Received 2 September 2022, accepted 1 October 2022, date of publication 10 October 2022, date of current version 17 October 2022.

Digital Object Identifier 10.1109/ACCESS.2022.3213033

APPLIED RESEARCH

A Text Classification Methodology to Assist a Large Technical Support System

ELENE FIRMEZA OHATA[1], CÉSAR LINCOLN CAVALCANTE MATTOS[2], SAMUEL LUZ GOMES[3], ELIZÂNGELA DE SOUZA REBOUÇAS[4], AND PAULO ANTONIO LEAL REGO[2], (Member, IEEE)

[1]Department of Teleinformatics Engineering (DETI), UFC, Fortaleza, Ceará 60355-636, Brazil
[2]Department of Computer Science (DC), UFC, Fortaleza, Ceará 60355-636, Brazil
[3]Department of Electrical Engineering (DEE), UFC, Fortaleza, Ceará 60355-636, Brazil
[4]Federal Institute of Ceará (IFCE), Canidé, Ceará 62700-000, Brazil

Corresponding author: Elene Firmeza Ohata (elene.ohata@lapisco.ifce.edu.br)

ABSTRACT Text-based tools for reporting technical issues and receiving support are widespread in commercial applications, such as customer services and internal corporate communication. Past issues recorded in such systems may provide valuable knowledge for better handling future interactions. Nevertheless, the predominance of short messages and the presence of specific domain subjects constitute additional challenges. In this work, we aim to build an assistant for a system operating in a large company that provides asynchronous services for technical support. It is known that some repetitive technical issues can be handled with simple standard messages, named *templates*. Thus, we propose a modular pipeline based on natural language processing and machine learning algorithms to enable raw text processing, feature extraction, and supervised learning to recommend suitable templates from a given textual description of the incoming issue. In a real-world scenario, the proposed pipeline achieved an average accuracy of 72.7%, a promising result for a setup with 9 classes and few labeled training instances. Moreover, a *post hoc* analysis shows how our methodology is able to correctly identify the words more closely related to the corresponding templates.

INDEX TERMS Natural language processing, text classification, technical support, automated assistant.

I. INTRODUCTION

Using Artificial Intelligence (AI) in daily company operations is a promising path to achieve several business benefits. In recent years, big companies have been building solutions to support tasks using Natural Language Processing (NLP), a subfield of AI that enables computers to understand human-generated information. Such text-based tools make it possible to speed up services, automate repetitive actions, reduce costs, and increase the accuracy and efficiency of people's work [1].

Natural languages are what humans use to share information with each other, for instance, English or Portuguese. Unlike computer programming languages, they are not intended to be translated into a finite set of mathematical operations, and the information is unstructured. Therefore, NLP usually follows several steps, such as language detection, misspelling correction, tokenization, and feature extraction, to convert natural language into numbers that can be processed. After extracting structured numerical data from natural language, we can apply machine learning (ML) algorithms to make conclusions, such as classifying the subject of a given piece of text [2].

NLP has been used in real-world scenarios to develop Medical Decision Support Systems (MDSS) for diagnosing and treating patients effectively and efficiently [3]. In such applications, an NLP software can scan the clinical text within seconds and synthesize the content into important points, which enables faster and more valuable decision support. This procedure frees up physicians and staff resources to focus on patients while also helping make insightful conclusions based on precise data and reducing the time spent on redundant administrative activities.

Law firms have also been using NLP for complex, automated analysis of document contents with the goal of cost and time reduction, as well as better human resources allocation [4], [5]. One possible application is the ability to check or find the relevant legal clause in accordance with

The associate editor coordinating the review of this manuscript and approving it for publication was M. Shamim Kaiser.

current regulations and corporate standards. Tools such as AQUILA [6] take a few seconds to fully analyze a given document.

Large telecommunication companies have also been using NLP to classify e-mails into distinct labels to improve customer support [7]. This enables support personnel to handle requests quicker and more efficiently, establishing stable and positive relations with the customer.

Following the above successful experiences, the current work aims to propose and evaluate an AI assistant using NLP and ML algorithms. The goal is to provide automatic assistance to an existing system operating in a large computer manufacturing company that provides text-based asynchronous services for technical support issues. Such a solution is tailored to be applied to real-world text data related to client technical support.

The proposed assistant classifies the issue type and, if possible, provides a suitable standard problem resolution in the form of a text *template*, previously written by a human expert. This solution intends to improve speed and assertiveness in resolving technical problems, reduce costs, and provide better client satisfaction.

We comprehensively evaluate several combinations of text feature extraction methods and learning algorithms using real-world data to obtain the most suitable solution, given complexities such as the predominance of short messages, the presence of domain-specific subjects, and class imbalance. The best solution is further analyzed by means of the LIME package [8], which enables verifying which input words lead to the recommendation.

Thus, the main contributions of this work are as follows:

1) The proposal of a modular pipeline based on NLP and ML methods to recommend suitable standard resolutions to technical issues;
2) The evaluation and comparison of several combinations of text feature extraction methods and machine learning algorithms with challenging real data, which is comprised mostly of technical wording and short sentences;
3) The analysis of the best solution using the LIME package.

The next sections are organized as follows. Section II shows an overview of the related work. Section III details the task of handling an incoming text describing a technical issue and how we aim to approach it. The proposed methodology to classify texts into standard resolutions is explained in Section IV. The experimental results are presented and discussed in Section V. Finally, Section VI presents the conclusion and further work.

II. RELATED WORK

Several approaches to help understand the human language to better respond to their demands have already been proposed in the NLP literature and achieved promising results [7], [9], [10], [11].

Borg et al. [7] proposed the use of ML to classify e-mails in 33 categories to improve customer support in a telecommunication company. The authors affirmed that the e-mails' categorization could support the employees in selecting the e-mails that better match their expertise, resulting in quicker responses. The authors investigated Bag of Words and Word2Vec embeddings for text representation. They evaluated traditional classification algorithms and deep learning methods. The authors affirmed that the Long Short-Term Memory (LSTM) network achieved the best result, reaching an F1-Score of 91%.

