Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 239 (2024) 1927–1934

Procedia Computer Science

www.elsevier.com/locate/procedia

CENTERIS – International Conference on ENTERprise Information Systems / ProjMAN – International Conference on Project MANagement / HCist – International Conference on Health and Social Care Information Systems and Technologies 2023

Port request classification automation through NLP

Samuel António Beecher Martins[a,*], Nuno Garrido[a,b], Pedro Sebastião[a,b]

[a]*Instituto Universitário de Lisboa (ISCTE-IUL), Av. Forças Armadas,1649-026 Lisboa, Portugal*
[b]*Instituto de Telecomunicações (IT-IUL), Portugal*

Abstract

This paper describes a suggested prototype to carry out the automatic classification of requests from a Port Help Desk. It intents to ascertain if the implementation of this framework is viable for this sector. For this purpose different models were employed, such as SVM, Decision Tree, Random Forest, LSTM, BERT and a SVM hierarchical model. To verify their efficiency these models were evaluated using Precision, Recall and F1-Score metrics. We obtained F1-Scores of 94.36% and 92.48% when classifying the request's category and group respectively. A F1-Score of 93.41% while using a SVM model for category classification when employing a hierarchical classification architecture.

Keywords: Help Desk; NLP; Request classification; Machine Learning; Port Systems.

1. Introduction

In the 21st, the Portuguese port sector has been consistently and rapidly digitizing. Over two decades, two generations of Information Systems have already been implemented and we are currently in the transition to the third generation.

* Corresponding author.
E-mail address: sabms@iscte-iul.pt

10.1016/j.procs.2024.06.376

This digital evolution has been characterised by the increased complexity of these systems. In the first generation of Port Systems only the direct actors involved in a ship's stay in port were included as direct participants on these systems. In the second generation, called the "Janela Única Portuária" (JUP), there was a great focus on communication with external entities as well this led to the inclusion of the railway module and the module for managing the stay of containers in the various Iberian logistics warehouses. This process of moving from a Port Community System (PCS) to a National Single Window (NSW) system led to the third generation, called "Janela Única Logística" (JUL), aims to cement the passage of the Information systems to the NSW concept, as well as evolve the PCS of each Portuguese Port Administration. These efforts will permit the centralization of the various support applications of each port and add the concept of national layer and will further the dematerialization process [1].

These applications have dematerialized, many of the processes that previously were executed directly by human intervention with other actors have been digitized, and this forced many of the operators to embrace the digital sector. As a result of this transformation, many of the actors are now users of JUL application and as such they have been encountering some constraints and/or issues in their day-to-day operations while using the port applications. Hence the Port Administrations have developed processes to support users so that the constraints can be overcome while maintaining user satisfaction and confidence in these applications. This development has culminated in the creation of help desk departments, and these departments are the front-end contact with the user. They identify the constraint and proceed to its resolution, satisfy the request and questions, as well as delegate the situations out of their scope to the relevant department, when the situation does so require [2].

In this context the users contacts the Help Desk Department of the Port Administration when they encounter the need for support via email and/or phone call. If they only contact via email this request creates a formal request for support. Once a formal request is created, it is categorized and manually assigned to a Help Desk team member, as shown on left side of Fig.1.

This study proposes the creation of a prototype to perform the distribution, and categorization of help desk requests, in an automatic way, to reduce the costs in human resources associated with this process and optimize the process of pre-analysis of the formal requests received. It also features development of a prototype that will allow for the collection and analysis of the results obtained and compare these to existing metrics of quality. Thus, transforming the process shown on the right side of Fig.1.

This paper is organized as follows: in section 2 we will look at the state of the art where NLP tools were implemented, with a special focused on unbalanced data sets; In section 3 we perform a brief overview of how the current manual classification is executed and where the suggested automatization will operate; in section 4 we review the available data set for this paper and how we have approached it; in section 5 we disclosed which NLP models where used and how they were configured; in section 6 we will review the results obtained and on section 7 we will cover the conclusions that were reached with this study.

