STAR: A System for Ticket Analysis and Resolution

Wubai Zhou[1], Wei Xue[1], Ramesh Baral[1], Qing Wang[1], Chunqiu Zeng[1], Tao Li[1,2],
Jian Xu[3], Zheng Liu[2], Larisa Shwartz[4], and Genady Ya. Grabarnik[5]
[1]School of Computing and Information Sciences, Florida International University, Miami, FL, USA; [2]School of Computer Science, Nanjing University of Posts and Telecommunications, Nanjing, China;
[3]School of Computer science and Engineering, Nanjing University of Science and Technology, Nanjing, China; [4]IBM T.J. Watson Research Center, Yorktown Heights, NY, USA; [5]Dept. Math & Computer Science, St. John's University, Queens, NY, USA

Introduction

In large scale and complex IT service environments, a problematic incident is logged as a ticket and contains the ticket summary (sys tem status and problem description). The system administrators log the step-wise resolution description when such tickets are resolved. The repeating service events are most likely resolved by inferring similar historical tickets. With the availability of reasonably large ticket datasets, we can have an automated system to recommend the best matching resolution for a given ticket summary.

In this work, we first identify the challenges in real-world ticket analysis and develop an integrated framework to efficiently handle those challenges.

Challenge 1. How to quantify the quality of the ticket resolution? Earlier studies generally assumed that the tickets with similar descriptions should have similar resolutions, and often treated all such ticket resolutions equally. However, the study [39] demonstrated that not all of the resolutions are equally worthy recommending. In order to develop an effective resolution recommendation model, low-quality resolutions should be ranked lower than high-quality resolutions.

Challenge 2. How to make use of the historical tickets along with their resolution quality for effective automation of IT service management? Although, it might be intuitive to search for historical tickets with the most similar ticket summary, and recommend their resolutions as potential solutions to the target ticket[39], such an approach might not be effective due to 1) the difficulty in representing the ticket summary and resolution, and 2) the avoidance of the resolution quality quantification.

Proposed Solutions

- Carefully identify relevant features and then build a regression model to quantify ticket resolution quality
- Train a deep neural network ranking model using tickets along with their quality scores obtained from the resolution quality quantification.
- Integrate sentence model into the neural network structure which can emit efficient representation for the ticket summary and resolution

