



# COMPARISON OF INFORMATION RETRIEVAL TECHNIQUES APPLIED TO IT SUPPORT TICKETS

Leonardo Santiago Benitez Pereira , *Student Member, IEEE*, Robinson Pizzio , *Member, IEEE* and Samir Bonho

**Abstract**—Institutions dependent on IT services and resources acknowledge the crucial significance of an IT help desk system, that act as a centralized hub connecting IT staff and users for service requests. Employing various Machine Learning models, these IT help desk systems allow access to corrective actions used in the past, but each model has different performance when applied to different datasets. This work compares eleven Information Retrieval techniques in a dataset of IT support tickets, with the goal of implementing a software that facilitates the work of Information Technology support analysts. The best results were obtained with the Sentence-BERT technique, in its multi-language variation distilluse-base-multilingual-cased-v1, where 78.7% of the recommendations made by the model were considered relevant. TF-IDF (69.0%), Word2vec(68.7%) and LDA (66.3%) techniques also had consistent results. Furthermore, the used datasets and essential parts of coding have been published and made open source. It also demonstrated the practicality of a support ticket recovery system by implementing a minimal viable prototype, and described in detail the implementation of the system. Finally, this work proposed a novel metric for comparing the techniques, whose aim is to closely reflect the perception of the IT analysts about the retrieval quality.

**Index Terms**—Information Retrieval, Machine Learning, Natural Language Processing, Support Tickets, Information Technologies.

## I. INTRODUCTION

Information Technologies (IT) has already become part of people's daily lives, who are increasingly dependent on it for educational, social, economic, or professional purposes [1]. To some extent, it is expected that technological resources "just work" and are available all the time, so that unavailability or problems with these technological resources end up disrupting the routine.

Within the business/institutional scope, so-called "IT support teams" are responsible for the proper functioning of IT resources [2]. When a user faces a problem with IT resources, they describe their problem by opening a support ticket in a management system. Then, an IT Analyst is responsible for solving the problem, communicating with the user, and recording in the management system what actions were taken to solve the problem [3].

The knowledge accumulated in these databases is a valuable asset for companies, as it can be used to improve the use of

digital technologies within the company. However, searching for this information in the management system databases is technically complicated and time-consuming [4]. Furthermore, in [5], the authors point out that IT support teams also face problems such as ticket overload, team turnover, use of inadequate and/or outdated technological tools, lack of qualified labor, among others, further complicating the use of these databases.

Many tickets have identical resolutions, so the analyst only needs to identify if a similar problem has already been solved in the past [6]. For this, the analyst can consult colleagues, read conversation histories, search directly in the database of already resolved tickets, among others. This process can consume a lot of the analyst's time, delaying problem resolution and harming the user.

An area of knowledge that can facilitate the use of these databases is Information Retrieval (IR), whose objective is to find documents of an unstructured nature (usually text) that meet an information need, from a large collection of materials, [7]. Such techniques allow searching for support tickets in the database, making it easier for the IT analyst to find the necessary information to solve a new ticket, which [6] argues can save the analyst a lot of effort and, thus, considerably improve the service provided by IT teams.

The project was carried out at the company Skylink, using a proprietary database where support tickets are described, and the solutions applied to these tickets are indicated. The eleven chosen IR techniques were applied to Skylink's database, aiming to define the best IR technique for a scenario where, given a new ticket, the system identifies, among the possibilities available in the database, the similar tickets previously solved to facilitate the work of the IT analyst. It was identified that the Sentence-BERT technique, in its multi-language variation distilluse-base-multilingual-cased-v1, obtained the best performance among the eleven with 78.7% relevant recommendations. In this context, the contributions of this research are:

- 1) Compare eleven information retrieval techniques specifically in the context of support tickets, a larger number than any other recent work in the same area;
- 2) make the dataset used and the code developed available for free;
- 3) propose a new metric to compare information retrieval techniques, which closely reflects IT analysts' perception of retrieval quality.

