Incident and Problem Ticket Clustering and Classification Using Deep Learning

FENG Hailin[1], HAN Jing[2], HUANG Leijun[1], SHENG Ziwei[3], GONG Zican[2]

(1. Zhejiang A&F University, Hangzhou 310007, China;
2. ZTE Corporation, Shenzhen 518057, China;
3. Huazhong University of Science and Technology, Wuhan 430074, China)

DOI: 10.12142/ZTECOM.202304009

https: //kns.cnki.net/kcms/detail/34.1294.TN.20231115.1421.002.html, published online November 16, 2023

Manuscript received: 2022-12-19

Abstract: A holistic analysis of problem and incident tickets in a real production cloud service environment is presented in this paper. By extracting different bags of words, we use principal component analysis (PCA) to examine the clustering characteristics of these tickets. Then K-means and latent Dirichlet allocation (LDA) are applied to show the potential clusters within this Cloud environment. The second part of our study uses a pre-trained bidirectional encoder representation from transformers (BERT) model to classify the tickets, with the goal of predicting the optimal dispatching department for a given ticket. Experimental results show that due to the unique characteristics of ticket description, pre-processing with domain knowledge turns out to be critical in both clustering and classification. Our classification model yields 86% accuracy when predicting the target dispatching department.

Keywords: problem ticket; ticket clustering; ticket classification

Citation (Format 1): FENG H L, HANG J, HUANG L J, et al. Incident and problem ticket clustering and classification using deep learning [J]. ZTE Communications, 2023, 21(4): 69 - 77. DOI: 10.12142/ZTECOM.202304009
Citation (Format 2): H. L. Feng, J. Han, L. J. Huang, et al., "Incident and problem ticket clustering and classification using deep learning," *ZTE Communications*, vol. 21, no. 4, pp. 69 - 77, Dec. 2023. doi: 10.12142/ZTECOM.202304009.

1 Introduction

For cloud service providers, maintaining an outstanding service level agreement with minimum downtime and incident response time is critical to the business. In order to provide such a prominent high-level reliability and availability, IT operation plays an important role. However, the emergence of modern computing architectures, such as virtual machines, containers, server-less architecture, and micro-services, brings additional challenges to the management of such environments[1 - 2].

Problem and incident tickets have been a long-standing mechanism in carrying on any issues reflected by customers, or any alerts raised by monitoring systems. According to the Information Technology Infrastructure Library (ITIL) specification, the incident, problem, and change (IPC) systems fulfill the tracking, analysis, and mitigation of problems[3]. Change requests are nowadays mostly managed differently due to the practice of DevOps. Incident and problem tickets often share the same system and process. An incident or problem ticket usually starts with a short description of the problem that has been originally observed. The ticket itself may be augmented by the personnel assigned along the debugging and resolution process. There are also multiple software platforms and services to help enterprises manage those tickets, including BMC Remedy, IBM Smart Cloud Control Desk, SAP Solution Manager, ServiceNow, etc.[4]

However, dispatching an incident or problem ticket is still basically a manual process depending on human knowledge. Some of the ticket management systems offer insights such as agent skill level, capacity, and relevance. There are some early works attempting to dispatch tickets based on the agent's speed from historical data[5]. Our observation reveals that dispatching to individual agents might be a secondary issue. Instead, finding the matching department for a specific issue appears to be a primary one especially if a prompt resolution period is the desired outcome. It is not uncommon for some tickets to go through multiple departments before it lands on the right one. For example, a service unavailable problem might be caused by security settings, networking, hosting services, applications, or even databases, and the specific problem may be resolved by one of the departments or by multiple departments. Therefore, it is essential to find the most likely department, es-

pecially at the beginning when the problem was initially reported to resolve the issue efficiently. The specific technical challenge of classifying an early ticket is that the only available feature is problem description.

2 Related Work

Since the day when computer systems were created, IT operation has been a critical issue. With the prevalence of online services, in order to minimize system downtime and maintain premium service level agreements, IT operation plays a central role in achieving such a goal. Especially in today's highly distributed multi-layered cloud environment, it is untrivial to effectively find the matching departments to resolve the issue.

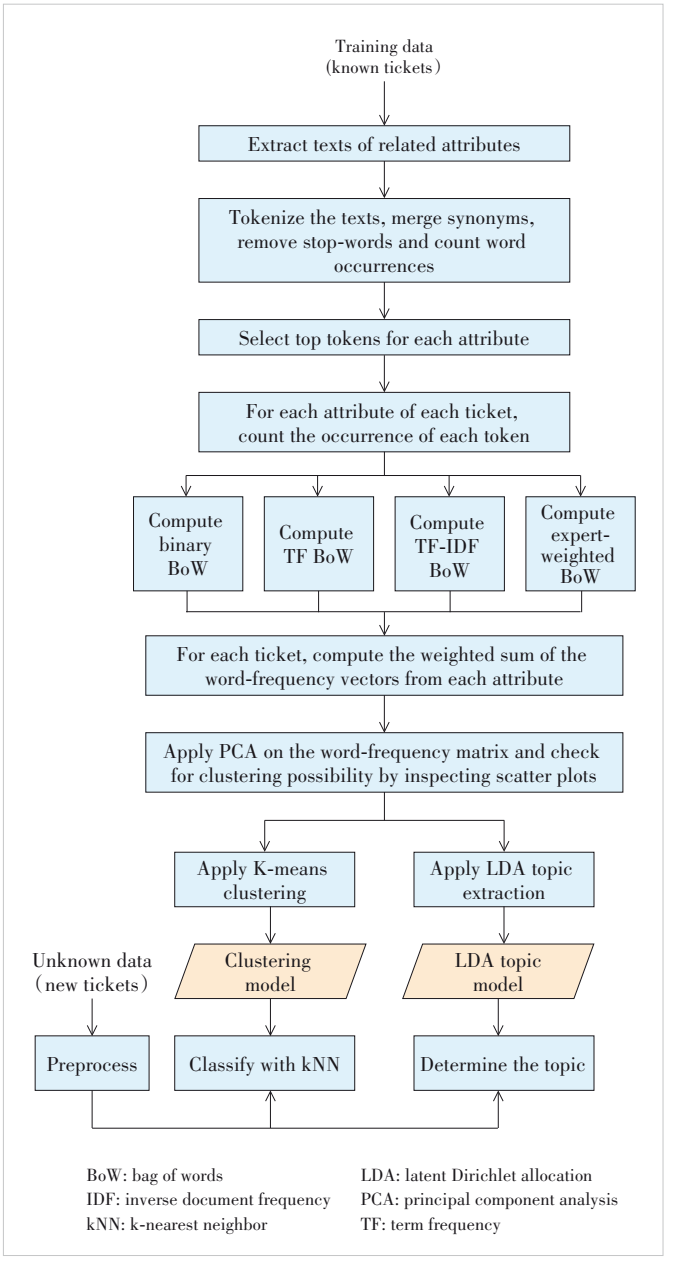
Artificial intelligence has been applied in IT operations, especially in anomaly detection[11 - 12], problem troubleshooting[13 - 14], and security[15 - 16]. A few works have attempted to improve the efficiency of ticket dispatching. BOTEZATU et al.[5] tried to find the most cost-effective agent for ticket resolution, rather than finding a matching group or department. SHAO et al.[17] focused on the transfer information in ticket resolution and formulated a model based on prior resolution steps. AGARWAL et al.[18] used a supported vector machine and a discriminative term to predict the matching department. While we use ticket descriptions and other attributes to find the best department, our solution is quite different from the previous works.