Background

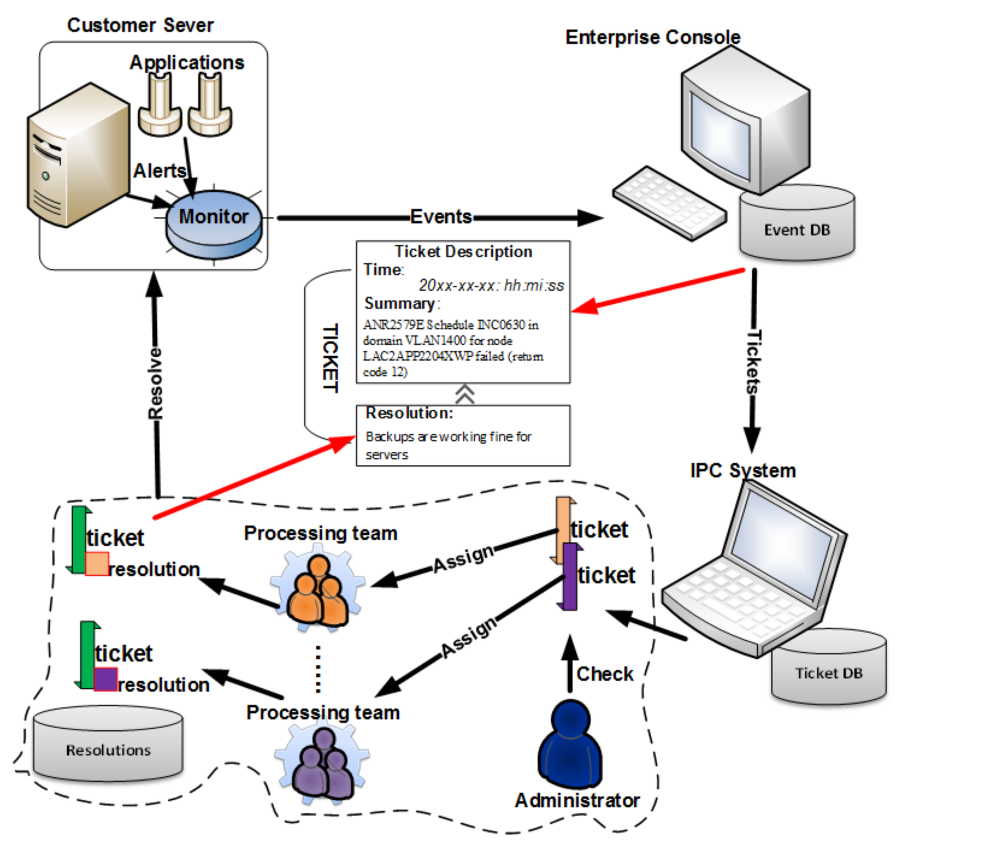

Figure 1: Information Technology Infrastructure Library (ITIL) Service Management System

A typical workflow of problem detection, determination and resolution in IT service management is prescribed by the Information Technology Infrastructure Library (ITIL) specification and is illustrated in Fig. 1.

1. A monitoring agent on a server keeps track of the system statistics and triggers an alert when a problem is detected.
2. If an alert persists beyond the specified duration, an event is triggered. Such events are consolidated into an enterprise console, which uses rule-based, case-based or knowledge-based engines to analyze the events and determines whether or not to create an incident ticket in the IPC system.
3. Incident ticket are manually or automatically assigned to domain experts for further system diagnosis and remedy.
4. The system administrators log the step-wise resolution description when such tickets are resolved

SEVERITY	FIRST-OCCURRENCE	LAST-OCCURRENCE
0	2014-03-29 05:50:39	2014-03-31 05:36:01
SUMMARY	ANR2579E Schedule INC0630 in domain VLAN1400 for node LAC2APP2204XWP failed (return code 12)	
RESOLUTION	Backups are working fine for the server.	
CAUSE	ACTIONABLE	LAST-UPDATE
Maintenance	Actionable	2014-04-29 23:19:25

Table1: A sample historical ticket

System Overview

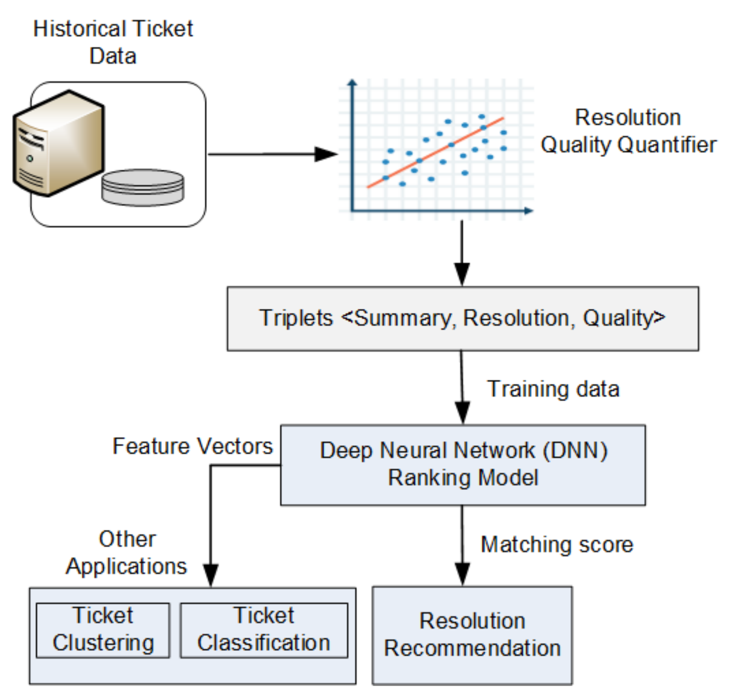

Figure 2: Overview of the proposed system

- The training data taken from the historical tickets dataset are first preprocessed in order to quantify and evaluate the quality of the resolution. The preprocessed result is then represented as a triplet of the ticket summary, its resolution text, and the quality score.
- These triplets are the training data for the proposed deep neural network (DNN) ranking model. The trained DNN model outputs a matching score of a quantified ticket resolution for an incoming ticket summary. The resolutions with the top N highest matching score can be recommended for an incoming ticket.
- The model's intermediate result is a feature vector for a ticket representation. Such vectors are used in other ticket analysis tasks, such as ticket classification and ticket clustering