Leonardo Santiago Benitez Pereira is with Skylink, Lithuania. e-mail:lsbenitezpereira@gmail.com

Robinson Pizzio and Samir Bonho are with the Electronics Department at Federal Institute of Santa Catarina, Brazil. e-mail:robinson.pizzio@ifsc.edu.br samir.bonho@ifsc.edu.br

This article is organized as follows: Section II presents a literature review in which the natural language processing (NLP) models used and related works are discussed. Details of the methods employed, involving the dataset, the IR algorithms, and the identification of the best technique are presented in Section III. Results and final considerations are presented in Sections IV and V, respectively.

## II. LITERATURE REVIEW

For a better understanding of the evaluated methods, a brief description of the NLP models used in this work will be given. Then, various recent works using IR techniques in the domain of IT support tickets are compared, which use a wide variety of methodologies and techniques.

### A. Techniques Used

The Latent Dirichlet Allocation (LDA) model is based on a probabilistic approach that assumes each document in a set of documents is a mixture of a fixed number of topics, and each topic is a mixture of words [8].

In the Term Frequency - Inverse Document Frequency (TF-IDF) model, the frequency of a term in the document is weighted by its occurrence in other texts in the set. The value of this weight is high when the term occurs frequently in the document in question and infrequently in the other documents in the set [9]. Similarly, the Best Match 25 (BM25) technique also considers the frequency of keywords in the document but penalizes very long documents [10].

The Word2Vec model [11] is a representation based on artificial neural networks (ANN) that relies on the premise that similar words have similar contexts, known as distributional similarity. One of its extensions, Doc2Vec, allows representing not only isolated words but entire documents in vector form [12].

The Bidirectional Encoder Representations from Transformers (BERT) [13] model is pre-trained on an extensive dataset and was pioneering in introducing the concept of bidirectional training, i.e., it takes into account the entire set of words in a sentence simultaneously.

The Sentence-BERT is an extension of the BERT model, which focuses on producing vector representations for sentences or expressions, as opposed to representing words or individual units [14]. The idea of this model is to capture the semantics and context of sentences, allowing the evaluation of semantic similarity between them. For this, a siamese neural network with a pooling operation at the output is used. Additionally, the model is trained on a dataset composed of pairs of sentences annotated with the relationship between them, so that it is "forced to learn" how to properly represent the two sentences. Both BERT and Sentence-BERT can handle new words (outside the training dictionary), unlike the other techniques compared in this work.

### B. Related Work

The work of [6] used pre-processed text from the ticket description with techniques such as lemmatization and stop word

removal, then applied TF-IDF and dimensionality reduction techniques to obtain a vector representation of the ticket, and finally compared the vectors using Cosine Similarity. The data annotation for training was performed with an existing system. The evaluation of the models was done by comparison with the existing system, using both the textual comparison method SS-Evaluator and a manual analysis by a support analyst.

In [15], embeddings from the BERT network and its derivatives (specifically: RoBERTa, DistilBERT, and DistilRoBERTa) were used to enable semantic search of tickets. Additionally, supervised models were used to classify the group/department of the company that should be responsible for the ticket and also the analyst who should handle the ticket. For all tasks, the top-k accuracy metric was used for evaluation.

The High Performance Computing Systems Research Center of the Los Alamos National Laboratory published in [16] their methods for automatically classifying tickets and suggesting similar tickets. They used 70,000 tickets, whose text was pre-processed (stop word removal, conversion to lowercase, among others) and then vectorized (using 3 different techniques: Latent Dirichlet Allocation, Latent Semantic Analysis, and Doc2Vec). The system was initially evaluated by comparison with two existing systems (the "more like this" functionality of the Elasticsearch software and an expert system that compares the percentage of common words); thus, 200 tickets were used for a qualitative manual evaluation by an expert, and no quantitative evaluation was performed.

When there is no prior system for comparison (e.g., [6] and [16]), it is common to use a small evaluation set. In [17], only 5 tickets were used, but the recommendations were evaluated in two different dimensions: whether the recommendations belonged to a similar area/category (examples of categories in this work are "video editing" and "graphical interface") and whether the recommendations had the same functional characteristic (in this work, functions such as "start", "save", among others were used). Based on this evaluation methodology, 6 vectorization techniques were compared (two variations of Word2Vec, two variations of Doc2Vec, TF-IDF, and BERT, using only non-retrained models), using the metrics of total score and average score in each of the two dimensions, where Doc2Vec and BERT obtained the best results.