In terms of ticket analysis, there are only a few works on alerts or ticket clustering. LIN et al.[19] used graph theory and similarity measures as Jaccard as the cluster mechanism. MANI et al.[20] proposed a technique combining latent semantic indexing and a hierarchical *n*-gram algorithm. AGARWAL et al.[21] used a mixture of data mining, machine learning, and natural language parsing techniques to extract and analyze unstructured tests in IT tickets. JAN et al.[22] proposed a framework for text analysis in an IT service environment. We examine the clustering characteristics to discover the content of the ticket descriptions specific to the system under investigation. Our approach is generic to all systems with minor adjustments of synonyms and user dictionaries. When it comes to clustering itself, we believe our dataset is also unique as it is from the latest container-based cloud environment which is more complicated than prior systems.

3 Design of Clustering System

We apply different topic modeling algorithms to cluster the tickets based on their descriptions and compare their performance by calculating their sum of square error (SSE) and silhouette scores. The clustering results indicate the number of major topics in the ticket description corpus. Since it is an unsupervised learning process, it saves great effort from data annotation. For ticket classification, word embedding models have shown much better performance. Therefore, we only adopt the supervised approach using a pre-trained BERT model[6] which is fine-tuned with domain-specific labeled data.

Fig. 1 illustrates the overall steps we perform ticket description clustering. First, data preprocessing is performed by extracting texts, merging synonyms, removing stop words, etc. After tokenizing the texts, we construct 4 types of bags of words (BoW), including binary BoW, term frequency (TF) BoW, term frequency inverse document frequency (TF-IDF)[7] BoW, and expert-weighted BoW. For each of the BoW, we apply principal component analysis (PCA) to check for clustering possibility and use K-Means to cluster the topics. We also perform latent dirichlet allocation (LDA)[8] for topic extraction and modeling.



▲**Figure 1. Data analysis flow**

Finally, we show some of the sample topics in the cluster.

4 Experiments

We use two datasets from an enterprise-scale cloud provider, comprising 468 infrastructure-level and 787 Platform as a Service (PaaS) -level incident tickets, respectively. Since both datasets have similar data formats, we use the same analysis methods, which are mainly unsupervised machine learning approaches such as K-means and LDA. Our goal is to learn and make use of the inherent homogeneity of the complicated ticket descriptions by analyzing them.

For model training, we use the number, title or subject, and description from the datasets, in which the title or subject is a summary of the incident, and the description is a detailed text describing the problem. Some of the description texts are in the semi-structured form. For example, more than half the infrastructure-level ticket descriptions consist of explicit attributes like symptoms, progress, network topology, conclusions, steps, and remarks. We focus on the symptom attribute rather than using the entire text body since prediction needs to be performed when the ticket only has a symptom description. Some of the corpus such as file names, URL links, and system logs are filtered as part of preprocessing.

4.1 Data Preprocessing

We extract the text of the symptom attribute from the ticket description. If the description does not contain an explicit attribute of "symptom", the whole text is used. For the symptom text, we utilize regular expressions to filter unwanted data like picture-attached file name, date, time, URL and also delete the system logs as many as possible. We also perform spell checking using a dictionary.

Our next step is to convert the symptom texts into individual word tokens. Since most of the incident descriptions are a mixture of both Chinese and English, we use different tokenization tools for each language. "Jieba" is used for Chinese and "spaCy" for English. We also remove stop words from the output token and merge synonyms, e.g., "db" and database are the same, so they are uniformly replaced by a database. The lists of stop words are from Baidu[9] and github[10]. We merge both and extend some ticket-specific stop words for the experiments.

Given that some titles are similar to the symptom in terms of interfering texts and marks, they are preprocessed in the same way. The process described above ultimately generates a list of most frequently used tokens in both the title and symptom token lists. We sample high frequency Chinese and English words from the results, which are shown in Table 1.

▼Table 1. Sample of high frequency words

Keywords	Frequency	Keywords	Frequency
node	538	tecs	328
defect	479	provider	198
version	463	daisy	150
symptom	329	nova	149
upgrade	317	dvs	139
alert	309	compute	123
conclusion	291	cinder	90
description	282	neutron	90
progress	275	sdn	79
operation	258	error	76
topology	256	host	69
note	253	nfv	69
cause	252	ip	64
site	235	agent	64
failure	229	ceph	62

4.2 Clustering Using BoW Models

First, we study the clustering characteristics of the incident tickets using the BoW model. For the tokens we extract during preprocessing, we choose the top high frequency words for title and symptom respectively. We combine the tokens from title and symptom based on a predefined weight so that each ticket is transformed into a word frequency vector, and accordingly, the dataset is represented by a word frequency matrix.