Barbosa and Godoy [11] proposed a BERT [12] model chatbot with structured data to increase customer support using an automated receptionist. The goal was to create a chatbot comprised of a finite state machine proposal that helps predict the customers' contact motivation. The authors stated that they obtained benefits in business impact compared to classical NLP models.

Hardalov et al. [13] presented a study comparing two different models used for automated customer support using a dataset collected from Twitter. The authors used a retrieval model and two generative neural networks: a sequence-to-sequence (seq2seq) model and a Transformer model [14]. They concluded that the seq2seq model outperformed the others regarding semantics and word overlap.

Liu et al. [15] proposed the use of customer complaints to identify the customers' expectations, mainly for service recovery, using NLP and ML methods. The dataset was composed of more than 68 thousand text-based instances, where each one had many features, including the main customer expectation. The authors proposed the reduction of six main expectations into three classes (solving problems, financial and psychological compensations). They stated that these new categories could be extended to other complaint contexts.

The work proposed by Park and Gates [16] shows that it is possible to measure customer satisfaction automatically by analyzing call transcripts. The authors use ML to predict the degree of customer satisfaction. This work used four classification methods: Decision Tree (DT), Naive Bayes, Logistic Regression, and Support Vector Machines (SVM). They affirmed that DT and SVM performed better when the full or half of the conversation was available. They reached 66.09% in accuracy considering 5-point satisfaction and 89.42% when executing 2-point classification.

Hoffimann et al. [17] proposed retrieving data in drilling reports and classifying them into three classes to accomplish accident identification and operation optimization. Some of the challenges were unfinished sentences, technical symbols, and abbreviations. The authors analyzed the use of a skip-gram model with three neural networks, in which the LSTM was the one with the best performance.

The proposed works by [7] and [17] used the LSTM network to resolve their problems. However, in our work, using a LSTM network would not be interesting since we do not have a large amount of labeled data, as will be detailed in Section IV-A. Moreover, the texts in our dataset are short

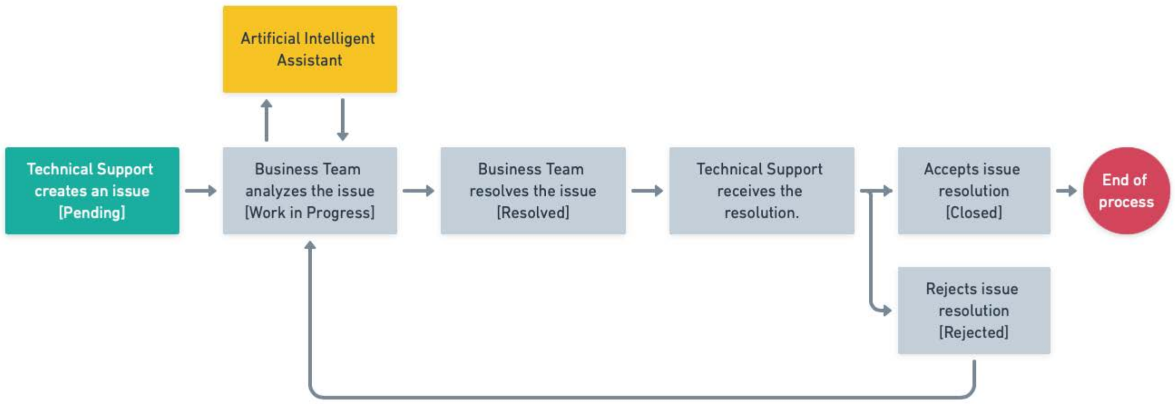

Artificial Intelligent Assistant
Technical Support creates an issue [Pending]
Business Team analyzes the issue [Work in Progress]
Business Team resolves the issue [Resolved]
Technical Support receives the resolution.
Accepts issue resolution [Closed]
End of process
Rejects issue resolution [Rejected]

FIGURE 1. Creation and resolution steps of an incoming issue. The issue is created by a technical support agent. The business team analyses the issue seeking a resolution. The technical support agent can accept or reject the resolution. Rejected resolutions require new resolutions by the business team until it gets accepted. The AI assistant acts by proposing templates (standard resolutions) for the business team.

and comprised mainly of technical wording. In Section III, we describe the real scenario in which our solution will be applied, and in Section IV, we detail the proposed methodology.

III. PROBLEM DESCRIPTION

In the company where this work has been developed, the information pertinent to the equipment or system component that requires maintenance is sent from the technical support agent team to the business team via a text-based tool. After a manual analysis, the business team sends a textual feedback to the agent based on his/her knowledge and experience. Then, the technical support agent can accept or reject the given resolution. If the resolution is rejected, the business team proposes alternatives. These steps are illustrated in Figure 1. It is important to emphasize that the complexity of such a process raises with the availability of more products and more incoming issues. This practical scenario motivates the use of an AI assistant.

A preliminary analysis of the available historical technical support data revealed that a significant amount of the reported technical issues were previously resolved or were part of groups of issues with similar solutions. The proposed AI assistant aims to recommend standard resolutions for the business team, as illustrated in Figure 1.

The data relating to resolved issues was subdivided into nine main groups. Then, eight standard responses were created in the form of templates: one for each group and one for issues where a standard solution would be inadequate. The standard resolution identified by the proposed AI assistant is then presented to the business team as a suggestion for handling the received technical issue. The expert may use the standard response, if it is appropriate, or he/she may customize the feedback text. The automatic identification of repetitive resolutions for similar technical issues is expected to bring greater agility to the support pipeline.

