Content-Aware Resolution Sequence Mining for Ticket Routing

Peng Sun[1], Shu Tao[2], Xifeng Yan[3], Nikos Anerousis[2], and Yi Chen[1]

[1] Computer Science and Engineering, Arizona State University
Email: {peng.sun, yi}@asu.edu
[2] IBM T. J. Watson Research Center
Email: {shutao, nikos}@us.ibm.com
[3] Computer Science Department, University of California at Santa Barbara
Email: xyan@cs.ucsb.edu

Abstract. Ticket routing is key to the efficiency of IT problem management. Due to the complexity of many reported problems, problem tickets typically need to be routed among various expert groups, to search for the right resolver. In this paper, we study the problem of using historical ticket data to make smarter routing recommendations for new tickets, so as to improve the efficiency of ticket routing, in terms of the Mean number of Steps To Resolve (MSTR) a ticket.
Previous studies on this problem have been focusing on mining ticket resolution sequences to generate more informed routing recommendations. In this work, we enhance the existing sequence-only approach by further mining the text content of tickets. Through extensive studies on real-world problem tickets, we find that neither resolution sequence nor ticket content alone is sufficient to deliver the most reduction in MSTR, while a hybrid approach that mines resolution sequences in a content-aware manner proves to be the most effective. We therefore propose such an approach that first analyzes the content of a new ticket and identifies a set of semantically relevant tickets, and then creates a weighted Markov model from the resolution sequences of these tickets to generate routing recommendations. Our experiments show that the proposed approach achieves significantly better results than both sequence-only and content-only solutions.

1 Introduction

Ticket routing is a critical issue in IT problem management. When a problem is reported to the IT service provider, a ticket is created to describe the problem symptoms and to serve as a token in the problem management process. Due to the increasing complexity of the reported IT problem, many tickets need to be routed among various expert groups, to search for the one with the right expertise to resolve it, i.e., the *resolver* group. Obviously, the goal of ticket routing is to quickly identify the resolver, so that the caused disruptions can be minimized.

Today, ticket routing is usually driven by human decisions. It is common that tickets can sometimes be mistakenly routed, which leads to unnecessary ticket routing steps. If this happens, not only resources are wasted, but also it would take longer time to close a ticket, possibly cause customer dissatisfaction. The goal of this study is to develop an approach to systematically reducing the number of ticket routing steps by mining historical ticket data.

Tickets typically are categorized based on the nature of the reported problems, e.g., AIX, Windows, DB2, etc. This categorization is rather coarse-grained and tells little about the problem details. Besides the problem category, two types of other information in the tickets can be utilized to improve ticket routing: (1) ticket content, which contains the text description of problem symptoms; (2) resolution sequence, which records the sequence of expert groups that have processed a ticket [23], including the final resolver group. A ticket example that contains both content and resolution sequence is shown in Table 1.

ID	Description
28120	GUI is failing with ``Unable to Logon: RT11844: Security exception: [IBM][CLI Driver] SQL30081N A communication error has been detected. Communication protocol being used: ``TCP/IP".Communication API being used: ``SOCKETS". Location where the error was detected.

ID	Time	Entry
28120	2007-05-14	New Ticket: GUI logon failure
28120	2007-05-14	Transferred to Group <u>SMRDX</u>
28120	2007-05-14	Check password correctness
28120	2007-05-14	Transferred to Group <u>SSDSISAP</u>
28120	2007-05-14	Check authorization of user account ...
28120	2007-05-15	Transferred to Group <u>ASWWCUST</u>
28120	2007-05-15	Web server checking
...	...	...
28120	2007-05-18	Transferred to Group <u>SSSAPHWOA</u>
28120	2007-05-22	Network checking...Resolved

Table 1. A ticket example with its problem description (top) and resolution sequence (bottom)

Previous works in this area have been focusing on mining only ticket resolution sequences [23, 22]. In [23], a Markov-model-based method was proposed to predict the next expert group that should diagnose the problem, based the groups previously processed the ticket. While this method was shown to be effective, the semantic information embedded in ticket content was ignored. Intuitively, the higher content similarity between a historical ticket and the new ticket, the higher similarity of their routing sequences. Thus different historical tickets can have different importance in guiding the routing of the new ticket. In this paper, we seek to extend the method in [23] by utilizing this information in ticket routing.

An intuitive way of using content information is to build text classifiers that can directly label each new ticket, based on its content, with its potential resolver group. As we shall see, such a method only works for tickets that are (1) rich in content, and (2) reporting very similar problems occurred in the history. As a result, it can only resolve a portion of the studied tickets and the resulting Mean number of Steps To Resolve (MSTR) is not always reduced, compared to existing solutions using the sequence-based method. The reason is as follows.

Each expert group corresponds to specific problem diagnostic steps. When a ticket is transferred among expert groups, corresponding diagnostic steps will be taken. The problem is resolved only when the ticket is transferred to a group that performs the diagnosis relevant to the root cause. By mining the resolution sequences of historical tickets (even though they are not reporting the same problem), the sequence-based method can increase the likelihood of finding the right expert group given that certain diagnostic steps have been taken and were not able to solve the problem. While for the content-only method, it can only try the resolver groups for those not-so-similar tickets, resulting in worse performance.