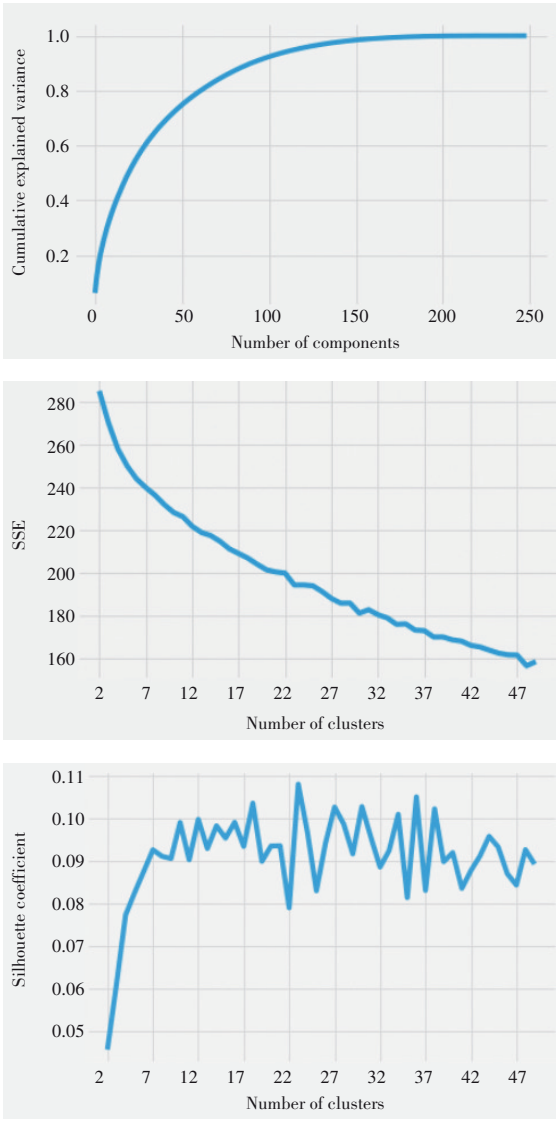
We apply principal component analysis (PCA) to the normalized word frequency matrix of the dataset, aiming to select the number of appropriate components using cumulative explained variance results. For example, Fig. 2 shows the PCA results of the symptom word frequency matrix, indicating that if 100 components are selected, and more than 90% of the variance can be explained.

After the number of the principal components is selected, we project the word frequency matrix to these components and use K-means for clustering analysis. For a given range of cluster numbers (i.e., values of K), we generate SSE and silhouette coefficient curves. As the best practice, the number of clusters is determined at the inflection point of the downward trend SSE curve or at the point when the upward trend silhouette coefficient curve becomes a plateau. The results are shown in Fig. 2. The SSE curve does not show an obvious inflection point, and the absolute value of the silhouette coefficient is too small even though the trend meets the demand (silhouette coefficient is between −1 and 1. The closer it is to 1, the more reasonable the clustering is). We conclude that it may not be a viable approach to evaluating the best cluster size by using PCA.

We also perform experiments using other models such as TF-IDF to generate a word frequency matrix, and the results are similar to PCA, indicating the word frequency matrix may not apply to incident tickets.

4.3 Clustering Using Latent Dirichlet Allocation (LDA) Model

In this section, we use LDA to extract dominant topics from

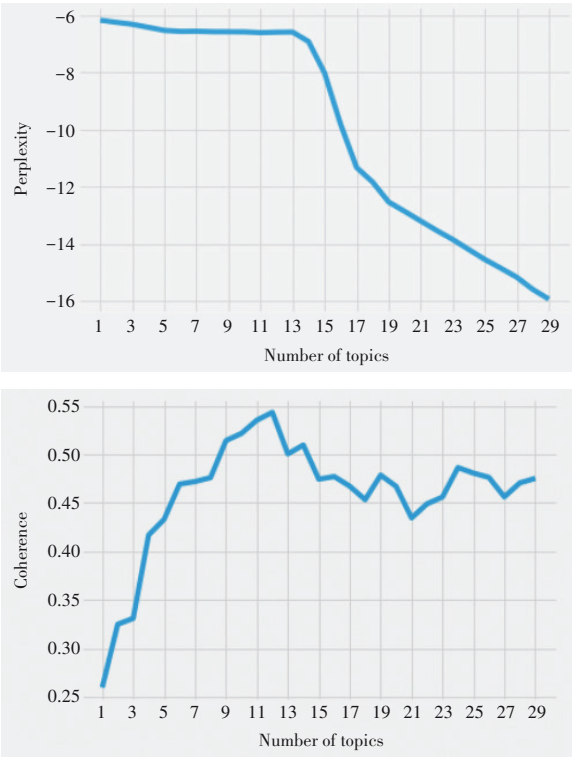


SSE: sum of square error

▲Figure 2. Principal component analysis (PCA) results, K-means SSE, and K-means silhouette curves using bag of words (BoW) model

the topic, symptom, and the combined token list. To determine the performance of the optimal number of topics, we compare different perplexity scores and coherence scores when applying different topic numbers. We select the topic number at the inflection point where the perplexity curve or the coherence curve turns.

Fig. 3 shows the results of the cloud infrastructure ticket data using LDA with 1 – 30 topics. Based on the characteristics of the curves, we can select the number of topics to be 14. The top ten keywords and the probabilities for each of the topics are shown in Table 2.



