IMPROVING AUTOMATED CATEGORIZATION OF CUSTOMER REQUESTS WITH RECENT ADVANCES IN NATURAL LANGUAGE PROCESSING

Filip Koukal[1], František Dařena[1]✉, Roman Ježdík[2], Jan Přichystal[1]

[1] *Mendel University in Brno, Czech Republic*
[2] *ALVAO, s. r. o., Žďár nad Sázavou, Czech Republic*

EUROPEAN JOURNAL OF BUSINESS SCIENCE AND TECHNOLOGY

Volume 10 Issue 2
ISSN 2694-7161
www.ejobsat.com

ABSTRACT

In this paper, we focus on the categorization of tickets in service desk systems. We employ modern neural network-based artificial intelligence methods to improve the performance of current systems and address typical problems in the domain. Special attention is paid to balancing the ticket categories, selecting a suitable representation of text data, and choosing a classification model. Based on experiments with two real-world datasets, we conclude that text preprocessing, balancing the ticket categories, and using the representations of texts based on fine-tuned transformers are crucial for building successful classifiers in this domain. Although we could not directly compare our work to other research the results demonstrate superior performance to similar works.

KEY WORDS

service desk systems, customer requests classification, transformer models, machine learning

JEL CODES

C89, L86

1 INTRODUCTION

Customer support is an activity that a company provides before, during, and after the sale of a service or product. It involves various forms to ensure customer satisfaction at any stage of the life cycle (Menken and Blokdijk, 2009). Similarly, users or employees might require support from IT or other staff to solve various issues related to their work (Al-Hawari and Barham, 2021). The support often has the form of a helpdesk or service desk, which provides a single point of contact with customers or users.

Requests are represented by so-called tickets that capture the whole interaction between a user and operator in the form of a conversation. Within the conversation, the user gradually specifies the problem and the operator responds

KOUKAL, Filip, DAŘENA, František, JEŽDÍK, Roman, and PŘICHYSTAL, Jan. 2024. Improving Automated Categorization of Customer Requests with Recent Advances in Natural Language Processing. *European Journal of Business Science and Technology,* 10 (2): 173–184. ISSN 2694-7161, DOI 10.11118/ejobsat.2024.010.

to the user's messages by asking additional questions, providing solutions, or redirecting the request to the appropriate department. Once a ticket reaches a certain state where the operator has a sufficient amount of information, a solver is assigned to it.

Help desks often have a tiered structure in which the user's first contact with the operator is the most significant for the early identification of the problem and subsequent routing of the ticket to the correct department. However, to be able to correctly redirect a ticket, the operator needs to know, what the request relates to. Ticket classification thus belongs to the main challenges in service desk systems (Jäntti, 2012).

Papers discussing the categorization of tickets in help desk systems often rely on traditional approaches based on sparse text representations (Zangardi et al., 2023). The modern artificial intelligence methods and neural networks typical in natural language processing (Qiu et al., 2020; Singh et al., 2022) that enable achieving state-of-the-art performance in many tasks are not examined.

The goal of the paper is to apply new approaches from the field of natural language processing (NLP) to the problem of ticket classification and derive useful findings and recommendations that would enable improved classification performance of real help desk systems.

2 CURRENT STATE

Landsman (2015) points out that from the efficiency perspective, the help desk system must not distinguish too many categories (hundreds) and recommends using about 20 categories. Al-Hawari and Barham (2021), Paramesh et al. (2018), Parmar et al. (2018), Herzig et al. (2013), and Eichhorn (2020) also investigated less than 20 categories. We can conclude that assigning a ticket to a category is typically a single-label multiclass classification problem, which is one of the most common applications of machine learning.

Only determining the ticket category is not sufficient to build a high-quality automated system. According to Olson (2018), operator response time and request resolution time are also very important indicators, and therefore assigning the correct ticket category as soon as possible after creation is essential to minimize operator delays.

We can expect that some problems are much more common than others, which leads to an imbalanced distribution of ticket categories. This fact poses significant problems for machine learning algorithms, as a sufficient number of documents from all categories is needed to train a good classifier (Liu et al., 2009). This is also confirmed by Paramesh et al. (2018), Parmar et al. (2018), Al-Hawari and Barham (2021), and Eichhorn (2020).

Although the field of natural language processing has experienced significant breakthroughs in recent years, many papers and applications still use traditional methods for applying machine learning to textual data. Specifically, for the field of automatic categorization of customer requests, several papers from recent years use sparse text representations with minimal use of modern artificial intelligence methods and neural networks.