The initial data analysis revealed that the data contains nonstandard texts due to abbreviations and jargon; we should highlight that some of these words are only used in the company environment where the data was collected. Moreover, the presence of several orthographic errors was also noticed. Another challenging aspect of the data is that sometimes different business agents solve similar issues in different forms. Section V-D presents four raw text samples and their preprocessed counterparts. In the next section, we detail the executed preprocessing steps that aim to mitigate these challenges.

IV. PROPOSED METHODOLOGY

The proposed data flow is illustrated in Figure 2, where the different modules to classify issues into templates are depicted. The methodology is comprised of multiple stages. First, the data is preprocessed with filters and standard NLP techniques. Second, text feature extraction is performed, followed by classification using ML algorithms. Finally, performance metrics are computed, and results are reported. Each step is explained in the following subsections.

A. DATASET

In collaboration with the business team that uses the system currently in operation, eight templates were developed to help resolve the issue. The dataset was built using historical information, in which 1049 issues were labeled with a corresponding template. A ninth ‘‘none’’ template class was also considered (totaling nine classes) since some issues require technical specific resolutions and a predefined template would be inadequate.

In Figure 3, the number of issues labeled in each template class is shown. It is possible to note that despite the templates being defined after the input of the business team, the problem is highly imbalanced. Moreover, we emphasize that the

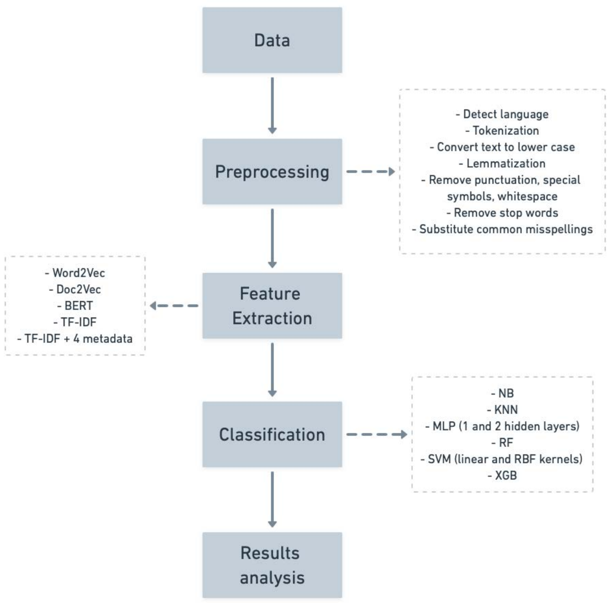


FIGURE 2. **Flowchart of the proposed methodology. The collected historical data is preprocessed using different NLP techniques. The "cleaned" data is encoded using one of the five evaluated text feature extraction methods. The features are classified with one of the eight chosen ML algorithms (including some variants). Finally, the classification results are analyzed.**

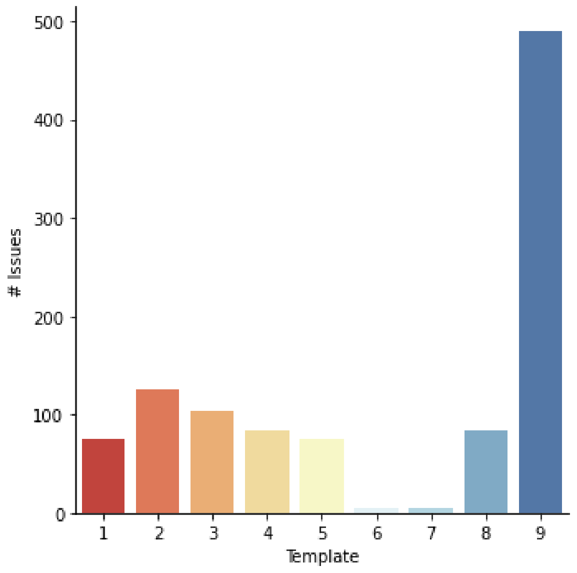


FIGURE 3. **Amount of labeled issues for each template. Notice the low quantities of examples for the classes "6" and "7", which are rarer. The class "9" refers to the "none" case, which should be redirected to manual analysis.**

small quantity of labeled examples constitutes an additional challenge.

We should note that more templates were considered during the data analysis. However, it was identified that some of them were rare cases; therefore, the business team would rarely use such templates.

B. PREPROCESSING

This work concentrates on classifying data from incoming technical issues into a template to assist the existing system operators. Nonetheless, the data is complex to classify due to its unstructured composition, such as text containing orthographic errors, truncated phrases, technical information, and technical abbreviations. Thus, the raw textual data requires preprocessing before being useful for the classification step. The preprocessing stage is also responsible for cleaning up the unstructured textual data and removing the noise from the data, such as undesirable abbreviations, special characters, symbols, and improper language.

In summary, the preprocessing techniques employed were: language detection (only English sentences are currently supported), tokenization, conversion to lower case text, lemmatization, common misspellings correction, removal of punctuation, special symbols, extra white spaces, and stop words.

C. FEATURE EXTRACTION

Before choosing a classification algorithm for the task, it is essential to transform the unstructured data into a numerical vectorized representation.

In the NLP and ML literature, such a step can be performed by many techniques [12], [18], [19], [20], [21], [22]. In this study, we consider four methods to encode the tokens of a given technical issue text into a vectorized representation: the well known term frequency-inverse document frequency (TF-IDF) [23], Word2Vec [18], [19], Doc2Vec [20], and BERT [12].

We apply a hyperparameter optimization step for the TF-IDF method using the same pipeline from the classification stage (see Section IV-D). The threshold for the removal of frequent tokens (max_df) is varied between 0.3 and 1. The number of n-grams was limited to unigrams, bigrams, or trigrams. Furthermore, the tokens that appeared in less than five issues were removed.