The insufficiency of considering ticket sequences only or content only motivates us to integrate both information and develop content-aware resolution sequence mining techniques.

In the hybrid method, we first identify a set of existing tickets that are similar to the new ticket in content. Then, we use the resolution sequences of these similar tickets to generate a weighted Markov model. Compared with existing approach [23], in this model tickets having different similarity levels are weighted differently. To evaluate content similarity, we extend the existing text-mining techniques [27, 6, 20]. Specifically, we develop a *Cosine-similarity-based* weight function for model generation. Our study shows that the parameters in these weight functions can make a salient difference of the model effectiveness. Thus for the weight function, we develop an algorithm to tune its parameters to optimally fit the new ticket based on the models built for the historical ticket that is most similar to this ticket. Furthermore, we observe the situation where there are a lot of tickets that are dissimilar to the new ticket, whose combined weight may low down the effect of the highly similar tickets. Thus, we performed a model normalization to generate a training set of tickets with uniformly distributed similarities, even though the original training set can have a skewed distribution on similarities.

We conduct extensive experiments on a set of 1.4 million problems tickets. The results show that the Cosine-similarity-based weight function with normalization outperforms the other alternatives. Overall, the proposed method can reduce the MSTR of a ticket by 12.23% over the sequence-only approach proposed in [23].

To the best of our knowledge, this work is the first attempt to combine both ticket contents and resolution sequences to generate optimal ticket routing recommendations. Our contributions in this paper include:

- We explore the potential of mining ticket content to complement the resolution sequence mining method proposed in [23] to improve the accuracy in predicting ticket routing.
- We develop a hybrid approach to mine resolution sequences in a content-aware manner, and design algorithms to normalize the training data, as well as to fine-tune the parameters to achieve the optimal prediction results.
- We conduct extensive experiments, with real-world ticket data, to verify the proposed method. Our study shows it can significantly improve the efficiency of ticket routing, hence reducing MSTR. Therefore, it has great potential in serving as an on-line recommendation tool for ticket routing.

The rest of this paper is organized as follows. We first formally define our problem in Section 2. We briefly review the sequence-only method proposed in [23] in Section 3. Then we present a content-aware sequencing mining approach in Section 4. We introduce an approach for training data normalization in Section 5, and the algorithm to generate ticket routing recommendations in Section 6. In Section 7, we evaluate the effectiveness and robustness of the proposed approach. The related works are reviewed in Section 8. Finally, Section 9 concludes the paper.

2 Problem Formulation

In this section, we formally define the ticket routing problem. We consider a ticket processing system that involves a set of expert groups $\mathcal{G} = \{g_1, g_2, ..., g_m\}$. A ticket t is a tuple $t = (\tau, s)$, where τ is the description of the problem and s is the resolution sequence that consists of an ordered list of groups that processed the ticket. Take the ticket shown in Table 1 as an example, the content describes the problem as "unable to logon", while its resolution sequence records the routing among several groups, who may leave comments regarding the diagnostic steps taken. To simplify the problem setting, this work does not consider diagnostic comments.

In many cases, a ticket could be resolved by only one group. However, it may be transferred and diagnosed by multiple groups before the resolver group is found. Although a problem can be attributed to many different causes, there could be only one that led to the reported problem. Using the ticket in Table 1 as an example, the logon failure problem can be due to wrong password, unauthorized user account, server down, or network outage. As illustrated in Figure 1, each cause is checked by a different expert group. The problem can be resolved only if the group responsible for checking the actual cause (in this case, `SSSAPHWOA`) receives the ticket. Note that in this process, the ticket can be routed to groups in different orders.

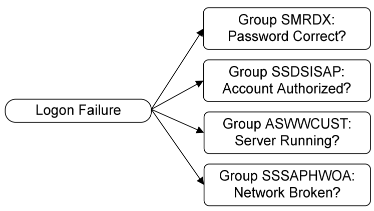


Fig. 1. Possible causes and corresponding expert groups for the ticket in Table 1

In this study, we focus on the problem of efficient ticket routing where tickets all have a single resolver. Given this assumption, our goal is to find the resolver as quickly as possible, so that the routing delay could be minimized. For a ticket that needs multiple resolvers, the algorithm proposed in this work is going to find its last resolver group. We measure the routing efficiency with the Mean number of Steps To Resolve (MSTR) [23]. Given a set of resolved tickets, $T = \{t_1, t_2, \ldots, t_n\}$, MSTR is defined as

$$\text{MSTR}(T) = \frac{\sum_{i=1}^{n} |s_i|}{n} - 1. \tag{1}$$

Note that we assume the initial group g_1 is given. Therefore, MSTR represents the average number of routing steps a ticket takes, starting from g_1, to reach its resolver group. Obviously, the smaller the MSTR, the more efficient the ticket routing method. For a set of new problem tickets (each contains problem description in its content and has a known initial group), T', the objective of this work is to *develop a routing system based on a training ticket set, T, which could predict the resolver of tickets in T', so as to minimize the MSTR of T'.*

3 Sequence-based and Content-based Ticket Routing

Given a resolved ticket dataset, there are multiple ways to model ticket routing among expert groups. A traditional approach is to build text classifiers based on ticket contents and then assign tickets to different experts. In [23], we proposed a routing algorithm based on resolution sequences. Since historical ticket resolution sequences provide rich information about the relationship and dependency between experts, the sequence-based approach has demonstrated good performance. In this section, we will briefly review the sequence-based and content-based approaches, and then discuss their strengths and weaknesses, respectively.