Parmar et al. (2018) used a sparse representation of documents using tf-idf with not very extensive preprocessing steps, mainly comprising cleaning the dataset from empty values. The authors used a very imbalanced dataset containing thousands of documents with twelve different categories, on which they tested a total of five different classifiers, namely Multinominal Naïve Bayes (MNB), Support Vector Machine (SVM), Decision Tree, Random Forest, and K-Nearest Neighbors (KNN). The best results measured by accuracy were achieved with an SVM classifier reaching 63% of accuracy. Such a high error rate, however, means that the classifier is not very useful.

Paramesh et al. (2018) also used a sparse representation of documents using tf-idf. The authors applied extensive preprocessing to the input data to remove unwanted names, email addresses, phone numbers, etc. They also re-

moved stop words and used oversampling and subsampling to balance the classes. Afterward, they used the χ^2 test to filter words with small importance. Multiple classifiers were combined using Bagging, Boosting and Voting Ensemble to improve the predictions. The best results were achieved using the Bagged Decision Tree classifier with an accuracy of 92.04%.

Eichhorn (2020) used traditional methods with sparse document representation (tf-idf) as well. Data preprocessing consisted mainly of lemmatization and stop words and punctuation removal. Categories with less than 100 tickets were removed which reduced the number of categories from 13 to 8 and thus greatly simplified the classification. On the other hand, the ability to predict less frequent categories was lost. From eight tested classifiers, Logistic Regression performed the best with an accuracy of 85%.

Al-Hawari and Barham (2021) used a dataset consisting of 1,254 manually labeled tickets from 13 categories related to technical support. The text data was cleaned from HTML tags, punctuation, and special characters and transformed using Weka's StringToWordVector filter with default setting to a structured representation (tf-idf) and classical machine learning models (J48, NaiveBayes, DecisionTable, and SMO) were trained. During evaluation, an accuracy of 81.4% was achieved.

It is evident that traditional methods and algorithms for classifying customer requests are still very relevant for building a robust system. However, new approaches and representations that usually enable reaching state-of-the-art results in many domains have not been investigated in the domain of the classification of help desk tickets. To identify relevant aspects of the process, related research needs to be examined.

Zhong and Li (2019) focused on the categorization of transcripts of customer calls (over 9,000 documents and 4 distinct categories). They explored several approaches using textual representations with static word embeddings. It was found that pre-trained GloVe vectors Pennington et al. (2014) provided the best basis for building the system. The transcripts were preprocessed mainly with text-cleaning techniques. The authors used a convolutional neural network (CNN) to create a classifier and achieved an F1 score of 93%.

Opuchlich (2019) focused on the classification of tickets from an SAP database containing several million documents and over 4,000 categories falling into two main areas, namely IT and HR. The author focused on the comparison and eventual combination of traditional and modern approaches using sparse and dense vector representations of tickets, namely tf-idf and fastText (Bojanowski et al., 2017). The paper also focused on the impact of various preprocessing techniques such as stop word removal, lemmatization, and infrequent category removal. As a result of the analysis of the impact of preprocessing techniques, it was found that removing sparse categories had a minimal impact on the performance of the classifier. The author created a two-tier classifier to separate the HR and IT tickets first (an accuracy of 97.46%), and then separate classifiers were trained for each domain. Since the used dataset had a very high number of categories, the author decided to provide the five most likely predictions of the classifier, which significantly increased the accuracy of the aggregate system. At all classification levels, the classifier trained with the traditional approach produced slightly better results than fastText. However, when combining both using a voting ensemble, the accuracy increased by 2–3% to 81.4% for the IT classifier and 78.9% for the HR classifier.

None of the previously mentioned works, dealing with customer support, used the latest innovations in the field of NLP. For example, Minaee et al. (2021) compared various deep learning models for some of the most common NLP applications, including multi-class classification. In their document topic classification experiment, they used the DBpedia dataset containing over 600,000 documents and 14 distinct categories. The authors tested a total of 9 different models and all of them achieved very good F1-scores greater than 98%. Two best-performing models were based on $\text{BERT}_{\text{LARGE}}$ (Devlin et al., 2018). In the task of

categorizing the 127,000 news article summaries into four distinct categories, the transformer-based models also provided the best results.

