Augmenting Collective Expert Networks to Improve Service Level Compliance

Dissertation

Presented in Partial Fulfillment of the Requirements for the Degree Doctor of Philosophy in the Graduate School of The Ohio State University

By

Kayhan Moharreri, M.S.

Graduate Program in Computer Science and Engineering

The Ohio State University

2017

Dissertation Committee:

Jayashree Ramanathan, Advisor

Rajiv Ramnath

Srinivasan Parthasarathy

Gagan Agrawal

Abstract

This research introduces and develops the new subfield of large-scale collective expert networks (CEN) concerned with time-constrained triaging which has become critical to the delivery of increasingly complex enterprise services. The main research contribution augments existing human-intensive interactions in the CEN with models that use ticket content and transfer sequence histories to generate assistive recommendations. This is achieved with a recommendation framework that improves the performance of CEN by: (1) resolving incidents to meet customer time constraints and satisfaction, (2) conforming to previous transfer sequences that have already achieved their Service Levels; and additionally, (3) addressing trust to encourage adoption of recommendations. A novel basis of this research is the exploration and discovery of resolution process patterns, and leveraging them towards the construction of an assistive resolution recommendation framework. Additional interesting new discoveries regarding CENs include existence of resolution workflows and their frequent use to carry out service-level-effective resolution on regular content. In addition, the ticket-specific expertise of the problem solvers and their dynamic ticket load were found to be factors in the time taken to resolve an incoming ticket. Also, transfers were found to reflect the experts' local problem-solving intent with respect to the source and target nodes. The network performs well if certain transfer intents (such as resolution and collective) are exhibited more often than the others (such as mediation and exploratory).

The assistive resolution recommendation framework incorporates appropriate strategies for addressing the entire spectrum of incidents. This framework consists of a two-level classifier with the following parts: (1) content tagger for routine/non-routine classification, (2) A sequence classifier for resolution workflow recommendation, (3) Response time estimation based on learned dynamics of the CEN (i.e. Expertise, and ticket load), and (4) transfer intent identification. Our solution makes reliable proactive recommendations only in the case of adequate historical evidence thus helping to maintain a high level of trust with the interacting users in the CEN. By separating well-established resolution workflows from incidents that depend on experts' experiential and 'tribal' knowledge for the resolution, this research shows a 34% performance improvement over existing content-aware greedy transfer model; it is also estimated that there will be a 10% reduction in the volume of service-level breached tickets.

The contributions are shown to benefit the enterprise support and delivery services by providing (1) lower decision and resolution latency, (2) lower likelihood of service-level violations, and (3) higher workforce availability and effectiveness. More generally, the contributions of this research are applicable to a broad class of problems where time-constrained content-driven problem-solving by human experts is a necessity.

To my parents, my wife, and my brother

for their boundless support, and encouragement every step of the way

Acknowledgments

Since the very first days of my graduate studies, I have realized that obtaining a Ph.D. in an engineering field is not just about technical competence but it requires strong willpower, major dedication, and an indefinite persistence. Maintaining these qualities would have been almost impossible without the constant guidance that I received from my advisors, mentors, and collaborators. Neither would this important milestone be achievable had I not been supported by my family and friends.

I would like to genuinely thank my advisors, Prof. Jay Ramanathan and Prof. Rajiv Ramnath for their tireless support throughout my Ph.D. years. I learned so many lessons from them, from foundations of scientific research, to sincere academic courtesy such as being gracious to not-so-friendly reviewers. Jay's passion for quality research, coupled with her in-depth analytical expertise, helped me to sketch and develop the course of this research. Rajiv's accurate critiques and directions have always been fruitful towards developing a presentable outcome. I will always be grateful for the opportunity that I was given to work under their advice.

I also would like to thank Prof. Srinivasan Parthasarathy, and Prof. Gagan Agrawal for serving on my dissertation committee, and providing invaluable guidance during the last stages of my Ph.D. Also, I was privileged to work with Prof. Ross Nehm on a multidisciplinary project which led to the development of an independent research study and

an educational tool; I am thankful to Ross for the opportunity that I was given. In addition, Prof. Micha Elsner and Prof. Alan Ritter were always kind, and helpful when I was approaching them with challenging research-driven questions, and they deserve a special thanks.

My collaborators were magnificent. I want to thank Sobhan Moosavi, Minsu Ha, Satyajeet Raje, Kaushik Prasad, Jie Zhao, and Alhad Sapre among many others. By working with them I learned how to be ambitious and pragmatic at the same time. An integral part of this research was the support that we received from our industry partner, Nationwide Insurance, I am especially grateful to Travis Lenocker, Conrad Kuiken, and Dave Daniel who believed in me and my research, and taught me to stay determined while pitching new research ideas in a business-driven IT environment.

Finally, there is no way I can concisely express my affection for my family; my inspiring parents, Mina and Mehdi, my intelligent and lovely wife, Fara, and my rock-solid brother, Ehsan. I am sincerely thankful for the love and support that I received from them during all these years. Without such a team behind me, I doubt that I would reach where I am today.

Kayhan Moharreri
Columbus, Ohio
June 15, 2017

Vita

2010	B.S., Computer Science, Shahid Beheshti University, Tehran, Iran
2012 – 2016	Graduate Research Associate, The Ohio State University, U.S.A
2013 – 2016	Data Science Researcher, Nationwide Insurance, U.S.A
2015	M.S., Computer Science & Engineering, The Ohio State University, U.S.A
2016 – 2017	Graduate Teaching Associate, The Ohio State University, U.S.A
2017	Ph.D., Computer Science & Engineering, The Ohio State University, U.S.A

Publications

Research Publications

K. Moharreri, J. Ramanathan, R. Ramnath, "Motivating Dynamic Features for Resolution Time Estimation within IT Operations Management". *IEEE International Conference on Big Data (Big Data)*, 2103–2108, December 2016

K. Moharreri, J. Ramanathan, R. Ramnath, "Probabilistic Sequence Modeling for Trustworthy IT Servicing by Collective Expert Networks". *IEEE International Conference on Computers, Software & Applications (COMPSAC)*, (1):379–389, June 2016

K. Moharreri, A. Spare, J. Ramanathan, R. Ramnath, "Cost-Effective Supervised Learning Models for Software Effort Estimation in Agile Environments". *IEEE International Conference on Computers, Software & Applications (COMPSAC)*, (2):135–140, June 2016

K. Moharreri, J. Ramanathan, R. Ramnath, "Recommendations for Achieving Service Levels within Large-scale Resolution Service Networks". *ACM Compute*, 37–46, October 2015

K. Moharreri, M. Ha, R. Nehm, "EvoGrader: Automated Online Formative Assessment Tool for Evaluating Written Explanations". *Evolution:Education and Outreach*, (7):1–14, Springer, 2014

Fields of Study

Major Field: Computer Science and Engineering

Table of Contents

Page

List of Tables

List of Figures

Chapter 1: Introduction

Within a complex IT enterprise IT service orientation includes one or more of the following challenges: (1) multiple technology silos/departments/configuration items are involved in the service delivery. (2) delivery of service is contingent upon completion of multiple non-trivial processes. (3) processes rely considerably on human experts and their collective knowledge. The dramatic rise of IT service complexity is a consequence of increased business process automation, and accommodation to customer needs. This has exposed a major pain point in IT support services. Providing technical expertise and support for complex service operations is disproportionately difficult to attain [17]. This has created a vital need for low-overhead support services in complex enterprise operations, in turn, demanding enhancement of traditional enterprise knowledge and expertise management. This demand is addressed in part by IT Service Management frameworks (ITIL [50], MOF [53], etc.). At its core these are a set standard practices for a service life cycle which entails service planning, delivery and operation [21]. In the following sections we present our research related to these frameworks prior to discussing the contributions of this research.

1.1 Need for Enhancement within IT Service Support

An integral part of an IT service life cycle is a set of service operation processes. These processes include but are not limited to event management, incident management, request fulfillment, and problem management. A common theme among all service operation processes is that they are dependent upon technical support staff and their expertise. Human collaborations are essential since the needed expertise is often across multiple IT service operation structures (data centers, cloud technology towers, network provisioning teams, call centers, help desks, disaster recovery units, etc.). These structures are used to initiate, report, and collaboratively resolve incidents arising from operational IT infrastructures. This paradigm shift from a technology focus to a more complex service-oriented enterprise has put a substantial burden on service operation structures and this has exposed a need for augmentation of human collaborations towards greater effectiveness.

To be more specific about effectiveness, we need to discuss the notion of 'cost of service' to the enterprise. Competitive business demands cost effective IT service delivery and IT service support. Efficiency of support and delivery services can be achieved by reducing the following costs: (1) cost for the enterprise to deliver a service, (2) cost for the customer to exploit the service, and (3) cost for the enterprise to support that service. The business goal of this research is to reduce the cost of service support (i.e. #3) by improving incident resolution processes via augmentation of human collaborations.

Adding more infrastructure components (i.e Configuration Items (CIs)) is common practice for service enhancement within typical IT operations environments. This is regularly performed because (1) there exists a desire for a more robust and fault-tolerant infrastructure, and (2) service functionality enhancement often requires dynamic provisioning (of both hardware and operating CIs). Thus, adding more infrastructure components

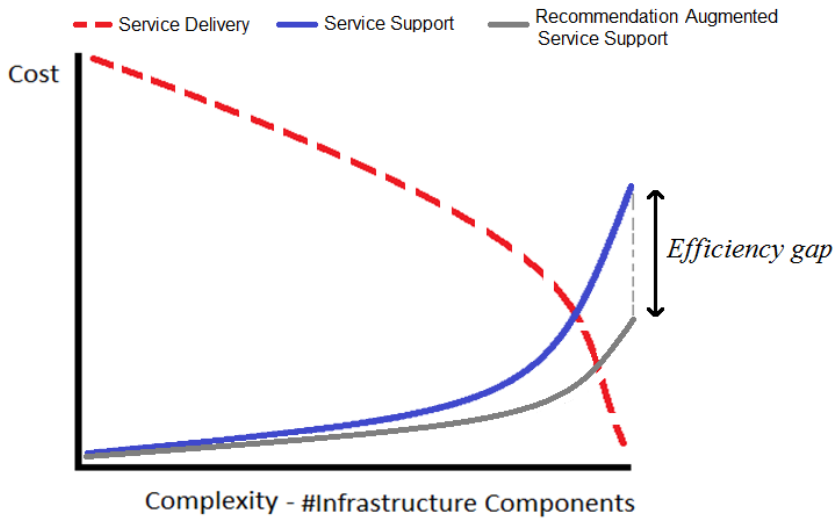


Figure 1.1: Research goal: support costs incurred due to complexity can be improved through recommendations

helps reduce the service delivery cost, but it also adds to the complexity of the service and thereby increases the service support cost. Figure 1.1 shows the relations between service complexity and the cost of service delivery and service support.

This research does not directly address the cost of service delivery (i.e. dashed line in Figure 1.1). On the other hand, it reduces the cost of service support and improves the efficiency. This is by providing adequate augmentation for the underlying expert collaborations, by addressing the incident resolution process. Therefore, our more specific goal is to leverage the full collective potential of the experts' knowledge and significantly reduce the cost of service support in IT operations environments.

1.2 Significance of Research in Assisted Problem-Solving

Collaborative problem solving – where the capability of the sum is greater than its parts – is leveraged today in a variety of ways. For instance, online question answering microblogs such as Stack Exchange (stack overflow , Cross validated, etc.) [64], Quora [54], and WebMD [84] have focused on taking advantage of wisdom of the 'qualified crowd'

in order to answer questions in respective domains. Going a step further, medical triage and health tracking systems such as TriageLogic [73], and InXite [27] have focused on resolving customers' complex medical treatment cases through collaboration between care providers. Software issue-tracking systems such as Bugzilla [9] and HP Quality Center [26] have also opened up new ways for software engineers to coordinate, relate and resolve program bugs. The field of IT Service Management has also resulted in an entire class of specialized collaborative enterprise systems as discussed next.

IT Service Management as the problem domain of this research imposes unique restrictions on expert collaborations. These restrictions by and large arise due to (1) process/workflow-driven dependencies, and (2) predetermined service delivery rules (i.e. Time Constraints in Service Level Agreements (SLA)). This research improves procedural support for collaborative problem solving that must meet both process dependencies and SLA time constraints. The unique challenges that arise in this context from a research perspective are due to the need for collaborative problem solving by a large number of individuals that must resolve problems with very complex technology infrastructures, within strict time constraints.

Briefly, the process of extracting, understanding, recalling and applying information obtained from heterogeneous sources is referred to as 'knowledge acquisition'. Human problem-solving expertise in a particular subject area gets developed through incremental iterations of the knowledge acquisition process [83]. The process by its very nature is limited to the individual's acquisition rate, cognitive load and memory capacity [69]. Even though performance of individual experts on simple problems with static subject matters is often reasonable, there are barriers to human cognition on more complex and dynamic problems. Therefore, collaboration between experts is essential on more complex problems over dynamically evolving environments. 'Collaborative intelligence' is the term for

retainable knowledge that emerges from collaboration of uniquely positioned individuals in order to achieve complex task completions [85]. 'Collective intelligence' has been introduced to identify the case where the individuals provide *non-overlapping expertise* to bear on the problem.

Exploiting the full potential of collective intelligence to address the efficiency gap of Figure 1.1 has become a critical subject for IT service organizations as they aim to reduce their workforce costs while delivering the needed technical expertise, dealing with evermore-complex infrastructures. To address this problem, a deep understanding of the individual's skills and knowledge, and a robust task delegation scheme becomes important.

Thus the goal of this research is to perform process discovery and prediction using machine learning and human computation combined in order to identify and recommend most efficient ways of delivering collective expertise. At a high level, this research enhances 'Collective Expert Networks' or CEN (as a special case of Expert Networks in the IT Service Management domain) by (1) developing a recommendation framework for efficient collective problem solving, (2) extracting problem solving expertise for the resolution process constituents, and (3) constructing a solution for resolution time estimation to offer time-efficiency and compliance with the time-limit requirements of the resolution process.

In the rest of this chapter, we provide a general background on Expert Networks and introduce more specific challenges and ways to improve on the existing methodology gap. We will also provide an overview of our research contributions.

1.3 Expert Networks as a Framework for Problem-solving

This section provides the motivation for the use of expert networks as the underlying framework for this research. Experts collaborate in order to complete complex tasks that

are not entirely achievable with a single individual's expertise. Therefore, interactions on complex tasks are due to non-overlapping and yet complementary knowledge needed at the time of the problem of solving. As a result, effective collaborations require experts to be able to answer two types of questions. First, they need to know how to contribute to the completion of the task. And second, they need to know who has the complementary knowledge to contribute to the completion of the task. Maintaining and applying these two types of knowledge – 'contribution knowledge' and 'interaction knowledge', respectively - is crucial in the task completion process. For purposes of framework development we next introduce a general and then a more specific type of network:

General Expert Networks (GEN) are constructed based on history of collaborations and task-transfers between experts. GENs in essence are directed graphs with historical interactions as edges between the expert nodes. Task completion is thus represented as a walk on the network and requires interactions.

Collective Expert Networks (CEN) are a special case of GEN in the domain of IT Service Management in which: (1) Experts provide the collective knowledge to resolve service-related incidents; (2) underlying well-defined workflows exist for the service operations and incident resolution; and, (3) there is a notion of time constraint associated with the resolution process.

In CENs, human experts are motivated to do as well as they are able (as we shall explain in Subsection 1.7.2), so this research aims to address the question of how machines can further boost the human problem-solving capabilities to address the efficiency gap. More specifically, how can we augment experts with recommendations to improve overall CEN performance? This research mainly targets recommendations for CEN interactions to

improve the expected time to resolve. We acknowledge that with some appropriate tuning many of our proposed solutions are generalizable to GENs as the more general case.

1.4 Incident Management Problem Solving Context

An 'incident' is an event that could lead to loss or disruption to organization's IT operations and services. Incidents are generally captured as 'tickets' and their content can be viewed as the 'footprint' of service inconsistencies. To achieve efficiency (Figure 1.1), it is essential to resolve the incidents within minimum elapsed time using only the necessary expert resources while meeting the Service Level Agreements. Service Level Agreements include predetermined time constraints defined based on urgency and impact of the corresponding service to the end-users. Experts are supposed to collectively problem-solve in order to resolve incidents. Figure 1.2 concisely illustrates the incident resolution process in IT Service Management, and the underlying model of a CEN, which represents experts as nodes that problem-solve on a given incident. More specifically, the Figure 1.2 (see corresponding labels) portrays:

1. Incident capturing procedure at IT Service Desk (ITSD); initiating ticket creation

2. Functional escalation to the CEN in case of unavailable pre-existing knowledge and need for technical expertise.

3. An example of CEN structure as a directed graph, and three suboptimal Ticket Resolution Sequences (TRS) as walks on the network leading to incident resolution. *Note*: well-established TRSs are also referred to as 'workflow' throughout this document.

4. Time to Resolve (TTR) and Service Level (SL) status after achieving resolution corresponding to each presented TRS.

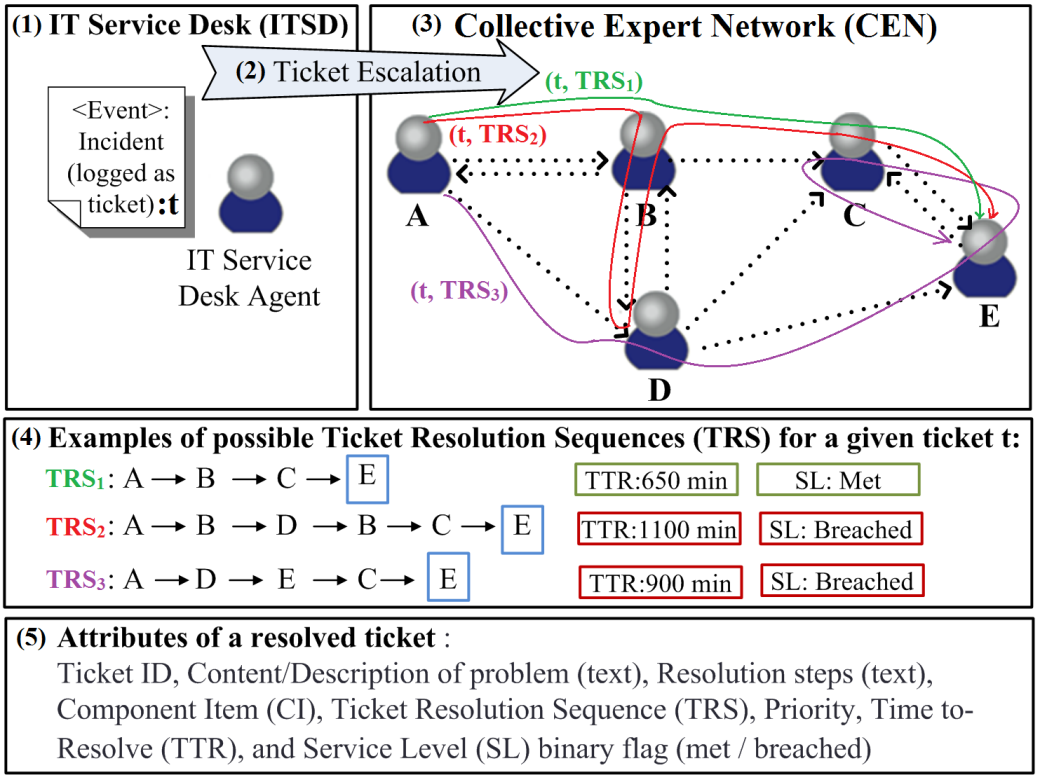


Figure 1.2: CEN terminology and resolution process overview

5. Other important ticket attributes such as Ticket content/description, priority, etc.

It is important to note that Ticket Resolution Sequence (TRS) with respect to a specific ticket is a sequence of transfers between the experts representing progressive discovery that leads to resolution for the ticket. In addition, as illustrated in the same figure, some TRSs help the ticket meet its SL goals while others do not. Here the objective is to minimize the resolution latency (i.e. TTR) of the tickets by recommending an optimal TRS given the history of tickets resolved by the experts in the CEN. This setting uniquely opens the opportunity to introduce a framework that facilitates incident management platforms with efficient ticket transfers between experts in any CEN.

1.5 Analysis of Challenges and Methodology Gaps

Service support challenges specifically pertain to (1) the event monitoring and incident capturing process and (2) the incident resolution process. Since the scope of this research is the incident resolution process as introduced in Section 1.4, it leaves the challenges related to incident capture as out of the scope. Also, fine-grained root cause identification for incidents also requires fine-grained embedded sensing and monitoring within the IT infrastructure and operations. Since IT Infrastructure and production operations are rarely available for research, and neither were they available to us at the time of this research, that too is out of scope. However, for completeness, in the following two subsections contextual facts and problems are discussed related to both incident capturing and resolution phases. In particular this helps us better discuss the problems that bound our research and proposed solutions.

1.5.1 Incident Capture Problems

- **Fact.1**: Existing monitoring systems auto-generate tickets based on rules that detect degradation of the vital signs against predefined thresholds in the operational event management systems.

- **Problem.1**: The threshold-based rules in these systems typically cause substantial volumes of *false alerts* (i.e. false positive events) that lead to resource misallocation and overstaffing.

- **Fact.2**: Many abnormal events that cause service disruption are not detected by the monitoring systems at the point of origin. Since an incident that is missed is not

captured, these events are referred to as *missing alerts* (i.e. false negative events) and these often lead to downstream user-reported incidents.

- **Problem.2**: Incidents originate from configuration items (or CIs) that are numerous within a complex infrastructure. It is almost impossible to achieve an accurate incident capturing system with a set of general monitoring rules on all CIs since every CI has its own configuration settings and environment specification. However, 'blanket' monitoring rules are commonly utilized since the alternative is the costly manual tuning of monitoring thresholds on each CI. Therefore, generally a large number of both false negatives and false positive events exist. Note that modifying the monitoring thresholds results in: (1) a trade-off between false negatives and false positives (2) a trade-off between cost of extra-resources (to deal with redundant or unnecessary tickets) and cost of outage recovery (to deal with unnoticed customer-impacting SLA violations)

- **Other research related to problem.2**: Recent work by Liang et al. [71] proposed a monitoring framework for enhancing the accuracy of alerts in an IT infrastructure, based on historically labelled data. Earlier work by Agrawal et al. [1] dealt with anomaly detection of constraint violations on network usage data. Both of these works are tailored to particular application domain and their deployed solution built on top of a well-sensed environment.

- **Fact.3**: User generated incidents lack fine-grained structure and are noisy textual reports about the observed behavior of the infrastructural defects and, not their causes.

- **Problem.3**: In these cases there is a prolonged incident investigation phase which could result in a SLA violation. Main contributing factors related to noisy user generated incidents are diverse user background knowledge and their linguistic variations, as well as user's partial view of the underlying problem.

1.5.2 Incident Resolution Problems and Research Approach

The problems identified in this section regarding incident resolution are the main focus of this thesis. Here we present an overview of our research approach while discussing some of the main challenges.

- **Fact.1**: Automated question answering and content-based recommendation systems can assist the incident resolution process by avoiding wasteful interactions between experts; based on a set of historically efficient interactions.

- **Problem.1**: Inference in these machine learning solutions is bounded by the operational logs where rare occurrence of the events, and scarcity among event features are commonly observed. In addition, the resolution process for a fraction of incidents heavily requires human engagement in order to perform investigation, diagnosis, and resolution. Thus, these incidents requires both trial-and-error and user-probing by human experts, none of which are recorded by the operational logs.

- **Non-routine Classification (research approach.1)**: This requires a *'non-routine'* incident handling system. We research our recommendation solution with a non-routine incident classifier dedicated for this propose. We looked for effective methods to separate frequently occurring incidents that are associate with well-defined resolution workflows from the rest of the incidents. The rare incidents that require

human-intensive resolution process are thus labeled as non-routine. Also note that we need to handle *all* incidents effectively due to the critical role of customer satisfaction.

- **Fact.2**: Experts have local knowledge of neighboring collaborators, but do not know the globally effective resolution paths. Thus they try to optimize the next interaction but struggle to identify the overall optimal resolution path.

- **Problem.2**: Lack of process-knowledge discovery solutions that could leverage historically resolved incidents to infer the entire global effective paths per incident content.

- **Routine Classification (research approach.2)**: This would require a *'routine'* incident handling system. We introduce our solution by proposing a routine incident classifier dedicated for this purpose.

- **Fact.3**: It is necessary for the business to ensure that the resolution process complies with Service Level Target (SLT: a fixed pre-agreed target time determined based on the priority of the incident).

- **Problem.3**: Resolution recommendations are not acceptable if SLTs are not *guaranteed to be met*. This relates to user trust - unreliable recommendations will slow down adoption.

- **Resolution Time Estimation (research approach.3)**: This requires a Time-to-resolve estimation system, which reports an *estimated time to resolve* for every recommended path. We developed resolution time estimation models leveraging dynamic characteristics of the CEN (i.e. Expertise modeling, expected high priority workload, etc.)

- **Fact.4**: IT Service Level Agreements are violated due to wasteful interactions between experts
- **Problem.4**: Absence of predictive measures for effectiveness of collaborative interactions
- **Transfer Intent Discovery (research approach.4)**: Need for explicit transfer intent identification. We outline a solution for intent characterization and discovery for ticket resolution paths.
- **Fact.5**: Organizational hierarchy is a meaningful representation of relationship between experts. This represents relationship between experts rather than 'management hierarchy'. Thus we call this: 'enterprise taxonomy'.
- **Problem.5**: Enterprise taxonomy has not been leveraged for predictive incident transfer models.
- **Utilization of Enterprise Taxonomy (research approach.5)**: (1) Need for further investigation about transfer localities on the enterprise taxonomy and its relationship with regularity of transfers. (2) Need for a regularization scheme for transfer models based on enterprise taxonomy, outlining a solution to battle the sparsity in the transfer model using the enterprise taxonomy.

Simply stated, we want a solution that is as sensitive as possible to when the CENs do well; and precisely identify and recommend those cases to augment the CEN when it is struggling. Next we wish to show that with such recommendations, the CEN does better.

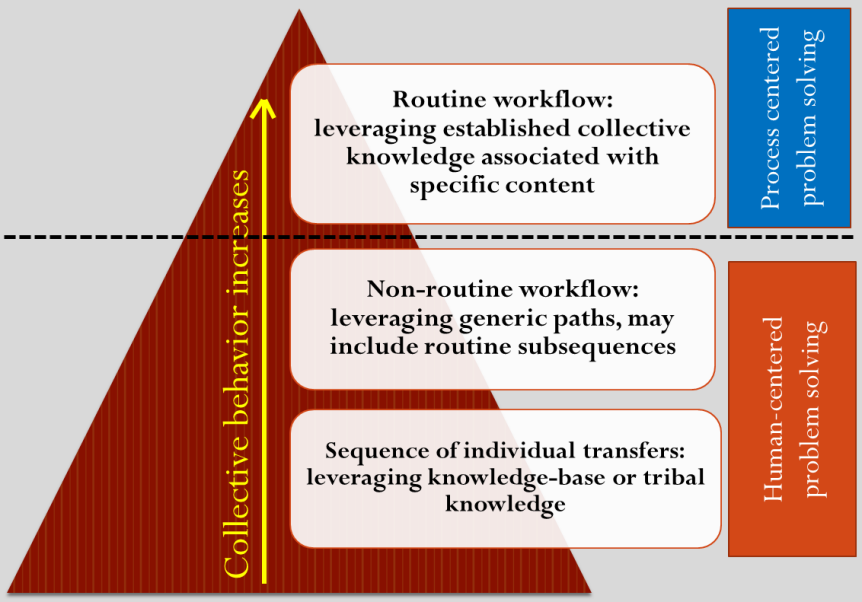


Figure 1.3: The spectrum of expert problem solving

We use Figure 1.3 to illustrate the entire spectrum of expert problem solving in the unassisted CEN as observed in the data. This ranges from human-centered (either non-routine workflows or sequences of individual transfers), to established process-centered (routine workflows leveraging collective routine knowledge).

Our unique research goal is *to improve collective problem solving within enterprise operations that also meets time and trust requirements*. Thus these are three research parts which together aim to integrate and improve both process-centric and human-centric resolution processes: (1) recommendations for transfer resolution sequences, (2) resolution time estimation, and (3) transfer intent discovery.

1.6 Research Hypothesis and Method

The main research hypothesis based on the approach identified above is as follows: *Every incoming ticket can be processed either by machine recommendations, or flagged for*

human-in-the-loop processing, establishing a way to ensure adoption of the framework by achieving service-level goals in enough cases to demonstrate compelling efficiency gains.

1.6.1 A Hybrid Method to Address Incident Resolution

The methods underlying the CEN recommendation framework take the content of an incident to robustly recommend a sequence of experts needed to work on the ticket in order to resolve it within the SL constraints. To achieve the goal, we present heuristics for identifying recurring resolution workflows associated with regular content. Then we build a classifier that is based on incident content and can distinguish whether the incident can be resolved by the recurring resolution workflows. If it can, then the resolution trajectory of the workflow on the CEN is recommended. Otherwise, human experts are required to fully identify the issue, diagnose and assign resources, and finally resolve the incident. Given a recommended resolution workflow, time to resolve estimation is critical for SL compliance, and it can create early warnings when an incident is expected to breach its SL. In order to provide an accurate resolution time estimation for a ticket on a recommended path, resolution and transfer expertise are estimated on the ticket; also workload (i.e. count of tickets in an expert's queue) is estimated, and are used together as indicative features for estimation of experts' response time. To discover more about human-centric resolution process, we propose a transfer intent discovery scheme in which we can identify the intention of experts' collaboration. This essentially helps to reduce unnecessary collaborations in the human-centric resolution models. Thus we implicitly incorporate effective workforce utilization.

Figure 1.4 is an overview to our resolution recommendation framework. As illustrated in Figure 1.4, the goal is to predict the full resolution path over the CEN. As an input, a new

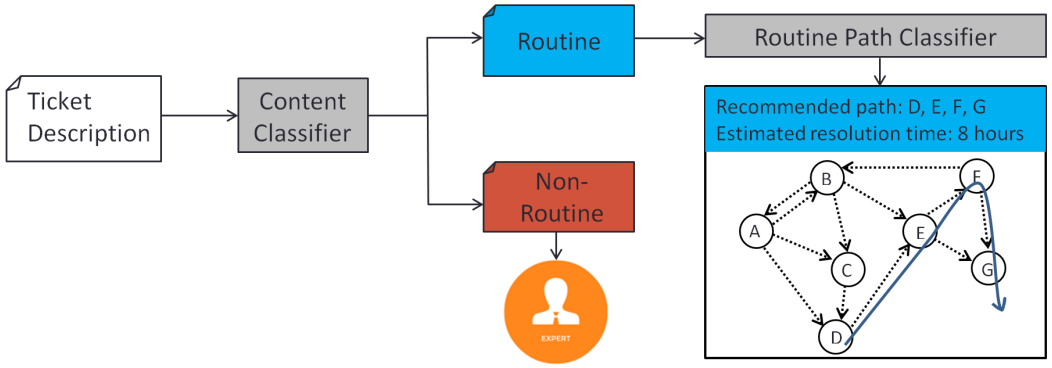


Figure 1.4: Overview of the recommendation framework for ticket resolution paths

ticket description will be passed to the recommendation model. First a content classifier identifies whether the content is routine. If it is, then a second level classifier determines its routine resolution path. Then for this predicted routine resolution path an estimation of the resolution time is provided. So the final output is either a recommendation of the full resolution path over the CEN along with SL time estimation, or a reference to human experts to find the resolution paths purely based on human problem solving. Next, the research contributions are briefly discussed.

Our final research goal was to make contributions that are more broadly applicable to problem-resolution environments (e.g. emergency response and triage, cloud-based service desks, supply chain resilience, software bug tracking). In these environments complex problems must be analyzed and solved within specific time constraints by networks of experts in order to meet the business or the social needs of the community. While our research in this thesis is based on a specific case of this general problem, the availability of extensive and detailed real-world enterprise data related to IT Service Management (defined by ISO 20,000 standard [50]) allowed us to develop a robust framework.

1.7 Main Research Contributions

Using real-world IT operational data, a set experiments were conducted to summarize our critical exploratory observations, and to evaluate our proposed resolution recommendation framework. Our main research contributions are now discussed.

1.7.1 Exploratory Observations

During the course of this research we conducted different exploratory studies, these will be provided in detail later (Chapter 3). An example is provided here to illustrates some of the evidence related to collective problem solving.

Relation between Enterprise Taxonomy Proximity and Collective Patterns

Our experiments and consequent transfer analysis on the structure of the CEN and its relation to the enterprise taxonomy resulted in the following findings: *Transfers within prominent collective problem solving units happen between experts that are structurally close to each other on the enterprise taxonomy*. As a case in point, experts in frequent transfers (i.e. probability of transfer occurrence greater than 0.012) are on average 18% closer on the enterprise taxonomy than the general-case transfers. This proves the fact that structural proximity (according to the organizational design) is often considered to leveraged collective problem solving by the CEN to achieve resolution. These findings opened a new avenue of research in which merits of structurally-motivated transfers are to be investigated. While in many cases we find it reassuring that the CEN dynamics for the problem-solving conform to the structural design of the enterprise, it raises a question of research: *what structural changes can be made to the enterprise taxonomy to optimize*

the transfer effectiveness (for example diameter of collective units, and average transfer elapsed time) to achieve collective problem solving?

1.7.2 Resolution Recommendation Framework

We created a two-level classification solution as shown in Figure 1.4 to address the entire problem-solving spectrum. To construct and evaluate our solution the following contributions were made. Further details of the framework and the experiments are provided in Chapter 4.

Labeling Strategy

We defined paths as the unit of problem solving. Then we separated tickets that had frequently-occurring content and were resolved by frequently occurring paths (i.e. *routine tickets*) from the rest of the tickets (i.e. *non-routine tickets*). This separation was experimentally done to maximize the distance of regularity of content for the routine tickets from that of the non-routine tickets. By the end of this stage, our experiment showed non-routine tickets are almost 4 times more likely to breach the SL targets. Thus the routine segment showed less anomalies, a window of opportunity for machine generated resolution recommendations.

R/NR Classification

We built a content tagger (routine/non-routine classifier) using a weight-normalized complement Bayesian classifier. Our classifier accurately distinguishing routine resolvable content from non-routine content: 80% precision is achieved on the routine class with a reasonable 20% coverage of the dataset, where precision of the routine class was a key

for the success of the framework. This played well into higher trust for adoption of the recommendations.

Path Classification

We built a routine path classifier using a weight-normalized complement Bayesian classifier. By only classifying routine content related to well-established resolution workflows we showed 34% performance improvement (based on R-precision) over existing content-aware greedy transfer model. We also estimated that there will be a 10% reduction in the volume of service-level breached tickets, and 7% reduction in the mean-time-to-resolve of all the incidents.

These advancements in developing resolution recommendation framework open a new avenue of research for studying human factors and their interactions with the augmented CEN. Particularly the recommendation framework can come under further scrutiny where usability evaluation, and improvement is taken into account with respect to human interactions.

1.7.3 Estimated Resolution Time Recommendation

To ensure SL-compliance for an incoming ticket given its recommended sequence of experts, there needs to be an accurate time-to-resolve (TTR) estimation. This percolates down to response time estimation for each expert in the resolution sequence. Our contributions with respect to resolution time estimation are summarized below, and further details are provided in Chapter 5.

