state-of-the-art metrics (i.e., *reference metrics*) perform when evaluating the quality of incident tickets?

RQ2 How does *ACQUA* perform when using *reference metrics* as features for ML classifiers in oder to predict the quality of incident tickets?

RQ3 How does *ACQUA* perform when using *deductive features* (i.e., semantic characteristics of the text) w.r.t. *ACQUA* using *reference metrics*?

RQ4 How does *ACQUA* perform when using *embeddings* (i.e., structural characteristics of the text) w.r.t. the above alternatives?

With the first research question we aim to understand whether state-of-the-art textual quality metrics are able to capture the quality of incident descriptions. Subsequently RQs 2, 3, 4 investigate how different types of features (reference metrics themselves, semantic and syntactic ones) perform when used in an AIIM approach.

The data used in this study were obtained (step 1) by exporting the tickets from the IM system (ServiceNow[1]) of BANK. Both customers and analysts of BANK are Dutch speaker therefore the tickets are mostly written in Dutch or a dialect[2]. The 77010 tickets collected in the dataset from September 2016 to April 2019 contain an average of 34 words each and one third of them (23874) required the analyst ask further details to the customer. Reference metrics selection (step 2) refers to the study of the literature in order to find existing metrics that could be used to obtain insights on the quality of text inputs.

Being tickets written by customers in natural language, we identified two metrics, the ones that obtained the highest similarity with the human perception of quality, widely used in the context of Natural-Language Processing: BLEU and ROUGE. These metrics evaluate the quality of *candidate text* (often machine generated) with respect to high-quality reference texts [13].

The original dataset does not contain any indications of the quality of the tickets. Therefore, in step 3 we defined five labels with an associated value between 0 (insufficient details) and 4 (well-described incident). Moreover, we manually labeled each ticket according to the comments left by the analyst and our perception of their quality.

4 Feature Engineering and Modeling

In this section we present the feature engineering and modeling steps (4–12) of *ACQUA*.

The selected dataset contains a large amounts of unstructured data requiring a preliminary processing phase (step 4). Indeed, the customer description of the incident, the comments between analyst and customer and the analyst closing notes are all written in plain text without any structure. For structured columns minor processing was necessary in order to reduce the noise and being able to properly compare values. The preprocessing consists in the following six activities: i) *filtering* to remove missing data, ii) *text transformation* to remove special characters and punctuation, iii) *domain transformation* to eliminate from the ticket text partial or blank parts, iv) *encoding* to transform values and labels

[1] https://www.servicenow.com/products/incident-management.html.

[2] A negligible amount of tickets contain also sentences (error messages) written in English.

onto pre-defined numbers, v) *tokenization* to obtain the list of words, and vi) *stemming* to normalized words to a root form.

4.1 Feature Extraction and Selection

ACQUA uses three types of features: reference metrics, deductive and embeddings. In step 5, we computed the value of BLEU and for ROUGE each ticket. These values, in addition to be evaluated as quality metrics in step 14, are then used them as input features for classifiers to understand whether they can provide additional insights during training.

Deductive features are features extracted from the description of an incident and are mainly related to the semantic of what the user describes (step 6). They are a set of boolean features, that indicate if a specific word is mentioned in the text. The intuition is that the occurrence of word like "error" or "warning" prelude to a detailed description of the problem. If incident related keywords (e.g.,"power drain" or 'restart") are included there could be high chances that the incident is explained. In addition to boolean deductive feature, we include also numerical ones that are related to the length of the description such as the number of tokens and the sum of the token length for a total of 13 deductive features.

Embeddings (step 7) are features encoded as sparse vectors obtained from words or documents. They help a machine understand natural language by placing similar text inputs close to one another [12,15]. *ACQUA* uses two embeddings techniques, namely Word2Vec, and Doc2Vec.

Word2Vec is a neural network language model that constructs a log-linear classification network that produces a vector [15] where each word is represented as a point in the space (a vector) and related words are located closely to each other. In *ACQUA* Word2Vec is used to create a machine readable feature from textual, unstructured data that can be used for further (algebraic) calculations.

Doc2Vec is another neural network language model that we used to create embedding features. While Word2Vec computes a feature for every word in a text corpus (e.g., an incident ticket), Doc2Vec computes a feature vector for every document/ticket [12]. This eliminates the need of a vector aggregation step as required by Word2Vec and facilitate the comparison among similar tickets. On the other hand, given that Doc2Vec reasons on a coarse granularity, it is less tolerant to word misspelling compared to Word2Vec. Since a significant amount of ticket description contains misspellings we used both the methods in *ACQUA*.

In the last step of feature engineering (step 8) we generated datasets containing the different types of features in order to answer RQ2, RQ3 and RQ4 in our future evaluation.

4.2 Service Ticket Modeling

The first step of modeling is the selection of classifiers that using the selected features can produce meaningful estimation models for the quality assessment of

incident tickets (step 9). Having to deal with different set of features, *ACQUA* does not rely on a single classifier but it uses 5 different types: *random forest* [1], *logistic regression* [2], *k-nearest neighbors* [1], *gradient boost* [5], and a dummy *most frequent* classifier.

Before training the models (step 12), the dataset is split in different parts. 20% of the data are removed and used for testing in step 13. On the remaining 80% *ACQUA* applies the stratified K-Fold [11] algorithm (step 10) to properly tune the classifiers parameters. Data are split into k consecutive partitions (or folds) each of them of approximately the same size. The training and validation sets (i.e., the dataset used to adjust classifiers parameters) are generated in k phase. On each phase one fold, in turn, is used as validation set while the other $k-1$ as training set. In *ACQUA* we used k equals to 10.