Ticket Resolution Quality Quantification

Shown in Table 1, a ticket resolution is a textual attribute of a ticket. A high quality ticket resolution is supposed to be well written and informative enough to describe the detailed actions taken to fix the problem specified in the ticket summary.

We've found that for a typical ticket, the ticket resolution quality is driven by the 33 features that can be broadly divided into following four groups: 1) **Character-level features, Entity-level features, Semantic-level features, Attribute-level features**

Concept	Pattern	Examples
Action	NOUN/NP preceded/succeeded by VERB	(file) is (deleted)
Problem	NOUN/NP preceded/succeeded by ADJ/VERB	(capacity) is (full)

Evaluated three of the most popular regression models (logistic regression, gradient boosting tree and random forest[3]) on the labeled real-world ticket dataset and found that the random forest performed best for the ticket resolution quantification and also for evaluation of the feature importance, as illustrated in the Table 2

Feature Group	Importance score		
Feature	Rank	Mean	Variance
Character-features			
uppercaseRatio	12	0.026123	0.008717
lowercaseRatio	10	0.049657	0.008206
punctuationRatio	11	0.036442	0.008710
whitespaceRatio	9	0.049123	0.008610
Entity-level features			
servernameNumber	13	0.018770	0.008553
Semantic-level features			
VERBRatio	7	0.079400	0.009091
NOUNRatio	4	0.088025	0.009420
ADJRatio	14	0.013885	0.009048
ADVRatio	5	0.084971	0.008327
DETRatio	8	0.055133	0.008147
PRTRatio	3	0.090921	0.022932
PUNCTRatio	15	0.008797	0.008228
problemNum	6	0.080322	0.008480
actionNum	2	0.147252	0.038538
Attribute-level features			
resolutionLength	1	0.152234	0.043585
MSE Avg.	0.010269	**MSE Var.**	0.004163

Table 2: Illustration of the top 15 ranked features and their rank evaluated by the random forest regression model. To best evaluate the feature importance score, we show the rank of average importance score, its mean and variance. The best performance in the metric of both MSE (mean square error) average and variance is attached of the end.

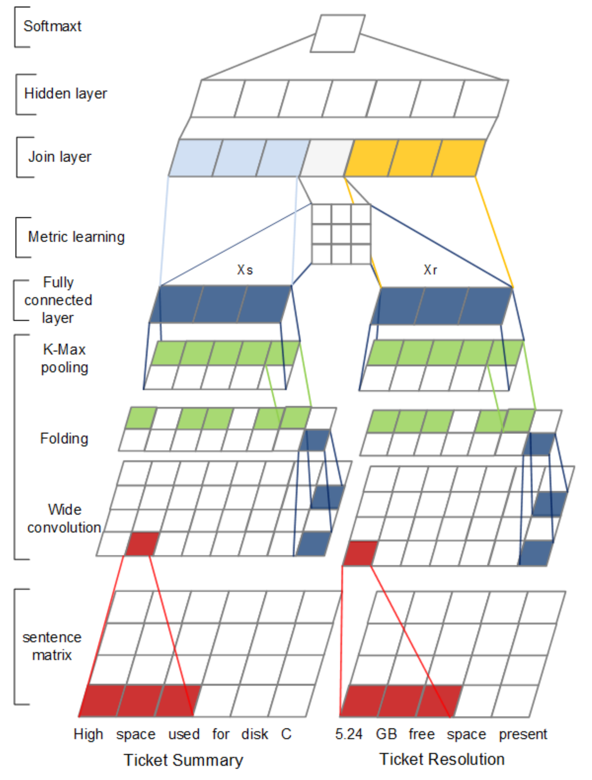

Figure 3: Ranking Model. The character level embedding is not shown for the sake of saving space.