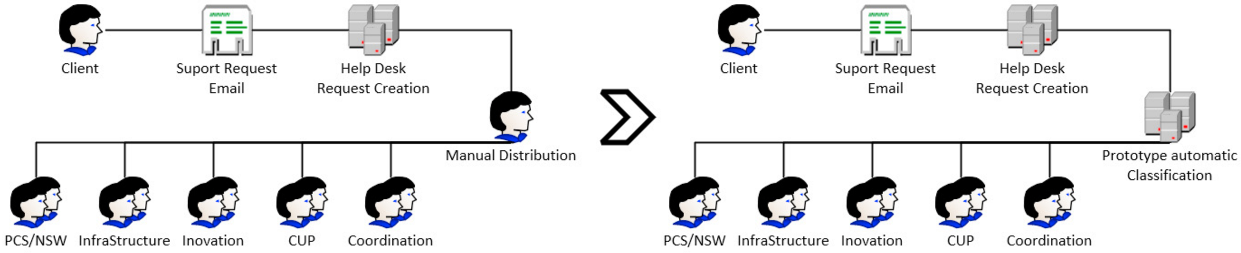

Fig. 1. Manual distribution as is vs suggested distribution.

2. Literature review

With the increased use of Customer Relationship Management (CRM) platforms, the implementation of Ticket Management Systems (TMS) is a valuable technical solution and tool both in the distribution, prioritization, and management of requests from customers [3].

Because the CRM system is the base for the classification process, many authors have tackled these in different manners. For instant the author [4] receives the customer information through the help desk application itself, being that the customer is enrolled in it. In this context an artefact was developed that applied vectorization to the term frequency–inverse document frequency (TF-IDF) model and subsequently to a Support Vector Machine (SVM) model for the classification of help desk requests for the German Jordanian University and achieved an accuracy rate of 83%, using description, subject and comments of the requests for classification.

The system developed in [5] performs classification by considering the description of the requests, and the there is a strong emphasis in comparison the classical SVM and Naive Bayes models. For this Corpora the Bagging-SVM model obtained the best results. As in preceding project, the authors in [6] performed a comparison between SVM, Naive Bayes, Logistic Regression and Multinomial Naive Bayes models. In this study, the authors verified that they obtained the best results for their Corpora, with about 87% accuracy using the SVM model.

In [7, 8] the authors developed two models for the Portuguese Navy, both addressing the issue of classification of emails sent to this institution. The models were developed with the intent of reducing the manual labour time occupied by this activity, and to reduce the human error in this activity. Although both authors shared the theme, the author [8] applied more traditional models. The best result was obtained by author [8] using the Linear Regression model with about 82,5% of accuracy. Author [7] applied Bidirectional Encoder Representations from Transformers (BERT) and Linear Support Vector Classification (SVC), obtaining an accuracy of 92%.

In [9, 10] the authors detected a significant constraint in the distribution of data that had a large number of records associated with one or two categories. They used undersampling methods to attempt to overcome this constraint. This method consisted in the random removal of records in classes with more representation in the training process. Another method used by these authors was the oversampling of the less populated classes to increase their weight in the training vectors. Another attempt was a classification by phases: in the first phase the less represented classes are labelled, and subsequently the more represented classes. In this way there is no direct competition of less represented classes with the more represented classes.

3. Help desk manual classification

This paper focus on the development of an add-on application to the current help desk system. This add-on will automatically classify the category and group of the request through the use of the “Subject” of the request. We are going to use three standard Natural Language Processing (NLP) models, a Neural Network and a Transformer in order to identify which model produces the best results regarding the data set. To have a larger data set and one that is less sensitive to seasonal issues, the data set being used was extracted from the actual help desk system for all 2022.