D. CLASSIFICATION

Given the available labeled samples, we are able to use supervised techniques to identify the template related to each of the issues. Therefore, we evaluate eight[1] ML methods based on: Support Vector Machines (SVM) [24], Random Forest (RF) [25], Multilayer Perceptron (MLP) [26], Naive Bayes (NB) [27], Extreme Gradient Boosting (XGB) [28], and k-Nearest Neighbors (KNN) [29]. We choose these methods due to their widespread use in practical NLP classification problems [30], [31], [32].

The classification stage is conducted in two steps: 1) training and 2) testing.

1) **Training:** In this step, 80% of the labeled data is used to perform the model training. Table 1 presents the

[1]We count SVM with linear and Radial Basis Function (RBF) kernels, and MLP with a different number of hidden layers as separate methods, totaling the 8 options presented in Table 1.

TABLE 1. Setups for each classifier in the hyperparameter optimization step.

Classifier	Macro Setup	Parameter	Setup
NB	Multinomial	-	-
KNN	-	number of neighbors	[3, 5, ..., 17]
MLP	one hidden layer (MLP1HL)	neurons in the hidden layer	2 to 1000 in steps of 50
	two hidden layers (MLP2HL)	neurons in each hidden layer	2 to 500 in steps of 25
RF	-	number of estimators	25 to 2000 in steps of 50
SVM	kernel: linear (LSVM)	C	2^{-5},2^{-3}, ..., 2^{15}
	kernel: RBF (RSVM)	C	2^{-5},2^{-3}, ..., 2^{15}
		γ	2^{-15},2^{-13}, ..., 2^{3}
XGB	-	maximum depth of a tree	[3,5,...,15]
		number of estimators	100 to 1000 in steps of 50

considered hyperparameters configurations. The hyperparameters are optimized with a 15-iteration random search procedure, following a 5-fold cross-validation, except for the NB classifier. Then, each text feature extractor+classifier with tuned hyperparameters results in a distinct model, which is stored in the processing machine.

2) **Testing:** The remaining 20% of the labeled data is destined for the testing phase. The model classifies each vectorized issue on one class (template) using the stored model from the training step. Then, the performance metrics are computed.

Finally, we execute ten repetitions of steps 1) and 2). Each repetition follows a distinct training/testing split.

E. PERFORMANCE METRICS

To evaluate the performance of the obtained classifiers, we validate the results by calculating their accuracy and the resulting F1-score. The accuracy metric corresponds to the proportion of correctly predicted classes. The F1-score is the harmonic mean between Precision and Recall. However, since this is a multiclass problem, we use the weighted F1-Score. In this metric, the F1-Score is first computed separately for each label. Then, a weighted average is performed depending on the number of samples in each class. We choose the weighted F1-Score because the consulted business team has reported that some issues are not very frequent, despite having a standard resolution.

V. RESULTS AND DISCUSSION

As follows, we present the results obtained by using different text feature extraction techniques in combination with distinct ML methods to recommend a resolution template for a given technical support issue. Table 2 presents the results of 10 runs, as described in Section IV-D. We executed 40 experiments; this amount is resulted from the five text feature extraction methods combined with the eight classification techniques. The algorithms were implemented using Python 3.8, and the main used libraries were: scikit-learn [33], gensim [34], nltk [35], and spacy [36]. The hardware environment was composed of a computer with Windows 10 as the operating system, with an IntelCore i5 and 16GB of RAM.

A. COMPARISON OF TEXT FEATURE EXTRACTORS

First, we can observe that Word2Vec achieved the worst results compared with the other feature extraction methods, independently of the paired classification method; the highest score is achieved using RF, in which the accuracy reached 54.1%. This result is somehow expected due to the low amount of labeled data and the content specificity, which may lead to unsatisfactory Word2Vec training.

Even though Doc2Vec+RSVM improved the best Word2Vec result in 9%, we also consider this accuracy unsatisfactory. We hypothesize that the improvement occurred because the generated Doc2Vec vector can better represent the issue context. However, the available data is still insufficient for its training.

Using BERT as a text feature extractor slightly improved the results obtained by Doc2Vec; BERT combined with LSVM achieved an accuracy of 65.3%. Nevertheless, it is important to note that instead of the used pre-trained BERT, the issues texts present very specific topics, including technical abbreviations and jargons.

From Table 2, one can observe that the TF-IDF representation outperformed the other methods. This representation reached 69.8% of accuracy when combined with the LSVM classifier. The use of the TF-IDF method was not hindered by the specific language present in the available data. Moreover, its application paired with a classifier enabled pointing out relevant tokens, including technical terms and abbreviations, as will be exemplified in Section V-D.

B. SOLUTION REFINEMENT WITH METADATA

Since TF-IDF obtained promising overall results, we opted to add four metadata related to the technical issue to perform the template classification. Considering that the primary purpose of the system is to report issues about equipment, their parts, and system errors related to equipment, the new features are: (*i*) the world region corresponding to the component part subject of the incoming issue; (*ii*) the family and (*iii*) the system where the issue is located; and (*iv*) a parent category that

TABLE 2. Accuracy and weighted F1-score for each combination of text feature extraction technique and classifier.