Transformer-based models (Vaswani et al., 2017) changed the field of NLP several years ago. The most famous one, BERT (Bidirectional Encoder Representations from Transformers) uses several encoders stacked in several layers (Rogers et al., 2020). The model created its own vector representations (embeddings) of input tokens based on the context in which they occur. The tokens are created using the WordPiece algorithm that adds the most frequent combinations of characters to the vocabulary (Wu et al., 2016). BERT can process only a given number of tokens, typically 512, while special symbols [CLS] and [SEP] are added to the beginning and end of the input. BERT can be trained from scratch in the Masked Language Modeling and Next Sentence Prediction tasks or fine-tuned in a task like classification. Depending on the number of model parameters, $\text{BERT}_{\text{LARGE}}$ (340 mil.) or $\text{BERT}_{\text{BASE}}$ (110 mil.) are typically used. A smaller all-purpose model DistilBERT can after fine-tuning in a specific task achieve performance comparable to larger models much faster (Sanh et al., 2019). There also exist models adapted for specific languages, like SlavicBERT (Arkhipov et al., 2019) for Slavic languages.

3 DATA AND METHODS

3.1 Data

The first dataset contains mainly technical support requests written in Czech and comes from the internal helpdesk of ALVAO, a leading provider of help desk systems in the Czech Republic. The requests are stored in the form of tickets consisting of the title of a request and all messages within the conversation between the user and the operator. Each ticket also has a category that the user selects when creating the request.

Tab. 1: Distribution of tickets in categories for the examined datasets

	Number of tickets	
Category number	**ALVAO**	**Endava**
1	3,987	34,061
2	1,595	9,634
3	368	2,628
4	272	921
5	237	612
6	116	239
7	65	191
8	57	137
9		72
10		45
11		4
12		3
13		2

The dataset contains a total of around 6,700 unique tickets, which are divided into eight categories. The distribution of the categories can be found in Tab. 1. It is evident that the data is strongly imbalanced – the majority category contains about 60% of instances, while the three smallest categories account for less than 4% of instances in total.

Individual messages in a ticket take the form of an e-mail. This means that a reply contains a new message, plus the original message to which the reply refers. It is, therefore, necessary to remove any duplicate pieces of messages, together with various auxiliary structures, such as greetings, signatures, or attachments that contain no relevant information for a category determination and would only decrease the quality of structured representations of the messages. Based on the preliminary experiments, removing these parts has a crucial impact on the classification performance.

The request title is usually a short text containing about 50 characters on average and less than 150 characters in 99% of cases. This corresponds to about 30 or 63 tokens when using the SlavicBERT (Arkhipov et al., 2019) tokenizer. 99% of the introductory messages of all tickets were not longer than 1,051 characters (the average length was around 213 characters). This corresponds to 488 tokens, which still fall

below the maximum number of tokens (512) that SlavicBERT can process. When combining the request name and the text of the first message, 551 tokens are needed for 99% of the tickets, which may imply a slight loss of context in the message representation.

The second dataset is the Endava public dataset. It is a technical support dataset that was originally used by Microsoft for the purpose of creating a web service for the automatic categorization of English tickets within Microsoft Azure (Żak et al., 2021). Requests are stored in the form of tickets containing the title and message text. Only the first request messages are available, from which various auxiliary structures, email headers, stop words, non-alphanumeric characters, specific names, and other unwanted words are removed.

The titles of the requests have 23 characters on average and 99% of titles have less than 68 characters. The messages of 99% of tickets are not longer than 1,900 characters (the average length is 266 characters). This corresponds to 455 tokens, which is within the limits of $\text{BERT}_{\text{BASE}}$ (512 tokens).

The dataset contains almost 50,000 tickets in 13 anonymized categories. The dominant category contains over 70% of all tickets and the second most represented category accounts for almost 20% of all tickets. At the same time, five categories have less than 100 tickets. It is, therefore, clear that some data balancing techniques need to be used to improve the quality of the classifiers.

3.2 Experiments

The title and body of the messages were concatenated (with a dot in between) to use as much available information from a ticket as possible. The experiments investigated both the traditional sparse tf-idf representation and the representation relying on static and contextual embeddings.

To produce the sparse representation, the new line characters were replaced with a space, all non-alphanumeric characters and stop words were removed, and the texts were converted to lower-case in the ALVAO dataset. Lemmatization or using the word bi- or tri-grams did not bring any improvements. The tf-idf representation had almost 25,000 dimensions.

We used fastText as the language model with static embeddings. All messages were preprocessed in the same way as in the case of the tf-idf representation and were used to train the CBOW model with 300 dimensions. The embeddings of all the words from a message were averaged to obtain a representation of the ticket.

BERT (Devlin et al., 2018) was used as the model for creating contextual embeddings. We used SlavicBERT for Czech texts and $\text{BERT}_{\text{BASE}}$ for English texts. According to Sun et al. (2019), fine-tuning a model on the end task is often beneficial for the quality of the embeddings for the given task. Thus, models without and with fine-tuning were investigated.