LDA: latent dirichlet allocation

▲Figure 3. LDA perplexity and coherence curves

▼Table 2. Topics and keywords in each topic

Topic	Keywords and Probabilities in Each Topic									
Topic 0	interface	daisy	NIC	file	maintain	virtualization	platform	zone	mode	increase
	0.107	0.055	0.034	0.033	0.025	0.023	0.023	0.022	0.021	0.020
Topic 1	OS	start	capacity	install	stack	service	blade	VM	mac	down
	0.113	0.090	0.072	0.054	0.029	0.027	0.025	0.017	0.017	0.017
Topic 2	provider	performance	device	login	director	memory	query	recover	bandwidth	occupy
	0.218	0.048	0.038	0.037	0.037	0.031	0.029	0.026	0.021	0.019
Topic 3	dvs	gateway	project	support	manage	business	resource	pool	add	thread
	0.081	0.041	0.038	0.038	0.033	0.032	0.031	0.028	0.027	0.023
Topic 4	network	blocked	power on	udm	update	management	information	change	packet loss	endpoint
	0.148	0.061	0.035	0.032	0.03	0.029	0.024	0.023	0.022	0.013

NIC: network interface card VM: virtual machine

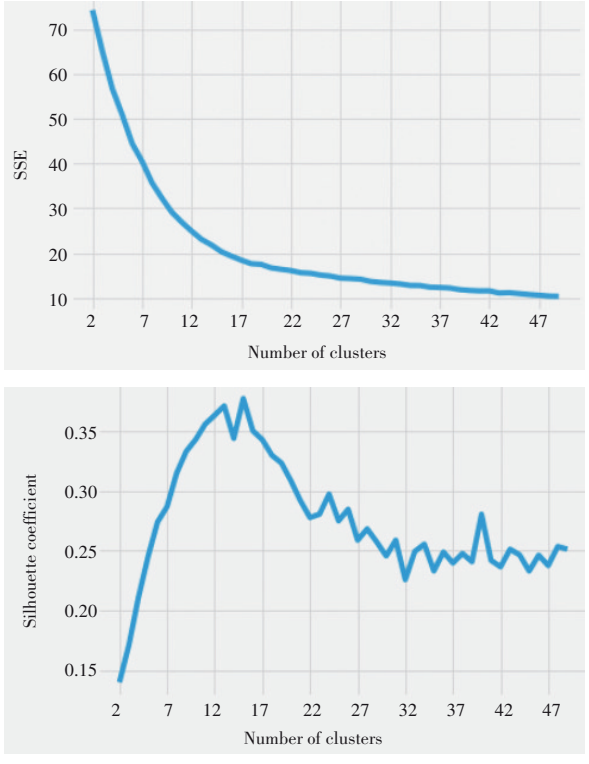
We evaluate the probability of topic appearance for each incident ticket and then cluster the topics using *K*-means. Fig. 4 shows the SSE curve and silhouette coefficient curve respectively. The curves demonstrate more significant turning points than the ones using the BoW model, indicating the LDA model is more suitable for incident ticket clustering.

With 14 topics, the SSE of LDA-allocated tokens is about 20, while the SSE of BoW is about 215, an order of higher magnitude. Though it is unpragmatic to map the SSE score to the exact accuracy, the lower the score the more accurate the prediction. Similarly, the silhouette coefficient for LDA results with 14 topics are about 0.37, compared with less than 0.10 using BoW models. As the score measures how apart of the cluster ranging from −1 to 1, a value close to 1 indicates clearly distinguished clusters.

Table 3 shows the titles of tickets in one cluster. Storage related problems consist of a majority of the tickets, especially during upgrade and backup stages. The next is networking issues.

4.4 Incident Ticket Classification and Prediction

Our ticket clustering experiments reveal that incident tickets do have clustering characteristics. In order to take full advantage of prior knowledge, e.g., to assign coming tickets to the same department which has resolved similar ones before, we study the classification and prediction of incident tickets in this section. We use a similar dataset with more fields, including ticket ID, ticket description, resolution, resolution groups, categories, sub-categories, and components. After removing null values, the categories and record numbers are shown in Table 4.



SSE: sum of square error

▲Figure 4. K-means SSE and silhouette curves using latent Dirichlet allocation (LDA) model

▼Table 3. Samples of title descriptions in one cluster

ID	Title Description	ClusterID
BC20200229-0052	Backup of version 6.10.10.08 failed during upgrading	5
BC20200307-0094	Problems of 3.17.15.06P2 trusted resource pool	5
BC20200309-0105	Backup of 6.10.10.P8tecs6.0 environment failed	5
BC20200402-0034	Nova service failed due to version 3.17.15.06 license problem during vHSS capacity increase	5
BC20200404-0051	Provider MariaDB failed to start after rebooting one of the control node in group one	5
BC20200409-0023	Keystone service abnormal in the 5GC test node in 3.17.14 control environment	5
BC20200411-0045	Two of the disks failed when batch creating 32 35T cloud disks reporting not sufficient space	5
BC20200507-0012	Mutual trust failed during Northeast 3.17.15.06P4 upgrading	5
BC20200511-0058	Part of the related information not updated after changing configuration of 3.17.15.06P3 on Daisy	5
BC20200520-0066	One of the VMs failed during start reporting volume not found after NFVINMA1Z station upgrade	5
BC20200603-0025	Failure of V03.17.15.06P4HP3 upgrading	5
BC20200603-0027	Multi-node upgrade failed due to 3.17.15.06P4 17.15.06P4HP3 mutual trust lost	5
BC20200618-0066	3.17.15.07_T2-daisy upgrade failed	5
BC20200629-0247	V3.17.14.P2Provider auto-backup service hang	5
BC20200630-0257	Network issues from two VMs on 3.17.15.06P4HP3	5
BC20200702-0060	Mirror file upload failed after V03.17.15.07T2-Provider upgrade	5
BC20200705-0002	Nova service down after 3.17.15.06P4HP3 upgrade	5
BC20200710-0121	3.17.15.08-OS distribution failed reporting mutual-trust issue	5
BC20200710-0123	40 VMs in shutoff status using 3.17.15.08 startup script	5
BC20200716-0095	Not able to apply new license after 3.17.15.06P6-license failed	5
BC20200722-0207	Tenant resource abnormal after tenants with the same name created	5

vHSS: virtual home subscriber server VM: virtual machine

There are 115 sub-categories and 49 of them contain 1 000 records or more. The 49 sub-categories consist of 96% of the total tickets, and 30 of them contain 3 000 records or more consisting of 87% of the total records. When it comes to components, there are 663 in total, among which there are 88 items with more than 1 000 records accounting for 79% of the total amount, and 34 items with more than 3 000 records ac-

