Less is more in incident categorization

Sara Silva
Instituto Universitário de Lisboa (ISCTE-IUL) Lisbon, Portugal
satsa@iscte-iul.pt

Ricardo Ribeiro
INESC-ID Lisboa
Instituto Universitário de Lisboa (ISCTE-IUL), Lisbon, Portugal
ricardo.ribeiro@iscte-iul.pt
https://orcid.org/0000-0002-2058-693X

Rubén Pereira
Instituto Universitário de Lisboa (ISCTE-IUL) Lisbon, Portugal
Ruben.Filipe.Pereira@iscte-iul.pt
https://orcid.org/0000-0002-3001-5911

Abstract

The IT incident management process requires a correct categorization to attribute incident tickets to the right resolution group and obtain as quickly as possible an operational system, impacting the minimum as possible the business and costumers. In this work, we introduce automatic text classification, demonstrating the application of several natural language processing techniques and analyzing the impact of each one on a real incident tickets dataset. The techniques that we explore in the pre-processing of the text that describes an incident are the following: tokenization, stemming, eliminating stop-words, named-entity recognition, and TF×IDF-based document representation. Finally, to build the model and observe the results after applying the previous techniques, we use two machine learning algorithms: Support Vector Machine (SVM) and K-Nearest Neighbor (KNN). Two important findings result from this study: a shorter description of an incident is better than a full description of an incident; and, pre-processing has little impact on incident categorization, mainly due the specific vocabulary used in this type of text.

2012 ACM Subject Classification Computing methodologies → Natural language processing

Keywords and phrases machine learning, automated incident categorization, SVM, incident management, natural language

Digital Object Identifier 10.4230/OASIcs.SLATE.2018.17

Category Short Paper

1 Introduction

An incident is defined by ITIL as “An unplanned interruption to an IT service or reduction in the quality of an IT service”. These incidents can be related with failures, questions, or queries and should be detected as early as possible [7]. It is crucial to have an appropriate incident classification, the process that assigns a suitable category to an incident, so they are routed more accurately [7]. Automating incident classification process means to avoid human error, reducing the waste of resources and avoiding incorrect routing due to wrong classification [3]. To automate the classification is relevant to take into account the structure of the incidents. The incident tickets are composed by several attributes, which are mandatory fields when the user is recording the incident in the Incident Ticket System (ITS). Two attributes are crucial for the categorization process: the short and full descriptions are the key attributes to

7th Symposium on Languages, Applications and Technologies (SLATE 2018).
Editors: Pedro Rangel Henriques, José Paulo Leal, António Leitão, and Xavier Gómez Guinovart
Article No. 17; pp. 17:1–17:7
OpenAccess Series in Informatics
Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

obtain a good classification performance. The dataset used to test and assess the algorithms contains information about real-word incident tickets provided by a specific company. Due to privacy questions, the company cannot be mentioned and the dataset is not available.

The remainder of this document consists of five sections that are structured as follows. Some related work is presented in Section 2. The proposed method is presented in Section 3. The implementation is described in Section 4. Section 5 presents the obtained results. Finally, the document closes with the conclusions and some possible future work.

2 Related Work

Over the years, approaches that automate Incident Management (IM) and particularly incident classification have being studied and developed. In this section, we describe relevant work developed in this area, used algorithms, and the results obtained with the respective implementations. To automate IM, the authors of [4] use Machine Leaning (ML) and information integration techniques to develop an algorithm for correlating incoming incidents. The authors used the incident description to extract keywords and their annotations as features. SVM is the algorithm used to attribute a category to an incoming incident. This approach generates the list of keywords which better identifies each category. In [1], the authors used for the same problem, incident tickets categorization, SVM, KNN, decision trees, and Naïve Bayes. They used four datasets with different categories. In this classification process, they present three approaches for each algorithm: accuracy results using TF-IDF, using only TF, and using boolean weighting. On average, SVM had accuracy results of 90% approximately, in the three approaches. KNN achieves 75% of accuracy, also with the three approaches. Decision trees have similar results in the three approaches, of around of 90%. Finally, Naïve Bayes was the only approach which presented different results for the three approaches: boolean weighting with 85% and TF-IDF and TF, both with 55% of accuracy.

3 Proposed Method

To propose the best approach to automatically categorize an incoming incident there are critical steps that have to be taken into account in order to ensure a good performance of the classifier. The automatic incident categorization is possible due to the data related to each incident. The main attribute for the categorization of an incoming incident is the description of the incident. There are two attributes related with the description assigned to each incident. These attributes are the short and the full description. Both consist in unstructured natural language text. The difference between both attributes is implicit in the name. Both descriptions are provided by the user who have created the incident in the ITS. The full description is a long and detailed description that contains all the relevant aspects to categorize the whole incident. The short description is much shorter than the detailed one, consisting in a summary, ordinarily one phrase that categorizes briefly the incident.