3.1 Sequence-based Routing

The sequence-based approach proposed in [23] relies on a Markov model to capture the transfer decisions made during ticket routing. In this model, each Markov state represents a group that processed the ticket. For the first-order Markov model, the transition probabilities between two groups A and B, represent the likelihood that group A transfers a ticket to B when A is not able to resolve it. In [23], we developed a more sophisticated approach using a variable-order Markov model.

Let $s_{(k)}$ be the set of k expert groups, i.e., $s_{(k)} = \{g_{(1)}, g_{(2)}, ..., g_{(k)}\}$. Given a resolved ticket dataset T, the total number of tickets that have been processed by $s_{(k)}$ is denoted as $N(s_{(k)})$; and the total number of tickets that are transferred to group g after all the groups in $s_{(k)}$ processed them is denoted as $N(g, s_{(k)})$. We could derive the conditional probability of transferring a ticket to g, given that it has been processed by $s_{(k)}$:

$$P(g|s_{(k)}) = \begin{cases} \frac{N(g,s_{(k)})}{N(s_{(k)})} & \text{if } N(s_{(k)}) > 0, \\ 0 & \text{otherwise.} \end{cases} \tag{2}$$

In [23], the conditional entropy is used to determine the optimal order of the Markov model. Using the above transition probability, we built a sequence-based routing algorithm, *Variable-order Multiple active state Search* (VMS) to predict the next group a ticket should be routed to.

VMS works as follows. Given the set of all groups that have processed ticket (L_v), VMS considers all its subsets $s_{(k)} \subseteq L_v$, and selects the next group from a candidate list L_c ($L_c \cap L_v = \varnothing$) to maximize the transfer likelihood:

$$g^* = \text{argmax}_g\ P(g|s_{(k)}), \forall g \in L_c, s_{(k)} \subseteq L_v, \tag{3}$$

using all of transfer probabilities calculated through Eq. 2. This prediction can be conducted interactively, at any stage of ticket routing, until the final resolver group is found. Note that if group g^* identified by Eq. 3 is not the resolver, it will then be added to L_v in the next iteration. More details of the VMS method can be found in [23].

The VMS method is a *sequence-based routing method*. It assumes that tickets related to similar problems are available to build the model. In [23], this is guaranteed by the manual ticket categorization by the experts. In practice, however, such categorization can be coarse-grained or inaccurate, which could undermine the effectiveness of this method.

3.2 Content-based Routing

Ignored by the sequence-based approach, ticket content contains informative descriptions of reported problems, such as where and when the problem occurred, the affected system, the phenomena etc. Intuitively, the content information should be very useful to identify the right resolver to a ticket. A straightforward approach of leveraging content information is to predict resolvers from ticket description. This is a classic text classification problem, for which various known algorithms (e.g. support vector machine (SVM), k-nearest neighbor, etc.) can be applied. For example, one could create a feature vector from the problem description. Each vector will then be mapped to a resolver group – the class label.

In our study, we trained an SVM classifier with the RBF kernel using the training ticket set. For each testing ticket, we use this classifier to generate a list of candidate groups, ordered by their matching probabilities. We then assign groups in this list, starting from the top, until the true resolver group is found. We refer to this method as the *content-based routing approach*.

3.3 Discussion

Figure 2 compares the cumulative prediction accuracy of the sequence-based and content-based methods as a function of the number of routing steps. The ticket dataset is obtained from the IBM problem management system, and related to the AIX operating system. It is clear that the prediction accuracy of the content-based method is better than the sequence-based method at the beginning. However, it barely improves in the following steps. As a result, it is outperformed by the sequence-based method gradually. A careful examination of the tickets shows that the content-only method performs better for those content-rich tickets than the sequence-only method. Unfortunately, for the tickets that are either not semantically rich in their descriptions, or their reporting problems never happened before, the content-based approach will perform poorly. In contrast, the sequence-based method can be more effective for those tickets since its decision is based on ticket transfer probability, which is complementary to ticket contents.

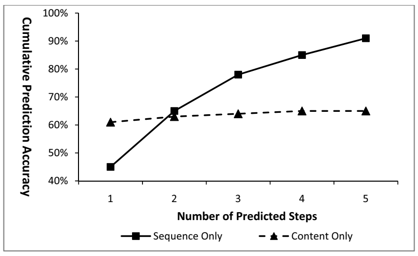


Fig. 2. Cumulative prediction accuracy as the number of routing steps allowed increases: sequence-only method vs. content-only method.

We illustrate this effect using the example in Table 1. If the reported problem has appeared in the training data and there is only one root cause of this problem, the content-based method will perform well. If either of the conditions is not met, it could make huge classification errors and needs more steps to resolve the problem. For the sequence-based method, it predicts the next step based on the actions that have been taken: if the actions represented by *SMRDX* and *SSDSISAP* have been taken, it predicts *SSSAPHWOA* as the most likely group to solve the problem. This decision process does not rely on the fact that the same or similar problems have happened before. Instead it can be inferred from many other ticket processing patterns in the training data.