The process of fine-tuning a BERT model involves the choice of the fine-tuning method and the data chosen. The pre-trained SlavicBERT was trained on the Masked Language Modeling and Next Sentence Prediction tasks like the original BERT. These two models thus cannot be used for ticket categorization without adding and retraining a classification head.

It is also possible to extract the embeddings of the last layer before the classification layer and use them to train a separate classifier. According to Choi et al. (2021), a document can be represented by either averaging or summing all token embeddings or by using a special [CLS] token placed at the start of each document as its representation.

We examined BERT as a model providing contextual embedding as follows:

- pre-trained embeddings from the last layer of the model (averaged and [CLS] token only) were extracted and used by a separate classifier,
- fine-tuned embeddings from the last model layer (averaged and [CLS] token only) were extracted and used by a separate classifier,
- a classification head using the CLS token from the last layer was added and trained on the classification task at the same time as the model.

3.2.1 Balancing the Datasets

Since the categories in both datasets were very imbalanced, various balancing techniques were explored. Aggarwal (2020) distinguishes two major approaches to balancing a dataset, namely undersampling of dominant categories and oversampling of minority categories. The SMOTE method (Chawla et al., 2022) is highlighted for oversampling, which produces synthetic samples after vectorizing the documents. To increase the number of instances from the minority classes, Coulombe (2018) recommends data augmentation using back-translation. The augmentation process consists of translating the text into another language and then translating it back into the original language of the input text. The idea behind this process is to replace some words with their synonyms or similar expressions as part of the machine translation. The result is thus a new synthetic text that is very similar to the original one.

In this work, we examined the impact of removing minority categories in the Endava dataset (three categories that contain 9 samples in total), random undersampling of dominant categories, random oversampling of minority categories (by duplicating some instances and using SMOTE), and the augmentation of minority categories by using back-translation utilizing the Microsoft Translator inside Azure.

To augment the ALVAO dataset, categories containing 150 or fewer tickets were augmented using English, French, and German. For augmenting the Endava dataset, categories containing 250 tickets or less were augmented using French, German, and Spanish. The same categories like during data augmentation were oversampled. In the end, the categories with less than 150 tickets were enlarged to have 250 tickets in the ALVAO dataset, and the categories with less than 250 tickets were enlarged to have 400 tickets in the Endava dataset. During undersampling, the sizes of the majority categories were decreased from 3,987 to 1,500, and from 1,595 to 1,000 tickets respectively in the ALVAO dataset. In the Endava dataset, the two majority categories were undersampled to 2,628 tickets to have the same number of tickets as the third biggest category.

3.3 Studied Classification Approaches

Several classifiers that proved to be successful for categorizing text data were examined. The implementation from the Scikit-learn library (Pedregosa et al., 2011) was used, except for XGBoost, which is implemented in a separate XGBoost library (Chen and Guestrin, 2016), and BERT implemented in the Simple Transformers library (Rajapakse, 2023), which is an extension to the Hugging Face Transformers library. Unless otherwise stated, classifiers were always trained with the default parameters of a given implementation. The examined classifiers include Gradient Boosted Tree (XGBoost), Support Vector Machine (SVM), Decision Tree, Multinominal Naïve Bayes (MNB), Logistic Regression, Random Forest, K-Nearest Neighbours (KNN) with $k = 3$, 4, and 5, Multilayer Perceptron (MLP) with three hidden layers and 100, 200, and 100 neurons in them, and BERT with a classification head (a linear layer on top of the pooled output from BERT, implemented as the BertForSequenceClassification method in the Transformers library by Hugging Face).

The separate classifiers were applied to all the text representations, i.e., tf-idf, fastText, and BERT embeddings. Classifiers that provided the best results, had their hyperparameters further optimized. The effect of combining the best-performing classifiers into a voting ensemble where the category with the most votes is picked was also investigated.

The data was split in the ratio of 85% for training and 15% for testing for all experiments except the experiments utilizing the full (not undersampled) Endava dataset, where the test data was first undersampled and a ratio of 96% for training and 4% for testing was used. Classification success was evaluated using a test dataset that was distinct from the one used for creating the model. Thus, the observed measures represent realistic expectations (Xu and Goodacre, 2018).

The quality of classifiers was measured by macro- and micro-averaged F1 scores (Goutte and Gausier, 2005).

4 RESULTS

Tab. 2 contains the F1 scores achieved for different combinations of classifiers, text data representations, and data balancing methods for the ALVAO dataset. No categories with very small numbers of instances existed and thus were not removed. Only the best results for each balancing method are presented. It is evident that data augmentation using back-translation brought the biggest improvement so it was further investigated. Separate classifiers also always outperformed BERT, although often used its embeddings as the input.