Beyond the retrieval of similar tickets, there are various applications of Machine Learning and related areas to facilitate the resolution of support tickets. In the work of [2], for example, 1585 tickets were used to train a model that classified the ticket into one of 13 categories. The concatenation of the title, description, and comments fields was used as input; additionally, pre-processing with stemming and TF-IDF was used to represent the tickets, and 4 models were tested: a rule-based system, J48, Naive Bayes, and Sequential Minimal Optimization (SMO). In [18], also conducted at the company Skaylink, a model is presented that classifies the ticket into 7 categories (using a fully connected 6-layer ANN), but no information retrieval techniques were applied. Beyond simple classification, the work of [19] performs multi-level classification, while [20] performs multi-label classification, assigning one or more categories out of 10 possibilities; both works use

a Bert neural network.

Several companies have also developed solutions for their internal ticket management systems, such as Uber, which in [21] presents a system that classifies tickets to define response templates that the analyst can use. The resulting models were evaluated with real users and reduced the ticket resolution time by 10%.

### III. METHODS

In this section, we present the methodological details used in the development of this article. Details about the dataset used, the IR algorithms compared, the definition of metrics, among others, are described in the following subsections. Details about the implementation of each technique, as well as the identification of the best among the selected ones, are also described in this section.

#### A. Dataset

The dataset used consists of information from 20,356 support tickets submitted between the years 2017 and 2022, recorded during the provision of services for a Skylink client company. The data was anonymized before the realization of this work, so no personal information was present.

Each ticket is described by nine variables: external\_ID (ticket identification in the management system), title (ticket title, as informed by the user who opened the incident), description (ticket description, also informed by the user), category (incident category), date\_open (ticket opening date), date\_close (ticket closing date), location (office the user belongs to), solution (solution applied to the ticket), and analysts (group of analysts responsible for resolving the ticket). Upon receiving the ticket, the analyst does not receive the variables category, date\_close, and solution, which are added to the records only after the ticket is closed.

For the purposes of this work, the title and description fields were concatenated and all other fields were discarded, so the system uses only the information provided by the user at the time of incident opening. The top 15 categories present in the dataset, ordered by the number of incidents, are as follows: Fileservices, Active Directory, Computer Services, Access Control, End-Of-Life, O365, Exchange, Create Account, Data Center, Identity Management, Fileshared, Telecom, Printer, Software general, and Security.

The tickets are written primarily in English (207 out of the 300 selected tickets); however, many are written in Portuguese (51 tickets), German (22 tickets), Spanish (19 tickets), French (1 ticket), and potentially the system in operation will receive tickets in other languages. It is noticeable that the dataset is highly unbalanced, and many tickets contain grammatical errors and abbreviations. The concatenation of title and description totals, on average, 224 characters. For illustration purposes, Table I presents a typical ticket (the external\_ID field, names, and dates have been used fictitiously to preserve the identity of the original user).

The process of identifying similar tickets is also called data labeling [22]; in this work, 300 tickets were labeled, chosen by the following methodology: from each of the 10 most

TABLE I: Example of a typical ticket.

Column	Value
external_ID	ABC123456
title	File Access
description	Good morning. I need access to Leonardo Benitez's computer. He has been dismissed from the company and I need the files left on his desktop. These are control spreadsheets and also emails. He has signed the authorization letter. Thank you
category	Fileservice
date_open	2022-01-01 10:23:19.000
date_close	2022-01-02 09:15:56.000
location	BRLM
solution	The client gained access and copied the files to their machine
analysts	Leonardo Pereira

frequent categories, 30 "representative" tickets were manually chosen, i.e., those that describe a single problem completely. The 300 tickets were randomly divided into 3 subgroups of 100 tickets, to facilitate labeling. For each ticket, the five most similar tickets (within the subgroup) were manually indicated. The decision to indicate exactly 5 similar tickets was made to simplify and make predictable the way the support analyst interacts with the system: many similar tickets are not needed to help solve a new problem, while at least some approximately similar tickets are more useful than none.

To simplify labeling, the graphical tool Miro [23] was used, placing the text of each ticket on a card arranged in a two-dimensional plane (Fig. 1), so similar tickets are positioned close to each other. The aforementioned procedure was carried out with 3 analysts labeling the data independently, each responsible for a subgroup, to reduce labeling bias.