From the above discussion, we can see clearly that both the sequence- and content-only methods have their own strength and weakness. This motivates us to develop a hybrid approach that combines these two methods together so that the predication accuracy of ticket routing can be maximized.

4 Content-aware Resolution Sequence Mining

In this section, we introduce a content-aware sequence mining method that customizes the VMS routing algorithm for each new ticket: an individual VMS model is derived for each new ticket t' based on the similarity between t' and the tickets in the training dataset. The basic idea is as follows: For a new ticket, we first evaluate the content similarity between the existing tickets and the new ticket; Then, the sequences of those similar tickets are used to learn a sequence-based routing model. In particular, the sequences of the training tickets weigh differently according to their similarity to the new ticket in content.

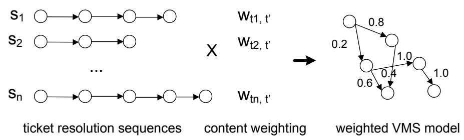


Fig. 3. Content-aware Weighted VMS Model

4.1 Overview

Given a training ticket set T and a new ticket t', we first evaluate the similarity between the new ticket and the existing ones, written as $w_{t_i,t'}$, $t_i \in T$. The similarity function will be discussed in Section 4.2. For each ticket t in T, we use $s_{(k)}$ to denote the set of k groups that have processed it in the past, and if a group g processed the ticket after all the groups in $s_{(k)}$, we denote it as $s_{(k)} \to g$. We then define $I(g, s_{(k)}, t)$ as the indicator function of whether $s_{(k)} \to g$ occurred in the routing sequence of t, i.e.,

$$I(g, s_{(k)}, t) = \begin{cases} 1 \text{ if } (s_{(k)} \to g) \text{ is found in } t \\ 0 \text{ otherwise.} \end{cases}$$

Similarly, we define $I(s_{(k)}, t)$ as the indicator function of whether a set of groups $s_{(k)}$ ever processed ticket t, i.e.,

$$I(s_{(k)}, t) = \begin{cases} 1 \text{ if } s_{(k)} \text{ found in } t \\ 0 \text{ otherwise.} \end{cases}$$

Thus, for a new ticket t', the weighted transition probabilities of the VMS model is defined as:

$$P(g|s_{(k)}) = \frac{\sum_{t_i \in T} w_{t_i,t'} I(g, s_{(k)}, t_i)}{\sum_{t_i \in T} w_{t_i,t'} I(s_{(k)}, t_i)}. \tag{4}$$

Here, the weight function $w_{t_i,t'}$ controls the contribution of ticket t_i to the calculation of transition probability. Figure 3 illustrated the content-aware weighted VMS model. When $w_{t_i,t'} = 1$, the learned VMS model will be the same to all new tickets. When $w_{t_i,t'}$ reflects content similarity between training tickets t_i and t', it becomes a customized model for ticket t'.

4.2 Content Similarity-based Weight Functions

To measure the content similarity, we adopt the vector space model that represents text as vectors [19, 4]. Vector-based similarity models have been reported to have limitations for representing long documents [12]. However, this is not an issue for our studied ticket data set, in which we found 96% of the tickets contain less than 80 words.

Before vectorizing tickets, we preprocess tickets using stopword deletion, word stemming [2], etc. After preprocessing, only 35,690 dimensions (distinct words) were left for all the studied tickets. Then the bag-of-words approach is employed to convert tickets to vectors. Formally, let V be the word set. For each ticket $t_i = (\tau_i, s_i)$ in a ticket dataset T, we have a $|V|$-dimension vector $\overrightarrow{\tau}_i = \langle v_{i1}, ..., v_{i|V|} \rangle$, where

$$v_{ij} = \log(c(w_j, \tau_i) + 1) \log(\frac{|T| + 1}{df_j}).$$

Here, $c(w_j, \tau_i)$ is the frequency of word w_j in ticket τ_i; df_j is the number of tickets in T that contain word w_j. Using this vector definition, we can compute similarity between tickets, and define weight functions.

In this paper, we examine the commonly used Cosine similarity function.

$$cos(\tau_i, \tau') = \frac{\overrightarrow{\tau}_i \cdot \overrightarrow{\tau}'}{||\overrightarrow{\tau}_i|| \cdot ||\overrightarrow{\tau}'||}, \tag{5}$$

where $\overrightarrow{\tau_i}$ and $\overrightarrow{\tau}'$ are the vectors derived from ticket contents of t_i and t', respectively.

Specifically, we define $w_{t_i,t'}$ as an exponential of the Cosine between two tickets.

$$w_{t_i,t'} = cos(\tau_i, \tau')^m, \tag{6}$$

where $m \geq 0$ is a parameter. When $m = 0$, the model falls back to the unweighted version. When $m \to +\infty$, the most similar ticket dominates the transition probability. Figure 4 shows the MSTR achieved for a testing data set with 1,000 tickets. The accuracy of the routing model is seemingly a convex function of m.

