Discovering accurate deep learning based predictive models for automatic customer support ticket classification

Paolo Zicari
University of Calabria
Rende (CS), Italy
p.zicari@dimes.unical.it

Gianluigi Folino
ICAR Institute, National Research Council
Rende (CS), Italy
gianluigi.folino@icar.cnr.it

Massimo Guarascio
ICAR Institute, National Research Council
Rende (CS), Italy
massimo.guarascio@icar.cnr.it

Luigi Pontieri
ICAR Institute, National Research Council
Rende (CS), Italy
luigi.pontieri@icar.cnr.it

ABSTRACT

Ticket Management Systems are widespread in disparate kinds of companies and organizations, as they represent a fundamental tool for handling customer requests and issues in an efficient and effective manner. In particular, accurately categorizing incoming tickets is a key task in real-life application settings (e.g., helpdesk/CRM systems and bug tracking systems), in order to improve ticket processing efficiency and effectiveness (e.g., in terms of customer satisfaction). In this work, we propose a comprehensive ticket-categorization analysis that relies on inducing and exploiting a heterogeneous ensemble of deep learning architectures, in addition to a range of functionalities for acquiring, integrating and pre-processing ticket-related information coming from different channels (e.g. *mail*, *chat*, *web form*, etc.). Experimental results conducted on the specific application scenario concerning the data of a publicly available ticket-mining dataset have proven the effectiveness of the framework in different ticket categorization tasks.

CCS CONCEPTS

• **Information systems → Data analytics**; **Data mining**; • **Computer systems organization → Neural networks**;

KEYWORDS

Automatic ticket classification and assignment, Automatic customer support, Ensemble of Deep Neural Networks

ACM Reference Format:
Paolo Zicari, Gianluigi Folino, Massimo Guarascio, and Luigi Pontieri. 2021. Discovering accurate deep learning based predictive models for automatic customer support ticket classification. In *The 36th ACM/SIGAPP Symposium on Applied Computing (SAC '21), March 22–26, 2021, Virtual Event, Republic of Korea.* ACM, New York, NY, USA, Article 4, 4 pages. https://doi.org/10.1145/3412841.3442109

SAC '21, March 22–26, 2021, Virtual Event, Republic of Korea

ACM ISBN 978-1-4503-8104-8/21/03.
https://doi.org/10.1145/3412841.3442109

1 INTRODUCTION

Customer support is a crucial task for enterprises, as witnessed by the widespread usage of Customer Relationship Management (CRM) systems. Indeed, with the aim of providing effective support to their customer base, many companies made up specialized customer service departments with dedicated personnel. Therefore, an accurate response is the key to ensure the customer satisfaction and reduce the risk of churning. As an example, the Uber's Customer Obsession team is assigned to design advanced tools to make the user experience quicker and more seamless across Uber services.

Ticket Management Systems (TMSs) are currently used to improve organization, efficiency and effectiveness of the custom support with relevant impacts on costs and revenues, customer retention, and public brand image.

Tickets, usually named cases or issues, are opened with a customer request through different channels. A multi-channel ticketing system collects all support tickets from different channels and organizes them in a single view. The most common channels for customer support use *emails*, *phone calls*, *web form*, *live chat*, and now also *social media* like Facebook and Twitter to have an easy and user-friendly communication with media increasingly used by young people. Thus, the integration of requests from different channels is a relevant issue in this scenario to improve the handling of the tickets and offer more personalized assistance.

The automatic ticket classification and assignment for customer support is a specific field of application of Natural Language Processing (NLP) and text classification. In literature, many solutions have been proposed in order to improve the capacity of customer support systems in solving ticket issues both in terms of accuracy and efficiency.

Most of the works correlated to our approach are based on machine learning. In [2], a machine learning-based help desk system for IT service management was proposed in order to improve the performance of the German Jordanian University IT staff to cope with technical ticket issues. A machine learning model performs the automatic ticket classification by exploiting title, description and comments of the ticket; specifically, TF-IDF is used for tokenization and for feature extraction, while the SVM algorithm is used for the classification task.