The analyzed dataset has incidents written in several languages like English, German, Spanish, French, and Portuguese. However, since the most common language is English, we have decided to choose English as the one to be studied and excluded all the incidents in other languages from our analysis. As referred in the previous section, there are other attributes that also contribute to the categorization, however, the short and full descriptions are the critical ones to obtain a good performance of the classifier, requiring a special process.

The dataset under study contains incidents that have several levels of categorization. This study only takes into consideration the first and second level. Moreover, we compare the performance achieved using the short and the full description, at both levels of categorization.

Table 1 Incident ticket example.

Short description	Adobe Reader XI
Full description	User cannot open Internet Explorer.Error message displayed.
Category	Software
Subcategory	Managed Software Workplace
Caller id	User X
Affected Location	Location Y
Severity	4 - Low

Concerning the second level of categories, we explored two different approaches. The first one is performing the categorization assuming that the first-level category is correctly assigned to the incident. Basically, we use the first-level category as an attribute to build the classifier that assigns the second-level category to a given incident. The second one does not take into consideration the first-level categorization. Therefore, the incident is categorized with the same data that we use in the first-level categorization. When building a classifier to automatically assign a second-level category, we are also automatically assigning the respective first-level category. The set of first-level categories is composed by the ten following categories: application, collaboration, enterprise resource planning (ERP), hosting services, network, security and access, output management, software, workplace, and support. After assigning a first-level category to an incident, a second-level category is assigned. The set of second-level categories is composed by 94 categories: 47 belong to application, 9 to collaboration, 3 to ERP, 7 to hosting services, 6 to network, 3 to output management, 4 to security and access, 2 to software, 3 to support, and, finally, 10 to workplace.

In this work, we analyze the impact of different natural language pre-processing techniques, as tokenization, TF-IDF, stemming, stop word removal, and entities recognition on incident classification based on their textual description. We explore two classification approaches commonly used for this task: support vector machine (SVM) and K-Nearest Neighbors (KNN). SVM is an appropriate technique for text categorization, proving to be more robust than other conventional techniques of text classification [6]. KNN is considered to be simple, easy to implement [10] and a popular one in text categorization [11]. The goal is compare the different approaches and conclude which performs better.

4 Implementation

To train the classifiers, we use a dataset composed by incident tickets correctly classified with an appropriate first-level category and a second-level category. In this dataset, each incident ticket has a short description, a full description, a caller id, which is the person who opens the ticket, a severity, and, finally, the respective first-level (category) and a second-level (subcategory) categories, as it is possible observe in Table 1. For the first-level categories, we used 2000 incidents per category. For the second-level, we use a dataset with 2000 incidents per subcategory whenever was possible. The most times it was, however, there are second-level categories with less incidents in dataset.

We start by analyzing the impact of using the short and full descriptions on the first-level categories. Then, we consider that the first-level categorization is correct and analyze the difference of using short and full descriptions on the second-level categorization. Finally, we study the how second-level categorization performs when not using first-level information and, again, the impact of using the short and full descriptions at this level.

The obtained results of the different approaches are presented in the next section.

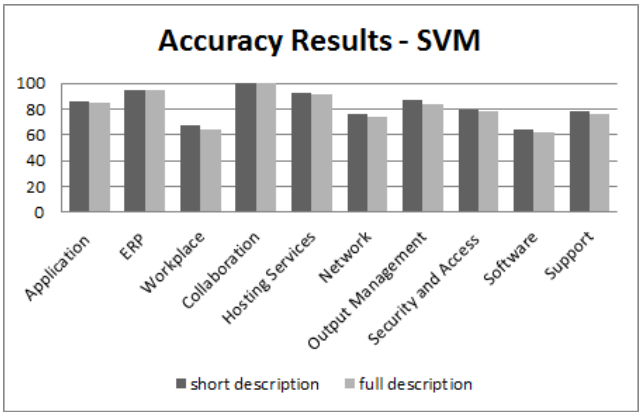


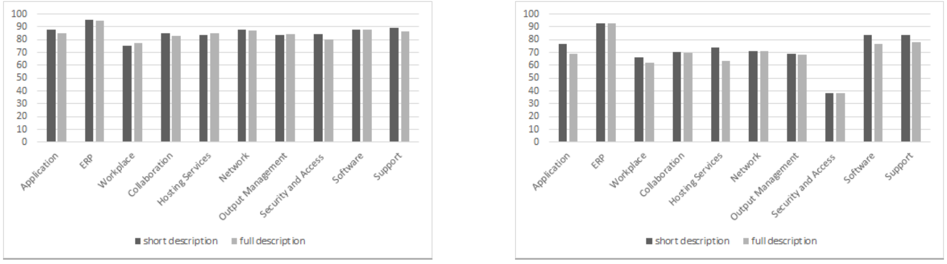
Figure 1 Short *vs* full description at the first-level categorization.