Tab. 3 contains the results achieved with different text representations with the augmented dataset. The representations based on the fine-tuned BERT embeddings occupy the top three positions.

Selected models (XGBoost and KNN classifiers using the CLS token of fine-tuned BERT and BERT with a classification head) were further optimized and combined in a voting ensemble. This improved the classification result expressed by the F1 score by no more than 1%, which is a negligible improvement, see Tab. 4.

Tab. 5 represents the effect of balancing the dataset Endava. The operations related to balancing were performed in the following order (corresponding to the table rows): undersampling, removing minority categories, and one of the techniques increasing the size of minority categories (i.e., oversampling and augmentation). Undersampling and removing minority categories improved the macro F1 score by almost 19%. The biggest improvement related

Tab. 2: Results achieved for different combinations of classifiers, text data representations, and data balancing methods for the ALVAO dataset

Balancing method	Representation	Classifier	Macro F1 score	Weighted F1 score
none	BERT fine-tuned, AVG	XGBoost	0.845	0.975
augmentation	BERT fine/tuned, CLS	XGBoost	0.864	0.977
oversampling	BERT fine-tuned, AVG	SVM	0.838	0.976
SMOTE	fastText	SVM	0.834	0.967
undersampling	BERT fine-tuned, AVG	KNN(3)	0.837	0.971

Tab. 3: Results achieved for text data representations with the augmented ALVAO dataset

Representation	Classifier	Macro F1 score	Weighted F1 score
BERT fine-tuned, CLS	XGBoost	0.864	0.977
BERT fine-tuned, AVG	XGBoost	0.855	0.975
BERT fine-tuned, CLS	KNN(3)	0.852	0.975
BERT fine-tuned	classifier head	0.846	0.973
fastText	SVM	0.824	0.971
BERT AVG	logistic regression	0.779	0.957
tf-idf	XGBoost	0.761	0.953
BERT, CLS	logistic regression	0.690	0.939

Tab. 4: Results achieved after hyperparameters tuning the ALVAO dataset

Representation	Classifier	Macro F1	Weighted F1	Accuracy
BERT (finetuned), CLS	XGBoost	0.869	0.976	0.976
BERT (finetuned), CLS, class. head	voting ensemble	0.866	0.976	0.976
BERT (finetuned)	classification head	0.857	0.975	0.975
BERT (finetuned), CLS	KNN(3)	0.852	0.975	0.975

Tab. 5: Results achieved for different combinations of classifiers, text data representations, and data balancing methods for the Endava dataset

Balancing method	Representation	Classifier	Macro F1 score	Weighted F1 score
none	BERT fine-tuned	MNB	0.473	0.751
undersampling	BERT fine/tuned	XGBoost	0.528	0.794
removing small categories	BERT fine-tuned, CLS	MLP	0.760	0.770
augmentation	BERT fine-tuned, CLS	XGBoost	0.702	0.790
oversampling	BERT fine-tuned, CLS	XGBoost	0.676	0.773
SMOTE	BERT fine-tuned, CLS	Random Forest	0.690	0.794

Tab. 6: Results achieved for text data representations with the augmented Endava dataset

Representation	Classifier	Macro F1 score	Weighted F1 score
BERT fine/tuned, CLS	XGBoost	0.702	0.790
BERT fine/tuned, CLS	Random Forest	0.670	0.793
BERT fine/tuned	classifier head	0.680	0.781
BERT fine/tuned, AVG	XGBoost	0.672	0.769
tf-idf	XGBoost	0.641	0.755
fastText	XGBoost	0.630	0.747
BERT, CLS	XGBoost	0.582	0.656
BERT, AVG	KNN(4)	0.577	0.632

Tab. 7: Results achieved after hyperparameters tuning the Endava dataset

Representation	Classifier	Macro F1	Weighted F1	Accuracy
BERT (finetuned), CLS, class. head	voting ensemble	0.703	0.787	0.790
BERT (finetuned), CLS	XGBoost	0.702	0.790	0.792
BERT (finetuned), CLS	random forest	0.700	0.784	0.787
BERT (finetuned)	classification head	0.693	0.780	0.783

to increasing the size of minority categories was brought by augmentation, similarly to the ALVAO dataset.

Tab. 6 shows the best results achieved with each text representation on the augmented dataset. The approaches using the fine-tuned BERT model provide relatively good results, while the approach using the CLS token of the tuned BERT model and the XGBoost classifier provided the best results. Approaches using the pre-trained BERT model provided the worst F1 scores (both macro and weighted).