Inefficiency of Static Expectation Solutions

We constructed a framework for assessing resolution time estimations of state-of-the-art static expectation model [80]. We assessed it as a baseline for time estimation modeling using historical routing time data. We identified poor estimations and the need for better Time to resolve modeling. We then used language modeling of the content and studied the impact of anomalous content on the baseline estimation error. We found that baseline estimation error increases too sharply when content gets more deviated from its recommended path. Thus we concluded that design of a more rigorous content-aware time estimation model is essential. Also, this is particularly important to engender trust in recommendations among the experts.

Expertise Modeling with Respect to the Ticket

Expertise on a ticket is evident based on whether the ticket is resolved or transferred, and this critical critical for resolution time estimation. The question we answered was: *Given the history of tickets resolved vs transferred by an expert, how likely is that specific expert in resolving vs transferring a specific ticket*? We showed that expertise can be learned from the transactional log. We introduced an Expectation-Maximization algorithm to learn log linear parameters and expertise vectors at the same time. Our results have shown a 7% F-measure improvement over similarity-driven expertise extraction baselines. As a result of our novel expertise modeling approach, we showed both low and high resolution expertise with respect to a ticket results in quicker response (i.e. Fast Resolution, Fast Transfer). This contribution not only helps towards the integration of an important feature (i.e. expertise) for a multivariate time-to-resolve estimation modeling, but also helps with

resolution and transfer knowledge estimation which are in tern important parts of transfer intent identification.

Additional features for multivariate time estimation are also studied. More importantly novel methods for *expected ticket load* was formulated, thus opening up a new avenue of research to statistically explore the dynamics of a CEN.

1.7.4 Novel Aspects for Framework Enhancement

By shifting our attention from the entire paths to the local dynamics related to expert pairs at each transfer, we are able to identify for the first time new ways of characterizing any expert's problem-solving intent at the time of the transfer. Further details are provided in Chapter 6.

Conceptualization of Transfer Intent Discovery

This is a new research focused on experts' intent in problem solving that begins with the following questions: Why do experts transfer tickets? And what kinds of transfers are necessary for resolution? The new avenue for CEN dynamics research above provides the methodological starting point for further understanding of experts' behavior, thus introducing a new subfield of large-scale collective expert networks. For example, in Figure 1.5 a ticket resolution sequence (i.e. a resolution path) is presented. Our task is to label the edges on the path with the appropriate transfer intent. We can now begin to identify four possible transfer intents: (1) Resolution (R), (2) Mediation (M), (3) Exploratory (E), and (4) Collective (C). Assuming that a model can now be developed to automatically label edges with the transfer intents, then incremental contributions to the resolution process can be discovered. In essence, this would lead to a rigorous solution to enhance human-centric resolution paths by eliminating non-contributing constituents from the path.

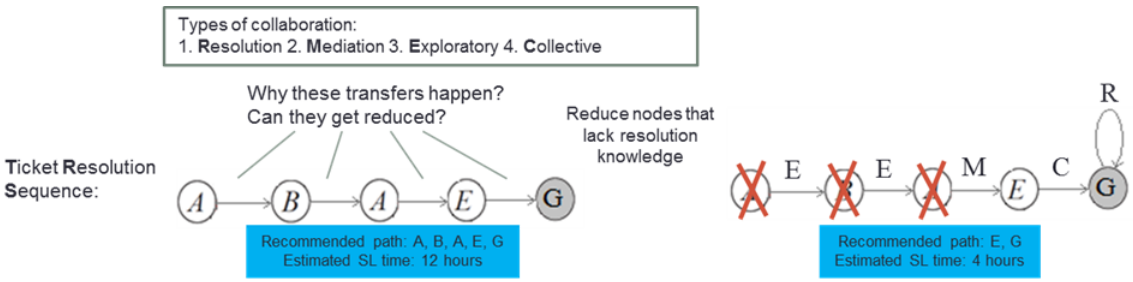


Figure 1.5: Intent discovery and resolution path reduction

1.8 Organization of the Rest of the Thesis

In subsequent chapters we provide details of related work, research and domain context, preliminary analysis, the research progress made thus far, and the research plan from here on out. This thesis is organized as follows:

Chapter 2 will introduce the broader area of research, and what all has been done by other researchers. Chapter 3 describes the required background for IT Service Management and provides our preliminary exploratory discoveries. Chapter 4 elaborates our muli-level classification framework and its impact on an IT operational environment. Chapter 5 discusses our solution for time-effectiveness of the proposed path recommendations. Chapter 6 summarizes our unique framework and discusses the future directions for this research, and its significance. Lastly, Appendix A presents prototypes of graphical user interface for the recommendation pilot in a live service management environment. Also, Appendix B shares the github repositories with code developed in the course of this research.

Chapter 2: Related Work

Conceptually related research is conducted under the broad areas of collaborative networks, expert finding, computer supported cooperative work, workflow process improvement, service science, and factors related to trust and adoption. Each of these aforementioned areas are reviewed next with respect to current research objectives.

2.1 Collaborative Networks

In recent years, social networks have attracted a lot of attention. Main body of work in Online Social Networks (OSN) focuses on social interactions and information exchange among users in large-scale networks. In OSNs, information generated at a source spreads by some growth factor into the network by users' forwarding activities. These forwarding activities diminish as the information loses its value [31, 86]. Also forwarding behaviors in the OSNs is typically made to influence other users. Collaborative networks in contrast, are dealing with information differently. Tasks are driving the information flow (i.e. resolution flow) in the networks. There is no branching in the information flow since transitions act as task handoffs. Task forwarding (transition) happens in order to find right experts to collaboratively work on the problem to resolve it.

Recent work by Miao et al. [41] studied the structure of collaborative networks (on two cases of developers network and service agents network), and presented that structure

affects problem resolution efficiency. They proposed a simulation-based approach to alter the network structure in order to achieve shorter routing sequences on the transformed network. Two shortcomings of this approach are (1) they considered information routing primarily as a stochastic process and (2) they ignored the real-world limits to reconstruction of the entire network, and the overhead costs.

Other work by Sun et al. [67] attempts to provide analysis of expert behaviors by defining expertise profile and expertise difference for transfers. They found that when task transfers happen there is some overlap of knowledge between experts. Recently statistical task routing has been researched in different domains, most related works are proposed in IT case and service management [60, 68, 39, 40], and in bug tracking systems [41, 68]. In earlier work by Shao et al. [60] a methodology was proposed for ticket routing by mining ticket resolution sequences without considering the ticket content. Then a Markov model was developed to statistically capture the decisions that have been made towards resolution. Later work by Sun et al. [68] enhanced the existing sequence-only approach by further mining the text content of tickets. The improved model was assessed based on reduction on *Mean Steps To Resolve* (MSTR).

Authors in [39, 40] decided to develop several generative models for ticket resolution process. These latest works inspired our approach as compared to earlier work since they decided to construct probabilistic models that can optimally generate the existing routing sequences, given the set of tickets. The main breakthroughs were their proposed transfer probability model and their greedy transfer model. Since we used a modification of their greedy transfer model as a baseline in Chapter 4, more details here are noteworthy. They modeled probability of a term given the CEN edges by using the Maximum Likelihood

Estimate (MLE) and defined that to be as follows:

$$P(w_k|e_{ij}) = \frac{n(w_k, T_{ij})}{\sum_{w_l \in W} n(w_l, T_{ij})} \tag{2.1}$$

Where w_k refers to the k^{th} word in the dictionary, and e_{ij} denotes the edge from expert group g_i to g_j. $T_{i,j}$ is the set of all the tickets transferred from g_i to g_j. This ratio of counts can be interpreted as: count of occurrences of the word in all tickets that took that edge by total count of tokens in all tickets that that took that edge. The greedy transfer model is developed to infer the most probable resolver upon the next transfer. The probability that ticket t is routed through the edge $e_{ij} = g_i \rightarrow g_j$ where $g_j \in G \setminus \{g_i\}$, is:

$$P(g_j|t, g_i) = \frac{P(g_j|g_i)P(t|e_{i,j})}{\sum_{g_l \in G} P(g_l|g_i)P(t|e_{i,l})} = \frac{P(g_j|g_i)\prod_{w_k \in t} P(w_k|e_{i,j})^{f(w_k,t)}}{\sum_{g_l \in G} P(g_l|g_i)\prod_{w_k \in t} P(w_k|e_{i,l})^{f(w_k,t)}} \tag{2.2}$$

Where: $P(t|e_{i,j}) = \prod_{w_k \in t} P(w_k|e_{i,j})^{f(w_k,t)}$ and $P(g_j|g_i) = |T_{ij}|/|T_i|$

$|T_{ij}|$ is the count of tickets transferred from group g_i to g_j , and $|T_i|$ is the count of tickets transferred from g_i to any group. The expert group $g^* = argmax_{g_j \in G} P(g_j|t, g_i)$ is selected to be the next expert group to handle ticket t. If g^* is the resolver, the algorithm terminates. If not, the algorithm gathers the information of all previously visited expert groups to make the next step routing decision. If a ticket t has gone through the expert groups in $R(t)$ and has not yet been resolved, the rank of the remaining expert groups in $G \setminus R(t)$ is:$Rank(g_j) \propto max_{g_i \in R(t)} P(g_j|t, g_i)$

Our baseline in Chapter 4 builds a path recommendation classifier based on consecutive expert-wise recommendations that are generated from the greedy transfer model discussed above. The main shortcoming of all the above statistical ticket routing models is the fact that they are assumed to work on any content based on probabilistic majority with

no accommodation for outliers as the irregular that also carry high impact to the IT organization. Another shortcoming is the validation of the recommendations for resolution in [60, 68, 39, 41, 40, 67]. Validation in all of the above work has been simplistically handled by an aggregate measure, MSTR, which is not truly representative of the degree to which the recommendation is resolution-achieving, especially when it comes to SL compliance. Information diffusion and propagation, as a common user behavior in social networks, has been widely studied, a few examples of which are [31, 12] on influence maximization, and [22, 86] on diffusion patterns in Twitter. Collective Expert Networks focus on routing information to the right experts (while minimizing the count of irrelevant hops and minimizing the expected resolution time) which poses different research questions than those in information diffusion and cascade models. In particular, deep domain knowledge of experts and their hierarchical dependence make the CEN routing problem completely a different research problem than information spread maximization problem.

Open issues: Models introduced above leave unanswered questions in terms of practicality of the solutions in a live IT environment as well as question regarding validation at a granularity of a single ticket. In a separate effort by Motahari and Bartolini [46] from HP labs, authors developed and deployed a domain-specific similarity-based resolution recommendation model for IT case management in which a finite state machine (FSM) was developed for possible steps in an IT problem case resolution process, and steps are predicted based on similarity of content to transitions on the FSM. The main advantage of their work as compared to others is the deployment and testing of the model in a live IT environment, but the downside is that their solution was domain-specific with a small-size

network (50 nodes). A salient feature of our model which has not been addressed by related research is its capability of estimating the completion time of a ticket on the predicted paths.

Online Social Networks vs General Expert Networks

Table 2.1: Comparison between OSN, GEN and CEN

	OSN	**GEN**	**CEN**
Goal	Understanding social communications	Problem solving and knowledge management	Problem solving (service recovery) within time constraints
Network analysis	Network structures and content extraction	Network structures, and content extraction wrt problem solving	Paths represent workflow, time constraints and enabling human computation
Network interpretation	Links are communications between individuals or nodes	Links are interactions on a task and nodes represent experts	Links are ticket transfers and nodes represent expert groups required for service recovery
Application example	Placing personalized ads	Expert finding queries	Service transactions and service level satisfaction

In this subsection, we make a comparison between General Expert Networks (GENs), CENs, and their prevailing counterpart, OSNs. This is also illustrated in Table 2.1. In the area of network mining and social applications, OSNs are considered predominantly rich entities for knowledge discovery as they carry large volumes of user related attributes, and their communication content that can lead to valuable outcomes based on interactive user

behavior mining, social anomaly detection, user evolution mining, and event correlations with the network structure. The OSNs are primarily considered to be non-collaborative networks. In other words users are taking part, not to propose or solve problems but to share personal or common interests [31, 86]. The OSNs links between users exhibit shared interests while links between experts in GENs represent collaborations/interaction. OSNs have no problem-solving objective to be met by each of the users. On the other hand in GENs, experts collectively have the responsibility to contribute to task completion and the commitment to resolve the problem for the benefit of an end-user. In OSNs, network structure carries meaningful concepts (structural communities, roles, and proximity patterns). On the other hand, in GENs, these very same structures have to be reinterpreted in terms of the problem solving goals. For example, network flows reflect task completion workflows.

Open issues: Finally, and importantly, in the special case of CENs (as first introduced in [13]), network flows represent potential incident resolution workflows that have *time constraints*. In practice, there might be various analytical goals for OSN mining such as efficient friend recommendations, social influence mining, or efficient personalized ad placements. In contrast the analytical goal for GEN mining is often to eliminate the overheads and optimize human collaboration efficiency for task completion. In the special case of CENs, the goal is to meet Service Level Agreement (SLA) or to minimizing Time-To-Resolve (TTR) for the incoming incidents. Here we also acknowledge the fact that certain graph mining applications such as frequent subgraph mining [19], and community detection [70] are fruitful to all of the above networks providing better understanding of the underlying structural properties.

2.2 Expert Finding

The expert finding problem is well-known in the information retrieval community. The incident resolution problem is related to the expert finding problem. That is given a query, find the most knowledgeable expert that can answer that query ([3, 20, 52]). Various approaches have been introduced to mine the information repositories in order to build personal expertise profile from experts' associated documents. These expert finding solutions have a common goal that is to propose algorithms that can accurately find the resolver of a problem or a query. The methods mainly fall into two categories: profile-based methods and document-based methods. For example, Balog et al. in [3] proposed two expert finding models, namely candidate model, and document model. In the candidate model, a textual representation is created to profile each expert's knowledge according to the documents associated with the expert. Then the probability of the query topic is assessed to rank expert candidates. The document model, on the other hand, ranks documents according to the query and then determines how likely a candidate is to be the needed expert by considering the set of documents associated. In a general case, profile-based methods [3, 20, 2, 37] first build a term based expertise profile for each candidate, and rank the candidate experts based on the relevance scores of their profiles for a given query topic by using traditional ad hoc retrieval models. In document-based methods [3, 20, 52, 4], instead of creating such term-based expertise profiles, the researchers use the supporting documents as a "bridge" and rank the candidates based on the co-occurrences of topic and candidate mentions in the supporting documents. These methods depend on rule-based methods to detect the candidate mentions in the supporting documents, to achieve reasonable retrieval accuracy. Most recently, social aspects of microblogs were leveraged [88] to solve expert finding in the context of enterprise social media.

Open issues: A common assumption in expert finding literature is that the expertise needed for the query can be found fully in an individual. In other words, although documents may be coauthored by multiple experts, the result of expert finding algorithms is a list of individual candidates for resolution. This assumption which is generally true for web and microblog documents makes the expert finding problem a narrow special case for collective incident resolution problem in the CENs.

This research not only considers the general case for collective incident resolution, but it incorporates a methodology for uncovering patterns of routing intents both of which are not part of typical expert finding research.

2.3 Computer Supported Cooperative Work

In the fields of Computer Supported Cooperative Work and Social Networks, coordination mechanisms that address the increasing complexity of collaboration has been extensively studied [10]. More recently the related concept of affordance that is 'individual', 'collective' and 'shared' has also been introduced and discussed extensively [29]. A relevant notion here is the 'collective' network behavior when individuals collectively achieve SL goals that they cannot achieve individually. Thus, it has been pointed out that shared affordances are essential to the performance improvement. However, here statistical methods for mediating shared affordances have not been researched. Also the notion of Work as a Service (WaaS) introduced in [49] proposes a hub to achieve responsiveness and address unpredictability. Other approaches to enhancing knowledge management through community aware strategies were provided in [29, 75]. In systems engineering, transitions are shown to add inefficiencies. A framework for measurement, traceability and improvement in service-oriented environments is presented in [55].

Open issues: In general, the design of statistical models to achieve SL has not been much addressed. Also in highly dynamic situations, statically defined transitions soon become obsolete. Thus the current research is complementary in that it uses probabilistic methods where static traceability cannot be relied upon resulting in discovery – oriented problem solving by CENs.

In addition to statistical research mentioned earlier analysis of human factors is critical for CEN augmentation and framework development. In order to find scenarios in which human errors are more likely to occur, analysis of these errors and mechanisms to prevent those have been studied in [65, 61] where common high-impact human errors are identified in the domain of service delivery. This is not limited to IT Services, and it is even more critical in the medical domain where humans are pressured to deliver emergency triage [7]. Similarly, human error identification, and preventive design were the cornerstones of our research. More specifically, in our research we first analyzed tickets that were incorrectly transferred to identify reasons for misrouting behaviors. Then we designed a learning framework to prevent expert from erroneous transfers.

2.4 Workflow Process Improvement

Workflows are designed to provide an infrastructure for execution, and monitoring of a defined sequence of tasks. The concept of workflow is tied to improvement of a business process during which information is passed around for action according to a set of procedural rules [33]. Data-driven process discovery [38] is a useful technique for the identification of the underlying workflows in a complex enterprise. The use of event logs to reconstruct the process model has been thoroughly studied in [76, 79] under the topic of process mining. The applications explored include process conformance checking [77] and

data provenance [78]. Motivated by these, in our research we have followed a data-driven approach where routine resolution worflows were first discovered and further exploited to improve the overall performance of the CEN. In particular, process discovery enabled prevention of anomalous problem-solving behavior on the routine content.

Open issues: Process time estimation has been commonly performed using aggregate measures over historically similar processes in the transactional logs [80, 58]. In this research, motivated by our earlier work [44], inefficiencies of the aggregate estimation methods, and lack of a context-aware time estimation model were assessed. Then a novel solution was proposed for resolution time estimation by taking into account the dynamics of the experts in the CEN (i.e. expertise on the ticket, and ticket load in queue).

While being on the subject of workflow improvement, it is noteworthy that existing probabilistic task routing models in GENs [60, 39, 13] do not take workflows into account. These models recommend the 'most probable' transfers between the experts by solving the inference on $P(transfer \mid task)$. They assume that the Markov property holds and thus recommend transfers only based on the previous node. In [40], a fixed short look-ahead subsequence was introduced to partially mitigate the memoryless modeling problem. However, all these transfer-based models lack a full consideration of workflows. We differentiate our own stream of research as 'workflow-based' models using $P(TRS \mid ticket)$, and will discuss the details and results in Chapter 4.

2.5 Service Science: Complex Enterprise Services

A related trend within the last decade has been the process-driven automation of customer-related services. The applications can be in different sectors of the economy, from financial

to the manufacturing and construction businesses. Service-oriented methods were developed to enable business processes with IT delivery processes [91]. Each delivery process is constructed on top of a bundle of technology components (software, applications, network, hardware, etc.) and maintained by specialized staff [50, 8]. A significant increase in business process automation in conjunction with more complex customer needs (driven by competitive markets) has resulted in an unprecedented increase in the prevalence of enterprise complexity [34].

Open issues: A number of challenges typically arise within complex enterprise supporting services. Here we present some of those challenges and the data-driven research conducted to address them: (1) The impact of negotiated SLAs on delivery cost is not easy to assess early in the service engagement process. A modeling approach is proposed [18] to estimate the impact of SLAs before and during the service engagement. (2) a considerable amount of human effort is needed for ticket categorization in IT incident management. Domain-specific classifiers are proposed [87, 90] to achieve accurate classification with minimum human-intervention. (3) Repeating auto-generated events generate similar tickets with similar resolutions. A scalable classification algorithm is proposed to recommend resolution text for tickets with historical evidence [93]. (4) Ticket volume and backlogs are critical to be monitored for proper resource allocation. Historical trends are leveraged to build time series in order to forecast the volume of the tickets [35].

Our research belongs to the same category, in the sense that it performs data-driven modeling to gain improvements on enterprise supporting services. However, our work builds a unique framework to address user-perceived tickets on the resolution workflows with a special consideration of SLs which has not been studied before.

Here we acknowledge the fact that other domains, such as medical triage [63], emergency response [47], and software issue tracking [28] have commonalities with time-constrained incident resolution which inspired the foundations of this work.

2.6 Recommendations and Trust

On-demand real time scoring and recommendation systems are becoming increasingly popular within the subfield of 'Decision Support Systems'. These systems are most effective where critical decisions are to be made in massive-scale within limited periods of time, or otherwise would suffer heavily from constrained and error-prone performance of humans. Their applications range from intelligent financial credit modeling [89], to automated response assessment [62]. In many of these applications there is a notion of trust which plays an important role in the adoption of such systems.

Open issues: Reliability of recommendations can be ensured by model transparency in addition to the accuracy. According to [57], although evaluation on annotated data is a useful pipeline for many applications, it may not correspond to performance "in the wild" and practitioners often overestimate the accuracy of their models.

In early stages of trust formation in a decision support system, knowledge-base and interactive design are found to be important factors for reliability of the system [82]. As suggested above, our research ensures trustworthy recommendations by (1) only acting on routine content, and (2) only recommending when the system evaluates an acceptably high confidence on its prediction.

Chapter 3: Background and Discovery of CEN Characteristics

In the first part of this chapter we present a detailed context for IT incident management, that is, the scope of the problem that is to be solved by this research and the objectives. Without understanding the supporting services in the IT enterprise, it is not feasible to construct an effective knowledge discovery and recommendation framework. Therefore, it is crucial to understand as-is enterprise-supporting services and the way operations are managed through human agents.

In the second part of this chapter, in order to deal with challenges regarding CEN resolution introduced in chapter 1, preliminary analyses in the ITSM domain is presented. These have guided the direction of this research and brought us to a better understanding of the domain and helped us build our experimental hypotheses.

3.1 Service Management and the Incident Domain

An incident in general has a two-phase life cycle: (1) capturing phase and (2) resolution phase. Capturing phase is the stage in which an issue is sensed and gets reported as a ticket. Resolution phase is the stage in which the ticket is dealt by the experts who collectively problem-solve. Also as we defined in Chapter 1, incidents can be captured via two distinct ways. We called these 'auto-generated' tickets and 'user-perceived' tickets.

Auto-generated tickets get captured through sensing with pre-defined monitoring conditions. These tickets are generated when monitoring thresholds are violated (e.g. "Memory usage beyond 95% on the Mail Server OH44-East-SA"). Monitoring systems aim to measure availability and performance on broad range of CIs such as servers, network components, and applications. Hence, each auto-generated ticket is naturally reported with its corresponding CI. On the other hand, these tickets can be mainly considered as noisy signals for the actual underlying incidents.

In the capturing phase, monitoring systems may misidentify incidents: generating false positive tickets (false incidents) and false negative events (unreported incidents). As discussed in section 1.5.1, this is a challenge which imposes overhead costs on the supporting services. False incidents unnecessarily consume expertise, and unreported incidents cause impactful infrastructure outages without notifications.

In the resolution phase, auto-generated tickets are not as problematic. In a typical service-oriented IT enterprise, supporting expertise is built around CIs (i.e., applications, servers, and network components). CIs are not necessarily indicative of the resolver experts; some experts resolve problems pertaining to certain CIs, while others deal with tickets with variety of diagnosed CIs. Finding accountable experts for auto-generated tickets which are already captured with their respective CIs, is not a challenging task and usually follows a straightforward process [93]. The auto-generated domain was initially studied by us but for the reasons mentioned earlier, we decided to shift our focus to user-perceived incidents and address the challenge of effective collaborative problem solving.

User-perceived tickets are reported by the users and are captured at the IT service desk. These incidents are communicated via phone calls and are immediately logged as a ticket with mandatory incident description text. Human agents at the service desk have access

Table 3.1: Incident management challenges

Incident types ↓ Phase →	**Incident Capturing**	**Incident Resolution**
Auto-generated	False incidents, unreported incidents	–
User-perceived	Partial user knowledge, Noisy user report, unidentified CI	Ticket routing on the CEN

to a knowledge base of well-known problems along with their resolution instructions. If neither knowledge base nor the tribal knowledge of the agent can solve the problem, it is then escalated to the experts in CENs. The ratio of received issues to the escalated incidents is often a measure for the effectiveness of the knowledge base and the technical expertise of IT service desk agents. It is also important to note that the cost for an incident sharply increases after escalation due to two main factors: (1) time that the ticket stays open, (2) more expensive expertise that is required to spend time on the ticket. What makes the capturing phase in the user-perceived case different from the auto-generated case is the fact that incident descriptions are provided in *natural language form* and are directly *reported by the end-users*. These unique properties of user-perceived tickets make it infeasible to associate them automatically with their corresponding CIs. As a result of undefined CIs, in the resolution phase, finding the appropriate expertise on the CEN is challenging and this requires complex discovery-oriented collective effort by the experts in the CEN. Table 3.1, summarizes typical incident management challenges in each phase with respect to different incident types.

Even though we explored the entire domain of incident management for a while, the final scope of this research is focused on the resolution phase of user-perceived tickets, which is concerned with the ticket routing problem and its SL compliance. Later in chapter 6, we will argue that ticket routing is not an isolated problem and can be improved by automated text-enrichment recommendations for the incident capturing phase. But for now the claim is that effective ticket routing on the CEN essentially results in substantial cost reductions for the supporting services. These reductions occur due to (1) *lower engagement of unrelated expertise in the resolution process*, and (2) *lower likelihood of SL violation*. From this point forward, for the matter of simplicity and terminology consistency we refer to 'user-perceived incident' and 'user-perceived ticket' simply as 'incident' and 'ticket' respectively. Expert nodes in the CEN are also referred as 'expert groups' or simply as 'experts'.

3.1.1 Resolution Achievement in Unassisted CENs

When a ticket is escalated from the service desk to the CEN, the initial expert in the CEN has to be identified. Typically the initial expert is determined at the service desk through tribal knowledge of the agent, or certain locally maintained workflow routing list. From there it is the job of CEN experts to provide collective problem solving in order to resolve the ticket. A ticket at any given point of its life cycle is typically worked by a single expert. Therefore, collective problem solving on a ticket can only be possible through ticket transfers between the resolving parties. The last expert that ensures the problem is resolved is the *'resolver'*. Thus, transfers create a *Ticket Resolution Sequence* (TRS) ending with the resolver. At the end of the resolution process, several attributes of the incident will become

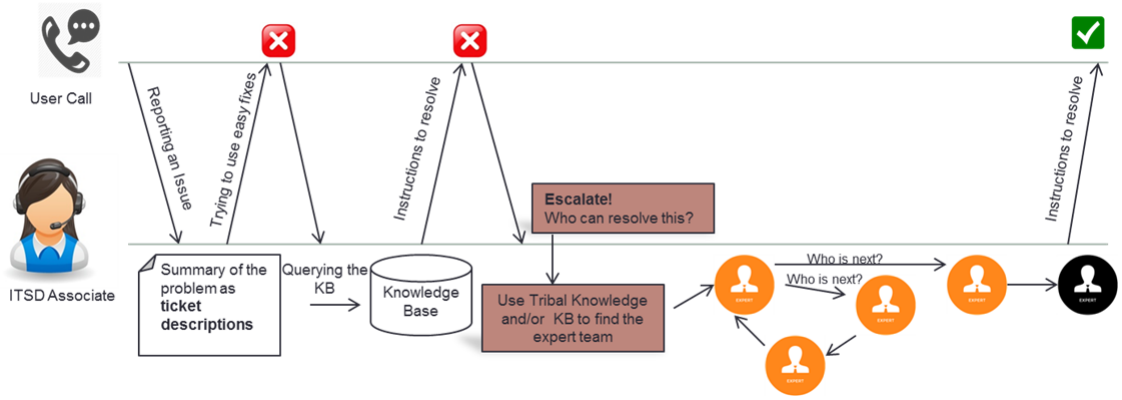
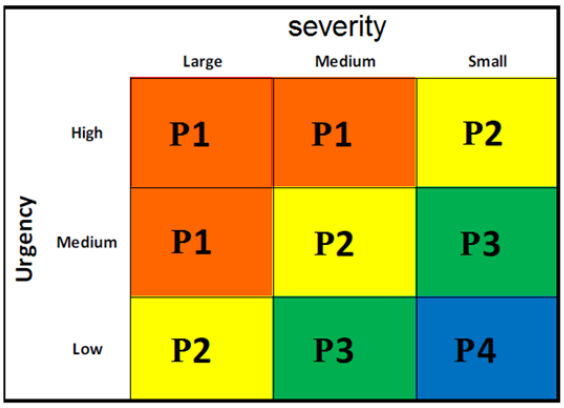


Figure 3.1: An example of an inefficient ticket resolution

known: SL met or breached, culprit CIs, incident resolution description, service downtime, ticket resolution time, impact to the users, etc.

As mentioned in the previous section, ticket transfers might cause wasteful interactions. Figure 3.1, portrays an incident resolution scenario from start to finish. Also in this specific scenario the ticket bounced around between several experts due to misidentification of the problem. Ticket bouncing, and CI misidentification are prominent issues causing delays in the resolution latency of the CENs.

Next we discuss the Service Levels (SL) and ticket priority. The holy grail of CEN problem solving is to meet the SL targets defined for each ticket. Breaching the SL target causes an outage, which significantly impacts the customers and related lines of business. These targets get defined based on ticket priority. For each incident, the priority is pre-determined in collaboration with the customer. The priority level is set between P1 (significant impact) to P4 (negligible impact) based on the severity and the urgency of the reported incident. In other words, priority is unifying two critical measures of SL compliance: (1) Severity and (2) Urgency. Severity signals the potential size of the impact to customers, and urgency signals the criticality of the disrupted service. Figure 3.2a shows how severity and urgency together produce the priorities for each ticket, and Figure 3.2b illustrates an example for SL



(a) Urgency/severity matrix

Priority	Business Impact	SL Target
1	Significant	14 hrs
2	Moderate	34.5 hrs (3 days)
3	Minor	46 hrs (5 days)
4	Negligible	115 hrs (10 days)

(b) SL Targets

Figure 3.2: Elements of service level agreement

target times per different priority classes. The SL clock runs per ticket, and time is shared among all the experts along the TRS.

3.1.2 Embedding a Recommendation Framework in the Incident Resolution Process

In this section the integration of our designed framework in a typical incident management environment is discussed. Our framework as black box, simply accepts an incoming incident description in natural-language form, and outputs a sequence of experts (referred as TRS, or resolution path). It estimates the expected time to resolve for the recommended resolution path. This functionality is to be integrated with existing IT incident management platforms (such as HP ServiceCenter [26]) enabling a new service available to IT service desk agents and the experts in the CEN. The prototypes for user interface are given in Appendix A. The resolution recommendation service is to assist the agents and experts but is not aiming to replace them. Agents/experts benefit from it, when uncertain about possible

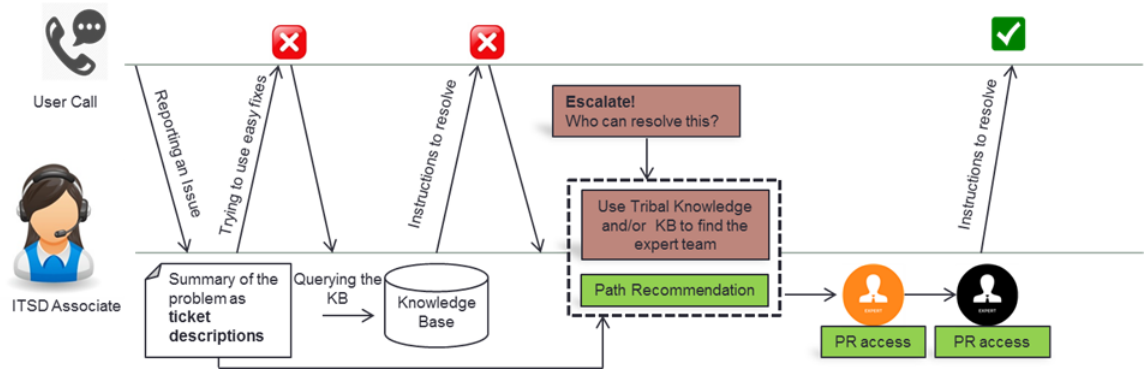


Figure 3.3: An example of ticket resolution augmentation using 'Path Recommendation' service

transfer options, to get an external confirmation on their transfer decisions and SL compliance. Maximum benefit occurs when new agents/experts are hired and have not built their tribal knowledge and are unfamiliar with routine resolution flows on the CEN. Figure 3.3 illustrates an incident resolution scenario from start to finish with the path recommendation service available to the agents and experts. As can be seen, knowledge base and tribal knowledge can still be used; in addition, path recommendation service is available to provide assistance leveraging history of CEN dynamics.

This integration avoids incorrect/wasteful transfers delivering reliable and effective resolution process. Furthermore, it reduces experts' decision time while enhancing capabilities for learning historical tribal knowledge at a global level.

3.2 Domain Data

The data for all the experiments in this study has been drawn from daily interactions within a large insurance company with an online business that serves over 18 million policyholders. This environment has more than 20 thousand staff members who interact with the IT Service Desk to report their IT related issues. IT Service Desk receives more than a

Table 3.2: Example of tickets and their important attributes

Incident ID (P.K)	**Description**	**Resolution**	**Priority**	**Time to Resolve**	**SL Compliance**	**CI**
IM01616342	internet explorer - crashing intermittently and giving debug errors. It keeps shutting down.	Reconfiguring the host file fixed the problem.	P1	1200 Min	Breached	BROWSER
IM01595817	Mail service will not oepn up, user is stuck on the splash screen.	Replaced the corrupt configuration file	P2	2880 Min	Met	LOTUS_NOTES (IBM E-mail service)

million calls per year. Many of them get resolved at the Service Desk leaving around 110 thousand user-perceived tickets to be resolved by the CEN. These tickets are often escalated because either (1) practical assistance is needed, (2) an unusual issue is reported that the Service Desk has not dealt with in the past, or (3) a recurring issue is reported but the resolution process is not defined explicitly in the knowledge base. The CEN in this environment consists of 916 expert groups that include 2476 technical support staff members. There are also more than 7400 Configured Items (CI) in the infrastructure. This research was also privileged to maintain an unfettered access to 220 thousand user-perceived incidents, with more than 380 thousand transfers, captured during a two year period (March 2014 to March 2016).

In this environment, incident management data is stored in a relational database. Some of the important attributes are: (1) ticket descriptions (text), (2) transfer sequence (list of nominals), (3) Configuration Item (nominal), (4) knowledgebase item (nominal), (5) ticket resolution (text), (6) time to resolve (numerical), (7) ticket priority (ordinal), and (8) SL Compliance (binary). All of our experiments were on the *closed tickets* where they were

Table 3.3: Example of tickets and their transfer sequence

Incident ID	***Expert (P.K)***	**Timestamp**
IM01049073	NSC-PCT-NORTHEAST	1
IM01049073	NSC-PCT-INCIDENT-BLUE	2
IM01049073	NSC-PCT-INCIDENT-INDIGO	3
IM01049073	NSC-PCT-INCIDENT-BLUE	4
IM01360487	NF-AES-IPS-MF-AEM	1
IM01054133	NI-FUSION-DESKTOP-SUPPORT	1
IM01054133	NSC-ITSD-AGENCY	2
IM01054133	NI-FUSION-DESKTOP-SUPPORT	3

resolved and for which all the above attributes are already stored. The recommendation model that is developed as a result of this research has to be used on the *open* tickets and in real time. Table 3.2 shows examples of escalated incidents and other attributes. Also Table 3.3 is an example of tickets and their transfer sequences.