Deep Neural Ranking Architecture

Given the triplets $\{<s_1, r_1, q_1>, <s_2, r_2, q_2>, \ldots, <s_n, r_n, q_n>\}$ from **Resolution Quality Quantifier** where s_i and r_i are ticket summary and ticket resolution for the ith ticket, and q_i is the quality score assigned by the quantifier.

Figure 3 shows the deep neural ranking model we proposed to solve the ticket resolution recommendation problem. The model consists of two sentence model[13] for mapping ticket summary and resolution to their vector representation, respectively.

Automating Ticket Resolution

Datasets

System	Training	Validation	Testing
Resolution Quality Quantifier	5000	–	1000
Ticket Resolution Automation	450,000	20,000	9,000
Ticket Clustering	10,000	–	2,000
Ticket Classification	20,000	–	3,000

System	p1	MAP	nDCG5	nDCG10
SMT	0.421	0.324	0.459	0.501
LSTM-RNN	0.563	0.367	0.572	0.628
Random Shuffle	0.343	0.273	0.358	0.420
CombinedLDAKNN	0.482	0.347	0.484	0.536
Our method	**0.742**	**0.506**	**0.628**	**0.791**

SMT: Statistical Machine Translation[28]

LSTM-RNN: sequence to sequence translation model[31]

CombinedLDAKNN[39]

Table 3: Overall Performance

Other Ticket Analysis Applications

Clustering

Category	Measure	Formula	Note
Surface Matching Similarity	Jaccard [9]	$S_{JAC}(T_1,T_2)=\frac{\lvert A\cap B\rvert}{\lvert A\cup B\rvert}$	A and B be sets of words in two ticket descriptions
	N-word overlap [1]	$S_{nwo}(T_1,T_2)=\tanh(\frac{overlap_{phrase}(T_1,T_2)}{n_1+n_2})$	A phrasal n-word overlap
	NLCS [11]	$S_{NLCS}(T_1,T_2)=\frac{(\lvert LCS(T_1,T_2)\rvert)^2}{\lvert T_1\rvert\times\lvert T_2\rvert}$	Considering the length of both the shorter and the longer string
Semantic Similarity	Leacock & Chodorow [15]	$S_{lch}(c_1,c_2)=-\frac{\log len(c_1,c_2)}{2\times D}$	Path-based method using wordnet
	RES [27]	$S_{res}(c_1,c_2)=IC(lcs(c_1,c_2))$	Information content-based method
	Word2vec-based [20]	$S_{w2v}(w_1,w_2)$	Word2vec based word similarity using wikipedia
Hybrid Similarity	ISLAM's measure [11]	$S_{STS}(T_1,T_2)=\frac{(\delta(1-w_f+w_f S_o)+\sum_{i=1}^{\rho}\rho_i)\times(m+n)}{2mn}$	Combining string similarity, semantic similarity and common-word order similarity
	Li's measure [17]	$S_{SSI}(T_1,T_2)=\delta\frac{s_1\cdot s_2}{\lVert s_1\rVert\cdot\lVert s_2\rVert}+(1-\delta)\frac{\lVert r_1-r_2\rVert}{\lVert r_1+r_2\rVert}$	Considering semantic similarity and word-order similarity
	SyMSS [22]	$S_{symss}(T_1,T_2)=\frac{1}{n}\sum_{i=1}^{n}sim(h_{1i},h_{2i})-l\times PF$	Considering semantic and syntactic info
	our method	$S_{TicDNN}(s1,s2)=cosine(x_{s1},x_{s2})$	x_s is the vector representation for ticket summary s

Table 4: The evaluated similarity measures including 3 categories and 10 measures. The distributed representation for tickets learned in our model capture both string and semantic similarity, thus we categorize it as hybrid similarity.

Measures	F1 score		
	Worst	Avg.	Best
S_{JAC}	0.4318	0.5677	0.7024
S_{nwo}	0.4763	0.5998	0.7043
S_{NLCS}	0.5325	0.6332	0.7221
S_{lch}	0.6823	0.7427	0.7866
S_{res}	0.6885	0.7576	0.7969
S_{w2v}	0.7538	0.8169	0.8693
S_{STS}	0.8048	0.8553	0.8953
S_{SSI}	0.8035	0.8497	0.8834
S_{symss}	0.8042	0.8503	0.8885
S_{TicDNN}	**0.8103**	**0.8595**	**0.9002**

Table 5: Comparisons of F1 scores using different similarity measures.