After tuning the hyperparameters of the models, only a negligible improvement over the best results of unoptimized models was achieved. Similarly to the ALVAO dataset, classifiers using the CLS token of the fine-tuned BERT model were selected. This time, XGBoost, Random Forest, and BERT with a classification head were studied. For the Endava dataset, combining the fine-tuned classifiers into a voting ensemble achieved the best macro-averaged F1 score of 70.28%, see Tab. 7.

5 DISCUSSION

In the experiments, we demonstrated that modern text data representations enable achieving results better than the traditional approaches. It was also found that to effectively use the embeddings obtained by the BERT model, it is beneficial to fine-tune the model on the given task. Even though BERT's added classification head did not provide the best results, the embeddings of the last encoder of this fine-tuned model were crucial to achieving the best results.

As transformer models are pre-trained on large text corpora and requests in helpdesk systems are highly domain-specific, traditional methods often bring satisfactory performance too (Campese et al., 2022).

It was also found that balancing the classes in the dataset often helps to improve the quality of the classifier. For the ALVAO dataset, augmentation using machine translation had the greatest impact, while the other examined techniques slightly degraded the results in this case. On the other hand, for the Endava dataset, the balancing techniques had a much greater impact and without them, the classifier would be practically unusable. Of the three examined techniques for increasing the size of minority categories, augmentation using machine translation was the most effective.

Using the best-performing classifiers, a macro F1 score equal to 86.94%, a weighted F1 score equal to 97.60%, and an accuracy of 97.61% were achieved for the ALVAO dataset. For the Endava dataset, a macro F1 score of 70.3%, a weighted F1 score of 79%, and an accuracy of 79.2% were reached, see Tab. 4 and 7.

For the ALVAO dataset, the main problems were caused by only one minority category that was very similar to another one. If these two similar categories were merged, a much higher macro F1 score could be achieved.

Much worse results were provided by the classifiers working with the Endava dataset. Since the dataset is anonymized, it is difficult to analyze the causes of the problem. The classes were also very imbalanced, much more so than in the ALVAO dataset, and even the balancing techniques were not able to fully resolve this issue. Moreover, the textual data provided here was already aggressively preprocessed, which may have reduced the effectiveness of contextual embeddings. Although the resulting classification metrics are quite low, they are much higher than those of Żak et al. (2021) who provided this dataset.

Although it is not possible to directly compare the results with other research based on different datasets, comparisons can still provide interesting information. For example, if we compare the results from the experiments with the ALVAO dataset to the work of Eichhorn (2020) using traditional approaches and similar data containing eight categories, we can see that the best approach from this paper achieves a 12.61% higher accuracy.

6 CONCLUSIONS

In this paper, we focused on the categorization of tickets in service desk systems. We explored modern neural network-based artificial intelligence methods and compared them to traditional approaches to find the potential for improvement and to address typical problems in the domain.

We demonstrated that modern text data representations, especially those provided by fine-tuned transformer-based models, enabled achieving results significantly better than the traditional approaches described in the literature.

During experiments with two real-world datasets, we concluded that text preprocessing, balancing the ticket categories, and using the representations of texts based on fine-tuned transformers were crucial for achieving classifiers with satisfactory performance.

7 ACKNOWLEDGEMENTS

The paper was created with the support of grant EG20_321/0023606 (Research and development of artificial intelligence software platform for digital office) provided by the Ministry of Industry and Trade of the Czech Republic.

8 REFERENCES

Aggarwal, C. C. 2020. *Data Classification: Algorithms and Applications*. Boca Raton: Chapman and Hall/CRC.

Al-Hawari, F. and Barham, H. 2021. A Machine Learning Based Help Desk System for IT Service Management. *Journal of King Saud University – Computer and Information Sciences*, 33 (6), 702–718. DOI: 10.1016/j.jksuci.2019.04.001.

Arkhipov, M., Trofimova, M., Kuratov, Y. and Sorokin, A. 2019. Tuning Multilingual Transformers for Language-Specific Named Entity Recognition. In *Proceedings of the 7th Workshop on Balto-Slavic Natural Language Processing*, pp. 89–93. DOI: 10.18653/v1/W19-3712.

Bojanowski, P., Grave, E., Joulin, A. and Mikolov, T. 2017. Enriching Word Vectors with Subword Information. *Transactions of the Association for Computational Linguistics*, 5, 135–146. DOI: 10.1162/tacl_a_00051.

Campese, S., Agostini, F., Pazzini, J. and Pozza, D. 2022. Beyond Transformers: Fault Type Detection in Maintenance Tickets with Kernel Methods, Boost Decision Trees and Neural Networks. In *2022 International Joint Conference on Neural Networks (IJCNN)*, 1–8. DOI: 10.1109/IJCNN55064.2022.9892980.