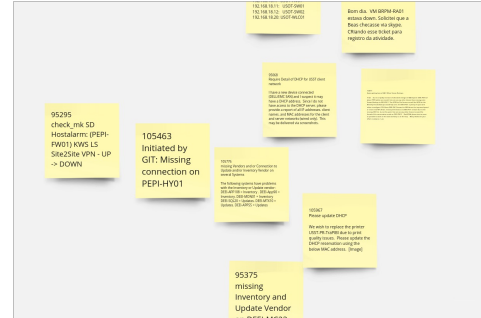


Fig. 1: Tool used for labeling.

The set of 300 tickets used is available at [24]. Before the data was published, all personal and/or sensitive information was removed (replaced by a tag indicating the original content, for example: the phrase "this text was written by Leonardo" is converted to "this text was written by [NAME]"). The removal was performed in three steps: first, the Machine Learning-based tool AWS Comprehend PII Removal was used; then, a sequence of custom regular expressions was applied; finally, all tickets were manually verified.

#### B. Chosen Techniques

From the study of IR techniques available to meet the objectives, it was chosen to compare several techniques that

are representative of the main existing approaches to the problem:

- representing traditional approaches that are highly dependent on the developer's domain knowledge, an Expert System was developed;
- representing well-established statistical approaches in the IR field, TF-IDF was used;
- representing well-established probabilistic approaches in the IR field, BM25 and LDA were used;
- representing approaches of neural networks with non-contextual embeddings, Word2vec trained on an English language dataset, Word2vec entirely trained with the Skaylink database, and its contextual variation Doc2vec also trained with the Skaylink database were used;
- representing approaches of neural networks with contextual embeddings, BERT trained with multi-language data, Sentence-BERT trained with multi-language data, Sentence-BERT trained with English language data only, and Sentence-BERT initially trained with multi-language data and then retrained with the Skaylink database were used.

Additionally, the technique of random selection - in which calls are selected randomly - was chosen to facilitate the interpretation of the obtained values.

### C. Implementation of Techniques

All techniques were implemented using the Python programming language. The vector generated by each technique was compared using the Cosine Similarity metric, defined as Eq. (1). The only exceptions were the expert system and BM25, which used their own similarity metrics. The system then recommends the calls with the highest similarity among the possible recommendations.

$$\text{Cosine Similarity}(A, B) = \frac{\mathbf{A} \cdot \mathbf{B}}{\|\mathbf{A}\| \|\mathbf{B}\|} \quad (1)$$

All techniques were implemented following an object-oriented programming structure, so they exposed the same generic interface. Thus, they were evaluated following the same implementation of metrics and methodologies.

1) *Expert System*: A system was developed that, based on the presence or absence of certain terms, generated a set of "labels" for each document. 116 IT jargon terms were used as terms, complemented with 141 synonyms. To increase the number of labels identified for each document, the text was preprocessed as follows:

- converting to lowercase;
- removing special characters —, ., ,, !, ?, \_ and \*;
- converting characters to their closest representation in unicode.

After preprocessing, the set of resulting labels was compared using the Jaccard Similarity metric, presented in Eq. (2).

$$\text{Jaccard Similarity}(A, B) = \frac{|A \cap B|}{|A \cup B|} \quad (2)$$

2) *TF-IDF*: For the TF-IDF technique, the implementation of the Sklearn library (in its version 0.23.2) was used. The dictionary was calculated based on the 20056 calls that were not used for evaluation (i.e., out of the total 20356 calls, 20056 were used to calculate the dictionary and 300 for evaluating the model). The same preprocessing as the expert system was used, along with the removal of stop words for the English language. The top 500 most frequent words from the dictionary were used, resulting in a 500-dimensional vector.

3) *BM25*: The implementation of the rank\_bm25 library (version 0.2.2) was adopted. The same preprocessing as the expert system was used, and the default parameters of  $k1 = 1.5$ ,  $b = 0.75$ , and  $\epsilon = 0.25$  were kept. Since the BM25 technique requires its own comparison metric, the implementation of the library was also used.