5 Results

To train and test the different techniques previously presented, we use cross-validation, which consists on dividing the dataset into subsets with the equal number of tickets. One subset is used to test the classifier, while the others are used to train the classifier. This process has the advantage of preventing overfitting, because the training data is fully independent of the test data [2, 5].

Figure 1 presents the accuracy results after applying the SVM algorithm, showing the difference between using the short and the full description. Previous work [9] showed that at the first-level categorization, the best results were achieved using an SVM (specifically, when compared to KNN). When using the short description, we achieve an accuracy of 83%. Surprisingly, when using the full description the accuracy decreases to 81%. Figure 2 presents the results related to the second-level categorization (considering the correct first-level categorization), also comparing the impact of using the short *versus* the full description. The same behavior is observed at this level, with both approaches, SVM and KNN, achieving better results using the short description. As we can see in Figure 2a, when using the SVM the overall accuracy for the short description is 86% and for the full description is 85%. Figure 2b shows the achieved results using the KNN algorithm: with the short description, we obtained 73% of accuracy and with the full description we obtained 69%. Overall, using the short description leads to better results, with the SVM consistently achieving the best results. The results of the second-level categorization when considering all the categories at this level without taking into account the first-level, as expected, decrease to accuracies of 75% using SVM and 65% using KNN.

Concerning the impact of pre-processing on the classification process, we start by analyzing two different tokenization strategies: alphabetic and word tokenization. The alphabetic tokenizer is a simple tokenizer that considers only tokens composed by alphabetic sequences. The word tokenizer is a standard word tokenizer that splits words according to predefined tokens, such as space, punctuation, etc. For this part of the work, we focus on the first-level categorization using the short description, since the best results were achieved using this attribute. Figure 3 presents the results of the application of the two tokenizers: with the alphabetic tokenizer achieving an accuracy of 83% and the word tokenizer achieving an accuracy of 82%.



(a) Results using a SVM.

(b) Results using KNN.

Figure 2 Accuracy at the second-level categorization (considering the correct first-level categorization).

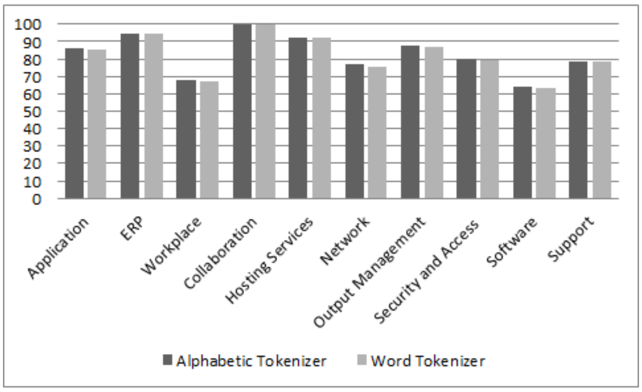


Figure 3 Accuracy of the alphabetic tokenizer *vs* the word tokenizer.

Another aspect that we have explored was the descriptions representation. In that sense, we represent descriptions as feature vectors of term frequencies (TF), $log(1 + f_{ij})$, f_{ij} is the frequency of word i in document j (other dampening strategies could be used); inverse document frequencies (IDF), log(num of Docs/num of Docs with word i); and, TF × IDF. TF × IDF increases with the number of times a term occurs in a document, but is offset by the document frequency of the term in the corpus. The results are shown in Figure 4. As it can be seen, the results are very similar (accuracies of approximately 82%). Our intuition is that as we perform stop word removal and use short descriptions (small documents), the impact of varying the weighting scheme in the feature representation is not significant.

We also explored stemming. Stemming consists on reducing the words to their base form, lowering the number of entries of the dictionary. We compare the use of the Porter Stemmer [8] to not using stemming at all. The use of stemming did not have any impact on the results. Our intuition is that description representation did not benefit of the use of stemming due to the specific vocabulary used in incident description.

Finally, we explored named-entity recognition focusing on the identification of organizations and use them as features to improve the categorization. This process did not impact the results.