Chawla, N. V., Bowyer, K. W., Hall, L. O. and Kegelmeyer, W. P. 2022. SMOTE: Synthetic Minority Over-sampling Technique. *Journal of Artificial Intelligence Research*, 16, 321–357. DOI: 10.1613/jair.953.

Chen, T. and Guestrin, C. 2016. XGBoost: A Scalable Tree Boosting System. In *KDD '16: Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 785–794. DOI: 10.1145/2939672.2939785.

Choi, H., Kim, J., Joe, S. and Gwon, Y. 2021. *Evaluation of BERT and ALBERT Sentence Embedding Performance on Downstream NLP Tasks*. ArXiv: 2101.10642v1. DOI: 10.48550/arXiv.2101.10642.

Coulombe, C. 2018. *Text Data Augmentation Made Simple by Leveraging NLP Cloud APIs*. ArXiv: 1812.04718v1. DOI: 10.48550/arXiv.1812.04718.

Devlin, J., Chang, M.-W., Lee, K. and Toutanova, K. 2018. *BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding*. ArXiv: 1810.04805v2. DOI: 10.48550/arXiv.1810.04805.

Eichhorn, G. 2020. *Predict IT Support Tickets with Machine Learning and NLP* [online]. Available at: `https://towardsdatascience.com/predict-it-support-tikets-with-machine-learning-and-nlp-a87ee1cb66fc`. [Accessed 2023, May 1].

Goutte, C. and Gaussier, E. 2005. A Probabilistic Interpretation of Precision, Recall and F-Score, with Implication for Evaluation. In Losada, D. E. and Fernández-Luna, J. M. (eds.). *Advances in Information Retrieval*, Lecture Notes in Computer Science, vol. 3408, pp. 345–359. Springer, Berlin, Heidelberg.

Herzig, K., Just, S. and Zeller, A. 2013. It's Not a Bug, It's a Feature: How Misclassification Impacts Bug Prediction. In *Proceedings of the 2013 International Conference on Software Engineering*, 392–401. DOI: 10.1109/ICSE.2013.6606585.

Jäntti, M. 2012. Examining Challenges in IT Service Desk System and Processes: A Case Study. In *ICONS 2012: The Seventh International Conference on Systems*, 105–108. ISBN 978-1-61208-184-7.

Landsman, I. 2015. *A Guide to Support Ticket Categorization* [online]. Available at: `https://www.helpspot.com/blog/a-guide-to-support-ticket-categorization`. [Accessed 2023, January 15].

Liu, Y., Loh, H. T. and Sun, A. 2009. Imbalanced Text Classification: A Term Weighting Approach. *Expert Systems with Applications*, 36 (1), 690–701. DOI: 10.1016/j.eswa.2007.10.042.

Menken, I. and Blokdijk, G. 2009. *Support Center Complete Handbook: How to Analyze, Assess, Manage and Deliver Customer Business Needs and Exceed Customer Expectations with Help Desk, Support Center and Service Desk*. Brisbane: Emereo Publishing.

Minaee, S., Kalchbrenner, N., Cambria, E., Nikzad, N., Chenaghlu, M. and Gao, J. 2021. *Deep Learning Based Text Classification: A Comprehensive Review*. ArXiv: 2004.03705v3. DOI: 10.48550/arXiv.2004.03705.

Olson, S. 2018. *10 Help Desk Metrics for Service Desks and Internal Help Desks* [online]. Available at: `https://www.zendesk.com/in/blog/top-10-help-desk-metrics/`. [Accessed 2022, November 13].

Opuchlich, P. 2019. *Text Classification of Support Ticket Data of SAP* [online]. Available at: `https://www.researchgate.net/publication/340492353_Text_Classification_of_Support_tiket_Data_of_SAP`. [Accessed 2023, January 20].

Paramesh, S. P., Ramya, C. and Shreedhara, K. S. 2018. Classifying the Unstructured IT Service Desk Tickets Using Ensemble of Classifiers. In *3rd International Conference on Computational Systems and Information Technology for Sustainable Solutions (CSITSS)*, 221–227. DOI: 10.1109/CSITSS.2018.8768734.

Parmar, P. S., Biju, P. K., Snahkar, M. and Kadiresan, N. 2018. Multiclass Text Classification and Analytics for Improving Customer Support Response through Different Classifiers. In *International Conference on Advances in Computing, Communications and Informatics (ICACCI)*, 538–542. DOI: 10.1109/ICACCI.2018.8554881.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M. and Duchesnay, É. 2011. Scikit-Learn: Machine Learning in Python. *Journal of Machine Learning Research*, 12 (85), 2825–2830.