4) *LDA*: To implement the LDA technique, the Gensim library (version 4.3.0) was chosen. The same preprocessing as the expert system was used, and the technique was configured for 300 topics. This dimension was chosen because it is the same value as the vectors from Word2vec, making the comparison fairer, besides being the value used in the work of [6] and being a usual value in the literature.

5) *Word2vec and derivatives*: The implementations from the Gensim library were adopted. For the Word2vec technique, a model trained on the Google News dataset, which contains approximately 100 billion words from the English language, was used. This model produces a 300-dimensional vector. A new Word2vec network was also trained using only the 20056 calls that were not used for evaluation (which totaled approximately 1.6 million words), also with 300 dimensions. The same preprocessing as the expert system was maintained.

Doc2vec was trained with the same 20056 calls, techniques for preprocessing data, and output vector dimensions. Training was performed with parameters of  $window = 10$ ,  $min\_count = 1$ , and  $epochs = 100$ , obtained after briefly testing various values and measuring the accuracy obtained. A pre-trained model was not used because there was no available model in the Gensim library, and another reliable source was not found to obtain a model.

6) *BERT and derivatives*: For the original BERT model, the BERT-base-multilingual-cased implementation from the HuggingFace library (version 4.25.1) was adopted, which was trained with 104 languages, taking the [CLS] special token output as the embedding, which has a dimension of 768 elements. The SentenceTransformers library was used for Sentence-BERT models, version 2.2.2. The multi-language model used was distiluse-base-multilingual-cased-v1, which was trained with 15 languages (including English, German, Spanish, and Portuguese), providing a 512-dimensional vector. When retraining the network with the Skaylink data, the same distiluse-base-multilingual-cased-v1 model was used as the base, and it was trained for 3 epochs with the 20056 calls not used for evaluation. The English language model chosen was all-mpnet-base-v2, which provides a 768-dimensional vector.

The BERT model and its variations tend to yield better results when the text is not preprocessed [25]; therefore, the original text was used.

#### D. Identification of the Best Technique

The similarity between the calls was calculated based on the positions of the cards, indicated during the data labeling, which were exported to CSV files (one file for each subgroup). For each model, these CSV files were loaded one by one, and the model under evaluation gave 5 recommendations for each of the 100 calls from each subset. The evaluation was performed using the precision metric, presented in Eq. (3). To measure precision, the 5 calls manually indicated during labeling were considered as "relevant documents," that is, the 5 calls closest in the two-dimensional plane.

$$\text{Precision} = \frac{\text{Number of relevant recommended items}}{\text{Total number of recommended items}} \quad (3)$$

After that, the average precision [7] among the 3 labeled sets was calculated, and the best technique was chosen as the one with the highest average precision. The recall metric [7] was not used because, as it was decided that each technique would retrieve five calls and there were also always five calls considered relevant, recall and precision inherently have the same value. Ranked metrics such as Recall@k [7] were not used because - for this specific IT support case - they do not adequately represent the utility of the system to the support analyst: since each call is briefly described, the analyst can scan the 5 calls for any useful information within a few seconds, making the order of documents less relevant.

Additionally, the "at least one accuracy" metric was created (formalized in Eq. (4)), where  $N$  is the number of calls,  $y_i$  is the set of relevant calls, and  $\hat{y}_i$  is the set of recommended calls), with the goal of having a metric that better reflects the analyst's perception of the system quality. This metric considers a "hit" if any of the five retrieved calls is indeed a relevant call. For example, if given a support call about "broken mouse," the system retrieves the previous calls "lost my mouse," "mouse not working," "can't log in," "air conditioner too cold," and "password recovery," such retrieval is considered a hit because at least the second call is relevant.

$$\text{Accuracy}_{\text{at least one}}(y, \hat{y}) = \frac{\sum_i \lambda(y_i, \hat{y}_i)}{N} \quad (4)$$

where

$$\lambda(a, b) = \begin{cases} 1, & |a \cap b| > 0 \\ 0, & \text{otherwise} \end{cases} \quad (5)$$

#### E. Prototype

Using the technique identified earlier as the most suitable one, a prototype software was implemented. The system architecture followed the proposal from [26, pg. 907] and was implemented as shown in Fig. 2. This architecture has two distinct flows: document collection and processing of new queries or documents.