The optimal value of m depends on the distribution of the similarity scores between t_i and t'. Hence, it should be tuned for each new ticket. Since the resolver of a new ticket t' is unknown beforehand, we are not able to directly tune the parameter m for

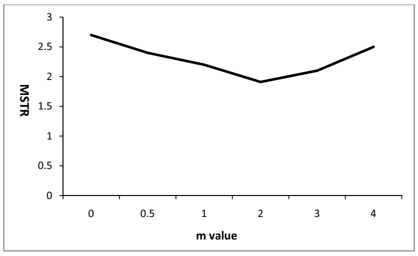


Fig. 4. The impact of m on the routing model

t'. Instead, we choose the training ticket (or a set of tickets to reduce noise) that is the most similar to t' to tune m. Specifically, we leave the most similar one t^* out as the new "testing" ticket, and use the rest of the tickets in the training set T to construct the model in Eq.(4). We gradually increase m starting from 0 with step 0.5 in each iteration, and compare the MSTR of t^* as m increases. The value of m that minimizes the MSTR of t^* will be chosen.

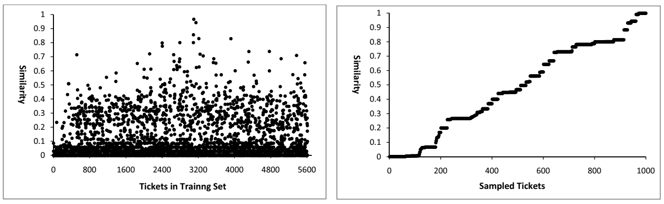
5 Training Data Normalization

The weighted VMS model proposed in Section 4 relies on a common assumption: the similarity between a new ticket and all training tickets is uniformly distributed. In practice, this assumption may not hold. For instance, Figure 5(a) shows the distribution of similarity between a randomly selected new ticket and 5,600 training tickets in the *AIX* problem category, using the Consine weight function. It shows that there are far more dissimilar tickets than similar ones.

As a consequence, even though the individual weight assigned to a dissimilar ticket is less than that assigned to a similar one, the overall transition probability can be overwhelmed by the dissimilar training tickets. For example, suppose there are 99 tickets that have the similarity value of 0.1, while only one ticket has the similarity value of 0.9. Ideally, this very similar ticket should dominate the routing recommendations generated by our model. However, if all the 99 less-similar tickets contain the same transition pattern, the transition probability for this pattern is likely higher than other patterns extracted from the most similar ticket. This might largely impact the accuracy of our method.

To overcome this problem, we use a *Bin-based Gibbs Sampling* method [7] to normalize the training tickets, so that their similarity to the new ticket is uniformly distributed. The approach works as follows.

First, we partition all the existing tickets into 50 buckets which are made by dividing the similarity range $[0, 1]$ into 50 equal size bins. We consider that all the existing tickets are represented by a n-variate joint probability distribution $p(\tau_i) = p(\tau_{i1}, \tau_{i2}, ..., \tau_{in})$, from which we wish to sample. Here, τ_i represents the content of ticket i, n is the number of dimensions in the word vector space. Suppose we choose a random ticket as the initial sample. In each step of Gibbs Sampling, we replace the value in dimension k (i.e., τ_{ik}) by a new value drawn from the distribution of that variable conditioned



(a) Original training set (b) Normalized training set

Fig. 5. Similarity distribution.

on the values of the remaining variables. That is, the value of τ_{ik} is replaced by τ'_{ik} drawn from distribution $p(\tau_{ik}|\tau_i\backslash k)$, where $\tau_i\backslash k = (\tau_{i1}, ..., \tau_{i(k-1)}, \tau_{i(k+1)}, ..., \tau_{in})$. One iteration of the sampling consists of n such steps that renew the values for all n dimensions. Then after each iteration, we find a ticket whose content vector is the most similar to the newly generated vector $(\tau'_{i1}, \tau'_{i2}, ..., \tau'_{in})$. The iteration continues until an equal number of tickets are sampled from each bucket.

The above method ensure that we obtain a set of tickets whose similarity to the new ticket is uniformly distributed. For example, Figure 5(b) shows the similarity distribution of 1,000 sampled tickets obtained from the original data set in Figure 5(a), using the proposed method. Clearly, the similarity distribution is now much closer to uniform than the original distribution was. Using these sampled tickets as the training data, we can then apply the models proposed in Section 4 to generate routing recommendations. As shown later in Section 7, this normalization can significantly improve the performance of our model.

6 Implementation

We implemented the proposed content-aware ticket routing algorithm in C#. The algorithm runs in three phases as described in Algorithm 1.

In Phase One, our algorithm first vectorizes the new ticket t as discussed in Section 4.2 and then calculates the similarity between t and the resolved tickets. Note that to reduce the delay of this step, all historical tickets are pre-vectorized and the resulting vectors are stored with inverted indices [19].

In Phase Two, it retrieves the most similar tickets to the new ticket t in the historical ticket dataset. Then, it finds the optimal parameter (m), based on the leave-one-out methods described in Section 4.

In Phase Three, our algorithm creates a weighted Markov model with the optimal parameter determined in Phase Two. Then, routing recommendations are generated for the new ticket t using the weighted VMS algorithm, given its initial group g.

We tested the performance of our code on a machine with 3.60 GHz CPU, 2GB memory. On average, the time spent in Phases One and Two in constructing the Markov model is about 11 seconds. Once the model is obtained, the time for computing the next routing group in Phase Three is negligible. Therefore, our algorithm can be readily used as an online recommendation system.

Algorithm 1 Content-aware ticket routing ($t = (\tau, \{g_1\})$)

ticket content: τ, initial group: g_1

Phase 1:
Build vector for ticket t: $\overrightarrow{\tau}$.
Evaluate similarity between $\overrightarrow{\tau}$ and the vector of each historical ticket.