In addition to the data characteristics mentioned above we want to draw the readers' attention to the Figure 1.2 where other important attributes such as acknowledgement alerts, and breach alerts are also introduced. This provided critical event data within the operational environment of the enterprise which is typically hard to attain. This data was essential for understanding the CEN dynamics as introduced in this research.

The rest of this chapter is dedicated to the data exploration, and analysis. We systematically cover all observed aspects of current CEN performance, CEN behavior characteristics with respect to content and transfers, and principles that guide beneficial assistance.

3.3 Exploratory Analysis of the CEN

Based on the direction of this research, in order to study the resolution process we particularly explore the TRS length (i.e. the number of experts used to achieve resolution) and

its impact on the following properties of the tickets (1) priority distribution, (2) SL violation, (3) time-to-resolve. Also, towards the end of this section, we study the distributions of time-to-resolve and time-to-acknowledge and compare them against their SL targets.

3.3.1 General Analysis of Escalated User-perceived Incident Tickets

In this subsection we will discuss our preliminary analyses of the environment and our more insightful findings. These analyses have been conducted with a focus on the resolution process. For this purpose we sampled 150 thousand tickets drawn from a time period of 24 months. These tickets yield a total of 254 thousand transfers. (Expected number of transfers per ticket = 1.7)

Priority of Tickets and Relation to TRS Length

Figure 3.4 demonstrates the distribution of prioritized tickets with respect to the length of the resolution path (i.e. TRS length) in the sample. As the TRS length increases, the ratio of P1 tickets to all decreases (Figure 3.4). This implies that experts try their best to resolve their P1 tickets within shorter TRSs. For P1s, CEN exhibits the urgency of resolving the tickets as soon as possible. In contrast, for looser SL targets (P2-P4) experts are less constrained and therefore the CEN is showing a tendency to execute more transfers on lower priorities.

We also discovered three types of transfer behaviors by the experts, happening when CEN is given relaxed SL targets: (1) "progressive delegation" that is transferring to another suitable expert to achieve effective progression towards resolution, (2) "evasive delegation" that is transferring to avoid resolution commitment (happens more for P4s), and (3) "delayed action" that is holding the ticket in the queue until it gets very close to its SL target (happens more for P4s). The cases 1 and 2 are more likely to cause SL violations. Also SL

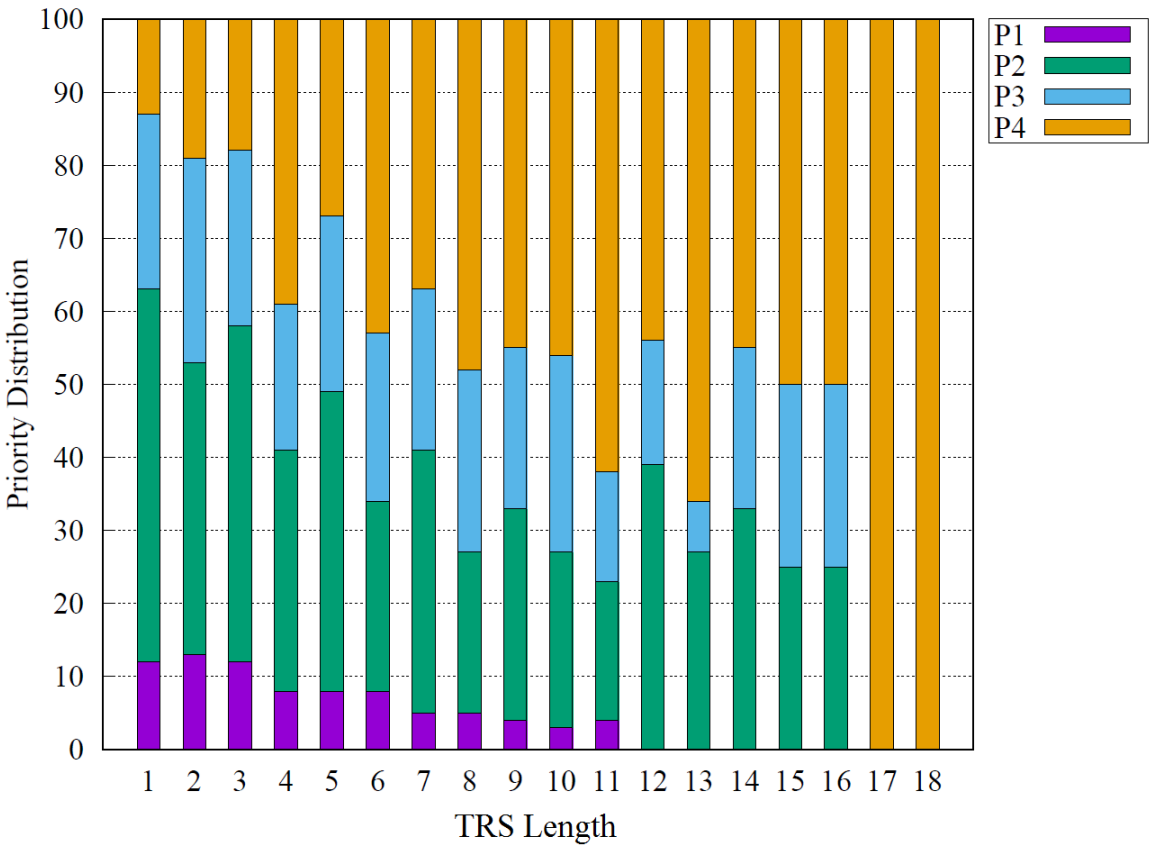


Figure 3.4: Distribution of ticket priorities over number of transfers

violations on single transfer P1s, are speculated to be due to initial misallocation which is typically followed by the expert's hesitation to transfer due to strict SL target.

Furthermore, Figure 3.5 shows cumulative distributions of prioritized tickets over the number of transfers. This is to show within how many transfers what portion of P1s, P2s, P3s, and P4s are expected to be resolved. From this cumulative plot it is concluded that higher priority tickets are expected to be resolved within less number of transfers. As an example, it is evident that experts are more freely transferring lower priority (especially P4) tickets. 64% of P1s are resolved at the first expert, while that fraction is only 52% for P4. In fact we generalize this observation as follows: The lower the ticket priority, the higher the expected number of transfers on the ticket.

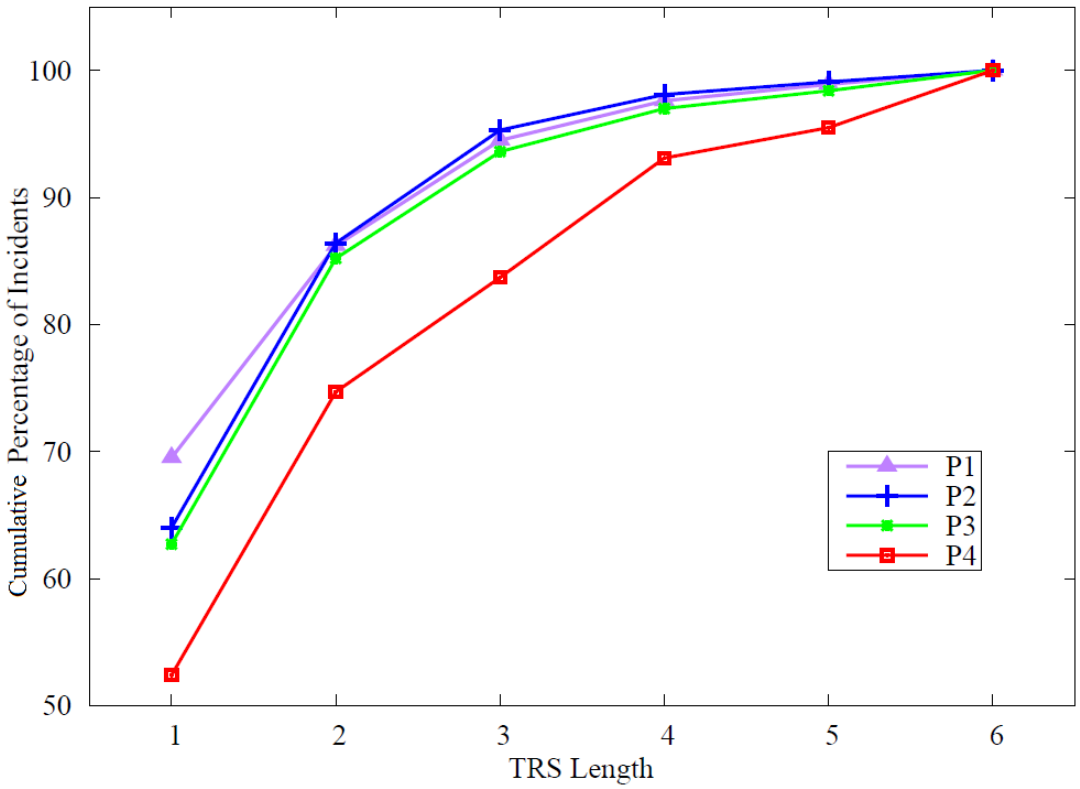


Figure 3.5: Cumulative distribution of tickets over the TRS length

3.3.2 Relating TRS to SLs and MTTR

In this section we present the effect of TRS length on the SL and MTTR. Figure 3.6 portrays distribution of tickets over different TRS lengths and their SL breach ratio. An exponential decay is observed on the volume of tickets as the TRS length increase. In the next subsection, properties of this distribution has been discussed. Also as the TRS length increases there is a linear growth on the expected probability of SL breach. TRS lengths more than 5, can be considered as the tail of the ticket distribution. More drastic breach ratios are associated with the tail of the distribution. Despite the fact that the tail hardly contains any tickets, the high breach ratio makes the tail costly both in terms of resources and business impact. In fact, 18% of tickets fall in the window of 3 to 10 hops where according to our earlier study [43] they cause 38% of the total resource cost.

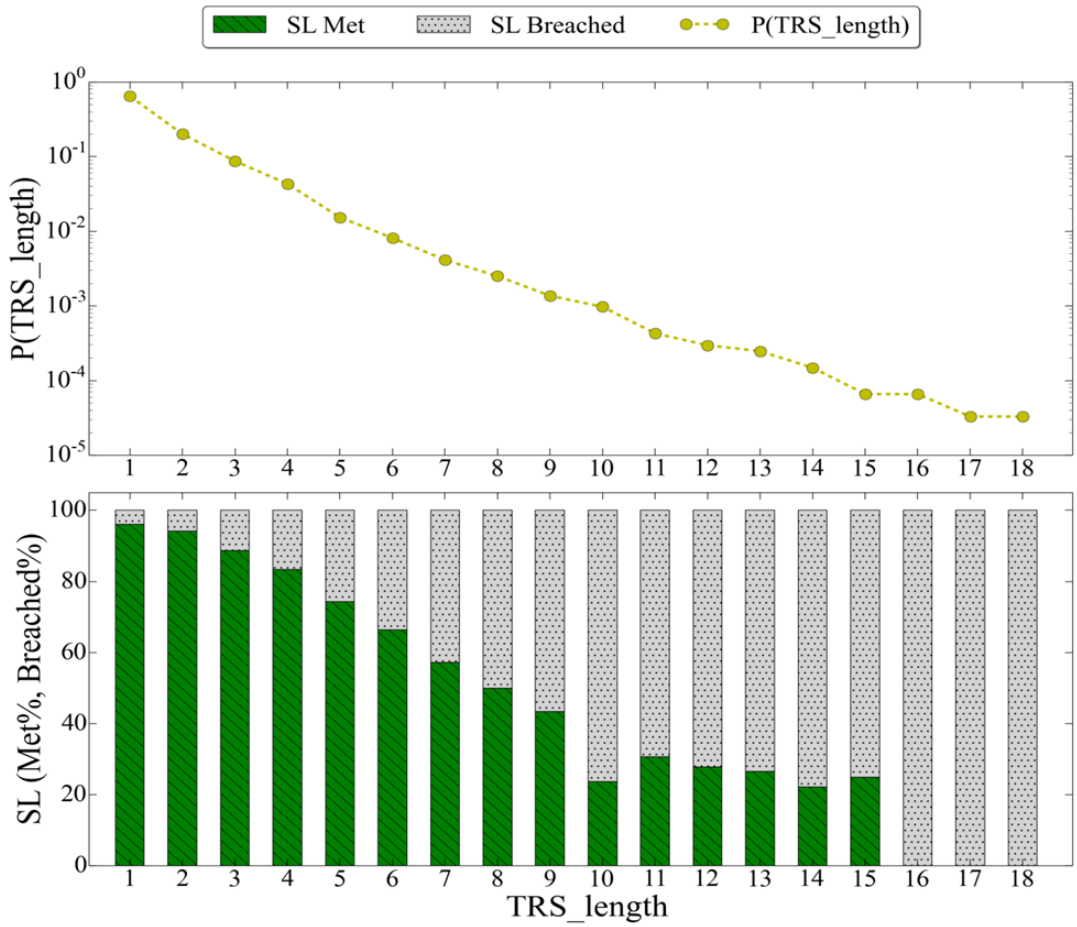


Figure 3.6: Distribution of tickets and SLs over number of transfers

Next, Figure 3.7 shows how an increase in TRS length leads to a linear growth on the Mean Time to Resolve (MTTR). Our sample was found not large enough to represent MTTR of the TRSs where we had more than 10 experts. In other words, the law of large numbers does not apply to the tail of our sample distribution. In order to estimate how MTTR grows with the TRS length (also referred as 'h' for count of hops) linear regression was used for the MTTR of the tickets with respect to number of hops. The R^2 coefficient is verifying the goodness of fit, and here came out to be 0.934. Linearity of MTTR can be interpreted as follows: Each expert that is added to the resolution sequence will add on average 45 minutes to the life-time of the ticket. This linear increase on the MTTR

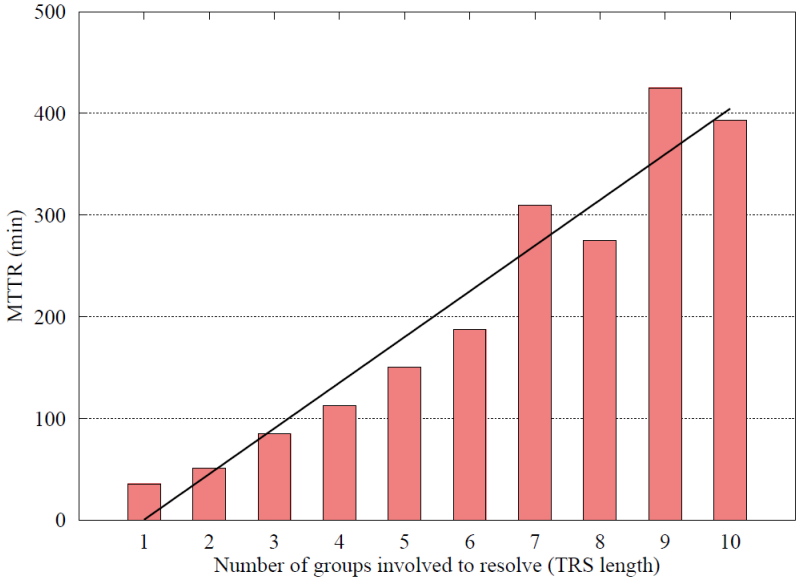


Figure 3.7: MTTR over number of transfers

is the cause of the increase on the probability of SL breach that is discussed above. It is important to note that shorter TRSs do not necessarily imply shorter time-to-resolve, but they are expected to do so. Therefore, at *the level of individual instances we found that MTTR estimation only based number of hops is an ineffective estimation model*. This is due to wide standard deviation of TTR values on different TRS lengths. In chapter 5, we introduce a more rigorous notion for time to resolve estimation that is the expected time to resolve on each specific resolution path.

3.3.3 Probability Distribution Fitting Based on Transfer Counts

As mentioned earlier, ticket mass has shown a sharp decline as the number hops increase (Figure 3.6). Here we want to formalize this, and construct a probability distribution

function as factor of number of hops. An interesting property has been observed in our sample that can be formulated as follows:

$$\forall h : P_{resolve}(h) = \sum_{i=h+1}^{N} P_{resolve}(i) \pm \varepsilon \tag{3.1}$$

This is a recursive probability mass function for any number hops. Here $P_{resolve}(i)$ denotes the probability of resolving a ticket in i hops, N denotes the maximum number of hops in the sample (in this case $N = 18$), and ε in our experiment is a small number less than 0.03. This means for any arbitrary h, the probability of tickets resolved in exact h hops, is approximately equal to the sum of the probability of tickets resolved in more than h hops. This is derived from the following inequality:

$$\forall h : 0.52 \leq P_{resolve}(h|\text{not resolved in previous } h-1 \text{ hops}) \leq 0.56 \tag{3.2}$$

That is for any arbitrary h, given that a ticket has reached its h^{th} hop (not resolved in the previous $h-1$), there is at least 0.52% and at most 0.56% chance of it being resolved right there. This forms a special case of binomial distribution that is known as the geometric distribution. *Experts transfer until they resolve where 'resolution' represents success, and 'making a transfer' represents failure.* A success will halt the trials, and failure brings on another trial. Therefore, here is a non-recursive probability mass function:

$$P_{resolve}(h) = 0.56 \times (0.44)^{h-1} \pm \varepsilon \approx 0.56 \times (0.44)^{h-1} \tag{3.3}$$

This can be written in an exponential form as follows:

$$P_{resolve}(h) = 0.56 \times e^{-0.82(h-1)} \tag{3.4}$$

This represents the probability mass function found for the sample, and can also represent the frequency function as:

$$F_{resolve}(h) = 0.56 \times n \times e^{-0.82(h-1)} \tag{3.5}$$

Where n is the total count of tickets in the sample. Thus the *exponential decay* which was hypothesized earlier is found as in Equation 3.4. This implies that count (or probability) of tickets with a particular TRS length can be estimated given the the size of the sample.

3.3.4 Performance of the CEN with respect to TTR and TTA

Next is to measure how quickly the CEN deals with different priorities. As defined by SLAs, there are two types of time measurements that are crucial for the incident management process. Time To Acknowledge (TTA) that is the time taken by the CEN to identify an expert to be held accountable with respect to a particular ticket, and Time to Resolve (TTR) that is the time taken by the CEN to resolve a particular ticket. There are predefined service level targets for both TTR, and TTA depending on the priority. Figure 3.8 represents the cumulative distribution function of tickets over their Normalized TTR (i.e. TTR devided by target resolution time). Below are some of the key findings:

1. Cumulative curves for the resolution process as compared to the acknowledgement process are less steep. This means the probability distribution of TTR for the tickets is less skewed towards lower values (double digit minutes) as compared to TTA. This also means that resolution process for a ticket is usually lengthier and not as straightforward as the acknowledgement process. In other words, quick acknowledgment on most of the tickets is easy to achieve but quick resolution on most of the tickets is not as much. Applying the same logic, it is easy to quickly acknowledge (solvability of) a hard problem, but it is difficult to quickly resolve one.

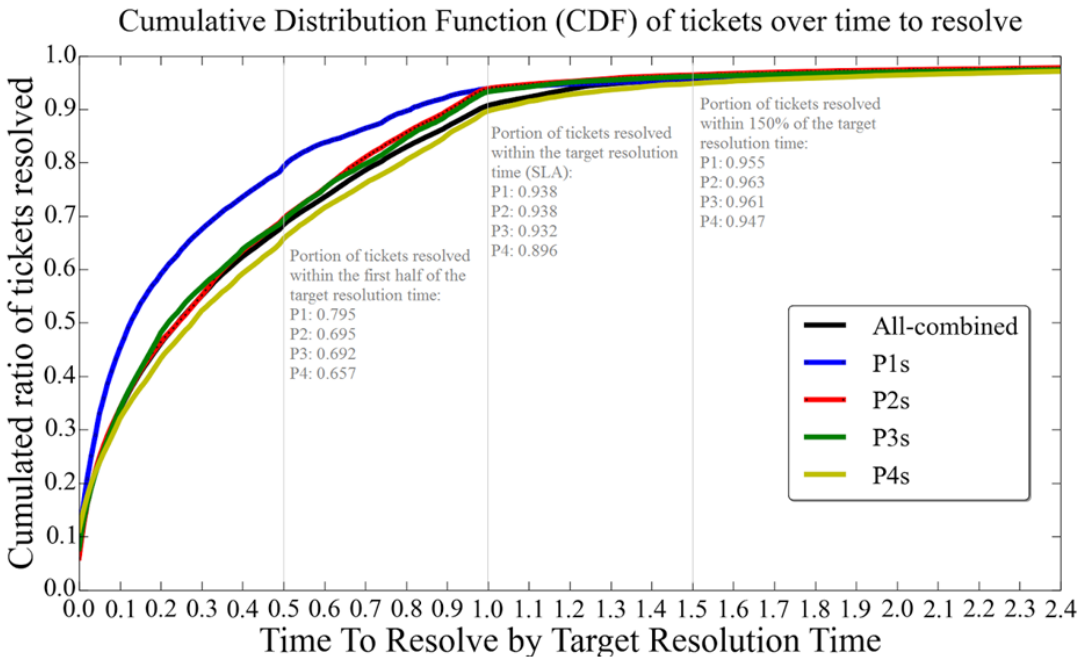


Figure 3.8: Cumulative distribution of time-to-resolve per priority

2. Low priority tickets (P3 and P4) are more likely to breach their resolution SL Target. Patterns of rush and procrastination are observed: Within the first half of the SL target time, P1 and P2 curves have steeper slope than P3, and P4 curves (experts are more rushed to resolve high priority tickets, sacrificing low priorities). Within the second half of the SL target time, P3 and P4 curves have sharper average slope than the P1 curve. (in the second half of the SL target time, experts try to lift and resolve more from the postponed P3s and P4s). More than 10% of P3s and P4s are breached. This is due to (a) experts' prioritization of P1s and P2s and (b) procrastination and underestimation of effort for P3s and P4s.

3. Once P3s and P4s get breached still a high positive slope is maintained indicating continuous resolution effort on low priority tickets (Experts act on these since these tickets can be resolved with a little more effort which was not achieved earlier due to

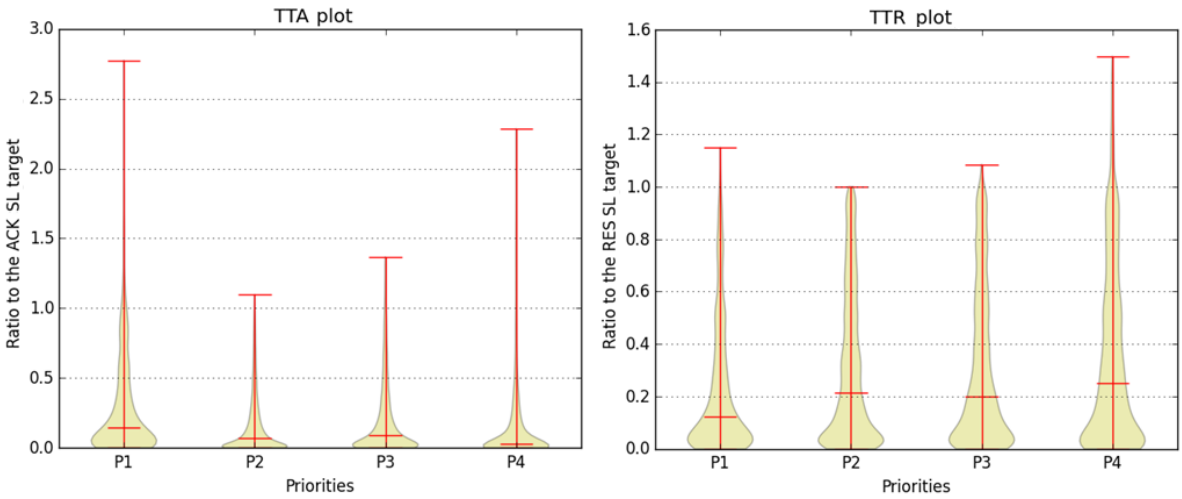


Figure 3.9: Distribution of TTR and TTA per priority

procrastination, or prioritization). On the other hand, once P1s, and P2s get breached, there is very little growth to both curves, and the tail flattens out (experts act as much to resolve these tickets since these are harder to be resolved given the fact that they have already been given enough effort from early stages of ticket creation).

4. A steeper slope within SL target time interval of $[0.9, 1.0]$ is observed for all priorities as compared to its left neighborhood ($[0.8, 0.9]$ and $[0.7, 0.8]$). This implies that the experts try harder to resolve once they get alerted that an SL is about to get breached.

In addition, according to Figure 3.9, we compared the distribution of TTAs against the distribution of TTRs. It is noticeable that acknowledgement is a quicker task as compared to the resolution since acknowledgement distributions are skewed more towards zero.

Based on the TTR violin chart, P2s and P3s are handled regularly in-bound and are shown to be well in control. P1s and P4s are showing more outliers, and a greater standard deviation was observed. This is due to the fact that most of the regular services are prioritized as P2s and P3s. After outlier removal, P1 distribution is effectively ahead of P2, and

P2 and P3 distributions are effectively ahead of P4, indicating experts' 'over-prioritization' on the tickets where higher priority tickets are handled *disproportionately* quicker than lower priority ones.

TTA data has shown more outliers. Acknowledgements have denser heads, and longer tails. Indicating the fact that acknowledgements are less content-sensitive.

3.4 Discoveries Related to Collective Behavior

In this section, TRSs are further studied to discover the collective behavior of their constituents. Particularly, proximity to the resolver, and expert repetition are explored in the context of TRSs.

3.4.1 Proximity to the Resolver

Does proximity to the final resolver on the TRS, imply more contributions to the ticket? To answer this question, we decided to compute the response time of experts on each and every TRS (using the physical timestamps on the transfer logs). We utilized expert's response time as an implicit signal for the amount of work that the expert contributed to the ticket. Figure 3.10 illustrates a key finding: as the distance from the resolver increases on the TRS, the average normalized response time of experts on the ticket decreases. This essentially supports the hypothesis that *most of the contributions towards resolution are often made by the last few experts in the TRS*. The farther the experts from the resolver, the more likely they are to make quick non-contributory transfers.

3.4.2 Repeating Experts: a Potential Signal for Collective Work

We observed that 58.07% of all executed TRSs with a minimum length of 3, contain a repeated node. When the minimum length is increased to 7, almost all tickets are resolved

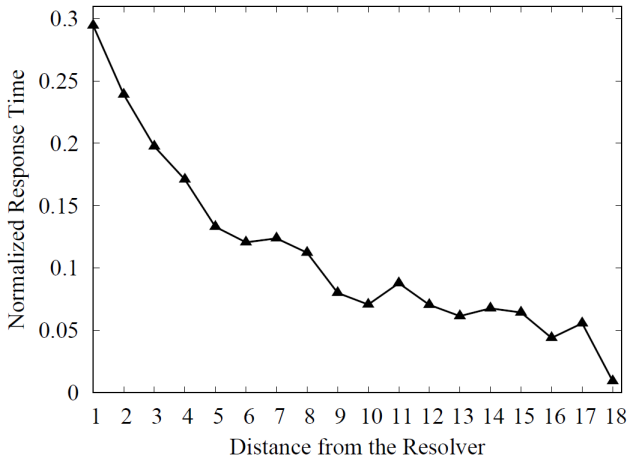


Figure 3.10: Mean normalized response time of Experts' vs. distance from the resolver

by TRSs containing repeated nodes. Also, we observed that 96.32% of executed TRSs that contain repeated nodes, meet their SL targets. Therefore, it is evident that repeated nodes are both *common and fairly effective*.

Also we decided to study the likelihood of repeated nodes in frequently executed TRSs (i.e. workflows). Interestingly, our results in Figure 3.11 suggest that TRSs that are used more often for problem solving happen to be more likely to have repeated experts in their sequences. Workflow-driven problem solving is executed according to a set of routine established processes. According to Figure 3.11, routine workflows are very likely to contain repeated experts, and we believe *experts in routine workflows that work on the ticket more than once are the process owners who moderate the collective resolution process between multiple parties*.

In Chapter 4 we will return to routine workflows by claiming that routine workflows are associated with frequently occurring content. Here, the fact that most routine workflows contain collective behavior makes us conclude that there exist a sizable subset of tickets that contain frequently occurring content, and collective behavior is utilized in resolving them.

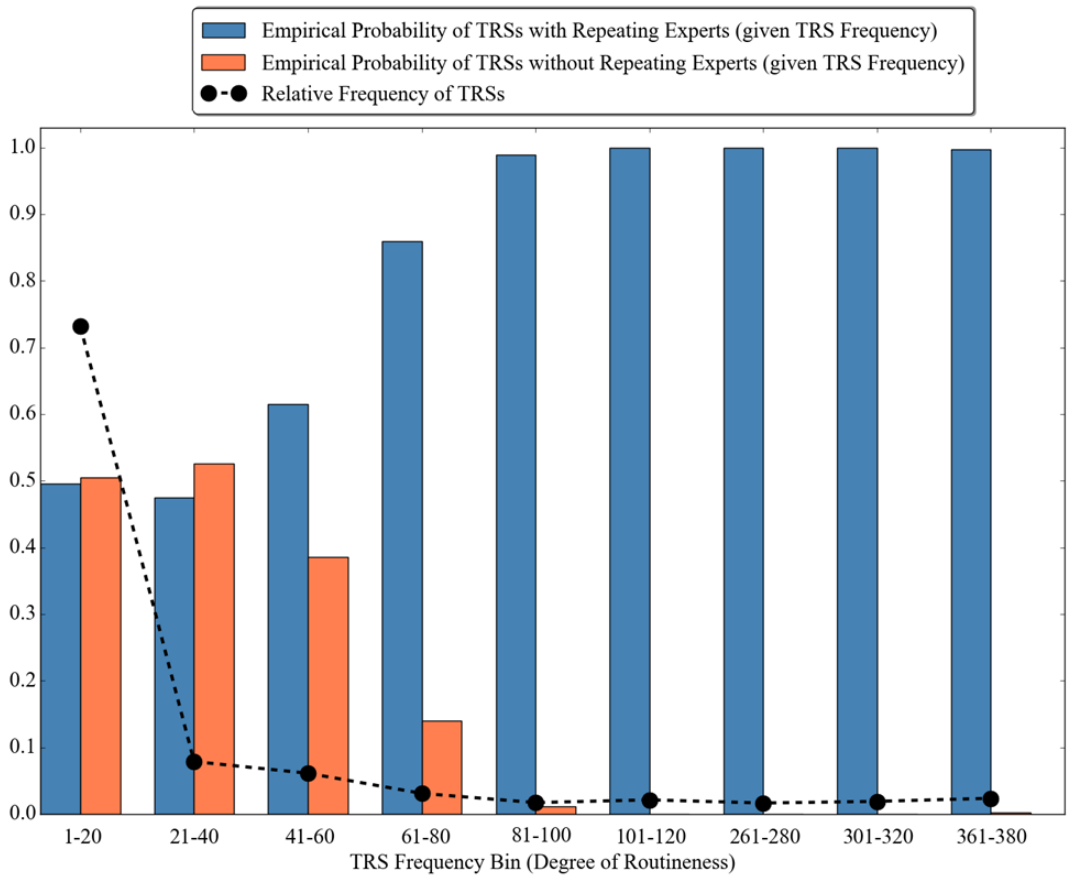


Figure 3.11: The relationship between TRS frequency and the probability of having a repeating expert

That subset will create a unique opportunity towards augmentation of the CEN. More about that will be discussed in Chapter 4.

3.5 Discoveries Related to CEN Structure

In this section we present the terminology and formalism for the CEN, and the Enterprise Taxonomy. Then the relation between the two is studied in order to compare dynamic problem-solving behavior to the static organizational design.

3.5.1 CEN Terminology and Formalism

This formalism has been motivated by our analysis and insights of collective behaviors that exist within the IT service support organization and can be captured by networks [14]. By presenting the conceptual model the insights can later be presented more succinctly.

To start with we define a Collective Expert Network (CEN) on a set of resolved tickets T as a directed graph where experts and transfers represent vertices and edges respectively:

$$CEN(T) = (Experts, Transfers) \tag{3.6}$$

For a ticket $t \in T$ a resolving sequence is $t.rs$:

$$t.rs =< e_{(1)}, e_{(2)}, ..., e_{(k)} > \tag{3.7}$$

Here $e_{(i)}$ is the ith expert which was working on t and received t from $e_{(i-1)}$, and transferred it to $e_{(i+1)}$ (<> denotes an ordered list). The last expert in the sequence achieved resolution and is noted as the 'resolver' ($t.resolver$). Note that the above definition accommodates duplicate elements in the sequence. Therefore, any resolving sequence is a *walk* on the CEN. It is important to note that in network theory the definition of a path does not entail duplicate vertices but here a resolving sequence does. So for simplicity we call any resolving sequence a *path* even though it contains duplicate vertices. The $Experts$ set is defined by the union of experts that have worked on at least one ticket in T:

$$Experts = \bigcup_{t \in T \,\wedge\, e_i \in t.rs} e_i \tag{3.8}$$

$Transfers$ is the set of expert pairs of the form (a, b) which is a directed edge that belongs to $Transfers$ if there is at least one ticket transferred in T from expert a to expert b. Formally:

$$(a, b) \in Transfers \quad if\ \exists t \in T \mid < a, b > \sqsubseteq t.rs \tag{3.9}$$

Here $< a, b >$ denotes an ordered pair and $\sqsubseteq$ is the notation we use for a 'contiguous subsequence'. Also note that we explicitly add self-loops to indicate resolvers as follows:

$$(a, a) \in Transfers \quad if\ \exists t \in T \mid t.resolver = a \tag{3.10}$$

Considering only edges and vertices as a base definition for CEN, next we enhance the directed graph to make it a *weighted* directed graph. Let the set of all tickets that got transferred from a to b be denoted as T_{ab}, then:

$$T_{ab} = \bigcup_{t \in T \wedge <a,b> \sqsubseteq t.rs} t \tag{3.11}$$

Now we define a weight for each edge (a, b) as the count of tickets in T that got transferred along (a, b): $w_{ab} = |T_{ab}|$ Also a self-loop (a, a) is weighted as w_{aa} and evaluates to the count of tickets resolved in a. In order to obtain insights about the CEN, we propose a transformation on the weights introduced above. This transformation yields a *Markov Chain* for the CEN which is a 'memoryless' probabilistic directed graph. The resulting Markov chain is also atypical (compared with [60]) since it contains self loops characterizing resolvers. w'_{ab} is the probability that a ticket was transferred to b after that the ticket was received at a. Formally that is evaluated as:

$$w'_{ab} = P(b \mid a) = \frac{w_{ab}}{\sum_{[c \in Experts \wedge (a,c) \in Transfers]} w_{ac}} \tag{3.12}$$

Also w'_{aa} can be interpreted as the probability that a resolves a ticket after receiving it. To illustrate the Markov representation of the CEN in our case, the Tarjan algorithm [72] was used to obtain strongly connected components. Figure 3.12 illustrates a strongly connected component of the CEN in which each vertex is reachable from any other vertex. Note that low-frequency edges ($w_{ij} < 60$) were removed upfront to focus on dominant transfer patterns. Some of the insights from this are: (1) the expert group 'Queue' almost

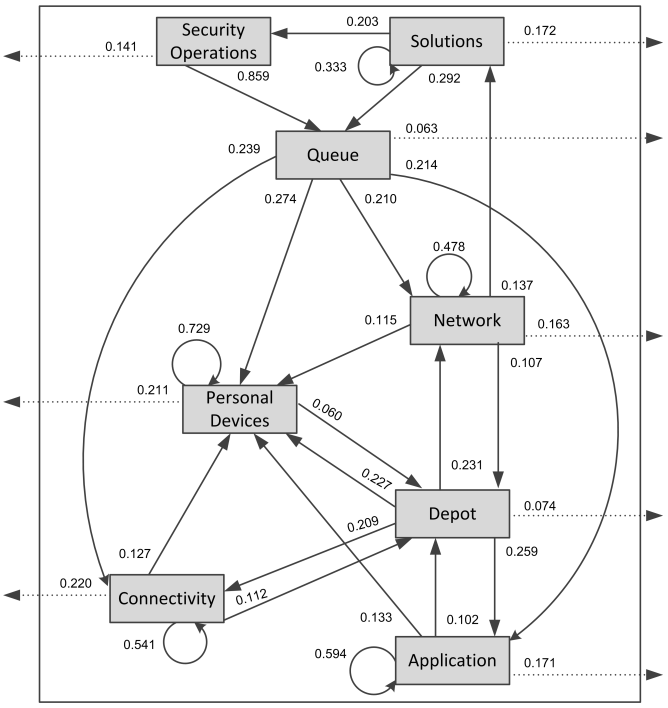


Figure 3.12: A strongly connected component of the CEN within the enterprise with edge weights as conditional transfer probabilities. Self-loops represent resolution

evenly distributes all of its tickets among 'Connectivity', 'Personal Devices', 'Network' and 'Application'. (2) 'Queue' does not resolve any tickets. (3) 'Application' resolves more than half (0.594) of all the tickets it receives, and transfers almost a third of its non-resolved tickets (0.133) to 'Personal Devices' which is then more than 70% likely to get resolved at 'Personal Devices'. Thus the figure captures dynamics of workflows from resolving sequences illustrating that the nodes of a CEN play specific roles in a more global context. For example, 'Depot' does not resolve tickets, it appears to mediate among four other experts. More detailed insights about roles are discussed next.