Feature Extraction	Classifier	Accuracy (%)	F1-Score(%)
Word2Vec	NB	46.52 ± 1.82	30.25 ± 1.77
	KNN	46.43 ± 2.44	39.20 ± 2.35
	MLP1HL	46.76 ± 1.75	29.82 ± 1.88
	MLP2HL	46.76 ± 1.75	29.82 ± 1.88
	RF	54.10 ± 2.23	46.90 ± 2.59
	LSVM	50.33 ± 2.02	37.59 ± 2.31
	RSVM	50.90 ± 1.92	39.64 ± 2.72
	XGB	52.24 ± 2.83	48.47 ± 2.64
Doc2Vec	NB	46.76 ± 1.75	29.82 ± 1.88
	KNN	55.33 ± 2.96	54.41 ± 3.40
	MLP1HL	49.1 ± 3.80	35.02 ± 8.03
	MLP2HL	58.81 ± 4.44	52.22 ± 7.43
	RF	62.95 ± 1.44	58.37 ± 1.81
	LSVM	59.10 ± 2.61	54.25 ± 2.77
	RSVM	63.14 ± 3.09	59.18 ± 3.76
	XGB	62.29 ± 2.32	60.45 ± 2.35
BERT	NB	45.95 ± 2.57	45.56 ± 2.30
	KNN	56.71 ± 2.43	52.57 ± 2.57
	MLP1HL	59.71 ± 4.88	52.97 ± 8.63
	MLP2HL	63.05 ± 3.05	59.21 ± 3.97
	RF	58.67 ± 1.52	51.62 ± 1.83
	LSVM	65.33 ± 2.05	61.37 ± 2.71
	RSVM	65.14 ± 2.10	61.53 ± 2.69
	XGB	61.62 ± 2.73	58.39 ± 2.73
TF-IDF	NB	56.33 ± 2.33	46.75 ± 3.06
	KNN	63.81 ± 2.84	60.49 ± 2.61
	MLP1HL	65.19 ± 2.03	61.17 ± 2.9
	MLP2HL	65.76 ± 1.50	61.61 ± 1.9
	RF	69.00 ± 2.38	65.74 ± 2.64
	LSVM	69.81 ± 1.97	67.41 ± 2.25
	RSVM	64.81 ± 2.63	58.85 ± 2.87
	XGB	66.24 ± 2.78	64.44 ± 2.35

TABLE 3. Accuracy and weighted F1-score obtained by the TF-IDF combined with four metadata.

Feature Extraction	Classifier	Accuracy (%)	F1-Score (%)
TF-IDF + 4 metadata	NB	59.76 ± 2.03	51.18 ± 2.34
	KNN	65.62 ± 3.21	63.41 ± 3.28
	MLP1HL	69.67 ± 1.46	66.26 ± 1.98
	MLP2HL	68.52 ± 0.99	64.89 ± 2.45
	RF	72.67 ± 2.04	69.18 ± 2.25
	LSVM	69.05 ± 2.63	66.83 ± 2.91
	RSVM	68.48 ± 2.64	66.23 ± 3.00
	XGB	69.33 ± 2.57	67.68 ± 2.72

the technical support agent selects when submitting the issue for analysis. These metadata were encoded using a one-hot encoding strategy.

The results of the combination of TD-IDF with the four metadata are reported in Table 3. We can note that the best accuracy improved 2.9%, meaning that the four metadata are indeed providing relevant information. The best metrics were achieved using the RF classifier, in which the accuracy reached 72.7%, and the F1-Score reached 69.2%; this combination is highlighted in blue in Table 3. Such a performance increase was expected to be even greater. However, the component part turned out to be less informative since it can be related to very distinct issues, resulting in different resolutions and thus different labeled templates. In addition, we have also noticed that the agents often select an inappropriate parent category for the submitted issue.

Figure 4 presents a confusion matrix of the best combination (TF-IDF with four metadata + RF classifier). The matrix is an average of each confusion matrix obtained in each of the ten separate runs. We can observe that most confusions happen between one class and class "9" (no template). Such confusion is expected since issues with similar words can either have a template related to it or be too specific and do not have a standard resolution. Importantly, even when the model wrongly predicts a class "9", it would not significantly impact the support pipeline since this behavior prevents false positives, i.e., it avoids recommending wrong predefined templates. On the other hand, the samples that belong to class "9" and are not correctly labeled by the classifier would still not have a significant impact. The latter is achieved by treating the recommended resolution as a guide to the business operator, who may manually modify it. We emphasize that the undesirable situation would be the model classifying an issue as a wrong standard resolution, i.e., a confusion between classes "1" to "8". However, as we can note from Figure 4, only a few samples were confused with a different template class.

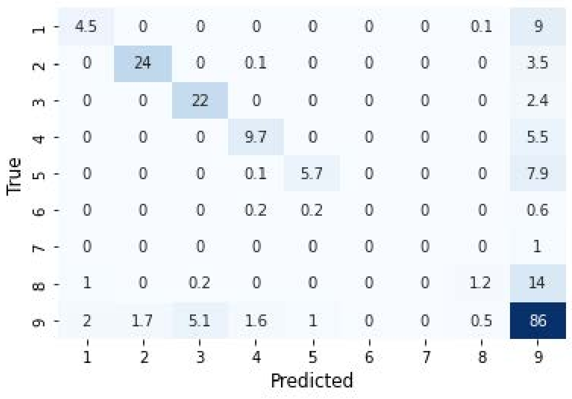


FIGURE 4. Confusion matrix of the combination TF-IDF with four metadata + RF.

C. STATISTICAL TESTS

We applied the Friedman test on the accuracy results from the ten iterations to verify the hypothesis that the classifiers have no significant difference between them. At the significance level $\alpha = 0.05$, we rejected the null hypothesis and concluded that there are significant differences among the classifiers (Friedman's statistics $F = 20.158$ and $p = 10^{-8}$). According to the *post hoc* Nemenyi test, the critical difference (CD) is 3.32. The CD diagram is presented in Figure 5, which shows that RF is significantly better than NB, KNN, RSVM, and MLP2HL.

Since RF was also among the best classifiers for all the text encoding methods, we additionally applied the Friedman test

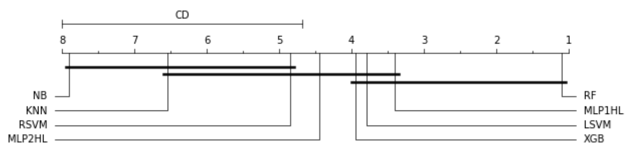


FIGURE 5. **Result of the Nemenyi test for TF-IDF + 4 metadata ranked by the classifiers.**

on the RF accuracy values to verify the hypothesis that the text encoding methods have no significant difference across the runs. At the significance level $\alpha = 0.05$, we rejected the null hypothesis and concluded that there are significant differences among the classifiers (Friedman's statistics $F = 40.0$ and $p = 4 \times 10^{-8}$). According to the *post hoc* Nemenyi test, the CD is 1.929. The CD diagram is presented in Figure 6, which shows that the strategy TF-IDF + 4 metadata is significantly better than Word2Vec, BERT, and Doc2Vec.