Phase 2:
Select the set of tickets T_s that are most similar to t.
Learn optimal m based on T_s.

Phase 3:
Generate a normalized training set for t.
Calculate the weighted transition probabilities using Eq. 4 with the optimal m (or σ) found in Phase 2.
Set initial routing sequence $s = \{g_1\}$
while resolution group is not found **do**
    Use the weighted VMS routing algorithm with input s to recommend the next group g.
    $s = s \cup \{g\}$.
end while

7 Experiments

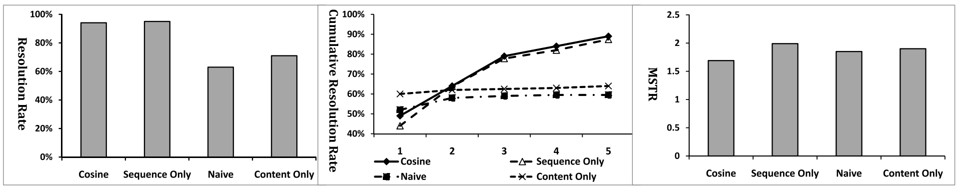
In this section, we evaluate the proposed approaches empirically. Our evaluation is based on 1.4 million problem tickets obtained from IBM's problem management system over half a 1-year period, from Jun. 1, 2006 to Dec. 31, 2006.

These tickets were pre-classified into 553 problem categories. This is the coarse-grained ticket categorization, typically done by the helpdesk when a ticket is first opened. In each problem category, 50 to 900 expert groups were involved in the ticket routing process.

Before explaining our experiment result, we will first introduce another ticket routing approach based on the resolution sequence only: the simplest ticket routing strategy, which we call *Naive Approach*. In this approach, the training dataset is composed of the pairs of initial group and the resolver group of each routing sequence in the historical ticket database. For an incoming ticket, based on its initial group, we calculate the transition probabilities of possible resolver groups. The resolver groups are ranked in the descending order of the transition probabilities, and are attempted in the order until the resolver group is found.

Besides MSTR, we introduce another effectiveness evaluation criteria: *Resolution Rate*. It measures how many tickets in the testing set can be resolved using a routing strategy. Specifically, for a testing set $T = \{t_1, t_2, \ldots, t_n\}$, resolution rate is defined as:

$$\mathrm{RR}(T) = \frac{\sum_{t_i \in T} R(t_i)}{|T|}. \quad (7)$$



(a) Resolution Rate (b) Cumulative Resolution Rate (c) Performance (MSTR)

Fig. 6. Comparing cosine based content-aware, sequence-only, content-only, naive approaches.

where the routing sequence of t_i has the last group denoted as g_i,

$$R(t_i) = \begin{cases} 1 & \text{if } g_i = g_i^* \ , \\ 0 & \text{otherwise.} \end{cases} \tag{8}$$

where g_i^* is the resolver of t_i determined by human decision.

Our experiments mainly aim to study the *effectiveness* of our approach: We first study the relationship between resolution rate and MSTR among four approaches: naive approach, sequence-only, content-only approach and our proposed cosine based content-aware approaches

Then we will focus on cosine based content-aware approach and sequence-only approach to show why content-aware approach is better than sequence-only approach.

7.1 Resolution Rate Comparison

First, we conduct the experiments to evaluate the resolution rate and MSTR of four different approaches. From 15392 AIX tickets, we randomly select 75% tickets as the training set, and use the rest of the 25% as the testing set to simulate new tickets that need to be routed.

The resolution rate of previous four approaches is shown in Figure 6(a). As we can see, while the other approaches have the resolution rate of at least 94%, the naive and content-only approaches only have a resolution rate of 63% and 71% respectively, which are not feasible to use in practice. Intuitively, since the naive approach only considers the correlation between the initial groups and the resolver groups, ignoring intermediate groups, it has a limited set of training instances and fails to transit many tickets to their resolvers. Content-only approach also only considers resolver groups, therefore it suffers a low resolution rate for the same reason as naive approach.

In addition, Figure 6(b) shows the cumulative resolution rate of tickets resolved within a given number of routing steps, when different routing approaches are used. As shown in the figure, cosine based content-aware approaches consistently outperform the sequence-only approach. This clearly demonstrates the benefit of incorporating ticket content information as well as sequence information for ticket routing: more tickets can be resolved within a given number of predicted steps.

Besides, the content-only and naive approaches outperform the other models at the first step of routing, showing that both of them are very effective for easy-to-resolve tickets. However, they become less effective as the routing continues. This indicates that for more difficult tickets, both content and sequence information should be considered to make effective ticket routing decisions.

7.2 MSTR Comparison

Figure 6(c) shows the performance of four approaches, where the cosine based approach has the best performance comparing to the other approaches.

Effect of Normalization. As mentioned in Section 5, to avoid lowing down the effect of the highly similar tickets, we generate uniformly distributed tickets as the training set. Now we evaluate the effect of the normalization.

In Figure 7, we know that normalization can reduce the MSTR about 15.28%. This shows that the normalization approach proposed in Section 5 is effectively when handling skewed data, and largely improves the MSTR.