Enterprise Taxonomy (i.e. ET) is a view of the CEN constructed from transfer labels obtained within the enterprise system. These labels were found to be related as a tree. Each internal node of this tree represents a conceptual scope of responsibility (abstract role) and

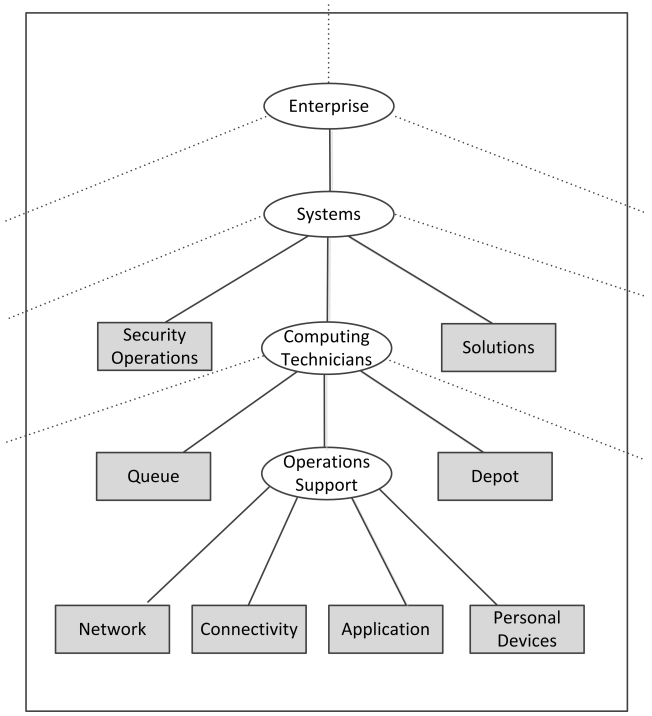


Figure 3.13: Enterprise taxonomy tree associated with the connected component in Figure 3.12

each leaf represents an expert group (concrete role in the CEN). In this structure, a child is a subarea of its parent. In knowledge representation terms, if a child and its parent are both internal nodes they have a 'part-of' relationship. Otherwise the child is a leaf and has an 'instance-of' relationship. Figure 3.13 illustrates an ET corresponding to the CEN of Figure 3.12.

Transfer distance: For each ticket t we define transfer distance $t.td$ which is the average pairwise distance on the taxonomy tree between consecutive pairs of expert groups in $t.rs$. Formally:

$$t.td = \frac{1}{\mid t.rs \mid -1} \sum_{j=1}^{|t.rs|-1} d_{ET}(e_{(j)}, e_{(j+1)}) \tag{3.13}$$

where d_{ET} is the pairwise distance function on the ET. In the next subsection, the transfer distance is used for further analysis.

3.5.2 ET structure and a Semantic Representation for a Transfer

CEN transfer knowledge has semantic dependency associations: We found that there are semantic association patterns in the ET of the CEN. To show this we contrast two CEN views - the network view (Figure 3.12) and the taxonomy-based view (Figure 3.13). With Figure 3.13 we found the ET tree labels identify expert groups based on semantics of the knowledge that they possess related to: (1) a technology or application, (2) a region of the physical facility, (3) a major enterprise project, (4) a mediator or resolution role, or (5) a virtual node representing a collection of sub nodes. The labels have emerged over time and are locally used by humans interacting with the workflow routing menu of the enterprise system (without any assistance). The labels were found to form the ET tree that makes explicit (in Figure 3.13) the knowledge associations that are not shown in the view of Figure 3.12. With this ET tree as the basis we also found that as the TRS length increases, the average $t.td$ also increases as shown in Figure 3.14. This implies that *tickets with longer TRSs are more likely to have long-distance transfers on the ET and this signals increased incident complexity due to expertise needed from distanced subtrees*.

3.5.3 Relating CEN Execution to ET Structure

A research question to be addressed here is: "Is there any relation between the ET, and the CEN problem-solving execution patterns?" This is an attempt to relate the IT organizational design with the IT process execution. To address this research question first we need to define metrics to map the CEN execution to ET structure and then use them to evaluate the efficiency of the existing ET. This could further lead to a set of re-organization recommendations to best suit the collective transfer patterns in the CEN.

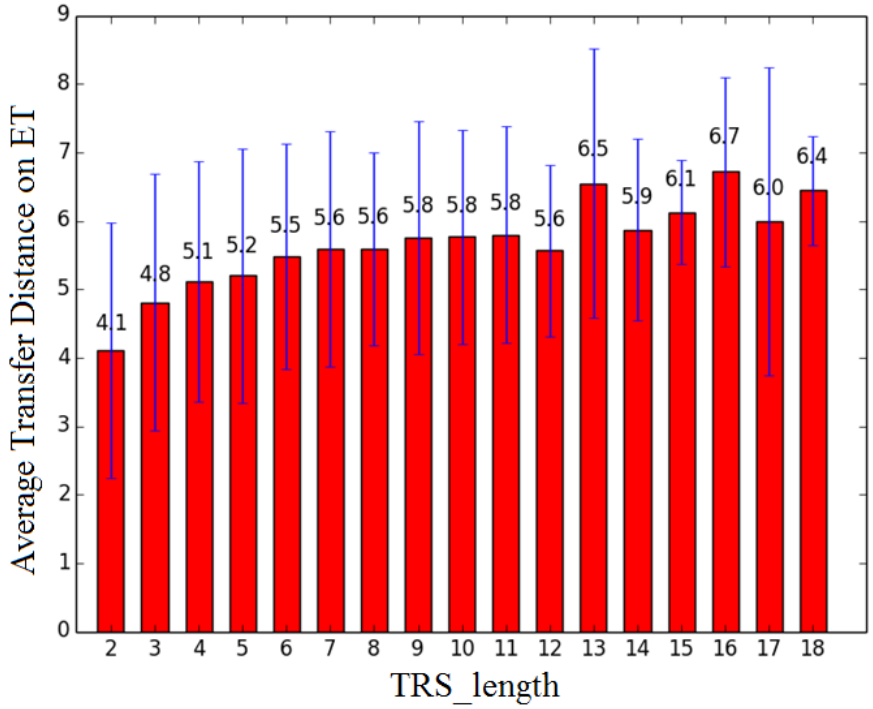


Figure 3.14: Average transfer distance on ET grouped by TRS length

Our solution is to (1) identify connected subgraphs in CEN such that each represents a 'collective problem-solving unit', and for each collective unit (2) define a 'diameter' on the ET. As a result of that, the relation between the 'connectivity strength' of the collective units and their corresponding diameter on the ET is discovered.

Collective problem-solving units are obtained by computing Strongly Connected Components (SCCs) on the CEN. Finding SCCs is performed using Tarjan's algorithm [32]. Connectivity strength is ensured by using a minimum edge weight, that is by retaining only the transfers in the CEN which satisfy the minimum edge frequency. Assuming that S is

a strongly connected component in the CEN, we introduced the diameter of S on ET as follows:

$$Diameter(S, ET) = (\mathbf{card}(edge\_set(S)))^{-1} \sum_{\forall(e_i,e_j)\in edge\_set(S)} d_{ET}(e_i, e_j) \qquad (3.14)$$

where: $edge\_set(S) = \{(a, b)|a \in S \wedge b \in S \wedge (a, b) \in Transfers\}$

A larger diameter value for a SCC implies that experts in that SCC are structurally more scattered across the ET. In contrast, a smaller diameter value for a SCC means that experts in that SCC are having a deeper closest common ancestor in the ET and thereby are less scattered across the ET. Algorithm 1 is showing how to obtain a list of ET Diameters corresponding to the SCCs of the CEN. Note that the algorithm takes a minimum frequency for the edge weights to filter out the low frequency edges at the beginning, and returns a list ET diameters corresponding to each collective unit (i.e. SCC) of the CEN.

Algorithm 1 `Diameter_Generator`

Input: $Experts$, $Transfers$, ET, min_freq
Output: $diam\_list$
$CEN = (Experts, Transfers)$
$filtered\_CEN = (Experts, \{(a, b)|\forall(a, b) \in Transfers, w_{ab} \geq min\_freq\})$
$SCC\_list = \mathsf{Tarjan}(filtered\_CEN)$
for *component in* SCC_list **do**
    if $component.size \neq 1$ **then**
        $diam\_list.add(\mathsf{Diameter}(component, ET))$
    end
end
Return $diam\_list$

Figure 3.15 illustrates that experts in strongly connected components of a higher-frequency-threshold CEN have a lower average diameter on the ET. This implies that *transfer in*

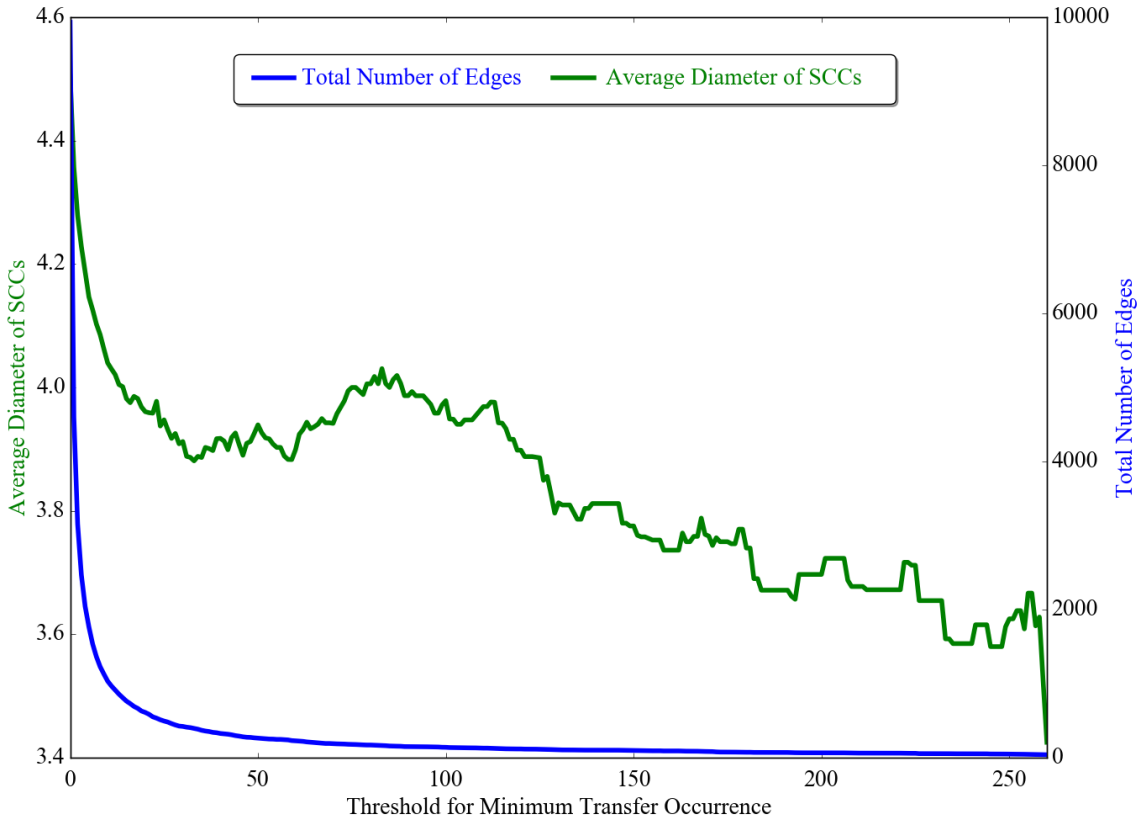


Figure 3.15: Average diameter of SCCs on the ET when varying the minimum transfer frequency

prominent collective problem solving units occur between experts that are structurally very close to each other. This finding clearly relates the CEN to the ET and addresses our earlier research question. From this vantage point, it can be concluded that *enterprise taxonomy (ET) is designed in such a way to encourage collective problem solving between experts that are close to one another* (which is mostly being followed by the CEN).

These findings also open a new avenue of research about "recommending a re-organization to the ET in order to reduce the collective problem solving cost on the CEN" where it can be framed as an optimization problem for re-structuring the ET with an objective of minimizing the average diameter of SCCs subject to structural constraints.

Chapter 4: Recommendation Framework with Routine/Non-routine Classification

As mentioned in the introduction, within today's complex cloud operations made up of layers of technology and application services, customer-perceived incidents often arise in the hundreds or thousands daily. These must be resolved by IT Service Management goals that are critical parts of any service level (SL) agreement:

- *Resolution Goal*: the problem must be resolved by restoring service to the business customer's satisfaction; and
- *SLT Goal*: the resolution process must meet time constraints set by Service Level Targets (SLTs) agreed-to with the customer based on the priority of the ticket.

The Resolution Goal is addressed by logging incidents as tickets and then transferring them to the knowledgeable experts (selected from among many) with skills to contribute to the problem resolution. In the real world, SL is a *time-and-satisfaction-based* metric that is defined for and contracted with customers of different lines of business.

Our technical goal is to develop a statistical learning framework that recommends the best set of transfers to guide experts to collectively work on a ticket and meet SLTs. The framework that is presented in this chapter is designed to be applicable to any service support environment characterized by *a small number of workflows that resolve a majority*

Table 4.1: Key terms

Term	Description
Expert	A technical support team with specialized knowledge and particular set of skills, and responsibilities
Ticket	t a ticket instance with content/attributes, T a ticket set
SLT	Target resolution time defined for each $t \in T$ chosen according to a predetermined priority level.
TRS	Ticket Resolving Sequence of experts for t. R-TRS is a TRS that is labeled as 'Routine'. RecTRS is a recommended TRS (i.e routing recommendation)
TTR	Time To Resolve (i.e. resolution time) of a ticket. Only defined for the resolved tickets.
ETTR	Expected Time To Resolve of a ticket. Used for arriving tickets to estimate their TTR.
MTTR	Mean Time To Resolve computed for T.
MSTR	Mean Steps (i.e. transfers) To Resolve computed for T.

of the tickets. In other words, the proposed solution benefits any environment with an observable *Pareto distribution* [48] of tickets over the resolution workflows.

The concept of 'workflow' is about specific experts that commonly work sequentially on a ticket and transfer it along to achieve resolution. Given a ticket, the sequence that results in the *resolution of the ticket* is referred to as a TRS for that ticket. A TRS of any ticket can be reflected as a *path* on the CEN (identified in Section 1.4). We assume that the TRSs captured in the historical incident-resolution database form a digital trace (i.e. the set of transfer sequences) of the best efforts of the experts thus far. We will show, however, these efforts often fail to meet service levels on longer transfer sequences. This leaves opportunities for CEN improvement with *automated recommendation assistance*. In Table 4.1 we present the key terms and common abbreviations used throughout Chapters 4, and 5.

In summary, in this chapter we establish that: (1) on frequent paths the SLs are very likely to be met, and (2) the frequent ticket content is associated with frequent paths (learned workflows) and therefore are also likely to successfully meet the SLs. Thus the research method is to make explicit the global knowledge exhibited by the CEN on frequent content and SL-achieving TRSs (i.e. paths that resolve) and use this to prevent ticket misrouting on frequent content. This is accomplished by splitting the digital trace into: (1) a trustworthy set which is used for probabilistic sequence learning and recommendations to the human experts, and (2) the remaining unreliable set which is used to signal anomalies in the content to draw early human attention within the resolution process. We implement this with a two-level classification framework that is experimentally shown to: (1) improve the precision of recommendations by 34% over existing content-aware sequence models; (2) improve Mean-Time-To-Resolve by 7%; (3) reduce SLT breaches by 10%; and (4) maintain a high level of trust. The validation uses held-out data generated in the production environment of the enterprise.

4.1 Analysis of the Unassisted CEN

In this section we add further details to pinpoint the causes for poor performance. Existing poor performance despite the enablement provided by current processes and enterprise systems provides an understanding of the opportunities for improvement. Then we extract principles for recommendations that address specific causes in a manner consistent with CENs own natural behaviors. With the data analysis below we systematically cover all observed aspects of current CEN performance, CEN behavior characteristics with respect to content and transfers, and principles that guide beneficial assistance and specific experiments.

Table 4.2: Priority of ticket related to the breach ratios

Priority	Business Impact	SLT	% of all	Breach Ratio%
1	Significant	14 hrs	10.8%	6.1%
2	Moderate	34.5 hrs	45.1%	7.6%
3	Minor	46 hrs	25.5%	8.1%
4	Negligible	115 hrs	18.6%	10.0%

4.1.1 Current CEN Performance

Incident Context and CENs Digital Trace: The digital trace of CEN problem solving has 7250 distinct paths that resolved incidents that were generated from over 7400 Configured Items (CIs) in the IT infrastructure. The operational data from the enterprise analyzed here consisted of 149,000 user-perceived tickets reported over a period of 13-months and resolved by 916 unique experts through 267,721 transfers.

The priority levels are set between P1 (highest priority and impact) to P4 (low priority and impact). The SLT is more relaxed for lower priorities. If the SLT is not met the ticket is said to 'breach' the SL. Table 4.2 depicts that the CEN more often resolves highest priority tickets within SL targets, and SL breaches occur more with lower priority tickets.

Collective problem solving and performance: Longer TRSs cause more difficulties in SL compliance. To prove this, our objective is to examine the TRS length of the tickets against their breach ratio. Given our ticket set, Figure 4.1 demonstrates (1) $P(|t.rs| = TRS\_length)$, that is the probability distribution of tickets on TRS lengths, and (2) $P(t.ttr > t.st \mid |t.rs| = TRS\_length)$, that is the breach ratio of tickets conditioned on their TRS length. Please note that for a sample ticket t, $t.rs$ denotes the TRS for t, $t.ttr$ denotes the time-to-resolve of t, and $t.st$ denotes SLT for t.

Observations: (1) The CEN is able to resolve most of its tickets via short TRSs that is $P(1 \leqslant |t.rs| \leqslant 4) = 0.79$. More generally, Figure 4.1 illustrates an *exponential decay* in

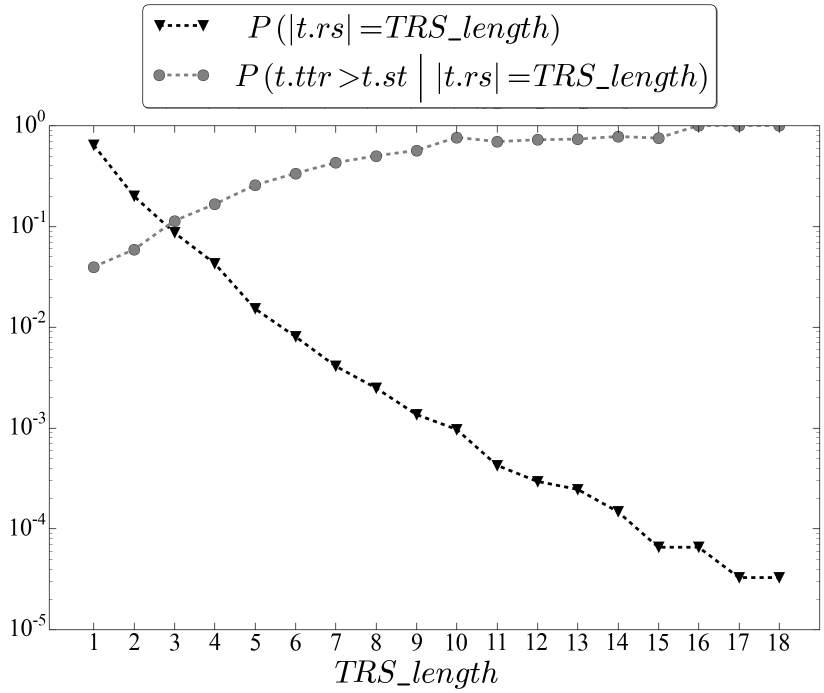


Figure 4.1: Distribution of tickets on TRS length, and breach ratio of tickets per TRS length

the volume of tickets as the TRS length increases. In other words, a transfer chosen by the CEN to be executed on a ticket is highly expected (with the probability greater than 0.5) to resolve that ticket. This establishes a *power law* [66] distribution. As mentioned in Section 3.3.3, we experimentally found the power law function that represents the probability of a ticket being resolved by a TRS in h hops where $h \in \mathbb{N}$:

$$P_{resolve}(|t.rs| = h) = 0.56\ e^{(-0.82(h-1))} \tag{4.1}$$

(2) Also per Figure 4.1 as the TRS length increases, the probability of the tickets breaching their SLT increases. Although longer TRSs are unlikely to occur, they are highly likely to breach their SLT. This presents an opportunity for improvement by avoiding wrong transfers which are the leading cause of longer resolving sequences, thus saving many tickets from inevitable SLT violations.

Also our earlier observations in Section 3.5.3 showed as the TRS length decreases the average transfer distance on the ET also decreases. Thus transfers on more frequent and shorter TRSs (where SLTs are met more often) are more likely to be aligned to the ET tree structure (i.e. low average transfer distance). In other words, more frequent TRSs are (1) shorter, (2) less likely to breach SLTs, and (3) better conforming to the ET structure. This provides the motivation for conditional conceptualization of the entire TRS as a collective problem-solving unit with global workflow characteristics.

4.1.2 Digital Trace Characteristics – Content & Transfer Knowledge

In the previous subsection we established that the CEN could better benefit the business from recommendation assistance on longer transfer sequences that are (1) more likely to breach SL goals, and (2) that the entire sequence has global associations that are tacit and also difficult for the CEN to exhibit. Given the observations, the next related questions are: (1) Is there machine learnable regularity exhibited in the paths of the CEN; (2) How are the content and the paths related? and (3) How do we ensure that the CEN trusts the recommendations?

Regularity of the paths: Many of the paths are very common reflecting the fact that the CEN's digital trace of *collective problem solving is not erratic*. The related analysis is in Figure 4.2. This figure also shows that the *Pareto Principle* holds: 5.5% of paths resolve 80% of the tickets. This skewed distribution of the tickets over the paths helps identify the subset of the paths that overcomes the challenge of *data sparsity*, leading to effective machine learning on that subset. Next we found that frequent content was also associated with frequent paths that are more successful.

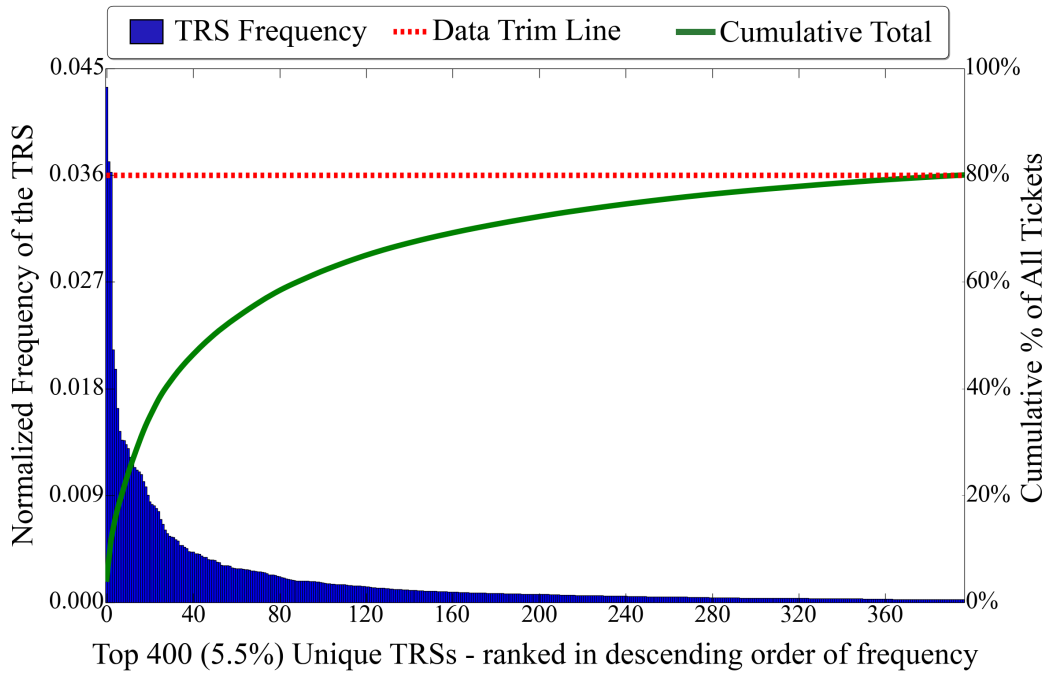


Figure 4.2: Normalized frequency of paths – pareto chart

Observations: Regularity of global network knowledge is exhibited by frequent paths (refer to as 'Routine' paths) that mainly resolve certain frequent content. From the machine learning standpoint, the goal is to choose a subset of all the paths, which contains frequent and well-separated paths (classes) for a multiclass classification algorithm resulting in a generalizable trustworthy classifier. From the standpoint of benefiting the CEN practically, the goal is to provide recommendations of routine paths that contain high-performing global patterns and thus prevent tickets with frequent content from taking suboptimal 'non-routine' paths.

4.2 Machine Learning Framework Goals: Trustworthy Recommendations

With the potential for beneficial assistance established above along with the business motivation, we next formulate the machine learning research problem to address the observations immediately above.

The goal is to develop assistive recommendations where the machine itself (1) determines the conditions under which it can learn and recommend based on some trustworthiness criteria; (2) learns the global network knowledge in terms of TRSs that can assist the CEN to meet SL; and finally (3) flags where trustworthy recommendations are not possible and in these cases increases the reliance on human problem solving (i.e. without recommendations) and put effort into dynamic knowledge creation. Thus this approach requires the machine to *differentiate* between the **Routine (R)** problem solving where it learns and recommends to meet SLs more effectively; from the **Non-routine (NR)** where the human experts do better to achieve SL and recommendations are not trustworthy.

Next, in Section 4.3 we will show that the breach ratio of the R-TRS class is found to be almost one-fourth of the NR-TRS class (2.26% to 8.25%). That is if a recommendation correctly saves a ticket from a non-routine path by recommending a R-TRS then there is 72% *reduction* in its likelihood to breach. In our study, 14% of all the tickets had regular content that got misrouted to a NR-TRS and the recommendation system could save these cases. This leads us to the conclusion that the expected breach ratio reduction overall through beneficial recommendation is $14\% \times 72\% = 10\%$.

4.3 Labeling Strategy – Regularity of Content vs Regularity of TRS

We first establish the critical hypothesis that the more likely inputs of the CEN are associated with the more likely outputs. To do this we first develop functions that *independently* quantify the regularity of the content (input) and of TRS (output). Note that if content can signal for strong association with frequent TRSs then predicting only among frequent TRSs leads to a more accurate classification outcome as opposed to predicting among all TRSs in the first place. This solution is particularly favored where a small portion of distinct TRSs are used to resolve majority of the tickets in the history. This is discussed further next.

4.3.1 High Likely Content is associated with Routine TRS

Here we first define routineness measures for content and distinct TRSs separately. Then we demarcate R-TRSs from NR-TRSs through the following ***labeling strategy***: we attempt to find a subset of distinct TRSs that their content likelihood distribution is in maximum distance from that of their complement set. Then members of the set with higher average content likelihood are labeled as R-TRS, and members of its complement are labeled as NR-TRS. To avoid trivial solutions, the aforementioned objective is subject to a constraint that enforces minimum ticket coverage on both subsets.

Routineness of TRS (discrete metric):We characterized regularity of a distinct TRS as the frequency of tickets resolved by that distinct TRS in the training set. The more frequent a distinct TRS is used, the larger its routineness value becomes. Our data illustrates the skewed distribution of tickets over distinct TRSs (Figure 4.2). This graph is a *Pareto chart* that represents normalized frequency distribution of distinct TRSs in descending order along with cumulative percentage of tickets. In the figure the top 5.5% of the most frequent distinct TRSs (400 distinct resolution sequences out of 7,250) are depicted which

are used as resolving paths for 81% of tickets. The other 19% of tickets use 94.5% of remaining less frequent distict TRSs. This follows the well-known *Pareto principle* that implies most of the ticket probability mass is accumulated on a small portion of TRSs. The Pareto principle is an inherent property of CENs. Also it is acknowledged in the Machine Learning community that more number of classes in a multiclass setup inevitably leads to higher misclassification rate [16]. Here the Pareto principle is leveraged to build a multi-class classifier that *only trains on a small portion of distinct TRSs denoted as R-TRSs, and can precisely recommend on a large portion of tickets.*

Routineness of content (continuous metric): To characterize the input content of the CEN we treat it as meaningful word sequences with a log-likelihood metric that measures the probability of the word sequence in the content of a ticket as *Content Log-Likelihood*(CLL):

$$CLL(t;\lambda) = \frac{1}{|t|}\sum_{w_i \in t} \log \hat{P}(w_i \mid w_{i-1};\lambda) \tag{4.2}$$

where:

$$\hat{P}(w_i = b \mid w_{i-1} = a;\lambda) = \frac{\#(ab)+\lambda}{\#(a*)+\lambda\,|V|} \tag{4.3}$$

Here w_i is the ith word token in a ticket t, and λ is a smoothing parameter. Normalization (i.e. division by $|t|$) is needed to establish a fair measurement for significance of words regardless of the number of word tokens in a ticket. Next, the probability of a word w_i in the context of w_{i-1} is computed. We use a *bigram* language model [11] where '$\#$' represents a function that computes the frequency of the given word phrase in the ticket corpus, and $*$ represents any word in the corpus dictionary. $|V|$ is the size of the corpus dictionary. In our data set, the CLL values of different tickets range from -4 to -14, with -4 signifying the most likely content.

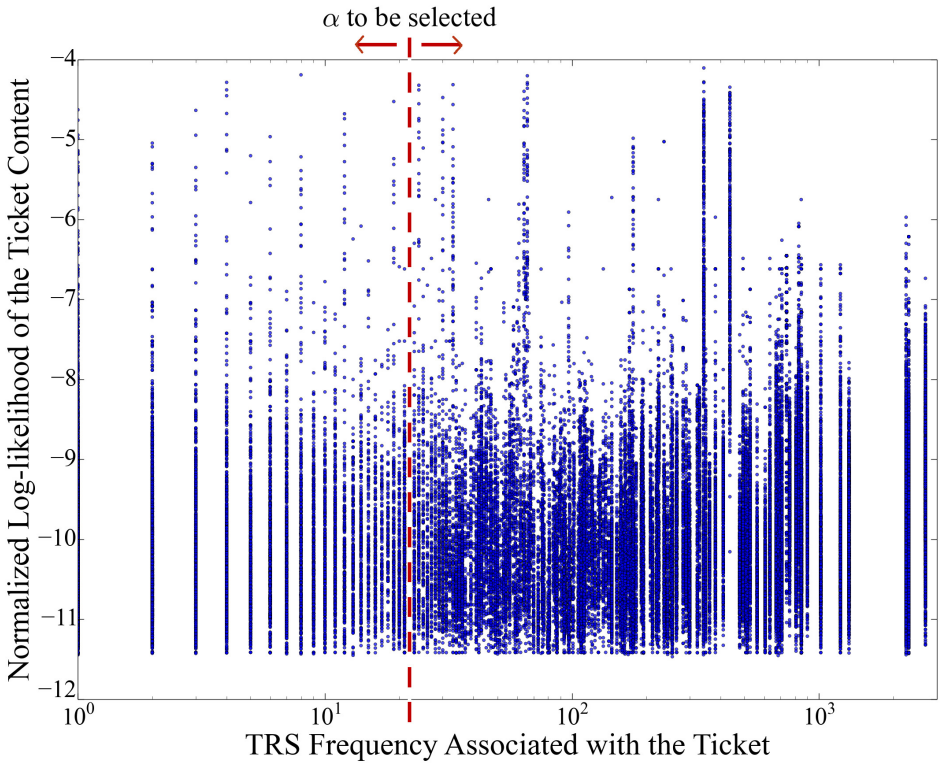


Figure 4.3: Projected tickets in CLL-TRS_frequency space – high density of tickets in the center right area means tickets with more frequent TRSs are very likely to have regular occurring content.

Figure 4.3 projects each ticket into a two dimensional space with the log likelihood of content (y-axis) and the TRS frequency (x-axis which is also on a log scale to accommodate the sparse tail of TRS distribution). As can be seen, there are considerable number of tickets that are having a highly likely content and are resolved by a highly frequent distinct TRS. Therefore, we can now look for an α-split on the x-axis that maximizes the distance between (1) CLL distribution of the tickets that have a TRS frequency less than α (to be labeled as NR) and (2) CLL distribution of the tickets that have a TRS frequency more than α (to be labeled as R). For simplicity, we define distance between two CLL distributions as the difference between the mode values of the two distributions.