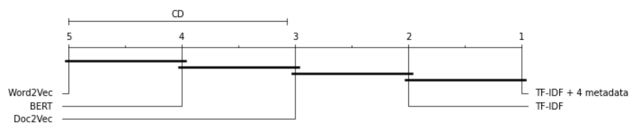


FIGURE 6. **Result of the Nemenyi test for RF classifier ranked by the text encoding methods.**

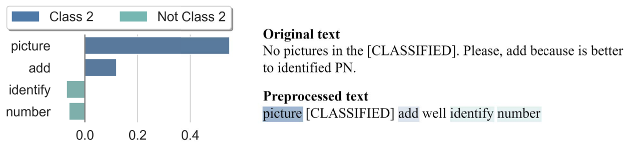
D. QUALITATIVE ANALYSIS

The previous results show the challenging task of classifying an issue into a template to aid the business team in their daily duties. Nonetheless, the combination TF-IDF with four metadata + RF classifier reached a significant result. Further qualitative analysis of the best solution is illustrated in Figure 7. For these evaluations, we use the LIME (Local Interpretable Model-agnostic Explanations) [8] package, which enables explaining individual predictions by ML classifiers. Since the data is private and contain sensitive information, the examples only show words that do not have sensitive content; the words with sensitive content were replaced with the "[CLASSIFIED]", "[DOT]", and "[COD]" tags. Then, in Figure 7, we present the results using LIME, the original, and the preprocessed texts from each example.

In Figures 7a and 7b we can observe two examples in which the template (class "2") is related to "pictures". As we can see in Figures 7a and 7b, the most relevant words according to the trained model are indeed related to pictures (image, photo). Thus, in addition to correctly classifying both examples, the recommendation can also be readily explained.

Figure 7c presents another example of an issue belonging to the class template "3", which is correctly classified. In this case, we can observe that the relevant technical abbreviations ("[DOT]" and "[COD]") were adequately identified by the solution.

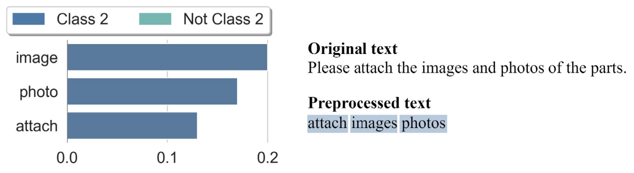
Figure 7d shows an example of a misclassification. We can note that the relevant words are the same as the example presented in Figure 7c, which yields a predicted template



Original text
No pictures in the [CLASSIFIED]. Please, add because is better to identified PN.

Preprocessed text
picture [CLASSIFIED] add well identify number

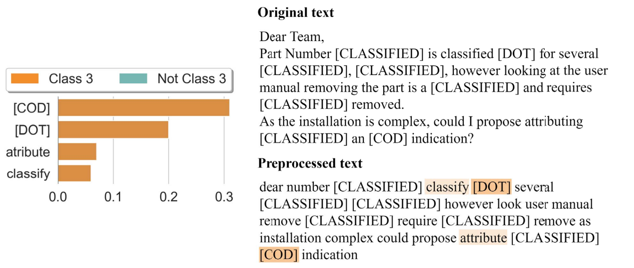
(a) An example of an issue correctly classified in its template (class "2"). Class "2" has "pictures" as the main topic.



Original text
Please attach the images and photos of the parts.

Preprocessed text
attach images photos

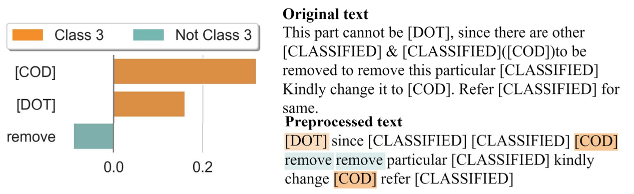
(b) Another example of an issue correctly classified in its template (class "2"). Class "2" has "pictures" as the main topic.



Original text
Dear Team,
Part Number [CLASSIFIED] is classified [DOT] for several [CLASSIFIED], [CLASSIFIED], however looking at the user manual removing the part is a [CLASSIFIED] and requires [CLASSIFIED] removed.
As the installation is complex, could I propose attributing [CLASSIFIED] an [COD] indication?

Preprocessed text
dear number [CLASSIFIED] classify [DOT] several [CLASSIFIED] [CLASSIFIED] however look user manual remove [CLASSIFIED] require [CLASSIFIED] remove as installation complex could propose attribute [CLASSIFIED] [COD] indication

(c) An example of an issue correctly classified in its template (class "3").



Original text
This part cannot be [DOT], since there are other [CLASSIFIED] & [CLASSIFIED]([COD])to be removed to remove this particular [CLASSIFIED] Kindly change it to [COD]. Refer [CLASSIFIED] for same.

Preprocessed text
[DOT] since [CLASSIFIED] [CLASSIFIED] [COD] remove remove particular [CLASSIFIED] kindly change [COD] refer [CLASSIFIED]

(d) An example of an issue wrongly classified in its template. The predicted template was class "3", but the correct one is class "9" (no template).

FIGURE 7. **Examples of issue classification explained using the LIME package. The raw issue texts and their preprocessed versions are also presented. The bar plots indicate the tokens that contribute the most, both positively and negatively, to the final template prediction.**

equal for both cases (class "3"). However, for the example of Figure 7d, it should be predicted as an issue that does not have a standard resolution (class "9", no template). Thus, although there are issues with similar words and contexts, their resolutions may vary, which increases the difficulty of the classification step.