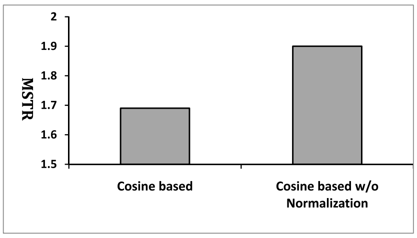


Fig. 7. The effect of normalization

7.3 Content-aware Approach vs. Sequence-only Approach

Using the same set of AIX tickets, we evaluate the MSTR of cosine based content-aware approach in improving the effectiveness of ticket routing, i.e., reducing MSTR, compared with the sequence-only approach.

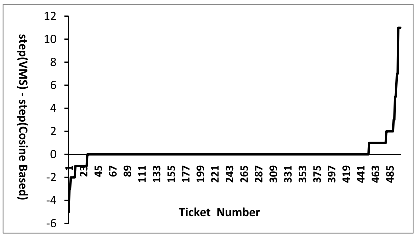
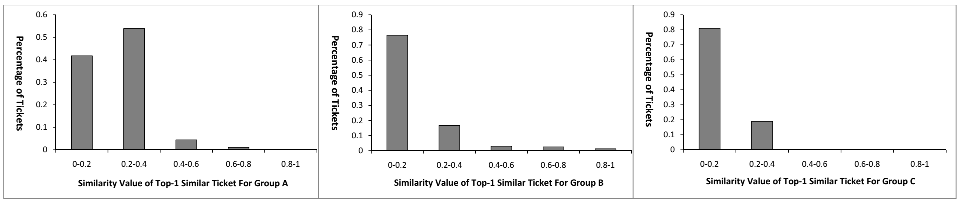


Fig. 8. The differences between sequence-only approach and the consine based content-aware approach in the resulting number of steps needed to resolve a ticket

Figure 8 shows the comparison between the sequence-only approach and the content-aware approach, in terms of the difference in number of steps needed to resolve a ticket in ascending order, for 500 randomly selected tickets. The figure shows that, for the majority of the tickets, the performance of cosine based content-aware approach is either



(a) Group A (b) Group B (c) Group C

Fig. 9. Top-1 similar tickets distribution for different groups

the same as or better than that of sequence-only approach (i.e., area with value greater than 0 in the figure). Nevertheless, the content-aware approach can still be outperformed by the sequence-only approach in some cases.

To illustrate why this happens, we first partition all the tickets into three groups based on difference of MSTRs for two approaches. In group A, cosine based content-aware approach performs better than sequence-only approach. In group B, both approaches have the same MSTR. In group C, the cosine based content-aware approach has a larger MSTR than that of sequence-only approach.

Then, we analyze the distribution for most similar ticket of each ticket in Figure 9. Figure 9 (a) shows the distribution for group A. As we can see, most of the top-1 similar tickets are located in the similarity range of [0.2, 0.4). While the top-1 similar tickets of group B (Figure 9 (b))and C (Figure 9 (c)) are mostly located in the range of [0, 0.2). Since the tickets in group A can find highly similar tickets to tune the parameters when building the routing model, they can be more effectively routed by our approach. Also, we can see that for group C, there is no ticket that can find a similar ticket with score larger than 0.4, resulting a larger MSTR. These observations confirm our intuition that the content-aware approach is more effective than the sequence-only approach when there are more tickets reporting on similar problems in the training data.

We have compared our content-aware approach with the sequence-only approach in all 533 problem categories, and found the former consistently outperforms the latter. Overall, the resulting MSTR of our content-aware approach is lowed by 12.23%, compared to that achieved by the sequence-only approach in [23].

8 Related Work

Text Mining. Related to this paper are the works on text mining [13, 17], which covers several important research areas, including text classification [20, 6], text association[16], topic modeling [27], etc. The Vector Space Model (VSM) applied in our system has also been studied before in the literature [19]. For instance, [15] first introduced SVM into the text classification applications based on the VSM model; the robustness of different text categorization methods was studied in [25]; [20] proposed methods to combine content and link information for document classification; [6] studied manifold methods; [6] introduced text classification using graph-based methods; [16] extracted word associations from text using synonyms or terms that tend to co-occur; [27] developed statistical models to find topics within a collection of documents. The weighted k-nearest neighbor classification method proposed in [14] extended the basic k-nearest neighbor algorithm by taking into account the distances to the nearest neighbors. In this paper, we extend these techniques in ticket routing applications, and address the unique challenges of parameter tuning and training set normalization in this context.

Expert Finding. The ticket routing problem is also related to expert finding: given a keyword query, find the most knowledgeable persons regarding the keyword query. Expert finding algorithms in [5, 10] use a language model to calculate the probability of an expert candidate to generate the query terms. [21] enhances these models by allowing candidates' expertise to be propagated within networks such as email networks, while [9] explores the links in documents such as DBLP [1]. Since most of expert finding algorithms are content-based, they have the same weakness as the content-based classification methods, as illustrated in Section 3.

Sequence Mining. The problem of sequential pattern mining was introduced by Agrawal *et al.* [3]. Various combinatorial algorithms such as SPADE [26], PREFIX [18], were developed for efficient mining in large sequence databases. Besides combinatorial solutions, probabilistic sequence mining was also studied in the literature [8, 11, 24]. For instance, Cook *et al.* [8] developed neural network and Markov approaches for mining software engineering processes. In the specific problem of ticket routing, [23] was the first work that proposed Markov-model-based methods to predict ticket routing steps. This paper extends that work to incorporate both sequence and content to accelerate ticket routing.