As mentioned in the introduction, the request in the help desk system is opened after a claimant sends an email to the service desk email server, or a member of the service desk opens one manually. In the first case, a service desk member must enter the created request and fill all the mandatory classification fields. Regarding the second case, as the service desk operator opens the request, he will also classify it. The prototype "Gestão Melhorada de Pedidos" (GMP) will focus on the first case, since in the second use case it is expected that the user classifies the request correctly.

4. Data set

The data set is reviewed employing a two-way approach. First, we will review how the data was extracted from Help Desk, then we briefly analyse the data format and its distribution in groups and categories. Finally, we will examine the data cleaning process that was used during the current experiments.

4.1. Data extraction

The selected data pertained to all the requests that were registered during the year of 2022 coming to total of 19897 formal requests.

The report functionality native to the help-desk software was used to extract the data. The file with the extracted data is a CSV file containing the requests. It has the following columns: "Request ID", "Subject", "Requester", "Description", "Category", "Subcategory", "Group", "Creation Date".

As mentioned in the previous section, we will mainly focus on using the Subject to determine the group and category of the request. The rest of the columns were extracted for other future developments so they will not be further used or mentioned in the current paper.

Regarding the Subject, this column can generally have all sorts of characters but does not exceed the length of 200 characters. In this column the Requester will briefly explain the motive of their enquiry, and depending on their background will resort to the use of coded sequences, such as a transport container plate and the respective applicational form on which analysis is required.

The "Category" column consists of the main 16 fields of Operation where the service desk operators have direct intervention. In this paper these categories will be identified as A, B, C, D, E, F, G, H, I, J, K, L, M, N, O and P.

The "Group" column has the five work groups that encompasses the reality of the service desk operators. This work groups tend to be specific to the operators' training and their current workstation at service desk operation. It is rare that an operator belongs to more than one group. The groups in this paper will be identified as G1, G2, G3, G4 and G5.

4.2. Data preparation

We carried out a series of text pre-processing methods which we will now list in order of execution: removal of the "RE:" prefix commonly associated to the response of a previous email; removal of the "FW:" prefix commonly associated to the forwarding of a previous email; lower casing of all characters; removal of Portuguese stop-words; removal of special characters; various standardization of maritime var-char sequences regarding the maritime business that we will clarify in the paragraph below, and lastly removal of numeric digits.

The maritime var-char sequences are coded sequences that offer specific information in a strict and direct manner, such as the vessel call number commonly used by the Port authority, e.g., "PTSIE123022272". that can be translated as: "PTSIE" -» Locode for port of Call; "1" means of transport that on the example means maritime, "23" is the year, and it ends with a six-digit sequential number. In Table 1 we give a full set of examples of these codes, as well of their meaning and we prepared them for tokenization. To be noted that we have applied this process to reduce the number of what would have been unique tokens, in a series of tokens that are more transversal in all the Corpora.

Table 1. Examples of Maritime sequences.

Meaning	Regular Expression	Conversion
Sines Maritime Transport	ptsie1[\d]{8}	ptsiea
Bobadela Rail Transport	ptbbl2[\d]{8}	ptbblb
Portimão Road Transport	ptprm3[\d]{8}	ptprmc
Bill of Lading	[a-z]{5}[\d]{7}	bill_lading
Container Plate	[a-z]{4}[\d]{7}	cn_plate
Customs Document	[\d]{2}[a-z]{2}[\d]{14}	digi_doc

Finally, as we will further explore below, we noticed that some of the categories that used by the operators did not correspond with the Group indicated. For example, this occurred with person identification software, that can have three categories, depending on which working group had to act upon the request (e.g. G1, G2, G3). This operation also involved some alterations in the classification regarding the group, that proved to help the data unbalance a little. Also, there were two categories that were renamed, by the company, during the year 2022 and as such there was the need to convert the previous nomenclature to the latest one.