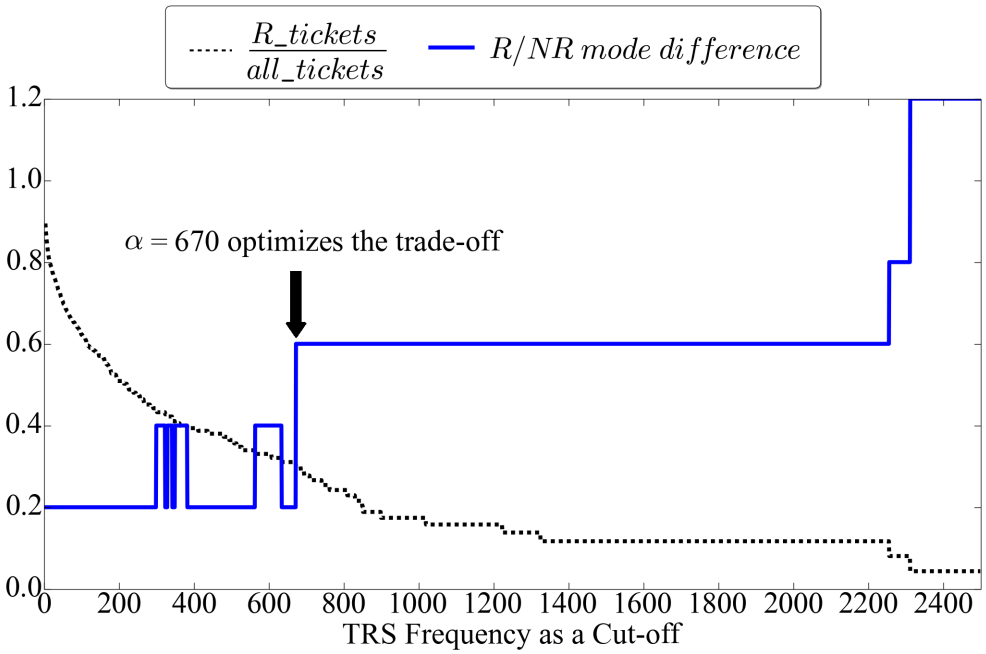


Figure 4.4: $\alpha = 670$ allows us to separate R from NR effectively.

Figure 4.4 illustrates the strategy to identify the optimal α by sliding the α–cut from low-to-high TRS frequencies on Figure 4.3. In Figure 4.4 we plot the mode difference of CLL distributions generated by different α–cuts. As we move α from left to right, generally the mode differences increase.

Viewed simply, the mode difference is maximized when α reaches maximum TRS frequency which is a trivial solution. However even though the largest α maximizes the mode difference between R and NR, the volume of R tickets is minimized at that point which is undesirable. This constitutes a trade-off between the ratio of $R\_tickets/all\_tickets$ and the mode difference of the two CLL distributions. We selected the value of α = 670 to address this trade-off, and with this the R tickets are 30% of the full data set, where this α has the mode difference of 0.6 between R and NR.

Establishing the main hypothesis: In Figure 4.5, for this α = 670, we depict the corresponding left and right CLL distributions that are now labeled as NR-TRS and R-TRS

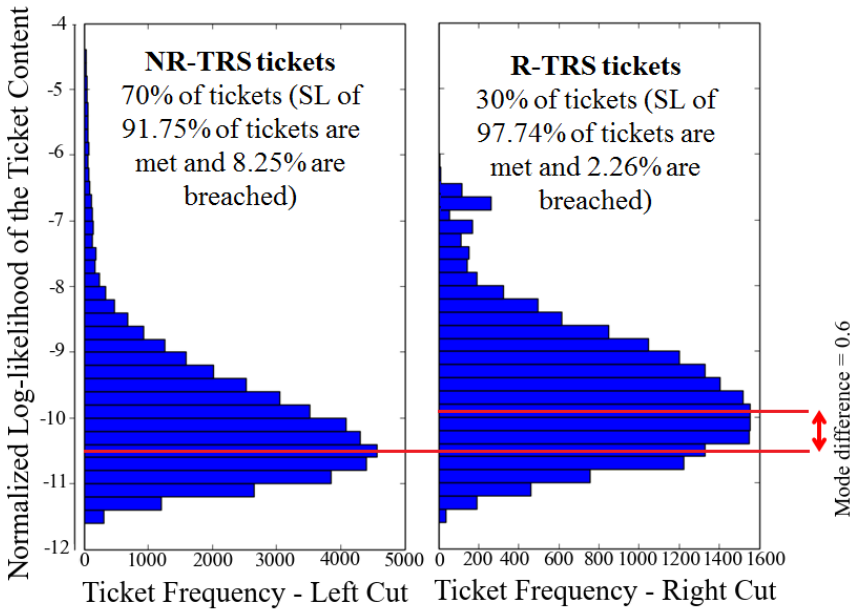


Figure 4.5: $\alpha = 670$ yielding optimal cut, two CLL distributions as NR-TRS (left) and R-TRS (right)

respectively. Note that R-TRS distribution is denser in the higher log-likelihood area as compared to NR-TRS.

Part (i): This establishes the hypothesis that a considerable number of tickets with frequent content (higher log-likelihood) are resolved by frequent TRSs. This also identifies the desired α supporting causality between frequent content and R-TRS and allows us to proceed with our two-level classification. Essentially, what we did in this step has introduced heuristics to label a subset of distinct TRSs as R-TRSs and the complement set as NR-TRSs. Later all of the distinct TRSs will be used to train the top-level R/NR classifier, and R-TRSs will be used to train the second level muliclass classifier.

Part (ii): Furthermore, we tie this optimal R/NR split with SL by calculating the SL breach percentage. As shown in 4.5, the breach ratio of the R-TRS class was found to be a fourth of the NR-TRS class (2.26% to 8.25%). This supports the hypothesis that

R tickets are more likely to meet their SL. Also, if R-TRSs are recommended then the recommendations are 97.74% likely to meet enterprise SL goals.

4.4 Enterprise CEN Deployment

The two-level framework is illustrated in Figure 4.6. The model developed is divided into offline training (left), and on-demand recommendations (right). Offline training includes computationally intensive operations and they lead to construction of the classification models. Formal details of training are in Section 4.5. On-demand recommendations apply the classifiers on the unlabeled data and recommend actions for achieving SL goal. Formalization details of recommendations and their validation are given in Section 4.6.

Offline Operations – Top Level Training: Here we use a Bayesian binary classifier that takes Natural Language (NL) content, and identifies whether it is associated with highly frequent paths (marked as Routine). As shown in Figure 4.6, NL features are extracted from training tickets and then used along with their R/NR annotations to perform Bayesian inference. Thus the top-level R/NR classifier is constructed for on-demand use.

Offline Operations – Second Level Training: Next we use a Bayesian multiclass classifier that takes NL content and identifies a Routine path that is most likely to resolve the incident. Here Bayesian inference is only performed for tickets with routine resolving sequences. NL features are extracted from content of those tickets and are annotated with their associated TRSs. Thus the second-level R-TRS classifier is constructed for on-demand use. We will discuss and address the underlying challenges of dealing with skewed class distribution in Section 4.5.

On-demand Operations: On the right of Figure 4.6 we show the two-level application of the method on an unlabeled ticket. First we determine if the NL content of the ticket

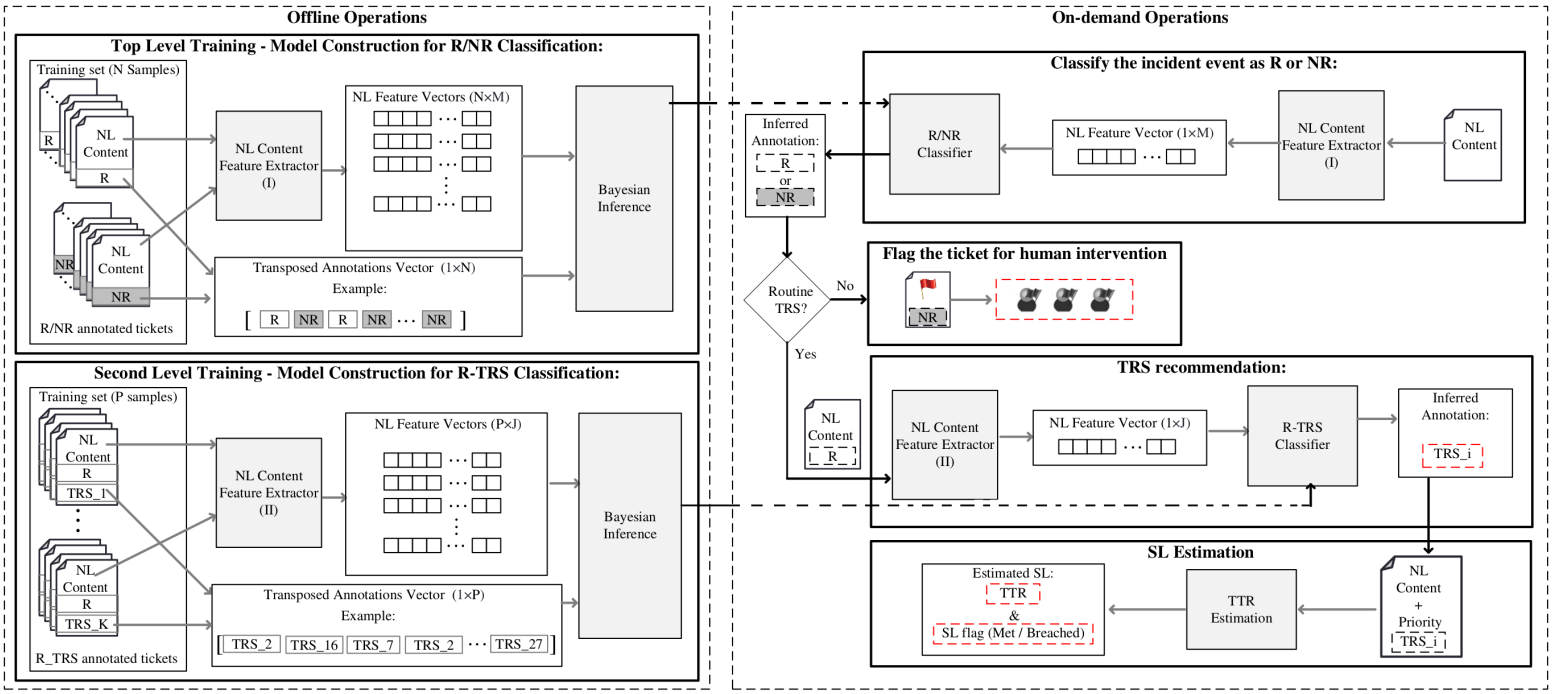


Figure 4.6: Dynamic CEN recommendation framework

is associated with either R or NR using the top level R/NR classifier. Second, if it is associated with R then the second level R-TRS classifier is applied to provide the path recommendation for the CEN. Also SL estimation is performed for the recommended path. If the content is associated with NR class then it is flagged and turned over to the CEN for resolving without assistance. In Figure 4.6 within the on-demand operations box, all of the dotted boxes are denoting predicted values. In Section 4.6 we discuss the validation and SL advantages of the framework.

4.5 Experiments Using the Two-level Classification Framework

Based on the existence of a strong relationship between frequent content and routine paths, we proceeded to build the classifiers. The learning algorithm we leveraged is the Transformed Weight-normalized Complement Naïve Bayes (TWCNB) [56] for both top and second level classifiers of the Framework introduced in Figure 4.6. This algorithm is designed to perform on skewed training data, and it incorporates effective weight normalization and feature transformations. Further rationale for selecting this method follows.

4.5.1 Training and Classification (R/NR and TRS Recommendation)

We modified TWCNB for path (R-TRS) classification as follows. Let:

1. $\vec{t}$ be the training set of routine tickets that previously got resolved by an R-TRS: $\vec{t} = (\vec{t_1}, \vec{t_2}, ..., \vec{t_n})$ and t_{ij} is the frequency of the j_th word of the dictionary in ticket $\vec{t_i}$.

2. $\vec{RS} = (r\vec{s}_1, r\vec{s}_2, ..., r\vec{s}_n)$ be the resolving sequences (i.e. TRS) corresponding to each of the training tickets.

3. $C = \{C_1, C_2, ..., C_s\}$ be the set of distinct paths.

4. $\vec{test} = (f_1, f_2, ..., f_m)$ be a test ticket where f_j is the frequency of the j_th word of the dictionary in the test ticket.

Then *train* and *predict*:

$$\omega = R\text{-}TRS\_Training(\vec{t}, \vec{RS}) \tag{4.4}$$

$$Predicted\_label(\vec{test}) = \arg\min_{c \in C} \sum_{j=1}^{m} f_j \cdot \omega(j \mid \overline{c}) \tag{4.5}$$

Algorithm 2 `R-TRS_Training`

Input: $\vec{t}$, $\vec{RS}$
Output: ω

for $j = 1$ *to* m **do**
  $IDF_j = \log \frac{n}{\sum_{k=1}^{n} \delta_{kj}}$
  for $i = 1$ *to* n **do**
    $TF_{ij} = \log(t_{ij} + 1)$
  end
end
for $j = 1$ *to* m **do**
  for $i = 1$ *to* n **do**
    $NC_{ij} = \frac{TF_{ij} \cdot IDF_j}{\sqrt{\sum_{k=1}^{m} (TF_{ik} \cdot IDF_k)^2}}$
  end
end
for $j = 1$ *to* m **do**
  for $h = 1$ *to* s **do**
    $\hat{P}(j \mid \overline{C_h}) = \frac{\lambda + \sum_{k:rs_k \neq c_h}^{n} NC_{kj}}{m\lambda + \sum_{k:rs_k \neq c_h}^{n} \sum_{p=1}^{m} NC_{kp}}$
    $\omega(j \mid \overline{C_h}) = \frac{\log \hat{P}(j \mid \overline{C_h})}{\sum_{k=1}^{m} \log \hat{P}(k \mid \overline{C_h})}$
  end
end
Return ω

The function call $R\text{-}TRS\_Training(\vec{t}, \vec{RS})$ is elaborated by Algorithm 2 which performs the training. It uses a set of transforms for term frequencies adapted from [56]. These transforms resolve different poor modeling assumptions of Naïve Bayes classifier including skewed word and class distribution. ω is the transformed weighted normalization function over $P(j \mid \overline{c})$ where j can be the index of any word in the corpus dictionary, and $\overline{c}$ can be complement of any class in the data set (distinct paths in this case). Some details of Algorithm 2 are: Line 2 constructs inverse document frequency transformation where $\delta_{kj} = 1$ if the j_th word of the dictionary is in ticket $\vec{t_k}$, otherwise $\delta_{kj} = 0$. Line 3: n is the number of tickets in the training set. Line 4: constructs term frequency transformation. Line 9: provides the length norm, where m is the size of the corpus dictionary. Line 13: s is the cardinality of the set C. Line 14: builds a smoothed probability function that estimates the probability of j_th word of the dictionary not in the class C_h. Line 15: is the log weight normalization of $\hat{P}(j \mid \overline{C_h})$.

Experimental Process Overview

For both classifiers in Figure 4.6 we extracted features from the NL content and the text was first transformed to vectors with weighted normalized values as discussed in the 'dampening the effect of skewed data bias' section 4.5. We dropped the stop words and removed low-frequency words, thus reducing the dimensions of our feature vectors to 4623. Next we randomly sampled 80% of <content, TRS> tuples (i.e. 119200 tickets) for end-to-end model training and 20%(i.e. 29800 tickets) for validation. That 80% was used to train the top level R/NR classifier, and the routine portion of it (i.e. 35776 tickets or 24% of all tickets) was used to train the second level R-TRS classifier. The training on each level was validated by 10-fold cross validation (i.e. rotation on 90%, 10% splits). After tuning parameters of each of the classifiers separately, we observed significant performance by

both classifiers in isolation. Then we measured the overall performance of the sequentially combined classifiers by using the 20% validation set.

4.5.2 Framework Evaluation Measure

Given our goal of achieving trustworthy recommendations we opted to increase the reliability at the expense of reducing the number of tickets for which assistive recommendations were presented. Assume the ground truth labels, *actual-R* and *actual-NR*. The human experts are capable to handle (1) all actual-NR tickets, and (2) actual-R tickets that got misclassified as NR. On the other hand, it is unfavorable for trust if an actual-NR ticket is misclassified as R, and is further recommended with an R-TRS. Therefore, in this application domain *the precision of the top-level classifier and the accuracy of the second-level classifier are more important for the overall performance than the coverage*. In particular from a SL achievement perspective, it is notable that the recall of the top level classifier is not as important as its precision since false negatives (misclassified routine tickets) will nevertheless get routed through the CEN and addressed directly by human experts (i.e. without recommendations). The performance of our two-level recommendation framework is evaluated by measuring the proportion of tickets that their TRS got correctly recommended, to all tickets that got recommended as R. Formally:

$$Overall\ R - Precision = \frac{\#(TRS\ correctly\ classified)}{\#(tickets\ predicted\ as\ R)} \tag{4.6}$$

4.5.3 Evaluating the R/NR Labeling Strategy

This is an unsupervised method that finds a non-trivial optimal cut that bifurcates the ticket set such that the distance between the two content distributions is maximized. The content distribution with the higher average log likelihood is then labeled as R and the other

distribution is labeled as NR. A path is labeled as R if and only if majority of the tickets that it has resolved in the history fall within the R content distribution. Otherwise that path is labeled as NR.

The above strategy in our experiments labeled most of the TRSs as NR (77.4%), which favorably conforms to our machine learning goal proposed in Subsection 4.2. Thus we called this labeling strategy as *'Low-rate Routine labeling' (LRL)*. To evaluate the optimal bifurcation strategy, we chose two alternative labeling strategies as baselines: (1) *'Balanced labeling' (BL)* where the most frequent paths are labeled as R in such a way that these paths together resolve 50% of all tickets, and the rest are labeled as NR. (2) *'High-rate Routine labeling' (HRL)* where the most frequent paths are labeled as R such that these paths together resolve 75% of all tickets, and the rest are labeled as NR.

Per each labeling strategy we constructed a top level R/NR classifier (using the TWCNB learning algorithm). Our goal here was to find the classifier that consistently outperforms the other two. Figure 4.7 presents the *Receiver Operating Characteristic (ROC)* curves corresponding to different labeling strategies. The concept of ROC was first introduced in [25] and it generally aims to show performance of a binary classifier as its decision threshold varies. In the context of this study the *true positive rate (TPR)* (i.e. recall) is the fraction of actual-R tickets that also got classified as R. The *false positive rate (FPR)* is the fraction of actual-NR tickets that unfavorably got classified as R. The perfect case is to have TPR at 1 and the FPR at 0. The ROC curves in Figure 4.7 are drawn as a result of varying classifiers' decision thresholds from 0 to 1. Performance of these classifiers are evaluated by the *area under the ROC curve (AUROC)*. Observably our adapted optimal cut strategy (i.e. LRL) outperforms both of the baselines. To be precise, AUROC for LRL,

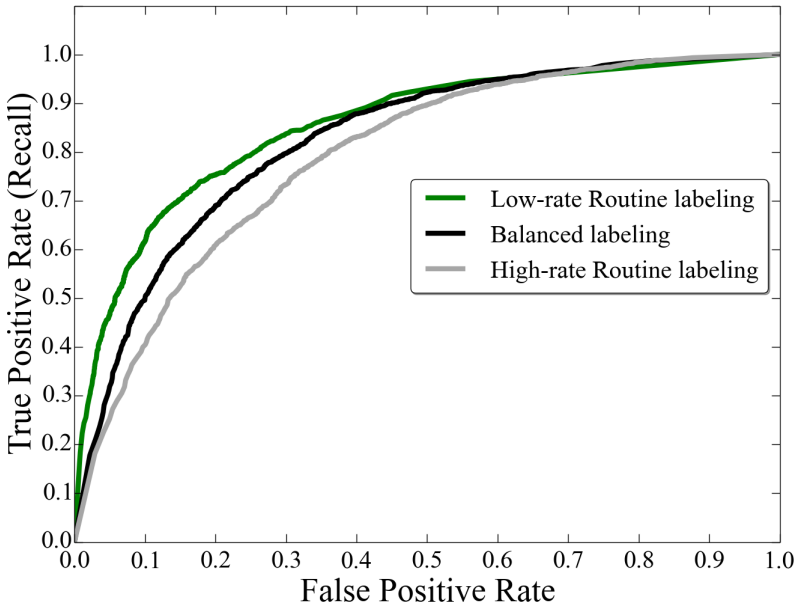


Figure 4.7: ROC curves for three variations of R/NR classifier

BL, and HRL are respectively 0.86, 0.83, and 79. Thus we continued to use the optimal cut strategy in construction of our top level classifier.

4.5.4 Tuning the Precision/Recall Trade-off for R/NR Classification

In this application domain, increasing the precision of the R class can significantly improve the SL performance overall. Therefore the goal here is to find an effective decision threshold which favors precision a bit more over recall. Based on equation 4.5 the decision threshold is used to classify a ticket as R based on: $\hat{P}(C = \text{`}R\text{'} \mid \tau) > \theta$

Here $\hat{P}$ is the inferred probability for a test ticket τ to be classified as R. θ is the decision threshold acting as the minimum acceptable probability value to classify a ticket as R. We used the LRL ROC curve from Figure 4.7 to pinpoint an effective decision threshold. After examination of the coverage of candidate decision thresholds we arrived at a point on the

ROC curve which yeilds a resonable high precision (through a low FPR) with an acceptable recall and coverage. More specifics of this sweet spot are as follows: recall=0.553, FPR=0.073, precision=0.802, and Routine Coverage=0.202. The decision threshold corresponding to this point found to be $\theta = 0.650$. Thus clearly resulting in a more conservative routine calls by the top level classifier.

4.6 Performance Evaluation on Variations of the Model and Baselines

By applying the same enterprise data, we compared two variations of the proposed framework, *Strict model* and *Flexible model*, against an existing sequence recommendation model called *Generative greedy model* taken from [39].

Strict and Flexible models: For the validation of path recommendations we define two different ways of claiming successful classification on a test ticket: (1) *strict* TRS matching: a ticket is called correctly classified if its predicted R-TRS matches exactly with its actual TRS. (2) *flexible* TRS matching: a ticket is called correctly classified if its predicted R-TRS is within the *congruence set* of its actual TRS.

The congruence set of a certain path like P consists other paths that are equally eligible to resolve same tickets that historically got resolved by P. Such replications exist by design among some of the routine paths in order to (1) balance the regular workload over more nodes in the network to improve the network throughput, and (2) make the network more tolerant against unavailability of certain nodes. Here for each of the routine paths in our domain, subject matter experts established a handcrafted congruence set representing corresponding qualified alternative paths, which we used for the flexible matching.

Baseline model - Generative Greedy: The Generative Greedy is considered a robust transfer prediction model [39]. This model is designed to make one-step transfer predictions and select the most probable resolver next. In our experiment Generative Greedy has shown effectiveness in predicting the final expert in the sequence for actual-NR tickets with long TRSs. To be able to compare the results, we re-defined the 'Overall R-Precision' for Generative Greedy: for any test ticket predicted as R, we let the Generative Greedy also predict n transfers at once where n is the length of the actual TRS. If the Generative Greedy matches the actual TRS, we consider it as correctly classified. The ratio of correctly classified TRSs divided by total number of predictions is considered as Overall R-Precision for this method.

Figure 4.8 shows the overall R-precision of the developed sequence models as the size of the training set grows. All three models converge to a stable precision before reaching to 60% of the size the training set. Many of the misclassified TRSs in the strict model are found to be within the congruence set of the actual TRS. Therefore as can be seen we achieved 17% improvement over strict model by allowing misclassification within congruence sets. Also the flexible model outperforms the baseline by 34%. (flexible:77%, strict: 60%, generative: 43%).

4.7 SL and TTR Estimation for Classified Tickets

For the fraction of test tickets that R-TRSs are recommended, we developed a simple expectation model to further estimate their TTR and SL compliance (SL Estimation in Figure 4.6). Let $T_{P,RP}$ be a subset of the training set that includes all tickets with priority P that were resolved by a particular routine path RP. For a test ticket τ with priority P

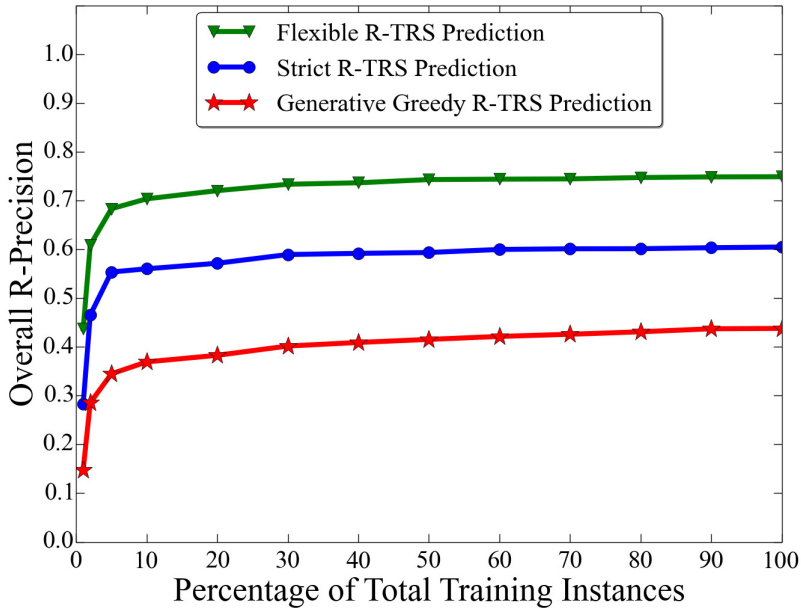


Figure 4.8: Overall R-Precision of flexible, strict, and greedy models.

(i.e. $\tau.p = P$) and recommended path RP the *Expected Time to Resolve (ETTR)* is estimated as the mean TTR of all tickets in $T_{P,RP}$. Formally:

$$\tau.ettr = \frac{1}{\mid T_{P,PR} \mid} \sum_{t \in T_{P,PR}} t.ttr \tag{4.7}$$

For ETTR evaluation, a held-out test set was used from which 1636 routine tickets eventually received recommended R-TRSs from the two-level classifier. Figure 4.9 illustrates a scatter plot of these tickets which compares ETTR of tickets against their actual time to resolve (ATTR). In order to present different ticket priorities within a unified scale we normalized all ETTR and ATTR values by their service time, thus generating NETTR and NATTR values. As a result of normalization any NETTR or NATTR value greater than 1 signals a SL breach. Also the diagonal line represents the *identity relation* between ETTR and ATTR. Tickets above the diagonal line imply ETTR > ATTR, and ticket below it imply

Table 4.3: Evaluation of expected time to resolve

Region#	ETTR	ATTR	ETTR>ATTR?	% of test tickets
1	Met	Met	TRUE	65.9% [1078]
2	Breached	Met	TRUE	0.2% [3]
3	Met	Met	FALSE	31.8% [521]
4	Met	Breached	FALSE	1.9% [31]
5	Breached	Breached	FALSE	0.2% [3]
6	Breached	Breached	TRUE	0.0% [0]

ETTR < ATTR. Therefore, there will be six regions on the scatter plot subject to further analysis.

In Table 4.3 the common SL and TTR properties of tickets in each region is presented. Also the last column reports the probability (and frequency) distribution of tickets over different regions.

The key insights reported in Figure 4.9 and Table 4.3 are as follows: (1) Almost all routine tickets that actually met their SL were also estimated to meet their SL based on their recommended R-TRS with an exception of tickets in region 2 (SL Recall = 0.998). (2) Most of the routine tickets that were estimated to meet their SL were also found to actually meet their SL with an exception of tickets in region 4 (SL Precision = 0.980). This confirms the fact that estimated SL compliance is a true indicator of the actual SL compliance. (3) Most of the tickets that actually breached their SL were estimated to *meet* their SL with an exception of tickets in region 5 (SL false positive rate = 0.911). Despite the common intuition that FPR is an error measure and has to be minimized, *here a high FPR is a point of* ***strength*** *for our estimation model*. The reason for high FPR is that in the absence of recommendations, human decision anomalies cause a fraction of routine tickets to take NR-TRSs. Our data has shown that 87% of all routine breached tickets were actually routed through NR-TRSs. However, nearly all of these tickets could have met their SL had they

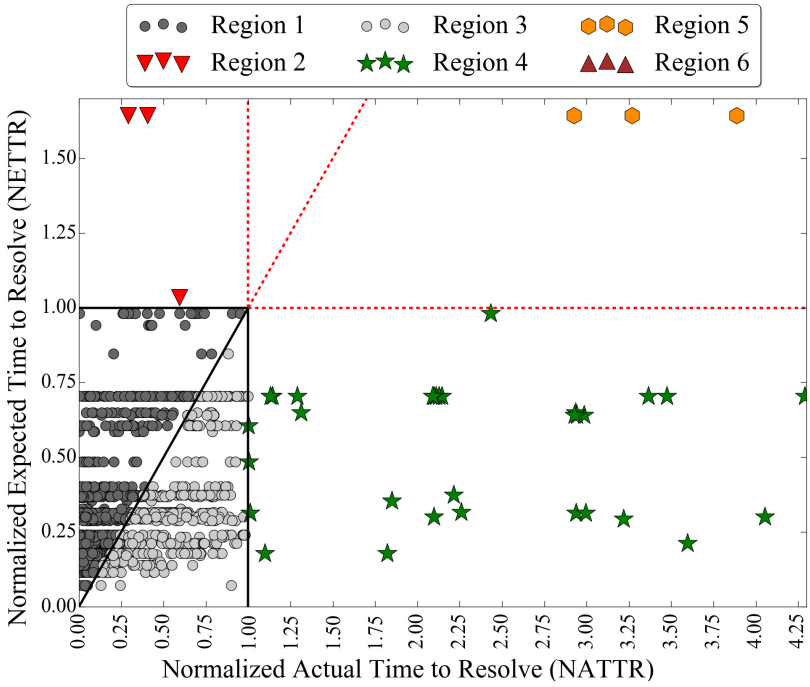


Figure 4.9: ETTR vs ATTR for tickets with a recommended R-TRS (refer to Table 4.3 for description of the regions)

taken their correct R-TRSs through recommendations. That is why ETTR is significantly lower than ATTR for most of the tickets in regions 4 and 5. *This clearly confirms the contribution of our statistical learning approach in reducing the negative impact of human decision anomalies.* (4) Based on ETTRs calculated above, recommendations significantly reduced the MTTR of the routine tickets by 34%. Viewing the system as a whole, the two-level classification method reduces the MTTR of *all* tickets by an average of 7%.

4.8 Configuration Items and Relation to Routineness

As discussed in Chapters 1 and 3, once an incident is captured at the IT service desk, an initial investigation results in an early speculation to identify the culprit CI. Here we explored the relationship between the change in the speculated CI, and classification label

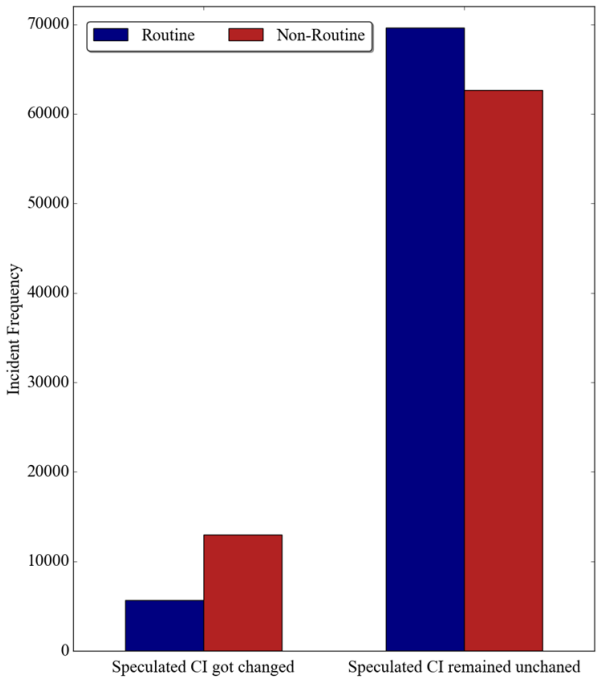


Figure 4.10: Change in a speculated CI represents non-Routine content

of incident content (routine/non-routine). In order to perform our analysis, we leveraged tickets in the test set that were correctly classified by the top-level classifier (R/NR), and we mapped the change in speculated CI against the incidents class (routine/non-routine) of content. According to Figure 4.10, it is observable that change in CI happens more with the non-routine content than with the routine one. On the other hand, fixed CIs happen more with routine tickets than than with the non-routine one. Here is another way of interpreting: when the content is routine, it is only 7% likely that the speculated CI gets changed during the resolution process, while in case of non-routine content, it is 20% likely that the speculated CI gets changed. A clear insight here is that resolution context mis-identification happens more when the initial content is unusual (i.e. non-routine) to the CEN.

4.9 Principles for Achieving SL Improvement and Summary

Since the real-world enterprise deployment goal (that is meeting SL) is particularly overlooked we next summarize the overall application logic from a business perspective. The analysis in the previous subsections has established the following principles and research goals:

1. *Business performance is related to improved MTTR:* That is, in contrast to MSTR of previous research, MTTR is a critical customer-facing measure and needed for ITSM. This is addressed with an objective function maximizing likelihood of meeting SLTs in Section 4.3.

2. *Assistive recommendations must be consistent with previous CEN behaviors along the entire TRS:* We note that R-TRS's are well-defined workflows throughout which incremental contributions are made in the context of achieving SLs. Thus the machine learning and recommendations are on the entire TRS. (i.e. No conditional independence assumption.)

3. *Trust is not achieved by noisy transfers:* Noisy transfer sequences with low probabilities for achieving SL goals are not to be used for machine learning and are to be filtered out through the R/NR classification. This is achieved with a first level for Routine (R)/Non-Routine(NR) inference and the second level for actual path recommendation to improve resolution within SLs.

4. *Trustworthiness of recommendations must be considered:* At the User Interface, the presentation of the specific recommendations is followed by the percentage of times it led their colleagues to successful resolution on that content (Appendix A). The human

is also notified when a trustworthy recommendation is not available and the knowledge base must be improved.

5. *Experiments must demonstrate improvement where the CEN struggles with content:* The CEN is actually performing as well as it can on frequent and high priority tickets and the SLs are being met. Thus experiments explicitly show that there is enough *other opportunity* to improve SL related to lower priorities or the longer TRSs due to (1) poor transfer knowledge, (2) content that is truly complex or new and the transfer knowledge is not explicit, and (3) lack of resources or training [67].

The related aspects of analysis above point towards and an opportunity for the assistive model to help with tickets that could be potentially misrouted tickets and where the SL breaches occur. We have established that there are enough such cases and there is adequate performance improvement. The business rationale provided is in the form of potential improvement in performance versus resources needed to achieve that improvement. The performance improvement metrics identified were: (1) improved Mean-Time-To-Resolve; (2) reduced SL breaches; (3) reduced number of transfers for specific priorities; and (4) maintaining a high level of trust to ensure the system is used and the investment is beneficial.

The recommendation framework improves performance by the Collective Expert Network in applications like the service desk within the enterprise. If a routine path on the CEN has historically achieved the SL by resolving the tickets within service time then it has met the time and customer satisfaction goals. Using this and other principles exhibited by the CEN in its digital trace, we developed the two-level framework suited for enterprise deployment. The path recommendation results are promising as they indicate 77% R-precision for the end-to-end model. The recommended R-TRSs are more than 96%

likely to meet the SL goals. The overall two-level classification model has also shown 10% reduction in average SL violation rate mainly by preventing frequent content from getting misrouted by the CEN.

Also there are three main contributions towards successful deployment within the enterprise ITSD. (1) The detailed analysis of extensive operational data to motivate the CEN conceptual model appropriate for time-constrained problem solving by expert networks as elaborated in Section 4.1. (2) Using CEN behavior insights obtained from the analysis to develop the principles that must be met by assistive and trustworthy recommendations in a two-level framework as explained in Section 4.4. And (3) The supervised learning model that meets the principles along with the experimental setup that show performance improvement as presented in Sections 4.5 and 4.6.