In [3], the authors proposed to adopt machine learning algorithms for ticket classification in Issue Tracking Systems (ITS), i.e.,

a software to capture and keep track of customer issues, and to automatically assign the issue tickets to the relevant person or unit in the support team. Tests were executed on a dataset collected from Istanbul Technical University ITU Issue Tracking System, a web application for answering on various requests to different departments within the university. The classification process contains two phases, one for the category and the other for the sub-category individuation.

A quite recent trend is the usage of *Deep Learning* architectures for ticket classification, which were proven as a powerful and convenient means for inducing accurate models from raw low-level data. For example, in [8], COTA (Customer Obsession Ticket Assistant), an intelligent system was integrated with Uber's customer support platform in order to improve the quality of their service for the tasks of ticket type identification and recommendation of the suitable reply template. This system uses deep learning algorithms in an Encoder-Combiner-Decoder architecture. In [11], a deep learning solution was proposed for the automatic classification of phone calls in an auto dealer company for customer service tasks. A convolutional neural network learning model was developed to classify the customer calls into four intent categories: sales, service, vendor or job seeker.

Clearly, it is not easy to choose among all the alternative Deep Neural Networks (DNN) architectures that could be used for ticket classification, each one needing long and careful design and tuning activities in order to work at best. In general, *ensemble classification methods* are widely recognized as a valuable tool for automatically combining different (weak) classifiers into a stronger integrated classification model, which improves all of its components in terms of generalization power.

In order to face ticket classification problems, we here propose a DNN-based framework hinging upon a novel ensemble of classifiers, which integrates heterogeneous DNN architectures by using a number of alternative combination schemes. The proposed approach, by integrating data coming from different channels (i.e., mails, chats, etc.) and exploiting an over-sampling technique for handling the unbalanced classes problem typical of this scenario, achieves accurate results in the prediction of category labels that are very relevant for improving core ticket routing tasks. An extensive experimentation conducted on a real-life dataset has confirmed the validity of our proposal in terms of predictive accuracy in different ticket classification tasks, compared to a number of baselines and competitors.

As a matter of fact, to the best of our knowledge, no approach based on an ensemble of DNNs has been proposed so far for this task.

The rest of this paper is organized as follows: Section 2 presents the ensemble of Deep Neural Networks adopted in this work for the task of ticket classification; Section 3 illustrates the parameters, the datasets and the experimental results; finally, conclusions are drawn in Section 4.

2 DEEP LEARNING ENSEMBLE MODEL

Our approach relies on inducing a novel kind of Ticket Classification Model, which takes the form of an ensemble of Deep Neural Networks (DNNs), and it is meant to ensure accurate and robust categorization performances.

Data preparation is a crucial step for improving the performances of ticket classification task. After the anonymization, the preprocessing includes the operations of cleaning the texts from errors (misspelling), as well as removing space and stop-words together with several tasks needed to convert texts into sequences of tokens. With *Lowercasing*, upper case letters in each word are converted into lower case ones, in order to normalize the representation of a concept, while also reducing sparsity and vocabulary size. Through *Lemmatizing*, tokens are replaced with the corresponding lemma, so that different forms of the same lemma are all uniformed to the same root form, representing a single distinguished vocabulary entry. *Stemming* replaces each token with the corresponding stem, representing a chopped-off form of the former. *Stopwords removal* purges off commonly used words, usually known as stopwords, since they are likely to convey a low amount of information; thus, it also reduces the number of features and the vocabulary size. Finally, by using *Noise removal*, characters, digits and pieces of text that can interfere with the analysis of the text are removed.

The word embedding model is adopted for the text representation, words are coded as dense vectors representing their projection into a continuous vector space. The position of a word within the vector space is learned from texts on the basis of its surrounding context words so that close distance embeddings vectors correspond to similar words or words that are often found in the same context.