In terms of Category request distribution of data is clearly illustrated in the last two columns of Table 2 where we can see that most requests belong to the categories "A" and "C" which have 11711 and 4655 requests respectively out of the 19897 total requests. A similar situation occurred when it came to the Group request Distribution. As can be seen in the first two columns of table 2, the groups G1 and G3 received the majority of the requests.

Table 2. Group and Category request distribution.

Group	Group count	Category	Category Count
G1	11717	A	11711
G3	6424	C	4655
G2	951	B	980
G5	281	D	825
G4	81	I	386
		E	370
		G	285
		F	238
		H	191
		K	95
		L	68
		P	30
		O	29
		M	20
		N	14

5. GMP Development

This section analyses the approaches that were used after the collection of the data and its preparation for model training and testing.

For this execution a standard TF-IDF Vectorization for all the models was used. We will be focusing on two classification architecture. The first architecture with pertains to the classification of the group and category as standalone features, with isolated NLP models. While the second architecture classifies the group and category in a hierarchical manner that we will further describe bellow.

5.1. Category and Group classification as Standalone

For the first experiments the request subject was used to the train the models to predict the classification of the Groups and the Categories. The models used for this classification were SVM, SVM with artificial oversampling, Decision Tree, Long Short-Term Memory (LSTM) algorithm and a BERT algorithm. For all these models the vectorization of the training data set was made using the standard "TfidfVectorizer" function, and the train and test split was applied using 20% of the data set as the test vector and with the random state as 100.

We used the base function, from “Sklearn”, with exception of the gamma set to auto in the SVM algorithm. We also added the function One versus Rest, as this function will help this model deal with the unbalanced data set, since it fits each class to the model while comparing it against all other classes, doing this one class at the time. We used this SVM configuration in order to test artificial oversampling, applying this function to the train and test vectors for this model, since this will inflate the values present on the less populated classes, to the same number as the biggest class.

In the Decision Tree algorithm we used entropy we used the entropy criterion, the max dep depth of three and the random state of 0. For the Random Forest Model, the model was set to have 400 estimators, entropy criterion and a random state of 0.

We collected the LSTM algorithm, from the “Keras” library. I had an embedding layer with 250 neurons, followed by a spatial dropout layer of 0.2. After these two layers the network arrives at the LSTM layer which is followed by a final layer with the same number or neurons as there are classes, e.g. 15 for Category and 5 for Group. The number of epochs was set to 100, but we also applied an early stop function with 10 epochs of tolerance. In general, for the category LSTM model the recurrent number of epochs was around 16 and for the group classification around six epochs.

The final model we used was the BERT base uncased model from the “Torch” library. Since it is a BERT algorithm we also used a tokenizer from this library and the respective label encoder and tensor. Regarding the model configuration we used the batch size of 16, the Adam optimizer and the learning rate of “2e-5” and. 8 epochs for training.

5.2. Category and Group classification as dependents

In terms of Category classification, we noticed that category "A" had a high number of false positives. There was also a dispersion of classifications regarding categories "B", "K" and "A", since these three Categories share the same core business, but pertain to from different areas of resolution. The Group classification was also affected but to a lesser extension. The models in the previous section operated in an "all classes versus all classes" fashion for category classification. We consider functionality could be improved if the subject of the request was classified by Group first and depending on this result: Category in order to reduce the number of classes in each category model.

Using the business rule that all categories aren't available for all service desk operators, and the selection of categories are highly dependent on the group of operators, we have developed a new model that uses a hierarchical architecture to classify the category, following the structure depicted in Table 3.

Table 3. Hierarchical model, categories distribution.