Chapter 5: Enhanced Framework: Recommendation with Rigorous Time Estimation

In this chapter we answer the following key technical questions:

- *Can we improve the performance of CEN with respect to resolution time by considering TRS workflows globally rather than just local transfers?*
- *Can we achieve a rigorous resolution time estimation model for recommended TRSs at the prediction time in order to improve users' trust in the recommendations?*

The research reported in the previous chapter only partially answered the first question. Validation found that while resolution accuracy increases by recommending R-TRSs (workflows), the TTR estimates (ETTR) are deviated from actual TTRs (ATTR) resulting in high time estimation errors. This discrepancy warrants further research addressed in chapter for the following two reasons:

- A discrepancy between ETTR and ATTR for ticket t could be desirable! Particularly, if ETTR is less than ATTR, that could signal an improvement resulted from taking a RecTRS (i.e. a notation for recommended TRS) that is more efficient. This is significant because currently SLTs are relaxed to accommodate worst case scenarios. By providing methods to improve the resolution time for individual recommendations, we also help improve the SLT goal setting and manage resources.

- When deploying a recommendation system within IT operations, the trustworthiness of the RecTRSs should be achieved by precisely estimating TTRs. We show that the results of the previous chapter have aggregate-level time estimation causing higher error, thus making the recommendations inaccurate with respect to TTR. This was identified therefore as a less trustworthy framework because the professionals whose performance is measured by meeting time constraints would not accept inaccurate recommendations.

To address this, Sections 5.1 and 5.2 present the integration of our previous and the above concerns into a more complete 'training, testing and assessment' framework as illustrated in Figure 5.1. By applying this *enhanced framework* we show (1) that for certain resolved tickets, their TTR is deviated from ETTR of their SLT-preserving RecTRS, and (2) that if this discrepancy is better understood it can provide opportunities for improving not only TTR, but also SL compliance in aggregate. To do this, the assessment method presented in Section 5.2 also analyzes the error of ETTRs and causes for estimation errors. This motivates the need for a more rigorous resolution time estimation modeling.

The next research step here is to understand the reaction of the experts to ticket content that is 'surprising' and consequent increases in resolution time. This is achieved by building a language model for each R-TRSs during training, and measuring the cross-entropy of a test ticket with respect to its RecTRS's language model. We show this allows us to verify whether high time estimation error is correlated with content that is deviated from the inherent language model of RecTRS.

Results in Section 5.3 show that tickets with high cross-entropy or 'surprising' content are strongly correlated with the high ETTR errors. This actually means we need a better estimation model that captures not only the dynamics of surprise, but also all other factors

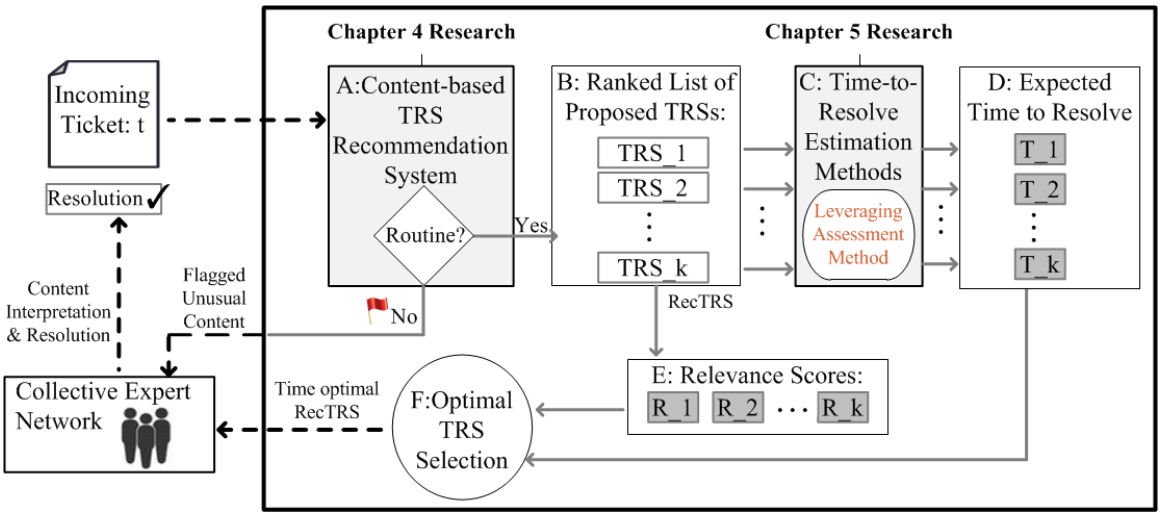


Figure 5.1: Enhanced framework: time-optimal TRS recommendations

contributing to the dynamic reaction of the experts leading to high estimation errors. Using this result as a basis, the rigorous resolution time estimation and its core components are presented in Section 5.4 and subsequent sections by considering the experts' dynamics.

5.1 Analysis of CEN Achievement of SLT Goals

Through exploratory analysis it was found that 24% of all the TRSs have three or more experts (sometimes repeated more than once) to provide add-on and contributory knowledge. We have identified this type of problem solving detected as collective problem solving [43]. As a reminder, the CEN is defined as a directed graph on a set of resolved tickets T where experts and transfers in T are represented by nodes and edges, respectively.

5.1.1 Summary of Developed Recommendation Framework and Enhancement

In an enterprise with the service desk as the first node that sets the clock, our research [43] discussed in Chapter 4, found that the CEN works hard at meeting SLTs and is more successful on *routine* or frequent ticket content. We discovered an important pattern that frequent ticket content is 'highly relevant' to certain frequent TRSs. Furthermore, of those frequent TRSs approx. 98% met their associated SLTs. This provided the basis for the enhanced framework in Figure 5.1:

Recommendations based on content classification: Incoming ticket is classified using a two-level classification framework introduced by our prior research in [45]. The top-level classifier labels the ticket content as Routine or Non-Routine (Figure 5.1 Box A). If the content is labeled as 'Non-Routine' then it is not used for further recommendations, but flagged for unassisted expert-driven resolution process. This helps retain only those tickets for which there is solid classification evidence ensuring greater accuracy to promote trust in the recommendations. If the ticket gets labeled as Routine, it will be followed by a second level classification which recommends a ranked list of TRSs based on the classification confidence for the incoming ticket (Figure 5.1 Box B).

Meeting the Resolution Goal: By recommending the Routine TRSs on frequent content, research in Chapter 4 established a 34% improvement in the accuracy of the recommendations when compared to the greedy baseline. In addition, it was found that the two-level TRS classification model has high precision (77%) when TRSs are recommended. Thus, establishing that RecTRS is an existing resolving sequence with a high likelihood to meet its SLT.

Meeting the SL Goal: Next there are two factors related to evaluating a proposed TRS: (1) SLT and (2) ETTR. However, our research in Chapter 4 was limited ourselves to SLT evaluation and found that 99.8% of the tickets in the history that achieved their SLT are also expected to achieve their SLT after taking RecTRS (i.e. SLT Recall = 0.998). This firmly established that the TRS recommendations are reliable and meet SLTs.

Moving to current research, we wish to identify opportunities to *improve on, and not simply meet* SLTs. Thus, also improving the aggregate SLT performance of CEN on T. This requires us to examine ETTR vs ATTR for RecTRSs. Previous time estimation models for ETTR are very approximate. The goal is to have better methods for time estimation for RecTRSs in Figure 5.1 Box C. Furthermore, since research in Chapter 4 does not well-address recommendation and validation against time-constraints, it thus became important to first conduct research into an error assessment framework for TTR estimation. This is reported next.

5.2 Understanding ETTR to Improve Resolution Time Estimation

We first show that by developing and using a method for assessment (leveraged in Figure 5.1 Box C) we can motivate the design of better features for time estimation for Box C which can then in turn be used for the selection of an SLT-optimal RecTRS (Circle F). This expands prior work by taking into consideration not only SLT achievement, but also the estimated time performance of recommendations validated over actual resolution time. For developing this assessment, a held-out test set of 3,560 tickets were used from which 1,636 tickets received recommended TRSs from box B in Figure 5.1 (the remaining 1,924 were flagged as Non-Routine). The performance of resulting recommended TRSs is

Table 5.1: Assessment of RecTRSs - ETTR vs ATTR

Assessment	ETTR?ATTR	Proposed	Actual	% test tickets
Investigate	1: $>$	SLT Met	SLT Met	65.9% [1078]
Ignore	2: $>$	Breached	SLT Met	0.2% [3]
Better	3: $<=$	SLT Met	SLT Met	31.8% [521]
Better	4: $<=$	SLT Met	Breached	1.9% [31]
Human	5: $<=$	Breached	Breached	0.2% [3]
Human	6: $>$	Breached	Breached	0.0% [0]

assessed and summarized in Table 5.1. Note that this table provides further assessment on our earlier evaluation in Table 4.3. Specific steps underlying this analysis are as follows:

- For the fraction of test tickets for which the TRSs are recommended, we used an expectation model to further estimate their TTR. We used $T_{P,R-TRS}$ to denote a subset of the training set that includes all tickets with priority P that were resolved by a particular routine TRS $R-TRS$. For a test ticket τ with priority P and recommended path $RecTRS$ the ETTR is estimated as the mean TTR of all tickets in $T_{P,RecTRS}$, formally:

$$\tau.ettr = \frac{1}{\mid T_{P,RecTRS} \mid} \sum_{t \in T_{P,RecTRS}} t.ttr \tag{5.1}$$

- In order to compare ticket with different priorities within a unified scale we normalized all ETTR and ATTR values by their corresponding SLTs, thus generating NETTR and NATTR values. As a result of normalization if NETTR>1 then recommended TRS is estimated to breach its SLT, and if NATTR>1 then its actual SLT was breached according to the ground truth.

- For a test ticket τ we define estimation error (squared error) as: $(\tau.nettr - \tau.nattr)^2$.

The resulting six regions are next subject to causal analysis that leads to design of better estimation models. To achieve a deeper understanding, Table 5.1 presents SLT, ATTR and ETTR properties of tickets within each region and an assessment in the first column. Note the last column reports the probability (and frequency) distribution of test tickets over different regions.

- **Region 1:** While meeting SLTs, t.ETTR > t.ATTR. This needs to be investigated due to the fact that the higher ETTR estimates could be due to inaccurate (means based on history) estimation method. This motivates the further analysis and potentially considering the CEN's dynamic features in the next section, and thus designing improved methods for Box D of Figure 5.1. The output of this can then be more accurate, resulting in reliable SLT achieving recommendations that take less time.

- **Region 2:** Here the proposed RecTRSs are not appropriate for recommendation and thus not investigated further.

- **Region 3:** Here the proposed RecTRSs are actually improving the TTR and used as RecTRSs utilized in final selection circle F in Figure 5.1.

- **Region 4:** Here the proposed TRSs are actually benefiting the business contractually by avoiding breaches and used as RecTRSs in final selection circle F of Figure 5.1.

- **Region 5:** Not used for recommendation, flagged and sent directly to humans in Box A of Figure 5.1.

- **Region 6:** Not used for recommendation, flagged and sent directly to humans Box A of Figure 5.1.

Using the assessment of the ETTRs in Table 5.1, our next goal is to further 'Investigate' TTR estimation methods for Box C of Figure 5.1 to gain insights and identify features that can yield more precise TTR estimation and thus improve the RecTRS selection process in Box D. This motivates the design of more rigorous ETTR models.

5.3 Evidences of Dynamic CEN Behaviors

Note that the 'Investigation' of tickets in Region 1 (motivated above) requires investigation of external features for new estimation models which will improve the framework of Figure 5.1 Boxes C, D, and F by selecting from high-confidence RecTRSs using reliably low TTRs. Thus, the result is a new TRS recommendation model which proposes a pareto-optimal TRS that is characterized by the optimal combination of high recommendation *confidence* (i.e. $P(TRS|t.content)$) and low t.ETTR.

5.3.1 Refined Hypothesis to Include CEN Dynamics

The estimation model in Section 5.2 leverages the *Mean* TTR of the RecTRS for a given priority, and thus lacks explicit consideration of dynamics of the experts in the CEN. We therefore ask: *Could this be a cause for inaccurate estimation?*

5.3.2 Content Deviation vs ETTR Error

Path-Priority Language Models: For each test ticket τ with priority P, and recommended path $RecTRS$, we aim to relate the language used in $\tau.content$ to the language of all tickets in the history (training data) which had priority P and got resolved by τ's $RecTRS$. The idea here is to measure how surprising the incoming content is to the $RecTRS$. Here we need a reliable model for the linguistic state of ($Priority$,TRS) pairs. Therefore, we define a *path-priority language model* for each ($Priority$,TRS) pair in the

training set. This is constructed using Bigram language models with Katz back-off smoothing [30].

Cross Entropy of Content: Next for a test ticket τ, we quantify its content deviation w.r.t. its corresponding language model for (P,$RecTRS$), using cross-entropy computation:

$$H(\tau, LM_{(P,RecTRS)}) = -\frac{1}{N}\sum_{i=1}^{N} logP_{LM_{(P,RecTRS)}}(b_i) \tag{5.2}$$

Here there are N bigrams in τ, represented as b_is and $P_{LM_{(P,RecTRS)}}(b_i)$ is the probability of the bigram b_i under the language model for (P,$RecTRS$). This measure is motivated by [15] where authors used a similar measure to quantify the difference between a user's language and that of the community in online discussion forums. Here a higher cross entropy for a ticket implies more deviation from the linguistic state of its RecTRS. For each test ticket τ we compared its min-max normalized cross entropy (NCE) against its time estimation squared error (SE). For training, we used the content of 41,800 natural language tickets to build 118 unique language models. Then 3,200 test tickets were carefully sampled for experimentation where each was ensured to receive accurate RecTRS (that is, matching its actual TRS). Analysis reveals insights:

- In the condition where there is a large estimation error for a ticket (SE>5), the normalized cross entropy also happens to be large. The correlation analysis for this case resulted in $R^2 = 0.5156$ which signifies strong positive correlation between time estimation error and normalized cross entropy. In other words, when the resolution time is mis-estimated by a large margin, ticket content is largely deviated from its RecTRSs' language models. With no conditions on the estimation error the normalized cross entropy is only weakly correlated with the estimation error. The positive correlations are shown in Figure 5.2 to illustrate the existence of a linear

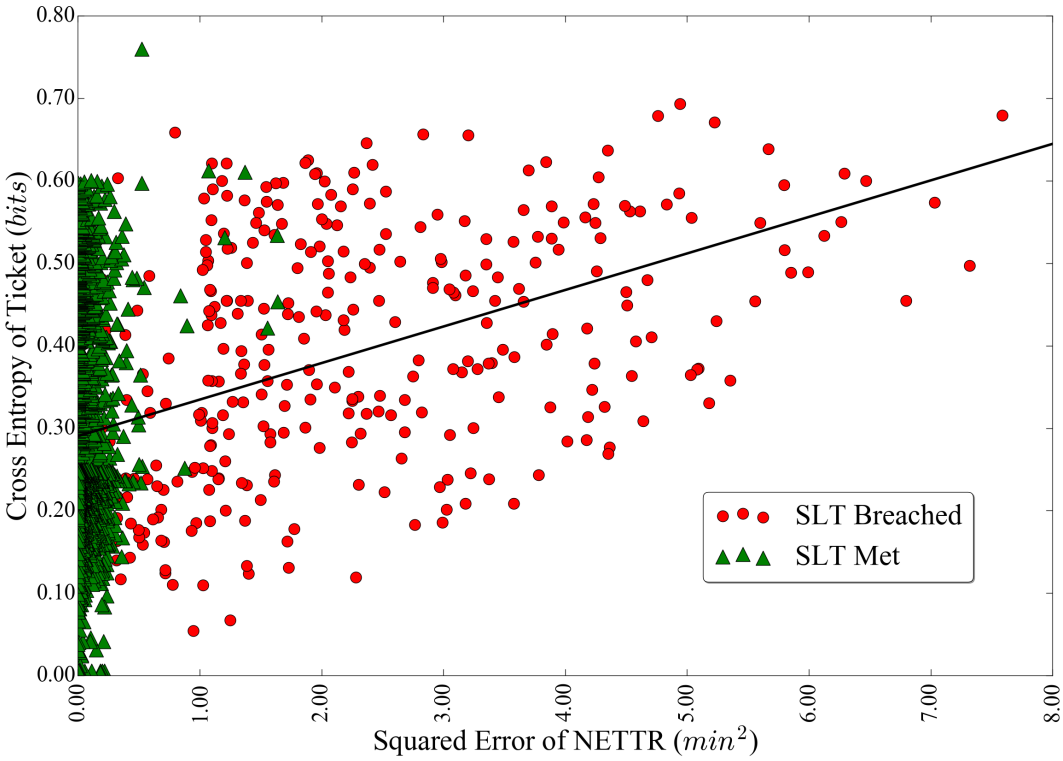


Figure 5.2: Squared error of NETTR vs normalized cross entropy (per each test ticket)

relationship between SE and NCE (regression line summarizes the relationship as $NCE = 0.0443SE + 0.2903$ with $R^2 = 0.1194$). This relationship is demonstrated more transparently on the aggregate level in Figure 5.3, where larger NCE results in higher Normalized Mean Squared Error. However, the unconditioned relationship here is not as strong as the relationship where SE>5, due to the fact that a considerable fraction of tickets (26.4%) with low estimation error (SE<1) happen to have high normalized cross entropy (NCE>0.3). This indicates that **not** all tickets with high linguistic deviation are inherently complex for the CEN. This also means the linguistic models of historical TRSs alone cannot capture factors contributing to time estimation.

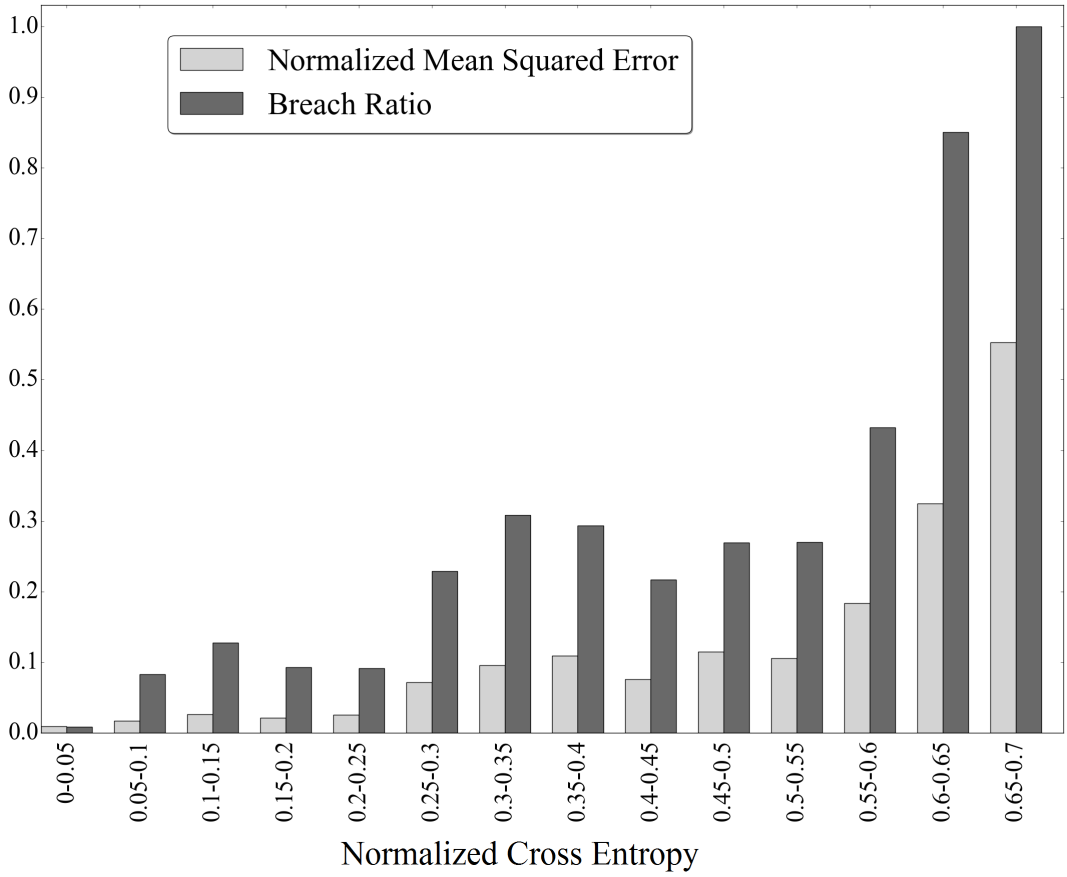


Figure 5.3: Normalized cross entropy vs. breach ratio, and vs. normalized mean squared error (aggregate level)

- High estimation errors mainly result from tickets with breached SLTs. 97.8% of tickets with SE>1 are breached (Figure 5.2). This uncovers a major pain point for path-based ETTR model in which 96.9% of tickets with an actual breached SLT will get estimated as meeting their SLTs. Therefore path-based ETTR model is incapable of (1) detecting such anomalous tickets, and (2) accurately estimating on them.

- SLT Breach ratio (likelihood to breach) increases as the NCE increases as shown in Figure 5.3. Thus, a dissimilarity metric between ticket content and the expertise of a TRS (such as NCE) qualifies as an important metric for better TTR estimation. This

is in conformance with [92], which suggests that a breach in SLT generally happens due to unusual, complex or ambiguous content.

These insights lead us to requirements for a better TTR estimation model that must leverage dynamic features available early in the resolution process to detect anomalies (such as surprising content) and use them in the estimation process.

5.4 Rigorous Response Time Estimation Modeling

In previous sections we identified the weakness of aggregate resolution time estimation model, in particular; we showed that more content deviation from the TRS language model results in higher mean squared error for time estimation. Here we decided to move beyond aggregate estimation measures by introducing a response time estimation model based on dynamic features of the CEN. Motivated by the above analysis, our next step was to achieve lower estimation errors using an ensemble multivariate regression model defined at an expert-level.

Note that this is opening up new opportunities by shifting to a consideration of CEN dynamics (in terms of load in queue, expertise, etc.) in order to estimate each expert's local contributions and its effect on resolution time of a potential TRS.

In our new approach that follows ETTR for a ticket is modeled as the sum of expected contribution time (ECT) in each intermediate TRS node, plus the expected resolution time (ERT) at the last node. Formally, for an arbitrary ticket τ that is recommended with a TRS $P = \langle e_{(1)}, e_{(2)}, ..., e_{(n)} \rangle$, The expected time to resolve is defined as:

$$ETTR(\tau, P) = \sum_{i=1}^{n-1} ECT(e_{(i)}, \tau) + ERT(e_{(n)}, \tau) \tag{5.3}$$

Figure 5.4 illustrates a rigorous process for time-to-resolve estimation armed with potential features that could capture the dynamics of CEN. To approximate ECT and ERT

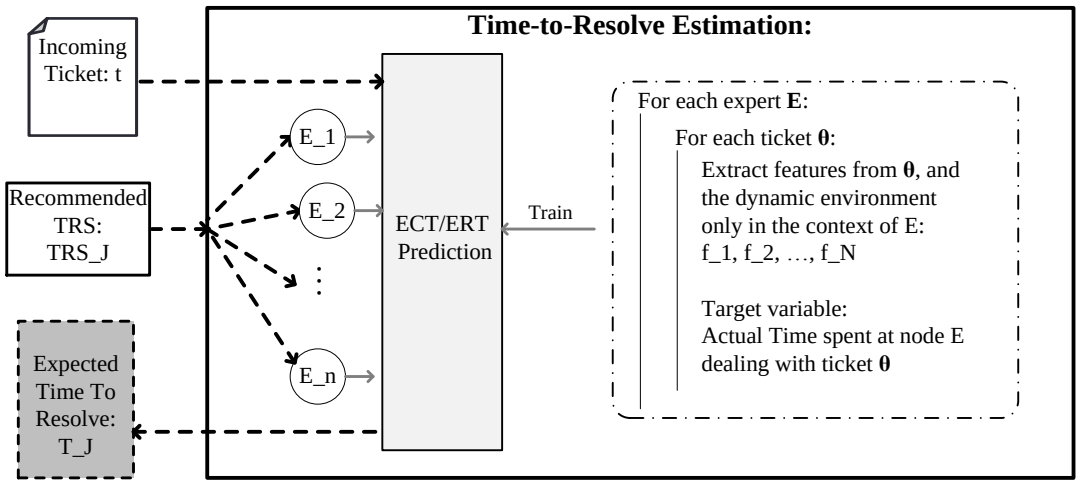


Figure 5.4: Time-to-resolve estimation, detailed expansion for Box C of Figure 5.1

functions for each expert, a multivariate regression model has to be constructed. The preliminary task for such regression modeling is to identify a set of explicit and inferable features that could affect each expert's response time. Below is a set of features that we introduced that are intuitively causal to experts' response time:

Features for TTR Estimation – considering dynamics of CEN :

- $e_{(i)}$'s **transfer/resolution expertise** on τ; This requires transfer/resolution profile extraction for each expert.
- $e_{(i)}$'s expected **queue load** when dealing with τ. This is to be obtained from the *ticket queue* which requires load estimation for each expert at any given point of time.
- Time elapsed divided by the SLT when the ticket is considered to be received by $e_{(i)}$.
- $e_{(i)}$'s **mean time to respond on the speculated CI**
- Other explicit features: priority, acknowledgement time divided by ACK target, SLT-breach alerts, etc.

5.5 Load Estimation Function

Here we introduce an estimation function that evaluates the expected number tickets queued at a particular expert using the transfer transactional log; we refer to it as *Higher Priority Workload* (HPWL) function. More formally, given the time interval that τ was at $e_{(i)}$, HPWL computes the expected number of tickets that were at $e_{(i)}$ (at least partially) during the same interval with sooner or equal SL deadlines:

$$HPWL(e_{(i)}, \tau) = \sum\nolimits_{t \in T} \frac{Length(I(t, e_{(i)}) \cap I(\tau, e_{(i)}))}{Length(I(\tau, e_{(i)}))} \tag{5.4}$$

Where $I(t, e_{(i)}) \cap I(\tau, e_{(i)}) \neq \emptyset$ and $t.TLTB \leq \tau.TLTB$. Here $I(\tau, e_{(i)})$ represents the time interval that ticket τ was at $e_{(i)}$. $TLTB$ is a timer on each ticket reporting the remaining time left until the SLT deadline.

The impact of workload on response time had to be studied, as a result of which workload can be deemed useful or bogus for response time estimation. Leveraging the $HPWL$ function, we measured the ticket load at all possible intervals in the history, and reported the average time to respond with respect to higher priority incident load (Figure 5.5). We summarize our findings as follows: *The larger the volume of higher priority incidents at an expert, the slower the experts' response*. Therefore, *Load can work as a useful signal for response time estimation*.

5.6 Expertise Modeling

Expertise modeling has been a subject of different studies in different application domains. For example in applications such as matching best reviewers for professional articles and proposals, a reviewer's expertise is modeled using the documents that were written by that reviewer, and the extracted expertise was then used for pairing the reviewer with

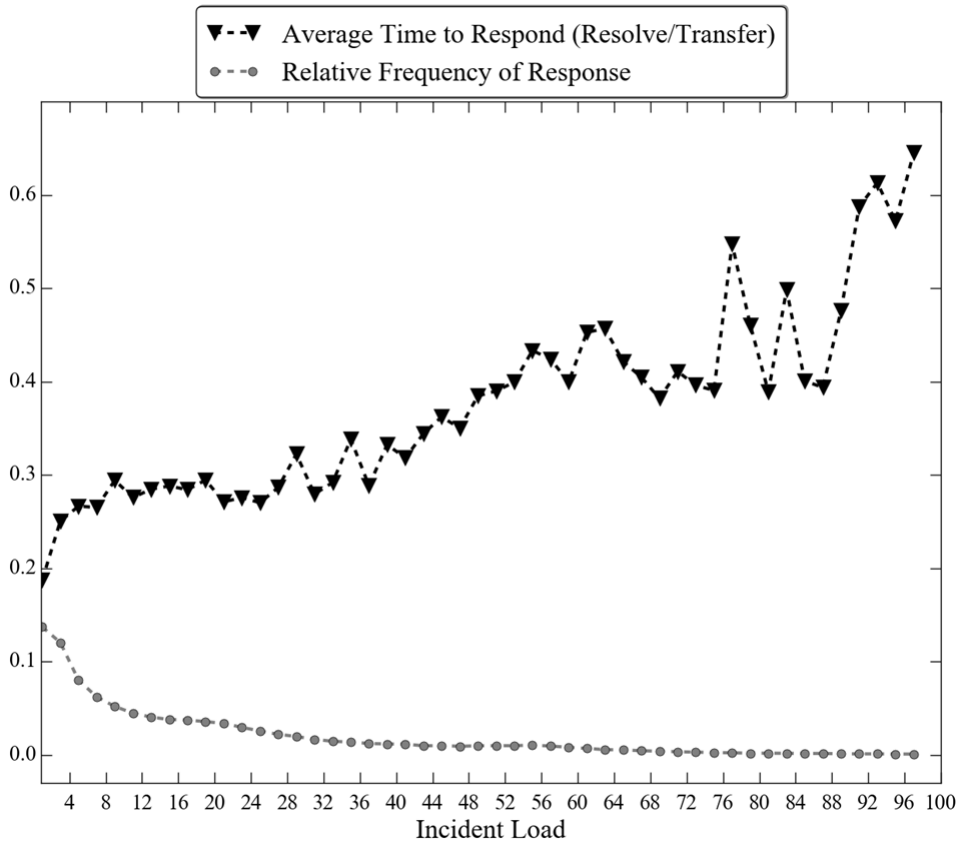


Figure 5.5: An increase in incident load results in an increase in response time

unassigned documents [36, 42]. Expertise has also been a prevailing research topic in question answering microblogs, for instance Zhou et al. [94] modeled expertise using language models based on the questions that a user contributes to. The subject has recently been worked in the social media domain where for social users' expertise is inferred based on structural links, and content [23, 81].

As introduced in Section 5.4, the problem that we aim to solve is: *Given the history of tickets resolved/transferred by an expert, how likely is that expert in resolving/transferring a new ticket?* Our goal thus is to (1) estimate transfer/resolution expertise for each expert

based on the tickets they historically worked on, and (2) report proximity of a new incoming ticket to a targeted expert's transfer/resolution expertise. As motivated by [24], we will leverage the fact that our goal can be achieved by framing the problem as a soft binary classification problem where given an expert, and a ticket, the goal is to estimate the probability that the expert resolves the ticket, or transfers it. In the rest of this subsection, we modeled and compare expertise extracted using several baselines, a related research, and our new approach.

5.6.1 Expertise Modeling – Baselines

To achieve the previous goal we extract the transfer and resolution expertise for any arbitrary expert e and report their proximity to any arbitrary ticket t. We define several baselines for expertise extraction and proximity calculation against which to compare our new approach.

Cosine TF-IDF: Here all tickets are transformed to TF-IDF word vector representation (first defined by Salton et al.[59]). Then for each expert e we constructed two word vectors, $e^{(Trans)}$ to represent the transfer knowledge, and $e^{(Res)}$ to represent the resolution knowledge. $e^{(Trans)}$ is computed as the average vector of all word vectors corresponding to tickets that historically got transferred by e. Similarly, $e^{(Res)}$ is computed as an average vector of all word vectors corresponding to tickets historically resolved by e. For a new incoming ticket t, the proximity of t to $e^{(Trans)}$ (or $e^{(Res)}$) is denoted as $Sim(e^{(Trans)}, t)$ (or $Sim(e^{(Res)}, t)$) and is evaluated using the Cosine similarity between $e^{(Trans)}$ (or $e^{(Res)}$) and t vectors.

Since in this application resolution is considered the flip-side of transfer, we decided to combine our proximity measurements, constructing a probability function:

$$P_{cos}(e \in Resolver|t, e) = \frac{Sim(e^{(Res)}, t)}{Sim(e^{(Res)}, t) + Sim(e^{(Trans)}, t)} \tag{5.5}$$

$$P_{cos}(e \in Transferer|t, e) = 1 - P_{cos}(e \in Resolver|t, e) \tag{5.6}$$

Here $P_{cos}(e \in Resolver|t)$ denotes the estimated probability that e resolves t. By definition if e is estimated as highly likely to be a resolver, it is highly unlikely for e to be a transferer (i.e. classes are mutually exclusive). This probabilistic setting makes it easy for us to validate the accuracy of expertise extraction on a hold-out transfer set.

Language Modeling: For this baseline, for each expert e we construct two bigram language models. Resolution Language Model, denoted as $e^{(RLM)}$, is constructed based on tickets resolved by e in the history log, and Transfer Language Model, denoted as $e^{(TLM)}$, is constructed based on tickets transferred by e in the history log. For a sample ticket t, we measure its language deviation from $e^{(RLM)}$ using the following cross entropy measure:

$$H(t, e^{(RLM)}) = -\frac{1}{N}\sum_{i=1}^{N} logP_{e^{(RLM)}}(b_i) \tag{5.7}$$

Here there are N bigrams in t, represented as b_is and $P_{e^{(LRM)}}(b_i)$ is the probability of the bigram b_i under the resolution language model of e. Similarly, by substituting all occurrences of $e^{(RLM)}$ with $e^{(TLM)}$, the language deviation of t from $e^{(TLM)}$ can be computed. To handle rare terms in the tickets, we used additive smoothing as suggested by Charniak

in [11]. Here we constructed a posterior probability function using these language models as follows:

$$P_{LM}(e \in Resolver|t,e) = \frac{\exp(H(t,e^{(RLM)}))}{\exp(H(t,e^{(RLM)})) + \exp(H(t,e^{(TLM)}))} \tag{5.8}$$

$$P_{LM}(e \in Transferer|t,e) = 1 - P_{LM}(e \in Resolver|t,e) \tag{5.9}$$

Here $P_{LM}(e \in Resolver|t)$ denotes the estimated probability that e resolves t using our language modeling baseline.

High-confidence ensemble: In an attempt to improve the quality of the binary Transferer/Resolver predictions, we defined a high-confidence ensemble which effectively switches between the two baselines to achieve the highest spread on the posterior distribution for each unlabeled ticket. Formally:

$$P_{ensemble}(e \in Resolver|t,e) = \tag{5.10}$$

$$\begin{cases} P_{cos}(e \in Resolver|t,e) & if\ |P_{cos}(e \in Resolver|t,e) - \frac{1}{2}| > |P_{LM}(e \in Resolver|t,e) - \frac{1}{2}| \\ P_{LM}(e \in Resolver|t,e) & o.w. \end{cases}$$

$$P_{ensemble}(e \in Transferer|t,e) = 1 - P_{ensemble}(e \in Resolver|t,e) \tag{5.11}$$

Log linear: For comparison purposes, we also implemented a related solution introduced by Sun et al. to perform expertise estimation in task completion networks [67]. They used a classic Log-linear model [6] which takes an expert's expertise vector and a task's (i.e. ticket in our case) word vector as input, and outputs the expert's capability to

solve (i.e. resolve) the task. In Log-linear model, the probability for expert e to resolve ticket t equals to:

$$P_{Log-Linear}(e \in Resolver|t, e) = \frac{1}{1 + exp(-(W_1 t + W_2 \Phi_e + b))} \tag{5.12}$$

Φ_e represents the expertise vector of e and is estimated as: $\Phi_e = e^{(Res)} \oslash (e^{(Res)} + e^{(Trans)})$, where $\oslash$ represents element-wise vector division. In plain words, expertise of an expert e is estimated as the average word vector of historical tickets resolved by e divided by average word vector of all historical tickets received by e. Here W_1 (vector), W_2 (vector), and b (scalar) are parameters that need to be learned globally in the training phase. Note that to generate training labels when framing the problem as a classification task, $P^*_{Log-Linear}(e \in Resolver|t, e) = 0$ when e transfers t, and $P^*_{Log-Linear}(e \in Resolver|t, e) = 1$ when e resolves t (P^* represents a training label). This is clearly providing some flexibility to the model (higher variance, lower bias) which should lead to more accurate classification since there are parameters that are learned to maximize the likelihood of the data. The logistic function introduced in equation 5.12 can be considered as a quasi-similarity function between a ticket and an expertise vector.