E. LIMITATIONS

The small quantity of labeled examples available in the experimental setting constitutes a challenge. Thus, it might be possible that a larger dataset might alter the ranks presented in Figure 6. Furthermore, in the proposed methodology, given new system issue types, corresponding to new templates, more labeled data related to the new classes are required to retrain the models. In practice, the new data could be acquired by running the standard support system for some time and storing the business team responses.

VI. CONCLUSION

In this work, we have developed and evaluated an NLP solution to aid the resolution of technical issues reported in a text-based asynchronous tool. The proposed pipeline starts from preprocessing raw textual data and vectorizing the resulting tokens. A machine learning classifier was then used to recommend a predefined template to handle simpler and repetitive issues or tag a "none" template, which requires manual resolution.

The proposed framework was evaluated with real data from a system in operation within a large company. The results indicate that the combination of a TF-IDF encoding, a few additional metadata information, and a Random Forest classifier obtained the best results, with 72.7% average accuracy and 69.2% average weighted F1-score.

The best solution was also evaluated qualitatively by means of the LIME package. It was possible to verify that the final model correctly identified business-related words more suitable for the classification task.

Further work involves enhancing the solution with additional metadata, leveraging unlabelled issues using semi-supervised learning, and building a sequential learning procedure to enable the inclusion and learning of new templates.

REFERENCES

[1] P. Suta, X. Lan, B. Wu, P. Mongkolnam, and J. H. Chan, "An overview of machine learning in chatbots," *Int. J. Mech. Eng. Robot. Res.*, pp. 502–510, 2020.

[2] H. M. H. Hobson Lane and C. Howard, *Natural Language Processing in Action: Understanding, Analyzing, and Generating Text With Python*. Shelter Island, NY, USA: Manning Publications, 2019.

[3] S. Anakal and P. Sandhya, "Clinical decision support system for chronic obstructive pulmonary disease using machine learning techniques," in *Proc. Int. Conf. Electr., Electron., Commun., Comput., Optim. Techn. (ICEECCOT)*, Dec. 2017, pp. 1–5.

[4] M. Michel, D. Djurica, and J. Mendling, "Identification of decision rules from legislative documents using machine learning and natural language processing," in *Proc. Annu. Hawaii Int. Conf. Syst. Sci.*, 2022, pp. 1–10.

[5] M. Y. Noguti, E. Vellasques, and L. S. Oliveira, "Legal document classification: An application to law area prediction of petitions to public prosecution service," in *Proc. Int. Joint Conf. Neural Netw. (IJCNN)*, Jul. 2020, pp. 1–8.

[6] I. T. Blog. (Jun. 2021). *SSW Pragmatic Solutions Law Firm Unveils Aquila—Artificial Intelligence for Automatic Analysis of Documents*. Accessed: Jan. 25, 2022. [Online]. Available: https://www.ibm.com/blogs/southeast-europe/ssw-pragmatic-solutions-law-firm- unveils-aquila-artificial-intelligence-for-automatic-analysis-of-documents/

[7] A. Borg, M. Boldt, O. Rosander, and J. Ahlstrand, "E-mail classification with machine learning and word embeddings for improved customer support," *Neural Comput. Appl.*, vol. 33, no. 6, pp. 1881–1902, Mar. 2021.

[8] M. T. Ribeiro, S. Singh, and C. Guestrin, "'Why should i trust you?': Explaining the predictions of any classifier," in *Proc. 22nd ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, San Francisco, CA, USA, Aug. 2016, pp. 1135–1144.

[9] B. Hartpence and A. Kwasinski, "CNN and MLP neural network ensembles for packet classification and adversary defense," *Intell. Converged Netw.*, vol. 2, no. 1, pp. 66–82, Mar. 2021.

[10] J. Palmer, V. Sheng, T. Atkison, and B. Chen, "Classification on grade, price, and region with multi-label and multi-target methods in wineinformatics," *Big Data Mining Anal.*, vol. 3, no. 1, pp. 1–12, Mar. 2019.

[11] A. Barbosa and A. Godoy, "Augmenting customer support with an NLP-based receptionist," in *Proc. Anais do 13th Simpósio Brasileiro de Tecnologia da Informação e da Linguagem Humana*. Nashville, TN, USA: SBC, 2021, pp. 133–142.

[12] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," in *Proc. 2019 Conf. North Amer. Chapter Assoc. Comput. Linguistics, Hum. Lang. Technol.*, vol. 1. Minneapolis, MN, USA: Association for Computational Linguistics, Jun. 2019, pp. 4171–4186. [Online]. Available: https://aclanthology.org/N19-1423

[13] M. Hardalov, I. Koychev, and P. Nakov, "Towards automated customer support," in *Proc. Int. Conf. Artif. Intell., Methodol., Syst., Appl.* Cham, Switzerland: Springer, 2018, pp. 48–59.

[14] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," in *Proc. Adv. Neural Inf. Process. Syst.*, 2017, pp. 5998–6008.

[15] Y. Liu, Y. Wan, and X. Su, "Identifying individual expectations in service recovery through natural language processing and machine learning," *Exp. Syst. Appl.*, vol. 131, pp. 288–298, Oct. 2019.

[16] Y. Park and S. C. Gates, "Towards real-time measurement of customer satisfaction using automatically generated call transcripts," in *Proc. 18th ACM Conf. Inf. Knowl. Manage. (CIKM)*, 2009, pp. 1387–1396.

[17] J. Hoffimann, Y. Mao, A. Wesley, and A. Taylor, "Sequence mining and pattern analysis in drilling reports with deep natural language processing," in *Proc. SPE Annu. Tech. Conf. Exhib.*, 2018, pp. 1–7.

[18] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient estimation of word representations in vector space," 2013, *arXiv:1301.3781*.