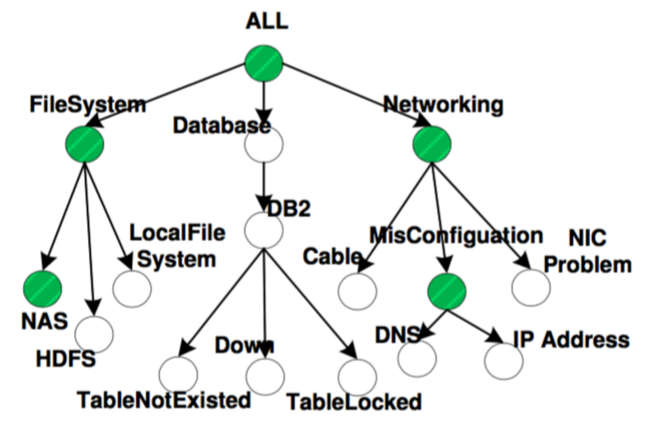

Figure 4: Hierarchical multi-label classification task.

Hierarchical multi-label classification[38]

Figure 4 visualizes the task of hierarchical multi-label classification.

We compared the performance of two classification algorithms on the original feature representation (GLabel and CSSA) and the derived feature representation (GLabel+ and CSSA+).

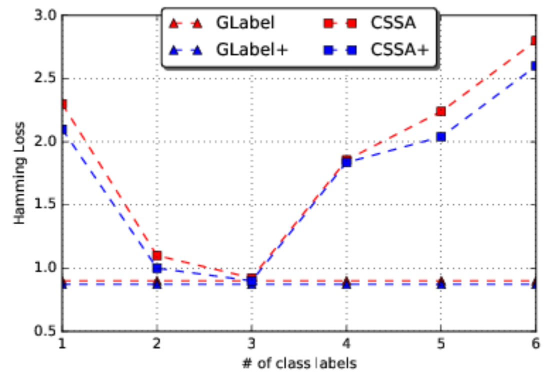

Figure 4: The lowest Hamming loss: GLabel gets 0.901 and GLabel+ 0.872; CSSA gets 0.923 and CSSA+ 0.901.

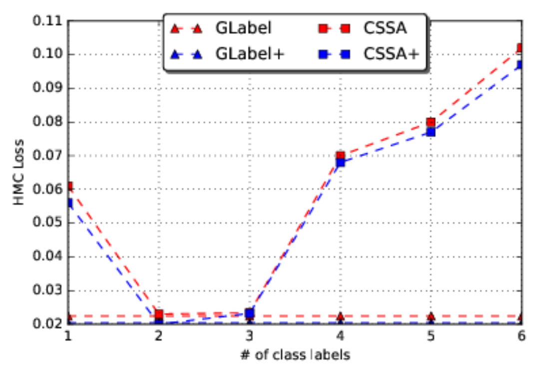

Figure 5: The lowest HMC-Loss: GLabel gets 0.022 and GLabel+ 0.020; CSSA gets 0.023 and CSSA+ 0.023.

Conclusion

In this work, we presented the major challenges in ticket resolution, such as quality quantification of ticket resolutions and consideration of resolution quantification in a recommendation problem. We de ned a deep neural network-based ticket resolution recommendation framework and evaluated it against a large real-world dataset. The evaluation demonstrated the effectiveness of the proposed model. Moreover, The distributed representation induced by the network is able to capture semantic relations of noisy ticket components, and can be applied to relevant fundamental applications in ticket analysis, such as ticket clustering, ticket classification and so on.

References

Acknowledgement

The work was supported in part by the National Science Foundation under Grant Nos. IIS-1213026, CNS-1126619, and CNS-1461926, Chinese National Natural Science Foundation under grant 91646116, Ministry of Education/China Mobile joint research grant under Project No.5-10, and an FIU Dissertation Year Fellowship