▼Table 4. Categories and record numbers

Category	Number of Records
Infrastructure	177 040
Operation product line	64 869
OA product line	55 570
EPMS intelligent service	22 454
iCenter application	19 849
PLM product line	15 870
AIOps group	14 121
Technical platform	1 232
Others	171
IT Wizard	19
Operation NOC	17
Network	5
Communication	2
Middleware	1
Security	1

AIOps: artificial intelligence for IT operations
EPMS: enterprise performance management system
NOC: network operations center
OA: office automation
PLM: product lifecycle management

counting for 54% of the total amount.

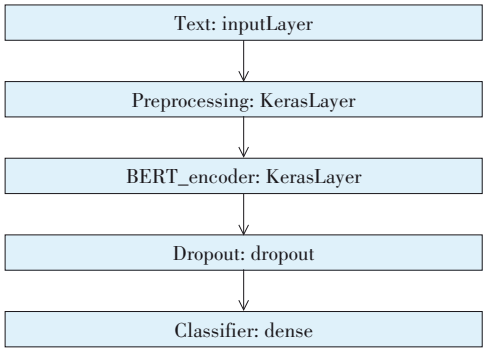
In order to achieve fine granularity of the classification, we use the combination of sub-categories and components as the label. There are 29 top labels with more than 3 000 records.

We compare multiple classification algorithms including TF-IDF, LDA and BERT. As expected, BERT achieved the best precision and recall for the same dataset. Both TF-IDF and LDA with the regression model yield a prediction accuracy of less than 80%. We build the incident classification model based on BERT which is shown in Fig. 5.

1) Architecture of our model

• The input layer is a text layer with preprocessed incident description text.

• The preprocessing layer is a Chinese processing model devised by Google (suited for the BERT model). Every ticket text is transformed into 3 vectors: input_word_ids, input_mask and input_type_ids with 128 dimensions respectively. Input_word_ids denotes the ID of the word. The lost elements of input_word_ids vector are filled with 0. For the corresponding numbers in an input_mask vector, they should be 1 while the remaining elements are 0. Input_type_ids can clarify different sentences. In this classification study, we set all of its elements to 0.



▲Figure 5. BERT classification network architecture

• BERT_encoder is an advanced BERT model devised by Google. BERT_encoder has 12 layers (bert_zh_L-12_H-768_A-12l) and the output of the BERT_encoder consists of pooled_output (each text corresponds to a vector of 768 elements), sequence_output (each word in each text corresponds to a vector of 768 elements) and encoder_outputs (output of inner units). We only focus on pooled_output in this experiment.

• The dropout layer aims at avoiding overfitting. The probability of dropout is set to 0.1.

• The classifier layer is a fully connected layer that outputs the probability of each ticket belonging to a certain classification in the labels.

2) Training and testing data preparation

We use the following steps as data preprocessing to generate training and testing data:

• Delete all the incident tickets containing null value category information or empty ticket descriptions.

• Modify the classification labels into lowercase and delete the redundant blank space. This operation is devised from observing the original data, where some categories and items are generally the same but only differ in lowercase and uppercase. For example, iCenter and Icenter.

• Delete tickets with ambiguous items and category labels like "other, others, to be classified, and other pending problems".

• Merge the item and category labels in the form of component. category such as intelligent maintenance.itsp serve website.

• After the merging operation, delete labels and their incident tickets data whose statistic number is less than the threshold (we set 3 000 in this experiment).

• Remove HTML formatting and redundant space (including line feed punctuation) from the incident description texts. For the English content, all the letters are also put in lowercase.

• Shuffle the resulting incident data. 70% of the dataset is utilized as the training set and the remaining 30% is used as the test set.

• Each classification label and its quantity of relevant incident tickets are given in Table 5 (29 classification labels with more than 3 000 records respectively are reserved).

As a result, 103 094 incident tickets are identified as training data and 44 183 incident tickets are collected as test data.

For training the model, we adopt the Sparse Categorical Crossentropy as the loss function, Sparse Categorical Accuracy for accuracy measurement and optimize the model with AdamW. The experiment sets the initial learning rate to 3e−5 and the epoch to 5. The original training data are partitioned

▼Table 5. Top labels (combination of sub-categories and components) and record numbers

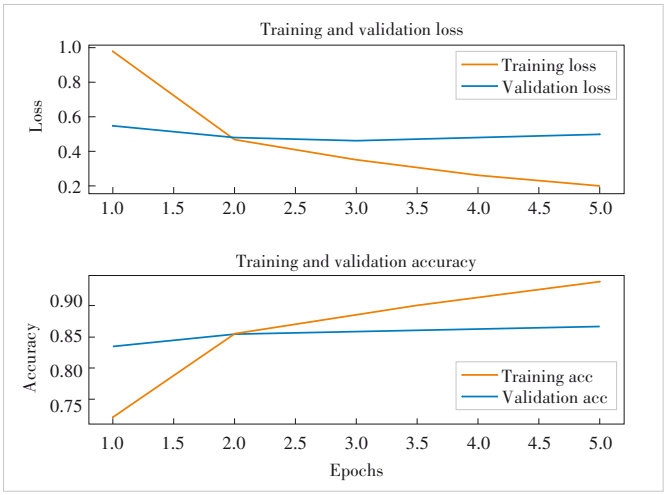
Classification Label	Numbers of Records	ID
AIOps - itsp service website	16 724	0
desktop cloud - linux desktop cloud	11 222	1
desktop cloud - OS issues	7 162	2
PC side zmail - operation issues	7 110	3
ifol finance - oerp enterprise resource planning	6 543	4
desktop cloud - intranet client-side login	6 531	5
ifol finance - fol finance online	6 381	6
network proxy - usage issues	5 511	7
iscp supply chain - iwms.wms cloud storage	5 086	8
iccp customer relationship - msm marketing	4 994	9
iccp customer relationship - ccg contract configuration	4 779	10
PC side zmail - account issues	4 622	11
iscp supply chain - iwms.mcs manufacture management	4 207	12
im instant message - usage issues	4 156	13
PC side zmail - account creation & login	4 043	14
ifol finance - cms contract management web	4 019	15
uds failure - security check	3 930	16
ifol finance - cms contract management form	3 886	17
icenter - ts team coordination	3 878	18
desktop cloud - blue or black screen	3 872	19
engineering domain - cca cloud code review	3 584	20
desktop cloud - client side login failed	3 551	21
OS issues - installation	3 531	22
ibcp human resource - hol online	3 258	23
iscp supply chain - isrm.srm supplier management	3 254	24
Mobil application - icenter Mobile side	3 234	25
individual network issue - restriction	3 213	26
ibcp human resource - elearning academy	3 178	27
uds failure - usage issues	3 008	28