During document collection, all 20356 calls from the database are registered, and their respective texts are vectorized. Once vectorized, these documents are stored in a database and thus become available for future queries.

During the processing of new documents, their vector representation is calculated; then, the 100 most recent registered calls are searched, from which the 5 most similar calls are selected and presented to the analyst through a graphical

interface. This restriction to the 100 most recent calls is for practical reasons, to avoid system overload, but a future implementation of the system may allow searching the entire database. Additionally, the new call is also stored in the system's database, allowing its use in future recommendations. The prototype provides a button to allow the analyst to give feedback, indicating whether the recommendations were helpful or not, and it was used in Skaylink's daily work.

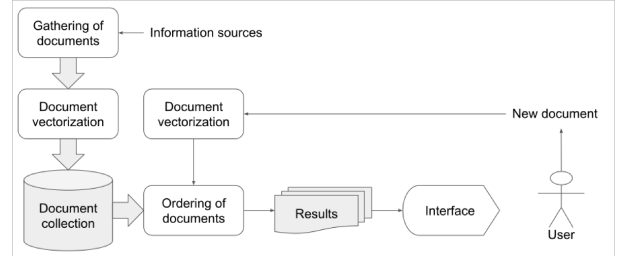


Fig. 2: Prototype architecture.

#### IV. RESULTS

The results were summarized in Table II. The time required to perform 100 recommendations (1 subgroup) was also included, calculated by the average of 3 executions (one in each subgroup), including the time to load the model from disk and not including the possible training time when executed on an Acer Aspire notebook. It can be observed that the multi-language Sentence-BERT technique presents the best result, with 35.1% precision and 78.7% at least-one accuracy (meaning, three out of four times the system recommends at least one previous call similar to the call being analyzed). Sentence-BERT is also the most recent technique, published in 2019 [14], so it was expected to have the best results.

Two techniques surprised by their low precision: multi-language BERT (17.2%) and Doc2vec (5.8%). Although both obtain better results than random selection (5.5%), the literature indicates that the use of these techniques usually results in precisions comparable to classical techniques like TF-IDF (29.6%). For Doc2vec, a possible explanation is that the training resulted in overfitting to the dataset, given the low number of calls used, 20056, and because it was not based on a previously trained model (i.e., training was done "from scratch"). Regarding multi-language BERT, a possible explanation was the choice of the [CLS] special token as the embedding, as there are other possible methods to extract embeddings from a BERT network.

Regarding the effect obtained by retraining the ANNs (from an existing base model), there were divergent results. Although retraining Word2vec improved its precision, retraining multi-language Sentence-BERT did not achieve the same result, and both had similar performance. This fact can be explained because BERT networks need a large volume of data to be trained, and the use of small datasets can even cause the network to "forget" part of what it learned previously.

#### V. FINAL REMARKS

Throughout this work, a comparison of eleven Information Retrieval techniques was carried out, applied to a dataset

TABLE II: Comparison of implemented techniques.

Name	Accuracy <sub>alo</sub>	Precision	Time(ms)
BM25	59.0%	23.7%	258
BERT multi-language	50.0%	17.2%	12781
Doc2vec	27.3%	5.8%	933
LDA	66.3%	20.9%	833
Random selection	26.0%	5.5%	199
Sentence-BERT English	74.3%	30.1%	10601
Sentence-BERT multi-language	78.7%	35.1%	6411
Sentence-BERT retrained	78.7%	32.7%	6450
Expert system	42.7%	17.2%	1101
TF-IDF	69.0%	29.7%	672
Word2vec English	58.3%	23.4%	49298
Word2vec retrained	68.7%	26.2%	49590

referring to IT support calls. These techniques included various approaches to IR, making it possible to clearly identify the possibilities to implement a system that, given a new support call, is capable of retrieving similar support calls.

The best result was obtained with the Sentence-BERT technique, in its multi-language variation distiluse-base-multilingual-cased-v1, where 78.7% of the recommendations made by the model were considered relevant. The two other variations tested from Sentence-BERT presented the second and third best results, followed by the TF-IDF technique.