9 CONCLUSIONS

In this paper, we study the problem of improving the efficiency of ticket routing by mining both ticket content and resolution sequences. We propose a novel content-aware sequence mining technique to build ticket routing models. Specifically, We build a weighted Markov model with tickets having different similarity levels weighted differently. Ticket content similarity is measured using a *Cosine-similarity-based* weight function, where the parameters are tuned to optimally fit the new ticket. Furthermore, our technique performs a normalization on training set to effectively handle the training set with diverse distribution on ticket similarity. Extensive experiments on real-world ticket data show that, because of the incorporated content information, our proposed approach is consistently more effective than both sequence-only and content-only approaches. In particular, comparing to the sequence-only approach in [23], our approach has almost the same resolution rate, with a significant reduction of 12.23% in MSTR, and thus effectively accelerates the ticket routing processes.

Acknowledgment

This material is based on work partially supported by the U.S. National Science Foundation under grants IIS 0915438, 0917228 and 0954125.

References

1. DBLP: http://www.informatik.uni-trier.de/∼ley/db/.
2. K. Aas and L. Eikvil. Text categorisation: A survey., 1999.
3. R. Agrawal and R. Srikant. Mining sequential patterns. In *Proc. ICDE*, 1995.
4. C. G. Atkeson, A. W. Moore, and S. Schaal. Locally weighted learning, 1996.
5. K. Balog, L. Azzopardi, and M. de Rijke. Formal models for expert finding in enterprise corpora. In *SIGIR*, pages 43–50, 2006.
6. M. Belkin, P. Niyogi, V. Sindhwani, and P. Bartlett. Manifold regularization: A geometric framework for learning from examples. Technical report, Journal of Machine Learning Research, 2004.
7. C. M. Bishop. *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Springer, October 2007.
8. J. Cook and A. Wolf. Discovering models of software processes from event-based data. *ACM Trans. Software Eng. and Methodology*, 7(3):215–249, 1998.
9. H. Deng, I. King, and M. R. Lyu. Formal models for expert finding on dblp bibliography data. In *ICDM '08: Proceedings of the 2008 Eighth IEEE International Conference on Data Mining*, pages 163–172, 2008.
10. H. Fang and C. Zhai. Probabilistic models for expert finding. In *ECIR*, pages 418–430, 2007.
11. W. Gaaloul, S. Bhiri, and C. Godart. Discovering workflow transactional behavior from event-based log. In *Proc 12th Int'l Conf. CoopIS*, pages 3–18, 2004.
12. E. Garcia. Description, advantages and limitations of the classic vector space model, 2006.
13. M. Hearst. What is text mining? In *http://people.ischool.berkeley.edu/ hearst/text-mining.html*, 2003.
14. K. Hechenbichler and K. Schliep. Weighted k-nearest-neighbor techniques and ordinal classification. Technical report, Ludwig-Maximilians University, 2007.
15. T. Joachims. Text categorization with support vector machines: Learning with many relevant features. In *Proceedings of ECML-98*, pages 137–142, 1998.
16. D. Lin. Extracting collocations from text corpora. In *In First Workshop on Computational Terminology*, 1998.
17. New York Times. Text mining. http://blogs.zdnet.com/emergingtech/?p=304.
18. J. Pei, J. Han, B. Mortazavi-Asl, H. Pinto, Q. Chen, U. Dayal, and M.-C. Hsu. PrefixSpan: Mining sequential patterns efficiently by prefix-projected pattern growth. In *Int. Conf. Data Engineering*, 2001.
19. G. Salton and M. J. McGill. *Introduction to Modern Information Retrieval*. McGraw-Hill, Inc., New York, NY, USA, 1986.
20. P. Sen, G. M. Namata, M. Bilgic, L. Getoor, B. Gallagher, and T. Eliassi-Rad. Collective classification in network data. Technical report, 2008.
21. P. Serdyukov, H. Rode, and D. Hiemstra. Modeling multi-step relevance propagation for expert finding. In *CIKM*, pages 1133–1142, 2008.
22. Q. Shao, Y. Chen, S. Tao, X. Yan, and N. Anerousis. Easyticket: A ticket routing recommendation engine for enterprise problem resolution. In *34th Int'l Conf. VLDB*, Auckland, New Zealand, 2008.
23. Q. Shao, Y. Chen, S. Tao, X. Yan, and N. Anerousis. Efficient ticket routing by resolution sequence mining. In *KDD '08*, pages 605–613, 2008.
24. R. Silva, J. Zhang, and J. G. Shanahan. Probablistic workflow mining. In *Proc. 1998 Int'l Conf Knowledge Discovery and Data Mining*, pages 469–483, 1998.
25. Y. Yang and X. Liu. A re-examination of text categorization methods. In *SIGIR*, 1999.
26. M. Zaki. SPADE: An efficient algorithm for mining frequent sequences. *Machine Learning*, 40:31–60, 2001.
27. C. Zhai, A. Velivelli, and B. Yu. A cross-collection mixture model for comparative text mining. In *KDD '04*, 2004.