5.6.2 Expectation-maximization for Expertise Modeling

The key idea here is that expertise can be inferred from training instances of transfer and resolution actions of experts. That is we estimate expertise vectors such that the cross entropy error on the training data is minimized yielding better results than using aggregated estimation methods.

We next introduce this improvement over the log linear expertise modeling, by *learning expertise vectors in addition to the global parameters*. We are using a two-step iterative approach. In step 1 (M-step), we learn the global parameters that maximize the likelihood

of the training data, and in step 2 (E-step), we using the learned parameters to estimate the expertise of all experts while maximizing the likelihood of the training data. We keep iterating until all expertise vectors converge to fixed sets of values. Our learning algorithm in details is provided below:

Iterative Expertise Extraction:

- **Step 0**: Initialize the the expertise matrix: $\Phi_E = [\Phi_{e_1}\Phi_{e_2}...\Phi_{e_M}]$ where for any arbitrary expert e_i we have $\Phi_{e_i} = e_i^{(Res)} \oslash (e_i^{(Res)} + e_i^{(Trans)})$
- **Step 1**: Given Φ_E, use Stochastic Gradient Descent (SGD) [74] to globally learn W_1,W_2, and b while minimizing the cost function for log linear model (cross-entropy function) on all training points:

$$\underset{W_1,W_2,b}{argmin} \sum_{\forall e_i,t_k:<e_i,t_k>\in<E,T>} [-L(t_k,e_i)logP(R|t_k,e_i;W,b) - (1-L(t_k,e_i))log(1-P(R|t_k,e_i;W,b))] \tag{5.13}$$

  Note that in this cost function, $< E, T >$ is the set of all (expert, ticket) training pairs. Also, $P(R|t_k, e_i; W, b)$ is a short hand notation for $P_{Log-Linear}(e \in Resolver|t_k, e_i)$ parametrized by Ws, and b, and $L(t_k, e_i)$ is a short hand notation for $P^*_{Log-Linear}(e \in Resolver|t, e)$ which is the function that generates the training labels (0 for transfer in log, 1 for resolution in log).

- **Step 2**: Given learned parameters W_1,W_2,b, for each expert e_i, locally estimate its expertise vector, Φ_{e_i}:

$$\underset{\Phi_{e_i}}{argmin} \sum_{\forall t_k:<e_i,t_k>\in<E,T>} [-L(t_k,e_i)logP(R|t_k,e_i;\Phi_{e_i}) - (1-L(t_k,e_i))log(1-P(R|t_k,e_i;\Phi_{e_i}))] \tag{5.14}$$

  Note that here $P_{Log-Linear}(e \in Resolver|t_k, e_i)$ is parametrized by Φ_{e_i}, and parameter estimation is separately performed by SGD for each expert.

- **Step 3**: if $\sum_{i=1}^{|E|} \left\| \Phi_{e_i}^{(k)} - \Phi_{e_i}^{(k+1)} \right\| > \varepsilon$ then go to Step 1.

  Here $\|\vec{v}\|$ is notation to show L_2 norm of $\vec{v}$.

- **Step 4**: Return Φ_E and W_1, W_2, b.

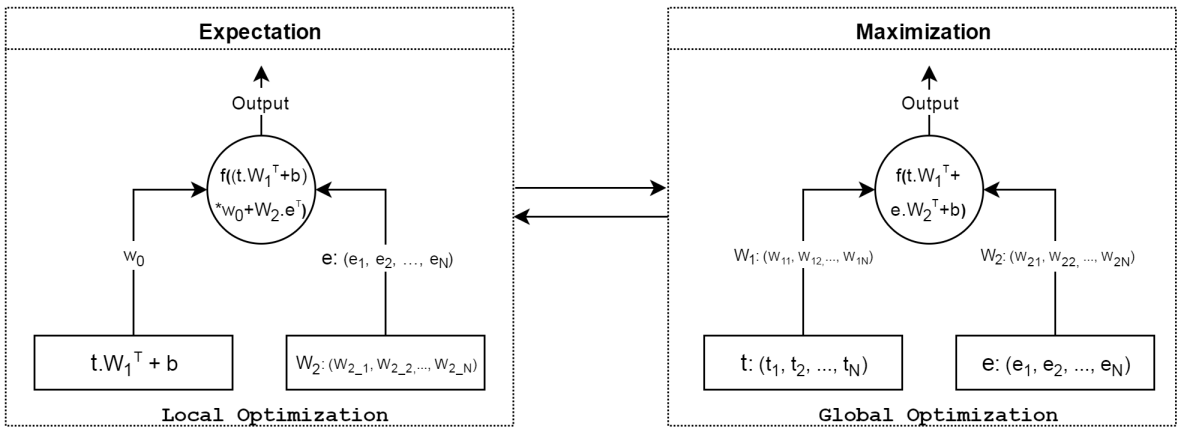


Figure 5.6: Perceptron representation of the E-M approach for expertise modeling

The EM algorithm that we laid out above can simply be captured in neural networks terms. Figure 5.6 is representing our EM algorithm using two perceptrons with logistic activation functions; The first one is the global perceptron running on the entire data set to estimate the global parameters (M-step), and the second one is the local perceptron defined for each expert on the training data corresponding to that expert to estimate his/her expertise (E-step). Algorithm 3 is illustrating the details of the M-step required to train the global perceptron, and algorithm 4 is portraying the details of the E-step required to train the local perceptrons. In both of these algorithms the function $TrainNetwork$ takes features, and parameters as input, and runs the update rule of the SGD algorithm, and return the updated parameters, and the error value.

The advantage of this algorithm over classic log-linear model is that the expertise vectors are now learned in accordance with the training data, this provides enough flexibility to the model which helps to get the most out of the training data and avoid under-fitting. Also a lower in-sample error is expected with this iterative approach. A disadvantage could be the running time of our E-M algorithm, since it runs SGD for as many times as the number

Algorithm 3 `Maximization Step`

Input: E, T,Φ_E, b, W_1, W_2, ε
Output: W_1, W_2, b
$error \leftarrow \infty$
while $error > \varepsilon$ **do**
  for $e \in E$ **do**
    for $t \in T$ **do**
      $\widehat{W_1} \leftarrow W_1$, $\widehat{W_2} \leftarrow W_2$
      if e *resolved* t **then**
        $\langle \widehat{W_1}, \widehat{W_2}, \widehat{b}, error \rangle \leftarrow TrainNetwork(t, \Phi_e, W_1, W_2, b, 1)$
      end
      else if e *transferred* t **then**
        $\langle \widehat{W_1}, \widehat{W_2}, \widehat{b}, error \rangle \leftarrow TrainNetwork(t, \Phi_e, W_1, W_2, b, 0)$
      end
      $W_1 \leftarrow \widehat{W_1}$, $W_2 \leftarrow \widehat{W_2}$, $b \leftarrow \widehat{b}$
    end
  end
end

Algorithm 4 `Expectation Step`

Input: e, T_e, b, W_1, W_2, ε
Output: updated Φ_e
$error \leftarrow \infty$
while $error > \varepsilon$ **do**
  for $t \in T_e$ **do**
    if e *resolved* t **then**
      $\langle nil, \widehat{\Phi_e}, nil, error \rangle \leftarrow TrainNetwork(nil, W_2, nil, \Phi_e, t.W_1^T + b, 1)$
    end
    else if e *transferred* t **then**
      $\langle nil, \widehat{\Phi_e}, nil, error \rangle \leftarrow TrainNetwork(nil, W_2, nil, \Phi_e, t.W_1^T + b, 0)$
    end
    $\Phi_e \leftarrow \widehat{\Phi_e}$
  end
end

Table 5.2: Characteristics of the data sets for experimentation

Data Set	# of Tickets	# of Transfer per ticket	(P1,P2,P3,P4)	Breach Ratio	Avg. Resolution Time by SLT
Training & CV set	121,184	1.644	(12.04%,44.14%, 24.34%,19.48%)	9.46%	0.714
Test set	30,036	1.651	(12.07%,43.72%, 25.05%,19.16%)	9.14%	0.723

of experts per iteration. As far as the running time, since we are only concerned with the training for once, we do not consider it a major drawback.

5.6.3 Expertise Modeling – Experiments and Results

In this subsection we provide more details about our experiments with regard to expertise modeling. We used a set of 151,220 tickets for our expertise-related experiments, 80% of which was randomly chosen for training and cross validation (CV) purposes (10-fold), and the other 20% of the data was used for testing, and performance measurement purposes. The details of our datasets, i.e., the number of tickets, the number of transfers per resolved ticket, the distribution of ticket priorities, breach ratio and average resolution time by SLT are provided in Table 5.2. As can be seen both data sets are portraying similar statistics; in other words, the training and cross validation (i.e. tuning) is performed on a set that is from the same distribution as the test set, making the final model not suffer from data inconsistency.

We considered 90% of the training & CV set as the training set. For all models we first needed to perform expertise estimation. In order to do so, we only considered experts in the training set that had transferred at least 100 tickets, and resolved at least 100 tickets, calling them 'solid experts'. This was because expertise estimation was practically infeasible (due to insufficient data) for experts with very few historically resolved/transferred tickets. At

the training phase, for solid experts we used content of their tickets and actions on tickets to extract resolution expertise (i.e. $e^{(Res)}$, $e^{(RLM)}$, and Φ_e) and transfer expertise (i.e. $e^{(Trans)}$, and $e^{(TLM)}$).

Tuning set is considered as 10% of the training & CV set and is used for in-sample validation and parameter tuning (i.e. decision threshold and regularization parameter). The test set is used for out-of-sample validation and final performance evaluation of the tuned model. In both tuning and testing phases, unseen data is used for model validation. Specifically, for each action (i.e. transfer/resolution) performed by a solid expert, we computed $P(e \in Resolver|t, e)$ through different models to represent the probability that e would resolve t, and used the actual action label in the log to validate the estimated probability.

Since all our models were probabilistic, the outputs of the introduced models were probability distributions over possible outcomes. To be able to leverage certain validation measures (Precision, Recall, ROC, etc.) we needed to predict a class as a final output based on the posterior distribution; therefore, we decided to tune the decision threshold (i.e. cut-off on resolution probability) to best classify the actions in the tuning set. To measure classification performance, we used F1 measure computed as a harmonic mean of precision and recall. Figure 5.7 is showing how F1 varies for different models as a result of changing the decision threshold. Triangles on the figure are showing where F1 is maximized per each model (Cosine:0.47, LM:0.48, Ensemble:0.46, Log-Linear:0.47, EM Log-Linear:0.48). We used these decision thresholds to move forward with tuned models and evaluate the performance on the test set.

After tuning the models, we tested the performance of the expertise models using traditional classification measures (F1, ROC curve, precision and recall) on the tickets of the test set. Here we summarized the performance results of all baselines and our novel EM

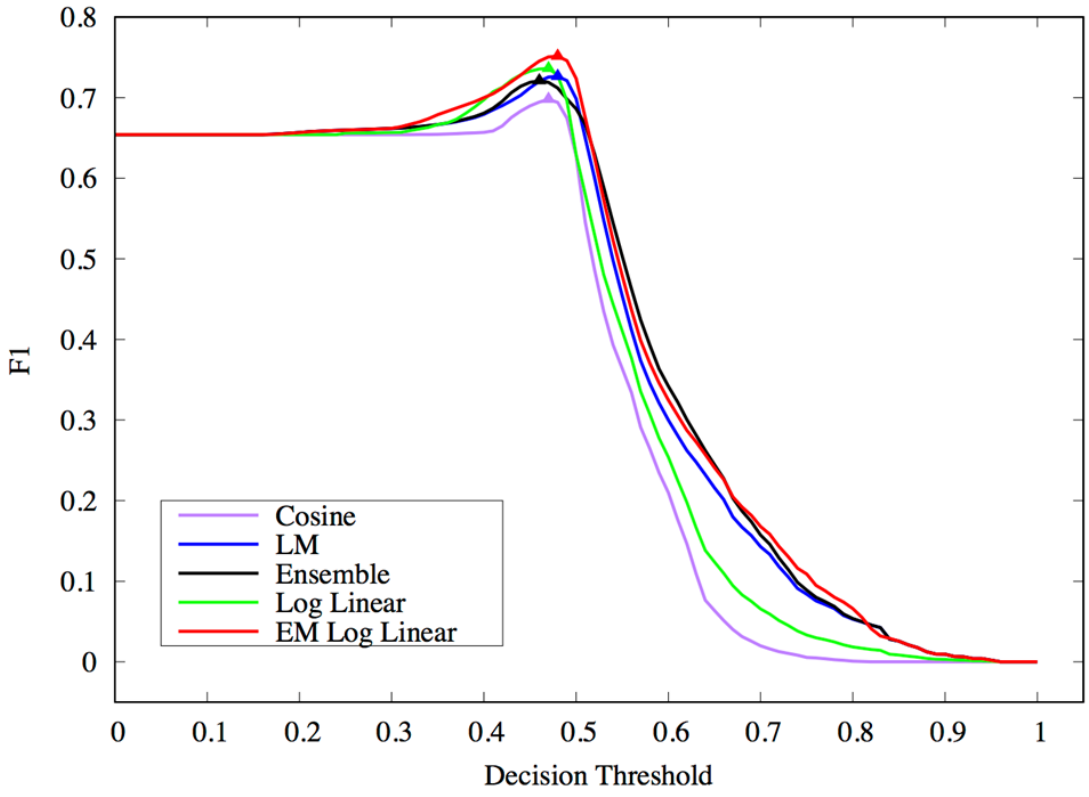


Figure 5.7: Tuning the decision threshold against F1

Log-Linear in Table 5.3. It is observable that our proposed solution (EM Log Linear) is achieving the highest F1 in predicting the experts' action labels. Precision in this application means how often is the model correct when it is predicting an expert's action to be 'resolution'. Recall in this application means how often is the model correct when an expert's actual action is 'resolution'. Results show that EM Log Linear is outperforming others in precision. One reason we found for low precision but higher recall on all classifiers is that models are somewhat aggressive in predicting resolution over transfer which is due to a relative resolution bias (Resolution prior probability is at 0.6 while transfer prior probability is at 0.4) in the training set. It is interesting to see that the Cosine baseline is achieving the highest Recall, but after error analysis we observed that the Coisne model

Table 5.3: Performance of expertise extraction models

Model	F1	Precision	Recall
(1) TFIDF + Cosine	0.677	0.559	**0.858**
(2) BigramLM + CrossEntropy	0.706	0.607	0.843
(3) Ensemble (1)+(2)	0.700	0.598	0.843
(4) Log Linear	0.716	0.631	0.828
(5) EM Log Linear	**0.731**	**0.645**	0.803

is producing resolution-skewed predictions due to relatively higher inverse document frequencies in the TFIDF vectors for resolution expertise (i.e. more specific vocabulary used for resolution than transfer). It is also note-worthy that the Ensemble classifier did not perform better than the language modeling classifier mainly because most (i.e. 86.3%) of the misclassified actions under the language model happened to be misclassified by Cosine; therefore finding the highest confidence between the two did not help in classification; in fact, in some cases the language modeling classifier has correctly classified an action, but cosine overruled it with an incorrect label just due to having a higher confidence.

Figure 5.8 is showing the ROC curve as a result of classifiers evaluation on the test set. Modifying the decision threshold on the posterior probabilities from 0 to 1 makes the classifiers go from the most conservative resolution labeling (i.e. bottom left corner) to the most aggressive resolution labeling (i.e. top right corner). Visibly, the largest area under the curve is produced by EM log linear model; this shows that EM Log Linear generalized well, and outperformed other models. Something to note here is that the EM log linear model does not top the other models when the decision threshold is too small, or too large. However, what matters the most is that within a reasonable neighborhood (i.e. ± 0.2) of the tuned decision thresholds, the EM log linear model surpasses the other alternatives.

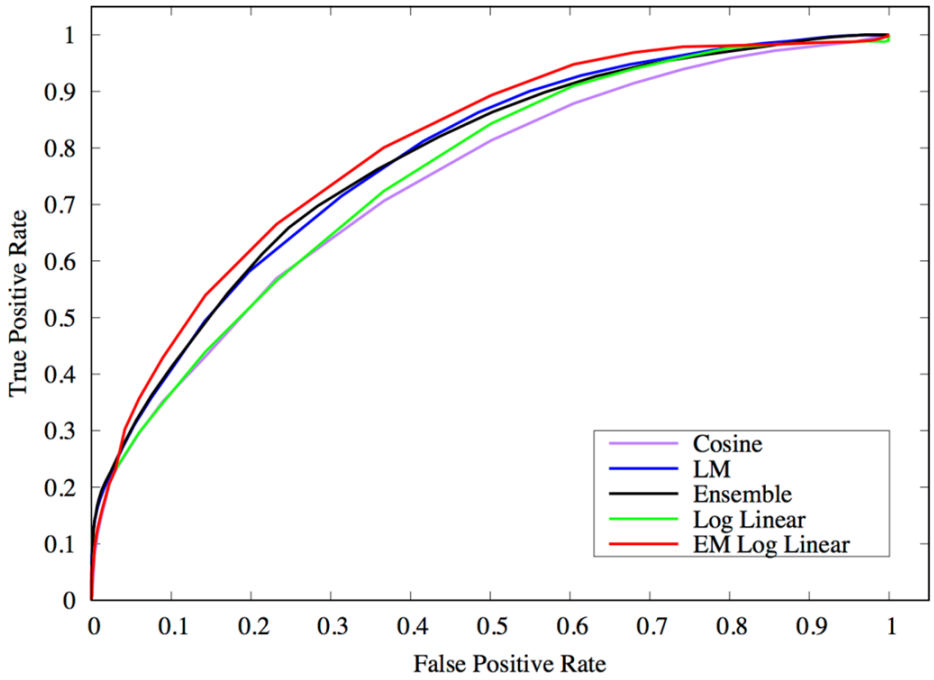


Figure 5.8: Validation of expertise modeling: true positive rate vs false positive rate

Another way of validation is to consider the estimated resolution probabilities as the outcome of the learning algorithm. The idea is to map the estimated resolution probability of the test tickets against their actual resolution ratio. Figure 5.9 is illustrating the distribution of the fraction of resolution actions to all in the test set (i.e. empirical probability of resolution according to the actual labels) over the estimated probability of resolution defined by each learning algorithm. This is effectively a validation for the estimated resolution probability using the empirical resolution probability. Ideally, the diagonal line ($y = x$) should represent perfect resolution estimation. As can be seen in the figure, EM Log Linear is the closest model to the diagonal line throughout all estimated resolution probability intervals. According to our EM Log Linear model, 90% of the estimated resolution probabilities are between 0.20 to 0.85, which indicates that even though tails (transfer

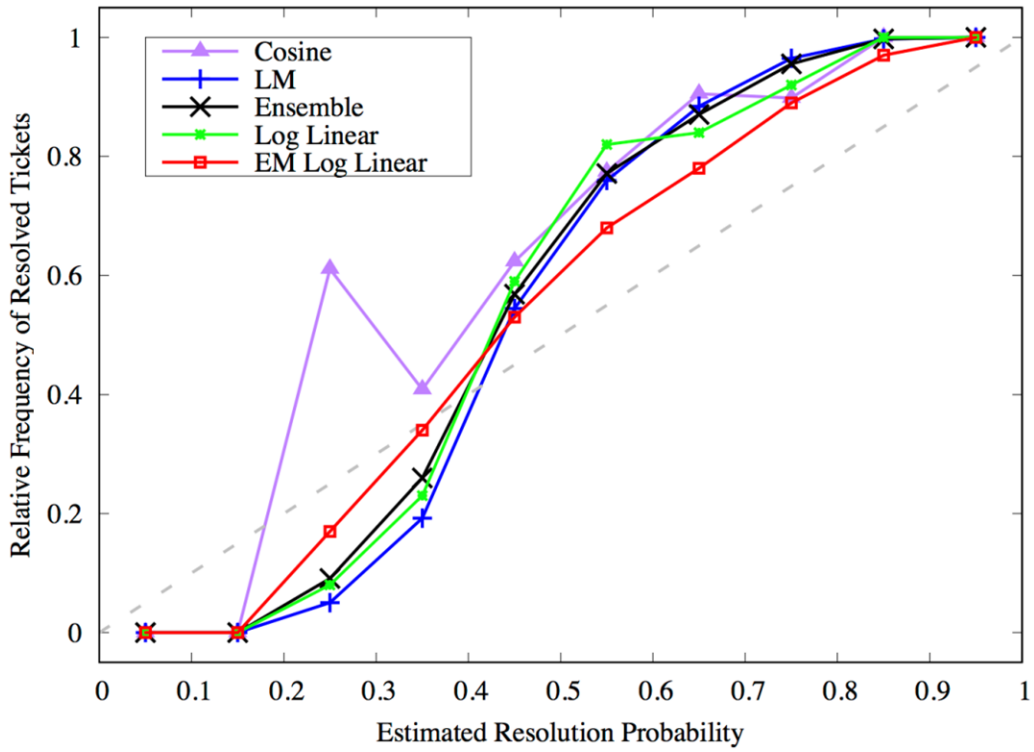


Figure 5.9: Validation of expertise modeling: empirical probability of resolution in log vs. estimated resolution probability

tail on the left, and resolution tail on the right) are higher certainty areas, most of the action population is carried elsewhere.

Next, to study the separation between the action classes, we used the test set to plot relative frequency distribution of transfers, and relative frequency distribution of resolutions over the estimated EM log linear resolution probability. Figure 5.10 is showing two well-separated distributions where all actions are mapped to their estimated EM log linear resolution probability. This separation signals for the fact that the EM log linear parameters are learned properly, and the model generalizes well on the test set. The figure also portrays where the two action distributions interfere resulting in a high-mass low confidence intersection area ($0.4 < P_{EM}(e \in Resolver|t, e) < 0.6$).

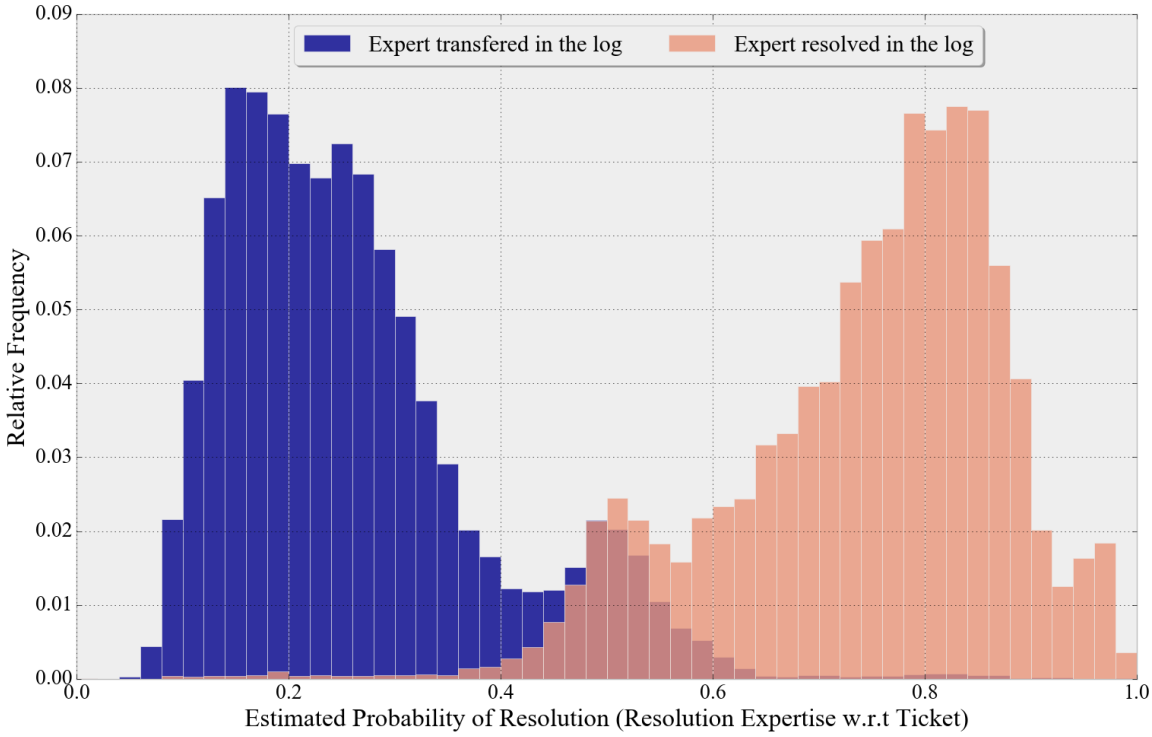


Figure 5.10: Resolution expertise clearly separating transfer actions from resolution actions in the log

Another insight based on this figure is that the resolution actions are having a larger spread as compared to the transfer actions; we believe this is due to the 'expert-training' phenomena, where tickets similar to those that have been transferred by experts in the history (i.e. training set) happen to to get resolved by the same experts after some period of time (i.e. in the test set).

So far we have shown that resolution/transfer expertise can be learned accurately from historical data. In order to tie this concept of resolution expertise to the resolution time estimation, we needed to study the impact of resolution expertise (i.e. estimated probability of resolution) on the response time of the experts. To conduct that study, for each action in the test data set we computed the estimated probability of resolution, and mapped it

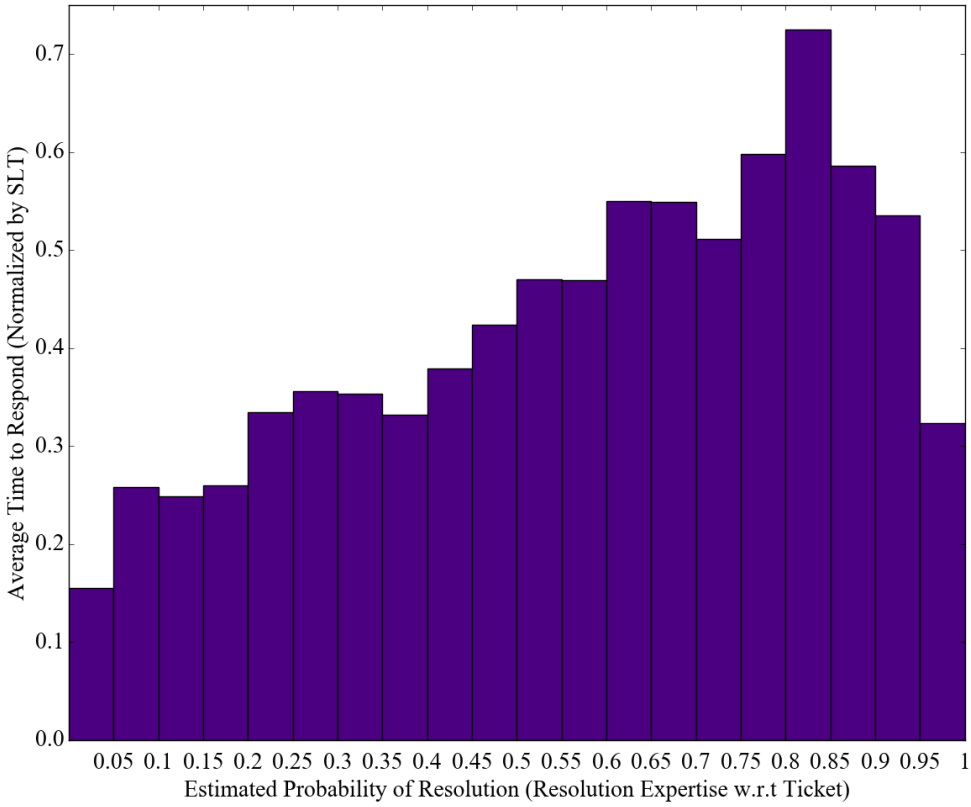


Figure 5.11: The impact of resolution expertise on the response time

against the response time of the expert for taking that action. Figure 5.11 is illustrating the average time to respond per each interval of the the resolution expertise. We discovered two interesting findings: (1) Up until a certain resolution expertise (i.e. $P = 0.85$) as the estimated probability of resolution increases on the tickets, the average time to respond on those tickets also increases. This indicates that the the less knowledgeable the expert on the ticket, the quicker the action on the ticket. Here, actions are predominantly transfers since for the most part the estimated resolution probabilities are less than 0.46. (2) After a certain resolution expertise (i.e. $P = 0.85$) as the estimated probability of resolution increases on the tickets, the average time to respond on those tickets sharply decreases. This indicates that the the highest degrees of resolution knowledge by an expert on a ticket

results in quicker action on the ticket. Here actions are predominantly resolutions since the estimated resolution probabilities are greater than 0.46. Therefore, we conclude that *estimated resolution expertise with respect to the ticket content could work as a useful signal (feature) for response time estimation.*

5.7 Putting it All Together: Enhanced Framework with Response Time Estimation

Finally, we wish to discuss how the response time estimation model will be used in the enhanced framework. The research presented in this section thus far establishes that (1) dynamic features of the CEN are detectable (due to the trends observed in Figures 5.5 and 5.11, and (2) dynamic features of the CEN are detectable are critical ticket-specific indicators for reduction of time estimation error for resolution.

These key results established the research hypothesis and in addition a basis for the remaining research and methods necessary to complete the rigorous TTR estimation framework (Figure 5.4). The approach is summarized below:

- Methods of this chapter (i.e. Load estimation and expertise extraction), applied for inferring/estimating additional features that affect the experts' response times. Per each expert, these features will feed into that expert's ECT/ERT model.

- ECT and ERT functions approximated for each expert, using a multivariate regression tree designed to use the following specific features:

    1. expert's transfer/resolution expertise on the ticket.

    2. expert's expected queue load when receiving the ticket.

    3. Time elapsed divided by the SLT when the ticket is received.

4. expert's mean-time-to-respond on the speculated CI.
5. Other explicit features: priority, acknowledgement time divided by ACK target, SLT-breach alerts.

It is important to note that the above features were initially identified by domain experts as essential factors in experts' response time. These features then got further studied by this research providing experimental observations to believe existence of relations between them and the target variable.

Chapter 6: Conclusions and Future Research in Time-constrained Problem-solving Networks

This chapter focuses on what this research has accomplished, and future research potential from this point onward. In Section 6.1 we briefly summarize our contributions, then in Subsections 6.2.1, 6.2.2, and 6.2.3 we provide descriptions about transfer recommendation enhancement, transfer intent discovery, and framework measurements in production as the immediate future directions for current contributions. Also in Subsection 6.2.4 we discuss use of data-driven analytical solutions in a broader domain to further benefit quality of supporting services in IT Service Management. Lastly, in Subsection 6.2.5 we present the implications of IT process discovery practices in order to assist educators and practitioners to systematically construct their service improvement cycles.

6.1 Establishing the Value of Recommendations for Time-constrained Problem-solving

Research results of this thesis establish the hypothesis by showing that every incoming ticket can benefit from machine recommendations achieving a 10% reduction in the volume of SL breach ratio. By flagging each incoming ticket for recommendation-assisted processing or for human-in-the-loop processing, the research ensures better adoption of the

framework by achieving service-level goals in enough cases to demonstrate compelling efficiency gains. The significance of flagging is to ensure that experts have greater confidence that recommendations will meet SL goals. This is important to promote trust and improve adoption.

The novelty of the research approach lies in the fact that the two-level recommendation framework is based on routine workflows. This allows us to achieve a 34% improvement over existing greedy transfer-centric models.

From a methodological perspective this research has the following contributions:

- Introduced Ticket Resolution Sequence (TRS) as a unit of problem solving; discovered most influential TRSs containing collective transfer patterns to resolve routine content.
- Built a content tagger (R/NR classifier) accurately distinguishing routine resolvable content from non-routine content. Resulted in higher trust in adoption of the recommendations.
- Built a path classifier resulted in early problem identification, resolution recommendation and shorter time to resolve.

While these contributions where shown to achieve an improvement over the baseline methods from the perspective of gaining the trust of users we found that the precision of resolution time estimation were unsatisfactory. This is because from a trust and adoption perspective, experts are looking for more precise SL compliance guarantees associated with recommendations. This led to the following contributions:

- Provided a solution to involve dynamics of CEN (i.e. Expertise, and ticket load) for a rigorous resolution time estimation to comply with Service level targets

- Introduced a novel expertise modeling approach to extract transfer and resolution knowledge at each expert, also introduced an expectation model to estimate the ticket load from the transactional logs

Thus these contributions established our research hypothesis aimed at benefiting the enterprise service support and delivery services by providing (1) lower decision and resolution latency, (2) lower likelihood of service level violations, and (3) higher workforce availability and effectiveness.

Even though our contributions where aimed towards continual improvement in IT Service Management, they can clearly have a broader impact. Some of our core contributions can be considered as reusable solutions for different application domains: Expertise extraction can be useful in information routing networks such task completion networks and question answering microblogs; estimation of higher priority workload can be useful in medical triaging applications, and emergency response management; Response time estimation based on workload and expected quality of action can benefit many downstream applications such as business process improvement, and optimized task allocation in cloud computing.

6.2 Towards Future Research in Time-constrained Problem-solving

Another value of the research presented here is a novel methodology for new learning models using intrinsic ticket properties as well as extrinsic dynamic properties of the CEN by uniquely combining these with supervised learning and regression methods. In this section we briefly discuss possible future research building on the existing contributions.