Typically, ensemble-based methods require the base models to be different enough, in order to ensure good and robust predictive power, even though the correlation between diversity and accuracy is still a matter of research and no strong conclusion has been drawn yet on this aspect [7]. In order to promote the diversity of the single predictions, in our solution a set of Neural Networks, $DNN_1, \ldots, DNN_n$, based on different deep architectures, are used in the first layer as base learners. More specifically, we devised a multi-representation ensemble model, schematically illustrated in Figure 1.

The following different DNN architectures are used to make up the ensembles: *(i) cnn*, a *Convolutional Neural Network* composed of a stack of *Residual Blocks* subnets [4], each one consisting of two instances of a *Convolutional Layer* linked one to the other by a skip connection; *(ii) cnn_gmp*, which is a variant of the *cnn* architecture that includes an additional *global max pooling* layer, meant to downsample the convolution results, *(iii) ffnn*, a *feed-forward neural network* composed of a stack of dense layers linked in residual block subnets and equipped with a *Rectified Linear Unit (ReLU)* [9] activation function; and *(iv) ffnn_gmp*, which extends the *ffnn* architecture with a *global max pooling* layer.

Nevertheless, in all these architectures, we alternate convolutional and dense layers with both: *(a) batch-normalization* layers, allowing for improving stability and performances [5]; and *(b) dropout* layers, for mitigating the risk of overfitting [6].

In Figure 2, a Residual Block including both Batch Normalization and Drop-out layers, is shown as an example.

Finally, the last layers of the above-described models are equipped with a *softmax* activation function for multi-class classification tasks.

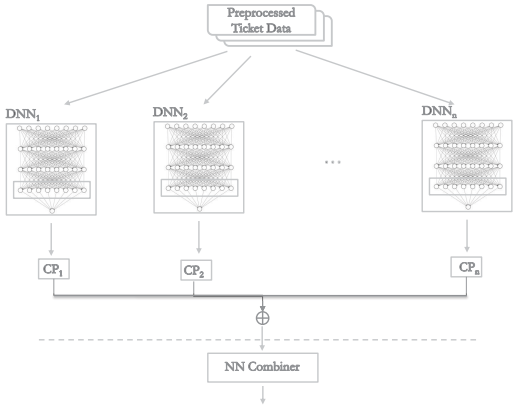

Figure 1: Deep Learning based ensemble model.

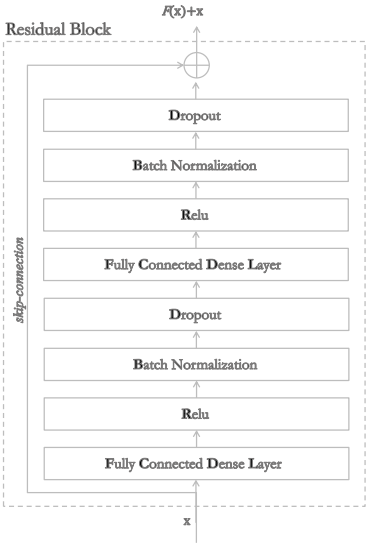

Figure 2: Residual Block architecture.

The last high-level layer of the ensemble model, named here *NN combiner*, is devoted to encoding the function for combining the outputs of the base DNN classifiers into an overall prediction.

Defining the strategy to combine the base models into a collective decision is a critical issue in each ensemble classification model. In our framework, two alternatives strategies can be used to instantiate the NN combiner sub-net of our model: *(a)* a simple *non-trainable* architecture that just average the predictions yielded by the base models, and *(b)* a *trainable* architecture implementing a variant of classical stacked generalization schemes, for learning a "context-aware" meta-classification scheme. Two alternative variants of the proposed ticket classification model are obtained when using the options *(a)* or *(b)* mentioned above, denoted hereinafter as *Avg ensemble*, and *Stack ensemble*. More precisely:

- In *Avg ensemble*, the combiner is a non-trainable sub-net encoding a fixed function that returns, for any ticket, the average of the class-membership probability among those estimated for it by the base DNN models.
- In *Stack ensemble*, the combiner is a trainable feed-forward sub-net, consisting of just one dense layer, that takes as input the predictions $CP_1, \ldots, CP_n$ provided by the base learners.