into a training set and a validation set at the ratio of 9∶1 in this pre-training procedure (i.e., the number of incident tickets used for model training is the number of preprocessed incident tickets × 70% × 90%).

Fig. 6 shows the training loss, training accuracy, validation loss, and validation accuracy of each epoch.

To verify our model after pretraining, we perform classification tasks on the test set. The assessment results are illustrated in Table 6. The overall precision is up to 86%. The confusion matrix of prediction results is shown in Table 7. The number in cell (i, j) denotes the number of tickets, the labels of which are i but predicted to be j in this model. Therefore, the numbers of correctly classified incident tickets lie on the diagonal while the number lying off the diagonal shows the discrepancies in classification.

$$\frac{N(i,j)}{\sum_{k=0}^{28} N(i,k)} \geqslant 5\%, i \neq j . \quad (1)$$



▲Figure 6. Training loss and accuracy vs validation loss and accuracy

▼Table 6. Prediction accuracy on test data

	Precision	Recall	F1-score	Support
0	0.91	0.91	0.91	4 951
1	0.85	0.85	0.85	3 289
2	0.67	0.65	0.66	2 071
3	0.80	0.85	0.83	2 146
4	0.88	0.88	0.88	1 931
5	0.73	0.75	0.74	1 974
6	0.93	0.93	0.93	1 872
7	0.91	0.93	0.92	1 619
8	0.93	0.92	0.93	1 543
9	0.91	0.93	0.92	1 470
10	0.91	0.90	0.91	1 500
11	0.84	0.80	0.82	1 381
12	0.90	0.91	0.91	1 231
13	0.91	0.93	0.92	1 266
14	0.80	0.82	0.81	1 188
15	0.80	0.80	0.80	1 206
16	0.89	0.90	0.90	1 228
17	0.79	0.77	0.78	1 159
18	0.93	0.94	0.93	1 119
19	0.67	0.71	0.69	1 173
20	0.95	0.95	0.95	980
21	0.87	0.89	0.88	1 046
22	0.89	0.88	0.89	1 060
23	0.90	0.89	0.90	925
24	0.91	0.91	0.91	978
25	0.95	0.93	0.94	994
26	0.87	0.77	0.81	960
27	0.95	0.96	0.95	955
28	0.79	0.69	0.74	968
Accuracy			0.86	44 183
Macro avg	0.86	0.86	0.86	44 183
Weighted avg	0.86	0.86	0.86	44 183

From Table 8, we can see that a majority of classification error cases occurs among different items of the same sub-categories. For example, it is confusing for the model to classify the items of some sub-categories like desktop cloud, PC side zmail incidents, ifol finance and uds failure. In addition, 7.4% tickets of OS issues - installation are wrongly predicted