6 Conclusions and Future Work

Text processing plays an important role in incident categorization. In this work, we analyze different natural language processing techniques and evaluate their impact on a real incident tickets dataset. An interesting outcome of this study was that the use of the short description

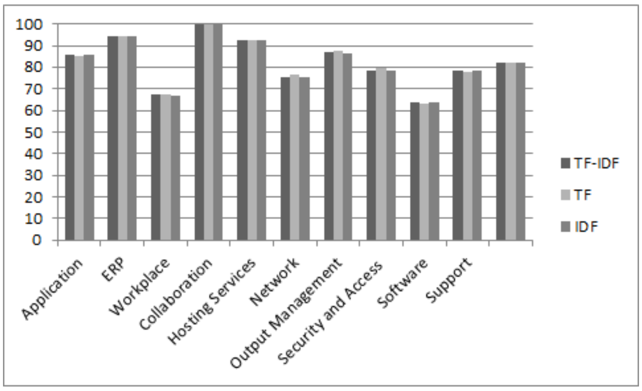


Figure 4 Accuracy when using TF×IDF, TF, and IDF.

of an incident leads to a greater accuracy than using the full description, on the different levels of categorization (first level, 10 categories; second level, 49 categories) under analysis. A possible reason that we found to justify such finding might be the fact that when the user describes an incident with limited text that results in a greater focus on explaining the incident. On the other hand, in the full description the user has tendency to disperse.

This analysis is critical to produce a positive impact in the categorization process and is determinant to obtain a correct assignment and consequently improve the whole incident route. This paper is part of a work related with the integration of a module in an Incident Ticket System in a specific company. So as future work we plan to carry out the integration and assess its impact by performing interviews to the IT teams responsible for the Incident Management (IM) process. It is also intended to extend the categorization to the whole activity of classification (which includes a third level) and initial support in the IM process, which includes automating the assignment of a priority and urgency to incidents. Moreover, we pretend to automate the resolution and recovery activities, finding and suggesting automatically a possible resolution to an incoming incident.

References

1 Muchahit Altintas and A. Cuneyd Tantug. Machine learning based volume diagnosis. In *International Conference on Artificial Intelligence and Computer Science (AICS)*, pages 195–207, 2014.

2 Sylvain Arlot and Alain Celisse. A survey of cross-validation procedures for model selection. *Statistics Surveys*, 4:40–79, 2010. doi:10.1214/09-SS054.

3 Rajeev Gupta, K. Hima Prasad, Laura Luan, Daniela Rosu, and Chris Ward. Multi-dimensional knowledge integration for efficient incident management in a services cloud. In *IEEE International Conference on Services Computing*, pages 57–64, 2009. doi:10.1109/SCC.2009.48.

4 Rajeev Gupta, K. Hima Prasad, and Mukesh Mohania. Information integration techniques to automate incident management. In *IEEE Network Operations and Management Symposium (NOMS)*, pages 979–982, 2008. doi:10.1109/NOMS.2008.4575262.

5 Chih-Wei Hsu, Chih-Chung Chang, and Chih-Jen Lin. A practical guide to support vector classification. *BJU international*, 101(1):1396–1400, 2008. doi:10.1177/02632760022050997.

6 Thorsten Joachims. Text categorization with Support Vector Machines: Learning with many relevant features. In *Machine Learning: ECML-98*, volume 1398 of *Lecture Notes in Computer Science*, pages 137–142. Springer, Berlin, Heidelberg, 1998.

7 John O. Long. Service operation. In *Itil Version 3 at a Glance: Information Quick Reference*, pages 55–74. Springer, 2008. doi:10.1007/978-0-387-77393-3_5.

8 Martin F. Porter. An algorithm for suffix stripping. *Program*, 14(3):130–137, 1980.

9 Sara Silva, Rúben Pereira, and Ricardo Ribeiro. Machine learning in incident categorization automation. In *Proceedings of CISTI'2018: 13th Iberian Conference on Information Systems and Technologies*, 2018.

10 Yang Song, Jian Huang, Ding Zhou, Hongyuan Zha, and C Lee Giles. IKNN: Informative K-Nearest Neighbor Pattern Classification. *Proceedings of the European conference on Principles and Practice of Knowledge Discovery in Databases (PKDD)*, pages 248–264, 2007. doi:10.1007/978-3-540-74976-9{_}25.

11 Bruno Trstenjak, Sasa Mikac, and Dzenana Donko. KNN with TF-IDF based framework for text categorization. *Procedia Engineering*, 69:1356–1364, 2014. doi:10.1016/j.proeng.2014.03.129.