A *Data Resampler* was included for mitigating the class unbalance issue (i.e., majority classes overwhelm minority ones). An over-sampling strategy was adopted to address the unbalancing occurring when the rare cases represent very small portions of the training set, which can be considered as exceptional concepts seldom occurring within these classes.

3 EXPERIMENTAL EVALUATION

This section illustrates a suite of experiments that we conducted to evaluate the performance of our ensemble-based approaches in comparison with different types of neural networks and with some state-of-the art ensemble techniques. All the experiments were performed on the Endava dataset, [1], a publicly available ticket mining dataset. Essentially, the Endava dataset is a collection of about 50K classified support tickets submitted as emails by users to the Endava helpdesk operators. Microsoft (Commercial Software Engineering team) in collaboration with Endava proposed a proof of concept project for text analysis and automatic classification through a web service in Azure cloud, in order to reduce the waste of time required by helpdesk operators to evaluate tickets and trying to assign values to some important properties.

The dataset includes 48,550 text messages, having an average length of 39.66 tokens, and associated each with several labels (manually assigned by the helpdesk operators): ticket type, category, sub category1, sub category2, urgency, impact, business service.

We considered the four-class classification problem of predicting the urgency label of a ticket. As it can be easily seen in Table 1, the class distribution is quite imbalanced.

As regards the base models, both dense and convolutional layers have been initialized with 64 computational units. Specifically, *Feed Forward* and *Convolutional Neural Networks* are composed of a stack of 5 layers and equipped with a ReLU activation function. Moreover, *Adam* [10] optimizer is used in the learning phase, while the dropout rate is set to 0.01.

Both the base and ensemble models were trained for a number of epochs equals to 32 and a batch size of 64. For each configuration of the model, we performed 30 runs.

The overall framework was implemented in Python (using the Keras framework on top of Tensorflow).

Standard metrics for evaluating a classifier (i.e., accuracy) are not really suitable for evaluating classifiers trained on class-imbalanced data, as the dataset used in this work. Therefore, in our experimentation, we also computed three further metrics, namely *F1*, *AUC* and *G-measure*, which are more suitable and informative in the case of imbalanced data.

Our ensemble-based approaches are compared in terms of the different metrics introduced, in comparison with 1) the different types of baseline neural networks and 2) with some state-of-the art ensemble techniques, i.e., gradient boosting and random forest,

Table 1: Class label distribution in the Endava dataset.

Label	Perc.
Urgency 0	3.40%
Urgency 1	13.90%
Urgency 2	11.39%
Urgency 3	71.31%

leveraging the respective implementations available in the *scikit-learn* library[1]. They were run using standard parameters, without performing any kind of parameter tuning and using 50 estimators for both the algorithms.

Table 2 compares the different configurations of our ensemble method, the base DNNs (as a sort of baseline), and some state-of-the art ensemble-based techniques (i.e., random forest and gradient boosting). Each of the F-measure, AUC and G-mean scores reported in the table was computed by averaging the results obtained in 30 runs.

Table 2: Comparison of our approach with other ensemble strategies: AUC, G-mean and F1 for the Endava dataset.