▼Table 7. Confusion matrix of prediction results

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28
0	4 520	5	6	22	4	14	8	20	1	1	7	41	2	28	42	5	16	1	26	2	0	5	58	24	7	35	13	4	34
1	12	2 789	166	24	1	118	1	16	1	3	0	0	1	26	16	1	0	0	4	60	4	13	5	1	4	0	7	4	12
2	8	216	1 352	6	0	233	0	2	0	2	1	0	1	5	2	1	2	1	0	203	1	15	8	0	0	0	7	0	5
3	17	21	6	1 826	1	1	3	5	0	1	2	139	1	1	106	1	0	1	6	1	1	0	1	1	0	0	0	3	1
4	2	2	0	0	1 706	0	49	0	19	2	2	1	56	1	1	37	0	33	0	0	7	0	0	4	6	0	1	1	1
5	10	70	242	2	0	1 473	0	0	0	0	1	0	0	1	2	0	3	0	1	113	2	47	0	0	0	0	6	0	1
6	6	5	0	3	61	0	1 745	1	4	5	3	1	2	0	0	9	0	2	3	0	2	0	0	6	9	0	0	3	2
7	21	27	1	3	0	0	2	1 510	2	0	2	0	0	1	4	1	1	0	5	0	3	1	1	1	2	0	10	5	16
8	4	1	1	2	22	0	8	2	1 421	1	6	1	8	0	0	15	1	11	4	0	0	0	0	0	27	1	1	2	1
9	7	0	1	2	2	0	6	1	3	1 374	39	1	0	1	1	7	0	5	5	0	1	0	2	7	3	0	0	2	0
10	5	2	1	3	3	0	2	3	2	66	1 352	0	5	3	1	7	2	30	1	0	7	0	1	0	2	0	0	1	1
11	28	6	1	196	1	1	1	0	1	0	1	1 103	1	1	27	0	1	1	2	0	1	0	0	1	2	1	1	2	1
12	6	3	2	0	29	0	2	0	25	3	4	0	1 117	1	0	4	0	12	1	0	4	0	3	4	8	0	2	0	1
13	14	24	3	7	1	0	3	1	0	1	0	3	2	1 173	2	4	0	1	1	0	1	1	4	5	4	2	2	2	5
14	31	16	2	125	0	5	1	2	0	0	1	12	0	3	969	0	0	0	2	0	1	2	0	3	1	2	4	0	6
15	1	0	0	3	51	0	7	1	10	13	7	0	1	3	0	966	0	132	6	0	1	0	0	0	3	0	0	1	0
16	30	4	6	2	1	4	0	1	0	0	0	2	2	3	0	0	1 107	1	0	0	3	1	4	1	1	0	4	1	50
17	0	0	0	0	40	0	0	0	15	13	29	1	23	1	1	140	0	892	0	0	2	0	0	0	1	0	1	0	0
18	19	0	1	12	2	0	6	1	2	1	0	4	1	3	1	0	0	2	1 050	0	1	3	0	4	2	2	0	1	1
19	1	42	174	4	0	93	0	0	0	1	0	0	1	2	0	0	0	0	0	832	1	17	3	0	0	0	2	0	0
20	5	9	1	6	3	0	2	3	2	2	4	1	2	3	0	0	2	1	1	0	928	0	0	2	1	0	2	0	0
21	7	11	15	1	0	50	0	3	1	0	0	0	0	1	2	0	0	0	0	17	0	926	2	1	0	1	4	0	4
22	78	5	9	2	2	1	0	0	0	1	2	0	3	2	1	0	6	1	0	6	0	1	933	0	1	0	2	1	4
23	42	2	0	2	2	1	12	0	0	3	2	3	0	6	0	0	0	0	7	0	0	1	0	826	5	5	2	2	3
24	6	4	0	4	6	1	12	1	16	5	9	1	7	2	2	2	0	5	4	0	0	0	2	1	886	1	0	1	0
25	35	5	0	3	0	0	0	0	0	0	1	1	0	3	1	0	0	0	4	0	0	0	1	7	1	927	0	4	1
26	25	9	18	3	1	28	1	71	0	2	0	1	0	1	6	1	2	1	1	6	5	5	6	2	0	0	736	3	26
27	4	4	3	4	0	1	5	2	0	1	2	2	0	1	1	0	0	0	0	0	0	0	1	6	1	0	3	914	0
28	44	11	5	7	1	5	6	6	1	1	1	1	1	12	17	0	99	0	1	2	1	22	8	6	0	1	40	3	666

▼Table 8. Sample labels of classification error

Ground Truth Category/Erroneous Category	Total Samples	Error Samples	
desktop cloud - linux desktop cloud	3 289		
• desktop cloud - OS issues		166	5.0%
desktop cloud - OS issues	2 071		
• desktop cloud - linux desktop cloud		216	10.4%
• desktop cloud - intranet client-side login		233	11.3%
• desktop cloud - blue or black screen		203	9.8%
PC side zmail - operation issues	2 146		
• PC side zmail - account issues		139	6.5%
desktop cloud - intranet client-side login	1 974		
• desktop cloud - OS issues		242	12.3%
• desktop cloud - blue or black screen		113	5.7%
PC side zmail - account issues	1 381		
• PC side zmail - account creation & login		196	14.2%
PC side zmail - account creation & login	1 188		
• PC side zmail - operation issues		125	10.5%
ifol finance - cms contract management web	1 206		
• ifol finance - cms contract management form		132	10.9%
ifol finance - cms contract management form	1 159		
• ifol finance - cms contract management web		140	12.1%
desktop cloud - blue or black screen	1 173		
• desktop cloud - OS issues		174	14.8%
• desktop cloud - intranet client-side login		93	7.9%
OS issues-installation	1 060		
• AIOps - itsp service website		78	7.4%
individual network issue - restriction	960		
• network proxy - usage issues		71	7.4%
uds failure-usage issues	968		
• uds failure - sercurity check		99	10.2%

AIOps: artificial intelligence for IT operations

to be AIOps-itsp service website and 7.4% tickets of individual network issues – restriction are wrongly predicted to be network proxy-usage issues.

5 Conclusions

In this paper, we demonstrate the semantic characteristics of problem and incident tickets. Taking the ticket data from a real production Cloud environment, we compare different text mining techniques. LDA and K-Means are applied to show the ticket clusters. We use BERT as the deep learning framework with fine-tuning to build a resolution department matching system. Using sub-category and component fields in the ticket description, our classification model achieves 86% accuracy when predicting the best match department to resolve the ticket.