Furthermore, this work sought to contribute to the academic community by providing, free and unrestricted, the dataset used and the implementation of the techniques, as well as proposing a new metric for evaluating IR techniques. These results meet the proposed objectives and guide the development of future work in the field.

When interpreting the results obtained, it may seem at first glance that the evaluation metrics indicate poor results. However, it is important to note that the evaluation methodology was defined strictly, and only 5 out of 99 calls were considered relevant (the 100th call being the one being evaluated). Moreover, the nature of the data itself (short texts, poorly explanatory, and with many technical jargon) makes the application of information retrieval techniques challenging. Nevertheless, all implemented techniques showed better results than random selection, indicating that they were able to capture the semantics of support calls.

As a complementary result, it was confirmed that the Sentence-BERT network presents better results in IR than the original BERT, as presented in the work of [14]. This superiority was maintained in all variations of Sentence-BERT tested and for all evaluation metrics used.

It is worth noting the positive result of the TF-IDF technique, which is simple to implement and computationally fast. The technique also proved to be robust in all experiments conducted, consistently presenting results, while other techniques did not perform equally well under all conditions. Thus, it is possible that the final system will be implemented with TF-IDF instead of Sentence-BERT.

Finally, it is worth mentioning that the expert system obtained considerably poor results. Despite its simplistic implementation, this demonstrates the difficulty of implementing manually crafted IR systems, justifying the use of more advanced Machine Learning techniques.

Based on the results obtained, it is possible to continue the work with, for example, the following topics: not restricting the prototype to search among only the last 100 registered calls; using a database with native support for vector similarity search (such as the ElasticSearch software); Development of variations of the test dataset, identifying under which conditions each of the techniques presents the best result; testing other techniques (such as FastText or GPT3); testing the Sentence-BERT model retrained for more than 3 epochs and on a larger dataset; combining embedding extraction techniques with classification and ranking methods to improve results; integrating the prototype with the Skaylink company's management system, facilitating the use of the system;

#### ACKNOWLEDGMENTS

The authors would like to thank Skaylink company for providing the data used in this study, as well as for their support during the implementation and evaluation process of this work.

#### REFERENCES

- [1] R. Stair and G. Reynolds, *Principles of Information Systems*. Boston, MA, USA: Course Technology Press, 9th ed., 2009.
- [2] F. Al-Hawari and H. Barham, "A machine learning based help desk system for it service management," *Journal of King Saud University - Computer and Information Sciences*, vol. 33, no. 6, pp. 702–718, 2021.
- [3] W. Zhou, L. Tang, C. Zeng, T. Li, L. Shwartz, and G. Grabarnik, "Resolution recommendation for event tickets in service management," *IEEE Transactions on Network and Service Management*, vol. 13, pp. 1–12, 2016.
- [4] F. A. P. Fialho, *Gestão do Conhecimento e aprendizagem*. Visual Books, 2006.
- [5] C. Silva and A. Vasconcelos, "Using the ideal model for the construction of a deployment framework of it service desks at the brazilian federal institutes of education," *Software Quality Journal*, vol. 28, 09 2020.
- [6] D. P. Muni, S. Roy, Y. T. Y. J. L. Chiang, A. J.-M. Viallet, and N. Budhiraja, "Recommending resolutions of itil services tickets using deep neural network," in *Proceedings of the Fourth ACM IKDD Conferences on Data Sciences, CODS '17*, (New York, NY, USA), Association for Computing Machinery, 2017.
- [7] C. D. Manning, P. Raghavan, and H. Schütze, *Introduction to Information Retrieval*. Cambridge, UK: Cambridge University Press, 2008.
- [8] D. M. Blei, A. Y. Ng, and M. I. Jordan, "Latent dirichlet allocation," *The Journal of Machine Learning Research*, vol. 3, p. 993–1022, mar 2003.
- [9] G. Salton and C. Buckley, "Term-weighting approaches in automatic text retrieval," *Information Processing & Management*, vol. 24, no. 5, pp. 513–523, 1988.
- [10] S. Robertson and H. Zaragoza, "The probabilistic relevance framework: Bm25 and beyond," *Found. Trends Inf. Retr.*, vol. 3, p. 333–389, apr 2009.
- [11] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient estimation of word representations in vector space," *CoRR*, vol. abs/1301.3781, 2013.
- [12] Q. Le and T. Mikolov, "Distributed representations of sentences and documents," in *Proceedings of the 31st International Conference on Machine Learning (E. P. Xing and T. Jebara, eds.)*, vol. 32 of *Proceedings of Machine Learning Research*, (Beijing, China), pp. 1188–1196, PMLR, 22–24 Jun 2014.
- [13] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," in *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, (Minneapolis, Minnesota), pp. 4171–4186, Association for Computational Linguistics, June 2019.
- [14] N. Reimers and I. Gurevych, "Sentence-bert: Sentence embeddings using siamese bert-networks," in *EMNLP/IJCNLP (1)* (K. Inui, J. Jiang, V. Ng, and X. Wan, eds.), pp. 3980–3990, Association for Computational Linguistics, 2019.