Algorithm	AUC	G-mean	F1
cnn	0.939 ± 0.0017	0.736 ± 0.0080	0.568 ± 0.0095
cnn_gmp	0.935 ± 0.0040	0.695 ± 0.0115	0.462 ± 0.0500
ffnn	0.941 ± 0.0014	0.734 ± 0.0051	0.571 ± 0.0076
ffnn_gmp	0.945 ± 0.0014	0.741 ± 0.0051	0.575 ± 0.0084
Gradient Boosting	0.944 ± 0.0010	0.692 ± 0.0040	0.532 ± 0.0060
Random Forest	0.944 ± 0.0010	0.723 ± 0.0040	0.582 ± 0.0070
Avg ensemble	0.949 ± 0.0010	0.749 ± 0.0041	0.593 ± 0.0060
Stack ensemble	0.949 ± 0.0012	0.752 ± 0.0041	0.594 ± 0.0066

By analyzing Table 2, it is evident that the ensemble-based strategies outperform all the baselines, while there is no substantial difference among the different strategies in combining the ensemble. The stack ensemble performs better than all the other techniques (ensembles and baselines) in terms of AUC, G-mean, and F1. The two proposed ensembles are comparable, but sensibly better than the baselines. Among the other ensemble techniques, the Random Forest obtains slightly better results in terms of G-mean and F1.

4 CONCLUSIONS

Increasing the level of automation in TMS systems produces several beneficial effects for both the company and customers. Indeed, this can help the company to reduce processing times, the number of mistakes due to human errors, and the amount of involvement of human resources in repetitive tasks; on the other hand, customer satisfaction can be improved, by providing customers with faster and more accurate answers to their requests.

In this paper, NLP and Deep Learning techniques have been employed to support issue classification and routing tasks. Experiments conducted on a real use case coming from a customer support helpdesk system demonstrated the accuracy improvements obtained by using the deep ensemble approach, in comparison with both the baseline DNNs and the other ensemble-based approaches.

[1]https://https://scikit-learn.org/stable/

For different reasons, such as introduction of new offers and new products, ticket data may change suddenly and quickly; therefore, future works will focus on extending the framework to cope with these changes and to handle concept drifts.

REFERENCES

[1] Endava dataset (2019), https://github.com/karolzak/support-tickets-classification
[2] Al-Hawari, F., Barham, H.: A machine learning based help desk system for it service management. Journal of King Saud University - Computer and Information Sciences (2019). https://doi.org/https://doi.org/10.1016/j.jksuci.2019.04.001
[3] Altintas, M., Tantug, A.C.: Machine learning based ticket classification in issue tracking systems (2014)
[4] He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 770–778 (2016)
[5] Hinton, G.E., Srivastava, N., Krizhevsky, A., Sutskever, I., Salakhutdinov, R.: Dropout: A simple way to prevent neural networks from overfitting. Journal of Machine Learning Research **15**, 1929–1958 (2014)
[6] Ioffe, S., Szegedy, C.: Batch normalization: Accelerating deep network training by reducing internal covariate shift. In: Proceedings of the 32Nd International Conference on International Conference on Machine Learning - Volume 37. pp. 448–456. ICML'15, JMLR.ORG (2015)
[7] Kuncheva, L.I.: Diversity in multiple classifier systems. Information Fusion **6**(1), 3–4 (2005)
[8] Molino, P., Zheng, H., Wang, Y.C.: 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 Data Mining. p. 586–595. KDD '18, Association for Computing Machinery, New York, NY, USA (2018). https://doi.org/10.1145/3219819.3219851
[9] Nair, V., Hinton, G.E.: Rectified linear units improve restricted boltzmann machines. In: Proceedings of the 27th International Conference on International Conference on Machine Learning. pp. 807–814. ICML'10 (2010)
[10] Ruder, S.: An overview of gradient descent optimization algorithms. CoRR **abs/1609.04747** (2016), http://arxiv.org/abs/1609.04747
[11] Zhong, J., Li, W.: Predicting customer call intent by analyzing phone call transcripts based on cnn for multi-class classification. 8th International Conference on Soft Computing, Artificial Intelligence and Applications (Jun 2019). https://doi.org/10.5121/csit.2019.90702