Reference

[1] FORELL T, MILOJICIC D, TALWAR V. Cloud management: challenges and opportunities [C]//IEEE International Symposium on Parallel and Distributed Processing Workshops and PhD Forum. IEEE, 2011: 881 – 889. DOI: 10.1109/IPDPS.2011.233
[2] MARTIN-FLATIN J P. Challenges in cloud management [J]. IEEE cloud computing, 2014, 1(1): 66 – 70. DOI: 10.1109/MCC.2014.4
[3] PEREIRA R, DA SILVA M M. Towards an integrated IT governance and IT management framework [C]//The 16th International Enterprise Distributed Object Computing Conference. IEEE, 2012: 191 – 200. DOI: 10.1109/EDOC.2012.30
[4] FARIA N E, SILVA M M. Selecting a software tool for ITIL using a multiple criteria decision analysis approach [EB/OL]. [2022-12-10]. https://aisel.aisnet.org/cgi/viewcontent.cgi?article=1241&context=isd2014
[5] BOTEZATU M M, BOGOJESKA J, GIURGIU I, et al. Multi-view incident ticket clustering for optimal ticket dispatching [C]//The 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. ACM, 2015: 1711 – 1720. DOI: 10.1145/2783258.2788607
[6] DEVLIN J, CHANG M W, LEE K, et al. BERT: Pre-training of deep bidirectional transformers for language understanding [EB/OL]. [2022-12-10]. https://arxiv.org/abs/1810.04805.pdf
[7] ROELLEKE T, WANG J. TF-IDF uncovered: a study of theories and probabilities [C]//The 31st annual international ACM SIGIR conference on Research and development in information retrieval. ACM, 2008: 435 – 442. DOI: 10.1145/1390334.1390409
[8] JELODAR H, WANG Y L, YUAN C, et al. Latent Dirichlet allocation (LDA) and topic modeling: models, applications, a survey [J]. Multimedia tools and applications, 2019, 78(11): 15169 – 15211. DOI: 10.1007/s11042-018-6894-4
[9] MO Z L. Stopwords [EB/OL]. (2019-12-18) [2022-12-10]. https://github.com/goto456/stopwords/blob/master/baidu_stopwords.txt
[10] LARS Y C. Stopwords [EB/OL]. (2011-12-06) [2022-12-10]. https://gist.github.com/larsyencken/1440509
[11] DU M, LI F F, ZHENG G N, et al. DeepLog: anomaly detection and diagnosis from system logs through deep learning [C]//The 2017 ACM SIGSAC Conference on Computer and Communications Security. ACM, 2017: 1285 – 1298. DOI: 10.1145/3133956.3134015
[12] ZHANG X, XU Y, LIN Q W, et al. Robust log-based anomaly detection on unstable log data [C]//Proceedings of the 2019 27th ACM Joint Meeting on European Software Engineering Conference and Symposium on the Foundations of Software Engineering. ACM, 2019: 807 – 817. DOI: 10.1145/3338906.3338931
[13] LIN J Y, ZHANG Q, BANNAZADEH H, et al. Automated anomaly detection and root cause analysis in virtualized cloud infrastructures [C]//IEEE/IFIP Network Operations and Management Symposium. IEEE, 2016: 550 – 556. DOI: 10.1109/NOMS.2016.7502857
[14] HARUTYUNYAN, A N, GRIGORYAN N M, POGHOSYAN A V, et al. Intelligent troubleshooting in data centers with mining evidence of performance problems [EB/OL]. (2020-09-20) [2022-12-10]. https://www.researchgate.net/publication/344251115_Intelligent_Troubleshooting_in_Data_Centers_with_Mining_Evidence_of_Performance_Problems
[15] BOU NASSIF A, ABU TALIB M, NASIR Q, et al. Machine learning for cloud security: a systematic review [J]. IEEE access, 2021, 9: 20717 – 20735. DOI: 10.1109/ACCESS.2021.3054129
[16] GRZONKA D, JAKÓBIK A, KOŁODZIEJ J, et al. Using a multi-agent system and artificial intelligence for monitoring and improving the cloud performance and security [J]. Future generation computer systems, 2018, 86: 1106 – 1117. DOI: 10.1016/j.future.2017.05.046
[17] SHAO Q H, CHEN Y, TAO S, et al. Efficient ticket routing by resolution sequence mining [C]//Proceedings of the 14th ACM SIGKDD international conference on knowledge discovery and data mining. ACM, 2008: 605 – 613. DOI: 10.1145/1401890.1401964
[18] AGARWAL S, SINDHGATTA R, SENGUPTA B. SmartDispatch: enabling efficient ticket dispatch in an IT service environment [C]//The 18th ACM SIGKDD international conference on Knowledge discovery and data mining. ACM, 2012: 1393 – 1401. DOI: 10.1145/2339530.2339744
[19] LIN D, RAGHU R, RAMAMURTHY V, et al. Unveiling clusters of events for alert and incident management in large-scale enterprise it [C]//The 20th ACM SIGKDD international conference on knowledge discovery and data mining. ACM, 2014: 1630 – 1639. DOI: 10.1145/2623330.2623360
[20] MANI S, SANKARANARAYANAN K, SINHA V S, et al. Panning requirement nuggets in stream of software maintenance tickets [C]//The 22nd ACM SIGSOFT International Symposium on Foundations of Software Engineering. ACM, 2014: 678 – 688. DOI: 10.1145/2635868.2635897
[21] AGARWAL S, AGGARWAL V, AKULA A R, et al. Automatic problem extraction and analysis from unstructured text in IT tickets [J]. IBM journal of research and development, 2017, 61(1): 41 – 52. DOI: 10.1147/JRD.2016.2629318
[22] JAN E E, CHEN K Y, IDÉ T. Probabilistic text analytics framework for information technology service desk tickets [C]//IFIP/IEEE International Symposium on Integrated Network Management (IM). IEEE, 2015: 870 – 873. DOI: 10.1109/INM.2015.7140397

Biographies

FENG Hailin received his PhD in computer science from the University of Science and Technology of China in June 2007. Since 2007, he has been working in the School of Information Engineering of Zhejiang A&F University, China. From 2013 to 2014, he was a visiting professor at Forest Products Laboratory, USA. He is currently a professor in the School of Mathematics and Computer Science and School of Information Engineering of Zhejiang A&F University. His main interest areas include computer vision, intelligent information processing, and Internet of Things.

HAN Jing (han.jing28@zte.com.cn) received her master's degree from Nanjing University of Aeronautics and Astronautics, China. She has been with ZTE Corporation since 2000, where she worked on 3G/4G key technologies from 2000 to 2016 and has become a technical director responsible for intelligent operation of cloud platforms and wireless networks since 2016. Her research interests include machine learning, data mining, and signal processing.

HUANG Leijun received his PhD in computer science from George Mason University, USA in 2008. Since 2010, he has been working in the School of Information Engineering of Zhejiang A&F University, China. He is currently a lecturer in the School of Mathematics and Computer Science. His main interest areas include computer networks, Internet of Things and data mining.

SHENG Ziwei received her BS degree in software engineering from Huazhong University of Science and Technology, China in 2022. She is currently pursuing her MS degree in electrical and computer engineering at Carnegie Mellon University, USA. In her master's program, she primarily focuses on the fields of engineering development and system design. Her ultimate goal is to advance technology and foster innovation in these domains.

GONG Zican received his master's degree in professional computing and artificial intelligence from the Australian National University in 2019. He has been a machine learning engineer in ZTE Corporation since 2020. His research interests include machine learning, professional computing and system architecture.