Group	Categories
G1	A, M
G2	B
G3	O, P, N, M, L, C, D, E, F, H, K
G4	F, K, D, E, G, H, N
G5	I, G, P, M, E, K

In this model, the first step was to separate the data set into two tables, one for training with 15918 requests (80%) and another with 3979 requests (20%) for test. Subsequently the SVM for Group classification model, as described in section 5.1, was trained using the train data set and the predictions from this model were added to a new column on the test data set denominated "Predictions". From this point forward, train and test data sets were again split depending on which group, they were labeled has in the column "Group" for the train data set, and the column "Predicted_Group" for the test data set. Each train-data set was used to train a group specific SVM model. The SVM model that we used had the same specs as the one described in the previous section, except for the train and test split function. After the generation of each model, the Group specific data set for each group was tested, and the predicted values were added in a new column "Predictions_Category". The only exception was the Group "B", since this Group has only one category and as such the value of the category "B" was simply applied to the column " Predictions_Category" of its test data set. Finally all the test data sets were merged, thus giving valid metrics for this model.This architecture was also applied using the same Decision Tree, Random Forest and BERT models with the same specs as described on section 5.1, but always having the group classification executed by the SVM model.

6. Simulation results

In this section we present two different simulation results: results for Group and Category Classification as Standalone models, and results for the hierarchical architecture. This approach allows a deeper analysis of the architecture of each model. The metrics used for the comparison of these models will be the F1-Score, since it has a balanced approach gauging the model results, when compared to precision and recall, especially due to the unbalanced nature of the that set.

6.1. Category and Group classification as Standalone results

Regarding the group classification as can be seen in the column “Group Classification 5.1” of table 4, the model with best results was the SVM model with an F1-Score of 94.36% followed by the BERT model with 93.92%. The SVM model with artificial oversampling, Random Forest model and the LSTM model had the F1-Score of 93.56%, 93.36% and 93.11% respectively. The Decision Tree Model was the weakest model with the F1-Score of 77.47%. In general, the Group models had better metric scores, when compared to the Category models.

The following results are the results for the category models, as described in subsection 5.1, and are available in column “Category Classification 5.1” of table 4. It was also visible that the SVM model has the best results with the F1-Score of 92.48%, followed by the BERT Model with 91.88%, the Random Forest model with 91.65%, the LSTM model with 90.84%, the SVM model with artificial oversampling with 88.75% and lastly the Decision Tree model with 73.93%.

Regarding the experiments in this section, in general the major concern was the distribution of the predicted request. Groups "G1" and "G2" had a significant number of false positives since these groups were equivalent to roughly 90% of the data set. This outcome was even worse with the category classification where categories "A" and "C" corresponded to around 85% of the requests in a total of 15 categories. Regarding the category classification another situation of concern, as mentioned in section 5.2, were the Categories “B", "K" and "A" as they pertain to the same application but have different resolutions depending on the Group. As the subjects of these requests are quite similar the system had difficulties dealing with these requests.

Regarding the experiments with SVM with an artificial oversampling, it was noticed that even though the number of false positives for the major groups and categories decreased, the number of false negatives had a major increase. This is especially evident in the category prediction when comparing the SVM with the SVM with oversampling models recall metric. The values were 92.84% and 86.83% respectively, meaning that the major classes lost a lot of requests, and this hurts the overall prediction results.

A major advantage of the more classical models (SVM, Decision Tree and Random Forest) was the fast-training time. In general, this was accomplished in less than 10 minutes. On the other hand, the LSTM model needed around one day to train and the BERT model two days, being more resource intensive. To be noted that these models were trained using CPU with 32 GB of RAM.

6.2. Category and Group classification as dependents results

While observing the results for these models in column “Category Classification 5.2” of table 4, it is noticeable that no models were made using the LSTM and SVM with artificial Oversample models. We didn’t use the LSTM model since previously it had not performed has well as the BERTH model and they have the same degree of complexity when setting them up for this architecture. The SVM with artificial oversampling was dropped since it wasn’t returning the expected results and increased the training time exponentially.