Pennington, J., Socher, R. and Manning, C. 2014. GloVe: Global Vectors for Word Representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 1532–1543. DOI: 10.3115/v1/D14-1162.

Qiu, X., Sun, T., Xu, Y., Shao, Y., Dai, N. and Huang, X. 2020. Pre-Trained Models for Natural Language Processing: A Survey. *Science China Technological Sciences*, 63, 1872–1897. DOI: 10.1007/s11431-020-1647-3.

Rajapakse, T. 2023. *Simple Transformers* [online]. Available at: `http://simpletransformers.ai`. [Accessed 2023, September 1].

Rogers, A., Kovaleva, O. and Rumshisky, A. 2020. A Primer in BERTology: What We Know About How BERT Works. *Transactions of the Association for Computational Linguistics*, 8, 842–866. DOI: 10.1162/tacl_a_00349.

Sanh, V., Debut, L., Chaumond, J. and Wolf, T. 2019. *DistilBERT, a Distilled Version of BERT: Smaller, Faster, Cheaper and Lighter*. ArXiv: 1910.01108v4. DOI: 10.48550/arXiv.1910.01108.

Singh, G., Mittal, N. and Chouhan, S. S. 2022. A Systematic Review of Deep Learning Approaches for Natural Language Processing in Battery Materials Domain. *IETE Technical Review*, 39 (5), 1046–1057. DOI: 10.1080/02564602.2021.1984323.

Sun, C., Qui, X., Xu, Y. and Huang, X. 2019. *How to Fine-Tune BERT for Text Classification?* ArXiv: 1905.05583v3. DOI: 10.48550/arXiv.1905.05583.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł. and Polosukhin, I. 2017. *Attention Is All You Need*. ArXiv: 1706.03762. DOI: 10.48550/arXiv.1706.03762.

Wu, Y., Schuster, M., Chen, Z., Le, Q. V., Norouzi, M., Macherey, W., Krikun, M., Cao, Y., Gao, Q., Macherey, K., Klingner, J., Shah, A., Johnson, M., Liu, X., Kaiser, Ł., Gouws, S., Kato, Y., Kudo, T., Kazawa, H., Stevens, K., Kurian, G., Patil, N., Wang, W., Young, C., Smith, J., Riesa, J., Rudnick, A., Vinyals, O., Corrado, G., Hughes, M. and Dean, J. 2016. *Google's Neural Machine Translation System: Bridging the Gap between Human and Machine Translation*. ArXiv: 1609.08144. DOI: 10.48550/arXiv.1609.08144.

Xu, Y. and Goodacre, R. 2018. On Splitting Training and Validation Set: A Comparative Study of Cross-Validation, Bootstrap and Systematic Sampling for Estimating the Generalization Performance of Supervised Learning. *Journal of Analysis and Testing*, 2, 249–262. DOI: 10.1007/s41664-018-0068-2.

Zangari, A., Marcuzzo, M., Schiavinato, M., Gasparetto, A. and Albarelli, A. 2023. Ticket Automation: An Insight into Current Research with Applications to Multi-Level Classification Scenarios. *Expert Systems with Applications*, 225, 119984. DOI: 10.1016/j.eswa.2023.119984.

Zhong, J. and Li, W. 2019. Predicting Customer Call Intent for the Auto Dealership Industry from Analyzing Phone Call Transcripts with CNN for Multi-Class Classification. *International Journal on Soft Computing, Artificial Intelligence and Applications*, 8 (3), 13–25. DOI: 10.5121/ijscai.2019.8302.

Żak, K., Glavota, F., Mironica, I., Dinu, B., Marin, B., Vinca, F., Raducanu, I. and Tipau, A. 2021. *GitHub – karolzak/support-tickets-classification/* [online]. Available at: `https://github.com/karolzak/support-tickets-classification`. [Accessed 2023, April 5].

AUTHOR'S ADDRESS

Filip Koukal, Department of Informatics, Faculty of Business and Economics, Mendel University in Brno, Zemědělská 1, 613 00 Brno, Czech Republic, e-mail: koukalfilip96@gmail.com

František Dařena, Department of Informatics, Faculty of Business and Economics, Mendel University in Brno, Zemědělská 1, 613 00 Brno, Czech Republic, e-mail: frantisek.darena@mendelu.cz (corresponding author)

Roman Ježdík, ALVAO, s. r. o., Hlohová 1455/10, 591 01 Žďár nad Sázavou, Czech Republic, e-mail: roman.jezdik@alvao.com

Jan Přichystal, Department of Informatics, Faculty of Business and Economics, Mendel University in Brno, Zemědělská 1, 613 00 Brno, Czech Republic, e-mail: jan.prichystal@mendelu.cz