[19] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean, "Distributed representations of words and phrases and their compositionality," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 26, 2013, pp. 1–9.

[20] Q. Le and T. Mikolov, "Distributed representations of sentences and documents," in *Proc. Int. Conf. Mach. Learn.*, 2014, pp. 1188–1196.

[21] J. Pennington, R. Socher, and C. Manning, "GloVe: Global vectors for word representation," in *Proc. Conf. Empirical Methods Natural Lang. Process. (EMNLP)*, 2014, pp. 1532–1543.

[22] A. Joulin, E. Grave, P. Bojanowski, and T. Mikolov, "Bag of tricks for efficient text classification," in *Proc. 15th Conf. Eur. Chapter Assoc. Comput. Linguistics*, vol. 2, 2017, pp. 427–431.

[23] A. Rajaraman and J. D. Ullman, *Mining of Massive Datasets*. Cambridge, U.K.: Cambridge Univ. Press, 2011.

[24] V. Vapnik, *The Nature of Statistical Learning Theory*. New York, NY, USA: Springer, 1999.

[25] L. Breiman, "Random forests," *Mach. Learn.*, vol. 45, no. 1, pp. 5–32, 2001.

[26] S. Haykin, *Neural Networks and Learning Machines*, vol. 3. Upper Saddle River, NJ, USA: Pearson, 2009.

[27] S. Theodoridis and K. Koutroumbas, *Pattern Recognition*, 4th ed. New York, NY, USA: Academic, 2008.

[28] T. Chen and C. Guestrin, "XGBoost: A scalable tree boosting system," in *Proc. 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2016, pp. 785–794.

[29] K. Fukunaga and P. Narendra, "A branch and bound algorithm for computing k-nearest neighbors," *IEEE Trans. Comput.*, vol. C-24, no. 7, pp. 750–753, Jul. 1975.

[30] L. Cabral, J. Monteiro, J. F. Da Silva, C. Mattos, and P. Mourão, "FakeWhastApp.BR: NLP and machine learning techniques for misinformation detection in Brazilian Portuguese WhatsApp messages," in *Proc. 23rd Int. Conf. Enterprise Inf. Syst.*, 2021, pp. 63–74.

[31] X. Deng, Y. Li, J. Weng, and J. Zhang, "Feature selection for text classification: A review," *Multimedia Tools Appl.*, vol. 78, no. 3, pp. 3797–3816, Feb. 2019.

[32] P. Meesad, "Thai fake news detection based on information retrieval, natural language processing and machine learning," *Social Netw. Comput. Sci.*, vol. 2, no. 6, pp. 1–17, Nov. 2021.

[33] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, "Scikit-learn: Machine learning in Python," *J. Mach. Learn. Res.*, vol. 12 no. 10, pp. 2825–2830, 2011.

[34] R. Řehůřek and P. Sojka, "Software framework for topic modelling with large corpora," in *Proc. LREC Workshop New Challenges NLP Frameworks*. Valletta, Malta: ELRA, May 2010, pp. 45–50.

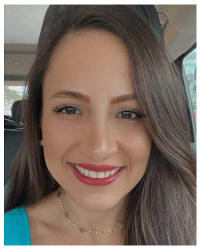
[35] S. Bird, E. Klein, and E. Loper, *Natural Language Processing With Python: Analyzing Text With the Natural Language Toolkit*, Sebastopol, CA, USA: O'Reilly, 2009.

[36] M. Honnibal and I. Montani, "spaCy 2: Natural language understanding with Bloom embeddings, convolutional neural networks and incremental parsing," Zenodo, 2020. [Online]. Available: https://zenodo.org/record/4317367.Y0QT19LMJk

ELENE FIRMEZA OHATA received the B.S. degree in mechatronics engineering from the Federal Institute of Ceará (IFCE), Fortaleza, Ceará, Brazil, and the M.S. degree in teleinformatics engineering from the Federal University of Ceará (UFC), in 2021, where she is currently pursuing the Ph.D. degree in teleinformatics engineering. Her research interests include NLP, machine learning, digital image processing, and computer vision. She received the sandwich period taken in the Algonquin College (Ottawa/Canada) through the Science Without Borders Program for the B.S. degree.

CÉSAR LINCOLN CAVALCANTE MATTOS is currently an Associate Professor at the Department of Computer Science, Federal University of Ceará (UFC), Brazil. He is also an Associate Researcher at the Logics and Artificial Intelligence Group (LOGIA). His research interests include the broad fields of machine learning and probabilistic modeling, such as Gaussian processes, deep (probabilistic) learning, approximate inference, and system identification. He has been applying learning methods in several research and development collaborations in areas, such as dynamical system modeling, health risk analysis, software repository mining, and anomaly detection.

SAMUEL LUZ GOMES received the B.S. degree in industrial mechatronics and the M.S. degree in renewable energy from the Instituto Federal de Educação, Ciência e Tecnologia do Ceará (IFCE), Fortaleza, Ceará, Brazil, in 2015 and 2017, respectively. He is currently pursuing the Ph.D. degree in electrical engineering at UFC. His research interests include signal and digital image processing, computer vision, and machine learning.



ELIZÂNGELA DE SOUZA REBOUÇAS received the master's degree in telecomunication engineering from the Federal Institute of Ceará (IFCE), Brazil, in 2017. She is currently a Professor at the Federal Institute of Science and Technology, Caninde, Ceará, Brazil. Her current research interest includes applications in computational vision.

PAULO ANTONIO LEAL REGO (Member, IEEE) is currently pursuing the Ph.D. degree in computer science from the Federal University of Ceará (UFC). He is also an Adjunct Professor at the Computing Department, UFC. He is also working in the area of computer science, with emphasis on computer networks and distributed systems. His research interests include mobile and cloud computing, vehicular *ad-hoc* networks, and edge and fog computing.