For each of the aforementioned classifiers, hyper-parameter (i.e., classifier parameters) tuning was manually applied in step 11 by taking into account the best practices when dealing with class imbalance and to avoid overfitting. The tuning consisted in an iterative process of training-validation-parameters adjustment-training until reaching satisfactory performance, as envisioned in the CRISP-DM standard process [4]. Finally, in step 12 models are trained with proper tuning parameters to be ready for being evaluated.

In our future work we will present how we tested the models (step 13) and how we compared with reference metrics (step 14) and the performance of the various classifiers (15).

5 Conclusions and Future Work

Incident Management and Service Continuity are key aspects of almost all the businesses to reduce or avoid the costs of downtimes. This paper presented *ACQUA*, a AIIM methodology for assessing the quality of incident tickets in order to minimize the long-lasting communications between customers and analysts. In the feature we plan to carry out an extensive evaluation of the approach.

Acknowledgment. We thank Dr. Jeffrey Vervoort for his valuable contribution to this work carried out during hist master thesis. Finally, some of the authors' work is partially supported by the European Commission grant no. 787061 (H2020), ANITA, European Commission grant no. 825040 (H2020), RADON, European Commission grant no. 825480 (H2020), SODALITE.

References

1. Aggarwal, C.C.: Text classification: basic models. Machine Learning for Text, pp. 113–157. Springer, Cham (2018). https://doi.org/10.1007/978-3-319-73531-3_5
2. Aly, M.: Survey on multiclass classification methods. Technical report (2005)
3. Cao, C., Zhan, Z.: Incident management process for the cloud computing environments. In: 2011 IEEE International Conference on Cloud Computing and Intelligence Systems, pp. 225–229 (2011)

4. Chapman, P., et al.: CRISP-DM 1.0 step-by-step data mining guide. Technical report, The CRISP-DM consortium, August 2000
5. Chen, T., Guestrin, C.: XGBoost: a scalable tree boosting system. In: Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 785–794. Association for Computing Machinery (2016)
6. Gupta, M., Asadullah, A., Padmanabhuni, S., Serebrenik, A.: Reducing user input requests to improve it support ticket resolution process. Empir. Softw. Eng. **23**(3), 1664–1703 (2018)
7. Gupta, R., Prasad, K.H., Luan, L., Rosu, D., Ward, C.: Multi-dimensional knowledge integration for efficient incident management in a services cloud. In: 2009 IEEE International Conference on Services Computing, pp. 57–64 (2009)
8. Iden, J., Eikebrokk, T.R.: Implementing it service management: a systematic literature review. Int. J. Inf. Manag. **33**(3), 512–523 (2013)
9. Janssen, P.: IT-Servicemanagement volgens ITIL, 3rd edn. Pearson Benelux, Gatwickstraat 1, 1043 GK, Amsterdam (2008)
10. Klems, M., Tai, S., Shwartz, L., Grabarnik, G.: Automating the delivery of it service continuity management through cloud service orchestration. In: 2010 IEEE Network Operations and Management Symposium-NOMS 2010, pp. 65–72. IEEE (2010)
11. Kuhn, M., Johnson, K.: Applied Predictive Modeling, p. 1. Springer, New York (2013). https://doi.org/10.1007/978-1-4614-6849-3
12. Le, Q., Mikolov, T.: Distributed representations of sentences and documents. In: International Conference on Machine Learning, pp. 1188–1196 (2014)
13. Lin, C.Y.: ROUGE: a package for automatic evaluation of summaries. In: Text summarization Branches Out, pp. 74–81 (2004)
14. Liu, R., Lee, J.: IT incident management by analyzing incident relations. In: Liu, C., Ludwig, H., Toumani, F., Yu, Q. (eds.) ICSOC 2012. LNCS, vol. 7636, pp. 631–638. Springer, Heidelberg (2012). https://doi.org/10.1007/978-3-642-34321-6_49
15. Mikolov, T., Chen, K., Corrado, G., Dean, J.: Efficient estimation of word representations in vector space. arXiv e-prints, January 2013
16. Mithas, S., Ramasubbu, N., Sambamurthy, V.: How information management capability influences firm performance. MIS Q. **35**(1), 237–256 (2011)
17. Motahari-Nezhad, H.R., Bartolini, C., Graupner, S., Singhal, S., Spence, S.: It support conversation manager: a conversation-centered approach and tool for managing best practice it processes. In: 2010 14th IEEE International Enterprise Distributed Object Computing Conference, pp. 247–256 (2010)
18. Papineni, K., Roukos, S., Ward, T., Zhu, W.J.: BLEU: a method for automatic evaluation of machine translation. In: Proceedings of the 40th Annual Meeting on Association for Computational Linguistics, ACL 2002, Stroudsburg, PA, USA, pp. 311–318. Association for Computational Linguistics (2002)
19. Rowley, D.D.: The fires that created an incident management system (2005)
20. Shao, J., Wei, H., Wang, Q., Mei, H.: A runtime model based monitoring approach for cloud. In: 2010 IEEE 3rd International Conference on Cloud Computing, pp. 313–320 (2010)
21. Wang, Q., Song, J., Liu, L., Luo, X., XinHua, E.: Building it-based incident management platform. In: 5th International Conference on Pervasive Computing and Applications, pp. 359–364, December 2010
22. Wold, S., Esbensen, K., Geladi, P.: Principal component analysis. Chemom. Intell. Lab. Syst. **2**(1–3), 37–52 (1987)