- [15] L. Feng, J. Senapati, and B. Liu, “Tadaa: real time ticket assignment deep learning auto advisor for customer support, help desk, and issue ticketing systems,” 2022.
- [16] A. DeLucia and E. Moore, “Analyzing hpc support tickets: Experience and recommendations,” 2020.
- [17] F. S. Dyrhovden, E. Norvang, and M. Sund, “Word embeddings for recommending semantically similar support tickets,” bachelor’s thesis, Western Norway University of Applied Sciences, 2021.
- [18] L. S. B. Pereira, R. Pizzio, S. Bonho, L. M. F. de Souza, and A. C. A. Junior, “Machine learning for classification of it support tickets,” 2022 (in press at the proceedings of the 2nd International Conference On Cyber Management And Engineering).
- [19] A. Zangari, M. Marcuzzo, M. Schiavinato, A. Gasparetto, and A. Albarelli, “Ticket automation: An insight into current research with applications to multi-level classification scenarios,” *Expert Systems with Applications*, vol. 225, p. 119984, 2023.
- [20] Z. Liu, C. Bengue, and S. Jiang, “Ticket-bert: Labeling incident management tickets with language models,” 2023.
- [21] P. Molino, H. Zheng, and Y.-C. Wang, “Cota: Improving the speed and accuracy of customer support through ranking and deep networks,” in *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD ’18*, (New York, NY, USA), p. 586–595, Association for Computing Machinery, 2018.
- [22] K. Faceli, A. C. Lorena, J. Gama, and A. C. P. d. L. F. d. Carvalho, *Inteligência artificial: uma abordagem de aprendizado de máquina*. LTC, 2011.
- [23] M. Developers, “Miro.” <https://miro.com>, 2022.
- [24] L. S. B. Pereira, “Semantic similarity of it support tickets.” Dataset on Zenodo, Dec. 2022.
- [25] U. Kamath, K. Graham, and W. Emara, *Transformers for Machine Learning: A Deep Dive*. Chapman & Hall/CRC Machine Learning & Pattern Recognition, CRC Press, 2022.
- [26] Mitkov, ed., *The Oxford handbook of computational linguistics*. Oxford [u.a.]: Oxford Univ. Press, 2003.



**Samir Bonho** received the B.Sc. and the M.Sc. degree in electrical engineering from the UFSC, in 2004 and 2006, respectively. He has experience in Biomedical Engineering, with emphasis on digital signal processing and data transmission over IP networks. He is currently a professor with the Electronics Department at the IFSC, Florianópolis Campus. He is father of Yannis and Isadora.



**Leonardo Santiago Benitez Pereira** received the B.Sc. degree in electronics engineering from the Federal Institute of Santa Catarina (IFSC), in 2022. He has extensive knowledge in software development, has carried out projects in different application areas, always with Machine Learning and data-driven solutions at their core. He currently works at Skaylink as a Machine Learning Engineer.



**Robinson Pizzio** received the B.Sc. and M.Sc. degrees in electrical engineering from the Pontifical Catholic University of Rio Grande do Sul (PUCRS), Brazil, in 1995 and 1998, respectively, and the Ph.D. degree from the Federal University of Santa Catarina (UFSC), Brazil, in 2018. He has worked as an assistant professor at PUCRS and University of Caxias do Sul (UCS). Since 2013 he has been an associate professor at IFSC, and since 2022 is the Director of the IFSC’s Innovation Hub.