The F1-Score for these models were 90.60% for the Decision Tree model, 92.98% for the Random Forest model, 93.41% for the SVM model and the result of 92.04% for the BERT Model. Regarding the results obtain by these models, we can clearly observe that even though the Decision Tree model has once more the weakest results, they have been improved from 73.93% to 90.60%. The BERT model only had a marginal improvement of less than 0.5%. Regarding the other models, the Random Forest model and the SVM model demonstrated an improvement of around 1% when compared to the category models discussed on section 5.1.

A major contributor to this situation was the fact that the categories "B", "K" and "A" were no longer in direct competition with one another, and that the categories "A" and "C" were now compartmentalized on their own group, diminishing the number of false positives that these categories had previously exhibited in the previous section.

One of the weaknesses of this architecture is that the category classification results are highly dependent on the results for the SVM model for group classification, since in this classification if a request that should have been classified as one group was classified as another, the following category classification is compromised.

Table 4. Models results.

Model	Group Classification 5.1 (F1-Score %)	Category Classification 5.1 (F1-Score %)	Category Classification 5.2 (F1-Score %)
SVM	94.36	92.48	93.41
SVM (art. oversample)	93.56	88.75	
Decision Tree	77.47	73.93	90.60
Random Forest	93.36	91.65	92.98
LSTM	93.11	90.84	
BERT	94.29	91.97	92.04

7. Conclusion

Our study shows that the SVM model for Group classification and the SVM in a hierarchical architecture for the Category classification have produced the best results with a F1-Score 94.36% and 93.41% respectively. It should be noted that when these two models are combined and adding the data preprocessing, the total time needed for this system to be functional was about 10 minutes, meaning that it gives good results with a low resource consumption. In general, these results prove that this System is a viable option for the classification processes of the requests.

Another promising option is the BERT models, since they have quite a good result, that with more fine tuning could have been improved. The major drawback of the BERT and LSTM models is the resource consumption necessary for their training, even when considering that the Corpora is quite limited, since only the subject of the request was used. Some points where this application can be improved are deeper fine-tuning of the models used to improve the models' training; gauging the end users' perception of this kind of application; addiction of more classification features like the subcategory of the requests.

References

[1] Pinto, C. Impactos organizacionais, informacionais e tecnológicos da implementa ̧c ̃ao da Diretiva 2010/65/UE: uma proposta de solução nacional. (Master's thesis, Instituto Politécnico de Setúbal. Escola Superior de Ciências Empresariais (2016))

[2] Santos, D. Serviço de Helpdesk Automático. Instituto Politécnico do Porto (2020)

[3] Zicari, P., Folino, G., Guarascio, M. & Pontieri, L. Combining deep ensemble learning and explanation for intelligent ticket management. Expert Systems With Applications. 206 pp. Article 117815 (2022)

[4] Al-Hawari, F. & Barham, H. A machine learning based help desk system for IT service management. Journal Of King Saud University Computer And Information Sciences. 33, 702-718 (2021)

[5] Larasati, P., Irawan, A., Anwar, S., Mulya, M., Dewi, M. & Nurfatima, I. Chatbot helpdesk design for digital customer service. Applied Engineering And Technology. 1, 138-145 (2022)

[6] Paramesh, S. & Shreedhara, K. Automated IT service desk systems using machine learning techniques. Data Analytics And Learning: Proceedings Of DAL 2018. pp.331-346 (2019)

[7] Neves, V. Automatic classification of correspondence from a public institution. (Master's thesis, Instituto Superior Técnico. (2021)

[8] Fazendeiro, A. Automatic Correspondence Distribution for a Public. (Master's thesis, Instituto Superior Técnico. (2021))

[9] Marcuzzo, M., Zangari, A., Schiavinato, M., Giudice, L., Gasparetto, A. & Albarelli, A. A multi-level approach for hierarchical Ticket Classification. Proceedings Of The Eighth Workshop On Noisy User-generated Text (W-NUT 2022). pp. 201-214 (2022)

[10] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A., Kaiser, L. & Polosukhin, I. Attention is all you need. Advances In Neural Information Processing Systems. 30 (2017)