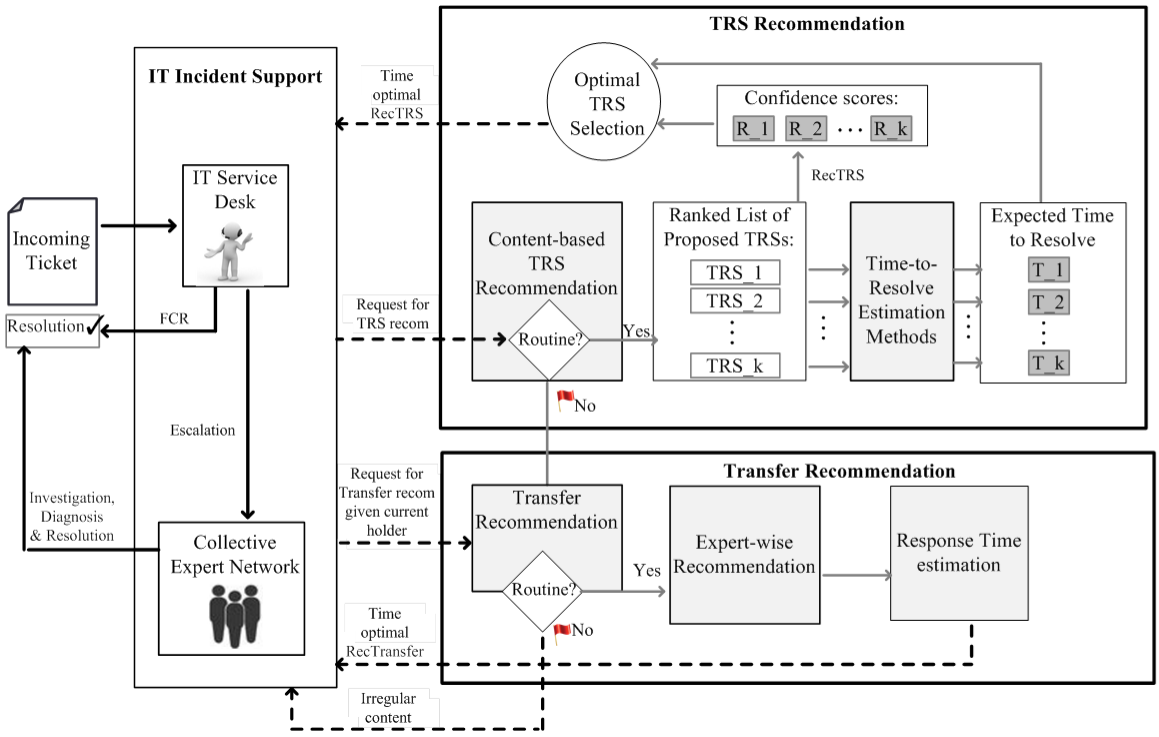


Figure 6.1: Transfer-enhanced resolution recommendation framework

6.2.1 Transfer-enhanced Resolution Recommendation Using Enterprise Taxonomy

Local transfer recommendations have been studied recently in task completion networks [60, 40]; some of the main shortcomings of those studies are: they do not accommodate SLTs in transfer recommendations; they do not consider mechanisms to battle poor recommendation accuracy on experts with sparse transfer history, and they do not deal with process-driven problem solving as a separate problem. As future work, in Figure 6.1 we outlined our enhanced CEN resolution recommendation framework in which the TRS recommendation framework is only a part of the overall solution. The enhancement is particularly about transfer recommendations which are triggered in one of the following

situations: (1) When the ticket content is predicted not to be associated with any established problem-solving workflow (i.e. Non-routine). (2) When an expert on a recommended routine TRS realizes that the rest of the experts on the recommended TRS are not going to help the resolution process, and seeks for local transfers to provide an ad hoc resolution.

The novelty and the rigor of our enhanced solution is achieved by (1) augmenting experts with sparse transfer history by leveraging enterprise taxonomy to perform regularization on rare and unseen transfers (2) response time estimation to help better reach SL compliance (3) enrichment of the transfer training data by removing non-contributing experts from historical transfers (using transfer intent identification). Here, response time estimation for transfers will be adopted from our earlier contribution presented in Section 5.6. Also transfer history enrichment will be discussed as part of our transfer intent discovery initiative in section 6.2.2. Thus, in the rest of this subsection we discuss our proposed regularization model to address the transfer sparisity challenge.

We discovered that it is fairly likely to miss many rare-transfers (from the long tail of transfers' frequency distribution) when sampling to build our training set. So we conducted a quick experiment in which five iterations of 80-20 splits for train and test got assessed. We observed an average of 18% unseen transfers in test data. Thus we found it essential to define a smoothing scheme that takes advantage of an external information source. If there is no transfer from expert A to expert B on the training set with the content ω then $P(A \rightarrow B|\omega) = 0$. This is problematic causing misclassification when there is unseen content. The general data sparsity problem happens when there is not enough training data to observe all possible events for at least a reasonable number of times. Our smoothing scheme leverages organizational neighborhood on the enterprise taxonomy in order to smoothen the transfer

probabilities. Formally:

$$\hat{P}\big(A \to B|\omega\big) = \big(1-\lambda\big)P\big(A \to B|\omega\big) + \frac{\lambda}{2}\Big(P\big(N(A) \to B|\omega\big) + P\big(A \to N(B)|\omega\big)\Big) \tag{6.1}$$

where $N(x)$ denotes neighbors (siblings) of expert x on the enterprise taxonomy. Of course the above smoothing technique has broad implications towards re-modeling the probabilistic transfer models. This can be considered as regularization (to avoid overfitting to the training data) which could now percolate down from ticket conditionals to word (bigram) conditionals. It is expected to significantly change the effectiveness of transfer models introduced earlier such as Equation 2.2. The proof of the merits and the significance of smoothed transfer modeling is yet to be determined after thorough experimentation on CENs with long tail transfer distribution.

6.2.2 Transfer Intent Discovery

Understanding CEN intentions for ticket transfers helps to (1) measure contribution of each expert in the resolution process (2) discover redundancies, and deficiencies in dealing with particular content shedding light on patterns of collaboration and problem solving behavior (3) enrich TRS and transfer training data by removing non-contributing experts to better achieve SLs by shortening the resolution sequences.

As discussed in the introduction, tickets not having a content associated with R-TRSs found to require human-centric resolution. Intent identification could mostly benefit the analysis of the non-routine tickets. Here the direction would be to discover more about the human-centric resolution process in the absence of routine workflows. More specifically, a transfer intent discovery model should target identifying intentions of experts' for collaboration. Note that intent is a property of a transfer which is the result of the knowledge

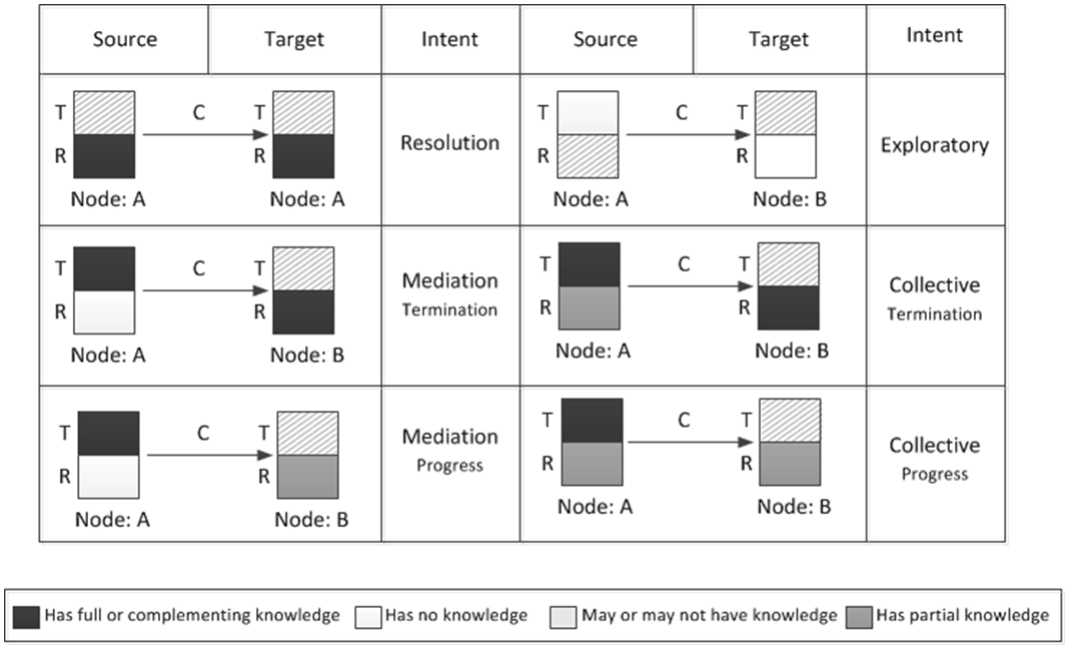


Figure 6.2: Intent characterization based on transfer and resolution knowledge

applied on content. Same edge between the same two nodes in the CEN may take different intents depending on the content of passing tickets. Figure 6.2 characterizes different types of intent identified based on the transfer and resolution knowledge at source and target experts. Please note that transfer knowledge and resolution knowledge on the ticket can be assessed by our language modeling solution introduced earlier in Section 5.6. Here the goal should be to build an intent tagger given the source, target and the transferred ticket. Construction of such a tagger helps (1) identify inefficient transfer behavior as they take place, (2) recognize the least efficient experts based on their historical transfer intent distribution, and (3) enrich the data for transfer recommendations by removing mediation/exploration intents from the training set.

6.2.3 Framework Measurements in Production

Thus far we have presented the qualifications of our resolution recommendation framework in terms of recommendations accuracy against the historical labeled data. The interactive use of a model exposes important aspects of that model that are not captured when only considering accuracy, most notably the computational cost of the features used by the model [51]. Thus, prediction accuracy is not enough to guarantee the adaptability of the recommendations. We believe deploying the framework as a pilot can enable valuable measurements on the users interaction with the system.

As future work we propose user feedback utilization after deployment. Particularly user feedback can help with usability evaluation, usability improvement, and content enrichment:

1. Usability evaluation: below we are introducing a set of cascading conditional measures which will help to analyze users' feedback:

- Tendency to use: How often do the users make request for recommendations?
- Recommendation rate on request: When a request is made, how often does the framework make a TRS recommendation?
- Adoption rate: When a TRS is recommended, how often is the TRS completely followed by the experts?
- Success rate after adoption (accuracy and R-precision in practice): When a recommendation is adopted, how often does it result in a resolution?

2. Usability improvement: For model improvement based on feedback measures a detailed error analysis on the untaken recommendations is required. Also user request data

can help us to target improvement in areas for which there are most number of requests, thus the recommendation framework can be re-engineered towards experts' needs. Here usability measures could lead to construction of user-content models aiming to maximize the overlap between (1) existence of statistical evidence for the content and (2) prevalence of demand for resolution assistance on the content. More details about the interface through which the feedback can be collected is provided in Appendix A. Figures A.1 and A.2 are presenting the prototypes of the pilot run interface and user feedback collection.

3. Automated follow up: content enrichment at the Service Desk: Tickets are being capture at the IT service desk and are summarized according to the calls received from service customers. It has been observed that many of the tickets do not contain TRS-indicative content. In other words, there is low confidence for any TRS given the content. Not surprisingly, these tickets are found to be the most puzzling ones for the agents at the time of the calls at IT service desk, where they are unable to link the content with any CI or any knowledge item at the first place before escalation. Part of this problem is due to customers' limited knowledge when reporting the problem. These tickets are also going to be costly for the CEN to resolve as the content is too general to be indicative of any resolution. Here there is a need for an on-demand real-time text-enrichment recommendations prompting agents with possible tags to append to the ticket in order to reduce the routing uncertainty. The question to be answered here would be: Given a non-indicative content, what tags can be added to increase the certainty in future routing? These tags will only get attached to the ticket if (1) they significantly increase the path prediction certainty and (2) the agent agrees to add them to the ticket. Here, advanced text-enrichment and domain-aided summarization methodologies should be studied. Solving the content quality challenges at the source of ticket generation can immensely reduce the misclassification error reported by the TRS and

Transfer recommendations. Also in term of benefits to the enterprise, it has direct impact on time-to-resolve improvement (and SL compliance), and on related knowledge retrieval and CI identification.

6.2.4 Towards Comprehensive Enterprise Decision Support

This work has entirely focused on augmenting CEN with respect to service levels, while delivering structural innovation to traditional human-intensive incident management. Finally, here we would like to briefly acknowledge some of the unexplored and open-ended avenues of research in IT Service Management; the following domains if provided with sufficient operational data can expand our current framework towards a more comprehensive enterprise decision support system:

- Service Operations Upstream:
  - Demand management – demand patterns discovery, and congestion prevention
  - Change and release management – risk assessment, incident prevention
  - Knowledge management – knowledge linking (dynamic to static) , automated request for knowledge creation/modification
- Service Operations Downstream
  - Problem and root cause – automated RCA recommendations using incident data
  - Incident management – supporting queue scheduling system to optimize experts' queue management

Lastly, we want to acknowledge the fact that new industry practices for software development and integration such as DevOps [5] can significantly alter the way in which change

and release management processes are being governed today. This could positively result in a more transparent operational data for incident management, and in turn boost the performance of data-driven recommendation frameworks (such as this work) across the entire spectrum of incidents.

6.2.5 Implications for Practice

To serve the community of practice and educators in the broad field of IT Service Management we provide here the process of knowledge discovery that was applied to Collective Expert Networks and is an important research process. We believe lessons learned in these learnings could enable other practitioners follow a similar set of practices to achieve service improvement and higher compliance and satisfaction guarantees at the time of delivery.

Critical Contextual Exploration of the IT Service Support Environment

- Learn the critical factors for contracted customers (i.e. associate cost against compliance).
- Determine primary target areas for improvement, they should portray significant impact on customers.
- For those target areas, identify the gaps in people, processes, and technologies. Study the limitations of the existing workforce. Find the areas that the human factors are deficient.
- Quantify the deficiencies with appropriate measures.
- Extract the process models using both (1) workflow documentation in the knowledge base and (2) process discovery tools based on the enterprise event data.

- Use process conformance checking to see if business-critical processes are being dealt by the staff as expected.
- If human errors exist, then targeted workforce re-training should be implemented.
- If process automation gaps are found, identify whether relying on machine learning and event log mining helps. If yes, proceed with the knowledge discovery pipeline below.

Construction of a knowledge discovery pipeline:

- Collect all the relevant event data from appropriate enterprise data marts (data identification and selection).
- Remove unnecessary attributes, estimate missing/unavailable data. (data preprocessing.)
- Perform data transformation and mapping (i.e. sessionize (ex. sort and group) and scale (ex. normalize)) to make it usable by the downstream units (data wrangling).
- Identify and count patterns, construct general rules from the data (model construction).
- Evaluate the extracted rules using standard measures. If acceptable, then process new data for interpretations. (model validation)
- If needed, re-iterate from data preprocessing step until the model passes the desired quality.
- Deploy the model as a pilot and collect users' feedback. Determine conditions under which the user will trust the results.
- Re-iterate from data selection step until reaching satisfactory feedback from users.

- Deploy the model to the production environment, and update periodically.

We believe the above steps with adequate resources for solution development, testing, and deployment can serve the IT service support and delivery ecosystem very well.

Appendix A: Prototypes of Resolution Recommendation Interface

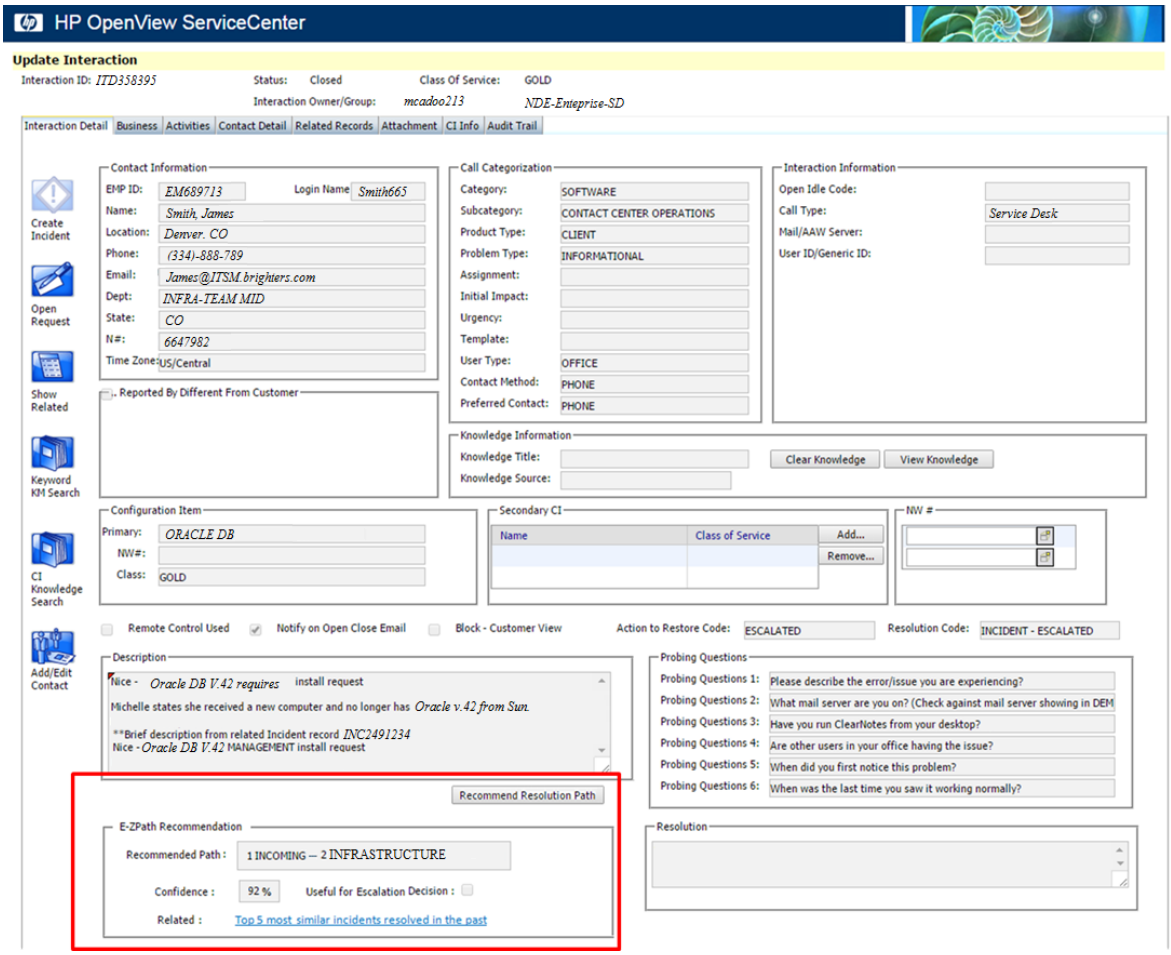


Figure A.1: At the service desk while capturing the ticket

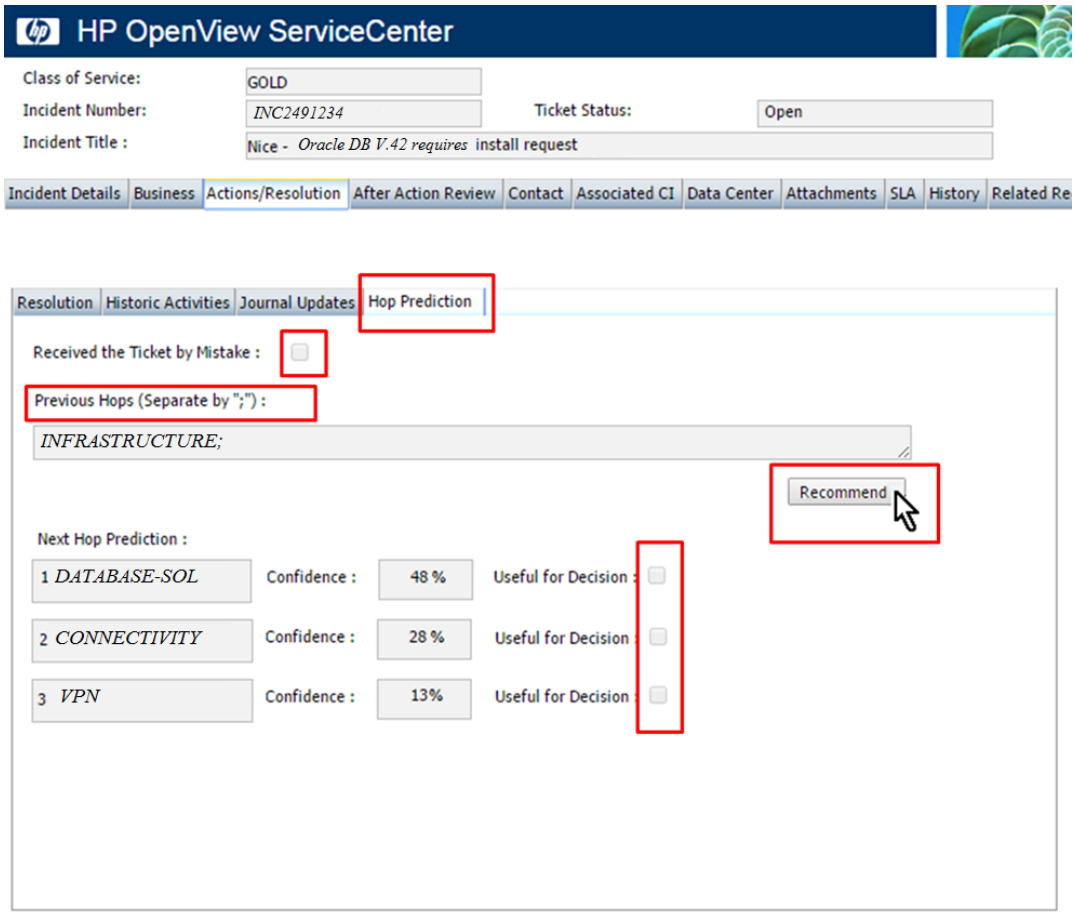


Figure A.2: At the expert groups after escalation

Appendix B: Code Repositories

All code repositories pertaining to the main contributions of this research are publicly available on GitHub: https://github.com/Kayhangamma/AugmentedCEN

For more information contact the author via: moharreri.1@osu.edu

Bibliography

[1] Shipra Agrawal, Supratim Deb, KVM Naidu, and Rajeev Rastogi. Efficient detection of distributed constraint violations. In *Data Engineering, 2007. ICDE 2007. IEEE 23rd International Conference on*, pages 1320–1324. IEEE, 2007.

[2] Leif Azzopardi, Krisztian Balog, Maarten de Rijke, et al. Language modeling approaches for enterprise tasks. In *TREC*, 2005.

[3] Krisztian Balog, Leif Azzopardi, and Maarten De Rijke. Formal models for expert finding in enterprise corpora. In *Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 43–50. ACM, 2006.

[4] Shenghua Bao, Huizhong Duan, Qi Zhou, Miao Xiong, Yong Yu, and Yunbo Cao. Research on expert search at enterprise track of trec 2006. In *TREC*, 2006.

[5] Len Bass, Ingo Weber, and Liming Zhu. *DevOps: A Software Architect's Perspective*. Addison-Wesley Professional, 2015.

[6] Christopher M Bishop. *Pattern Recognition and Machine Learning*, volume 1. Springer, 2006.

[7] Anthony Bleetman, Seliat Sanusi, Trevor Dale, and Samantha Brace. Human factors and error prevention in emergency medicine. *Emergency Medicine Journal*, 29(5):389–393, 2012.

[8] Sabine Buckl, Alexander M Ernst, Florian Matthes, René Ramacher, and Christian M Schweda. Using enterprise architecture management patterns to complement togaf. In *Enterprise Distributed Object Computing Conference, 2009. EDOC'09. IEEE International*, pages 34–41. IEEE, 2009.

[9] Bugzilla. `http://www.bugzilla.org/`.

[10] Federico Cabitza and Carla Simone. Computational coordination mechanisms: A tale of a struggle for flexibility. *Computer Supported Cooperative Work (CSCW)*, 22(4-6):475–529, 2013.

[11] Eugene Charniak. *Statistical language learning*. MIT press, 1996.

[12] Wei Chen, Yajun Wang, and Siyu Yang. Efficient influence maximization in social networks. In *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 199–208. ACM, 2009.

[13] Yi Chen, Shu Tao, Xifeng Yan, Nikos Anerousis, and Qihong Shao. Assessing expertise awareness in resolution networks. In *Advances in Social Networks Analysis and Mining (ASONAM), 2010 International Conference on*, pages 128–135. IEEE, 2010.

[14] Reuven Cohen and Shlomo Havlin. *Complex networks: structure, robustness and function*. Cambridge University Press, 2010.

[15] Cristian Danescu-Niculescu-Mizil, Robert West, Dan Jurafsky, Jure Leskovec, and Christopher Potts. No country for old members: User lifecycle and linguistic change in online communities. In *Proceedings of the 22nd international conference on World Wide Web*, pages 307–318. ACM, 2013.

[16] Ofer Dekel and Ohad Shamir. Multiclass-multilabel classification with more classes than examples. In *International Conference on Artificial Intelligence and Statistics*, pages 137–144, 2010.

[17] Kemal A Delic and Jeff A Riley. Enterprise knowledge clouds: Next generation km systems? In *Information, Process, and Knowledge Management, 2009. eKNOW'09. International Conference on*, pages 49–53. IEEE, 2009.

[18] Yixin Diao, Linh Lam, Larisa Shwartz, and David Northcutt. Modeling the impact of service level agreements during service engagement. *IEEE Transactions on Network and Service Management*, 11(4):431–440, 2014.

[19] Mohammed Elseidy, Ehab Abdelhamid, Spiros Skiadopoulos, and Panos Kalnis. Grami: Frequent subgraph and pattern mining in a single large graph. *Proceedings of the VLDB Endowment*, 7(7):517–528, 2014.

[20] Hui Fang and ChengXiang Zhai. Probabilistic models for expert finding. In *European Conference on Information Retrieval*, pages 418–430. Springer, 2007.

[21] Stuart D Galup, Ronald Dattero, Jim J Quan, and Sue Conger. An overview of it service management. *Communications of the ACM*, 52(5):124–127, 2009.

[22] Adrien Guille and Hakim Hacid. A predictive model for the temporal dynamics of information diffusion in online social networks. In *Proceedings of the 21st international conference on World Wide Web*, pages 1145–1152. ACM, 2012.

[23] Ido Guy, Uri Avraham, David Carmel, Sigalit Ur, Michal Jacovi, and Inbal Ronen. Mining expertise and interests from social media. In *Proceedings of the 22nd international conference on World Wide Web*, pages 515–526. ACM, 2013.

[24] Fangqiu Han, Shulong Tan, Huan Sun, Mudhakar Srivatsa, Deng Cai, and Xifeng Yan. Distributed representations of expertise. In *Proceedings of the 2016 SIAM International Conference on Data Mining*, pages 531–539. SIAM, 2016.

[25] James A Hanley and Barbara J McNeil. The meaning and use of the area under a receiver operating characteristic (roc) curve. *Radiology*, 143(1):29–36, 1982.

[26] HP Quality Center. `http://www8.hp.com/us/en/software-solutions/quality-center-quality-management/`.

[27] Inxite. `http://www.inxitehealth.com/`.

[28] Gaeul Jeong, Sunghun Kim, and Thomas Zimmermann. Improving bug triage with bug tossing graphs. In *Proceedings of the the 7th joint meeting of the European software engineering conference and the ACM SIGSOFT symposium on The foundations of software engineering*, pages 111–120. ACM, 2009.

[29] Victor Kaptelinin and Bonnie Nardi. Affordances in hci: toward a mediated action perspective. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 967–976. ACM, 2012.

[30] Slava Katz. Estimation of probabilities from sparse data for the language model component of a speech recognizer. *IEEE transactions on acoustics, speech, and signal processing*, 35(3):400–401, 1987.

[31] David Kempe, Jon Kleinberg, and Éva Tardos. Maximizing the spread of influence through a social network. In *Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 137–146. ACM, 2003.

[32] Donald Ervin Knuth. *The Stanford GraphBase: a platform for combinatorial computing*, volume 37. Addison-Wesley Reading, 1993.

[33] Ryan KL Ko. A computer scientist's introductory guide to business process management (bpm). *Crossroads*, 15(4):4, 2009.

[34] Heesung Lee, Jay Ramanathan, Zahid Hossain, Praveen Kumar, Ben Weirwille, and Rajiv Ramnath. Enterprise architecture content model applied to complexity management while delivering it services. In *Services Computing (SCC), 2014 IEEE International Conference on*, pages 408–415. IEEE, 2014.

[35] Ta Hsin Li, Rong Liu, Noi Sukaviriya, Ying Li, Jeaha Yang, Michael Sandin, and Juhnyoung Lee. Incident ticket analytics for it application management services. In *Services Computing (SCC), 2014 IEEE International Conference on*, pages 568–574. IEEE, 2014.

[36] Xiaoyong Liu, W Bruce Croft, and Matthew Koll. Finding experts in community-based question-answering services. In *Proceedings of the 14th ACM international conference on Information and knowledge management*, pages 315–316. ACM, 2005.

[37] Craig Macdonald, Ben He, Vassilis Plachouras, and Iadh Ounis. University of glasgow at trec 2005: Experiments in terabyte and enterprise tracks with terrier. In *TREC*, 2005.

[38] Laura Măruşter, AJMM TON Weijters, Wil MP Van Der Aalst, and Antal Van Den Bosch. A rule-based approach for process discovery: Dealing with noise and imbalance in process logs. *Data mining and knowledge discovery*, 13(1):67–87, 2006.

[39] Gengxin Miao, Louise E Moser, Xifeng Yan, Shu Tao, Yi Chen, and Nikos Anerousis. Generative models for ticket resolution in expert networks. In *Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 733–742. ACM, 2010.

[40] Gengxin Miao, Louise E Moser, Xifeng Yan, Shu Tao, Yi Chen, and Nikos Anerousis. Reliable ticket routing in expert networks. In *Reliable Knowledge Discovery*, pages 127–147. Springer, 2012.

[41] Gengxin Miao, Shu Tao, Winnie Cheng, Randy Moulic, Louise E Moser, David Lo, and Xifeng Yan. Understanding task-driven information flow in collaborative networks. In *Proceedings of the 21st international conference on World Wide Web*, pages 849–858. ACM, 2012.

[42] David Mimno and Andrew McCallum. Expertise modeling for matching papers with reviewers. In *Proceedings of the 13th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 500–509. ACM, 2007.

[43] Kayhan Moharreri, Jayashree Ramanathan, and Rajiv Ramnath. Recommendations for achieving service levels within large-scale resolution service networks. In *Proceedings of the 8th Annual ACM India Conference*, pages 37–46. ACM, 2015.

[44] Kayhan Moharreri, Jayashree Ramanathan, and Rajiv Ramnath. Motivating dynamic features for resolution time estimation within it operations management. In *Big Data (Big Data), 2016 IEEE International Conference on*, pages 2103–2108. IEEE, 2016.

[45] Kayhan Moharreri, Jayashree Ramanathan, and Rajiv Ramnath. Probabilistic sequence modeling for trustworthy it servicing by collective expert networks. In *Computer Software and Applications Conference (COMPSAC), IEEE 40th Annual*. IEEE, 2016.

[46] Hamid Reza Motahari-Nezhad and Claudio Bartolini. Next best step and expert recommendation for collaborative processes in it service management. In *International Conference on Business Process Management*, pages 50–61. Springer, 2011.

[47] Robin R Murphy. Emergency informatics: Using computing to improve disaster management. *Computer*, 49(5):19–27, 2016.

[48] Mark EJ Newman. Power laws, pareto distributions and zipf's law. *Contemporary physics*, 46(5):323–351, 2005.

[49] Daniel Oppenheim, Saeed Bagheri, Krishna Ratakonda, and Yi-Min Che. Agility of enterprise operations across distributed organizations: A model of cross enterprise collaboration. In *SRII Global Conference (SRII), 2011 Annual*, pages 154–162. IEEE, 2011.

[50] Brady Orand and Julie Villareal. Foundations of it service management with itil 2011: Itil foundation course in a book. *c. August*, 2011.

[51] Kayur Patel, James Fogarty, James A Landay, and Beverly Harrison. Investigating statistical machine learning as a tool for software development. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 667–676. ACM, 2008.

[52] Desislava Petkova and W Bruce Croft. Hierarchical language models for expert finding in enterprise corpora. *International Journal on Artificial Intelligence Tools*, 17(01):5–18, 2008.

[53] David Pultorak, Clare Henry, and Paul Leenards. Mof 4.0: Microsoft operations framework 4.0. *Zaltbommel: Van Haren Publishing*, 2008.

[54] Quora. `http://www.quora.com/`.

[55] Jay Ramanathan, Rajiv Ramnath, and Sreeram Ramakrishnan. Achieving'handoff'traceability for complex systemimprovement. In *Proceedings of the fifth annual IEEE international conference on Automation science and engineering*, pages 641–646. IEEE Press, 2009.

[56] Jason D Rennie, Lawrence Shih, Jaime Teevan, David R Karger, et al. Tackling the poor assumptions of naive bayes text classifiers. In *ICML*, volume 3, pages 616–623. Washington DC), 2003.

[57] Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. Why should i trust you?: Explaining the predictions of any classifier. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 1135–1144. ACM, 2016.

[58] Andreas Rogge-Solti and Mathias Weske. Prediction of remaining service execution time using stochastic petri nets with arbitrary firing delays. In *International Conference on Service-Oriented Computing*, pages 389–403. Springer, 2013.

[59] Gerard Salton, Anita Wong, and Chung-Shu Yang. A vector space model for automatic indexing. *Communications of the ACM*, 18(11):613–620, 1975.

[60] Qihong Shao, Yi Chen, Shu Tao, Xifeng Yan, and Nikos Anerousis. Efficient ticket routing by resolution sequence mining. In *Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 605–613. ACM, 2008.

[61] Larisa Shwartz, Daniela Rosu, David Loewenstern, Melissa J Buco, Shang Guo, Rafael Lavrado, Manish Gupta, Pradipta De, V Madduri, and Jai K Singh. Quality of it service delivery—analysis and framework for human error prevention. In *Service-Oriented Computing and Applications (SOCA), 2010 IEEE International Conference on*, pages 1–8. IEEE, 2010.

[62] Gursimran Singh, Shashank Srikant, and Varun Aggarwal. Question independent grading using machine learning: The case of computer program grading. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 263–272. ACM, 2016.

[63] Tasnim Sinuff, Neill KJ Adhikari, Deborah J Cook, Holger J Schünemann, Lauren E Griffith, Graeme Rocker, and Stephen D Walter. Mortality predictions in the intensive care unit: comparing physicians with scoring systems. *Critical care medicine*, 34(3):878–885, 2006.

[64] Stack Exchange. `http://stackexchange.com/`.

[65] Douglas M Stewart and Richard B Chase. The impact of human error on delivering service quality. *Production and Operations Management*, 8(3):240–263, 1999.

[66] Michael PH Stumpf and Mason A Porter. Critical truths about power laws. *Science*, 335(6069):665–666, 2012.

[67] Huan Sun, Mudhakar Srivatsa, Shulong Tan, Yang Li, Lance M Kaplan, Shu Tao, and Xifeng Yan. Analyzing expert behaviors in collaborative networks. In *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 1486–1495. ACM, 2014.

[68] Peng Sun, Shu Tao, Xifeng Yan, Nikos Anerousis, and Yi Chen. Content-aware resolution sequence mining for ticket routing. In *International Conference on Business Process Management*, pages 243–259. Springer, 2010.

[69] John Sweller. Cognitive load during problem solving: Effects on learning. *Cognitive science*, 12(2):257–285, 1988.

[70] Lei Tang and Huan Liu. Community detection and mining in social media. *Synthesis Lectures on Data Mining and Knowledge Discovery*, 2(1):1–137, 2010.

[71] Liang Tang, Tao Li, Larisa Shwartz, Florian Pinel, and Genady Ya Grabarnik. An integrated framework for optimizing automatic monitoring systems in large it infrastructures. In *Proceedings of the 19th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 1249–1257. ACM, 2013.

[72] Robert Tarjan. Depth-first search and linear graph algorithms. *SIAM journal on computing*, 1(2):146–160, 1972.

[73] Triage Logic. `http://www.triagelogic.com/`.

[74] Yoshimasa Tsuruoka, Jun'ichi Tsujii, and Sophia Ananiadou. Stochastic gradient descent training for l1-regularized log-linear models with cumulative penalty. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP: Volume 1-Volume 1*, pages 477–485. Association for Computational Linguistics, 2009.

[75] Nadeem Ur-Rahman and Jennifer A Harding. Textual data mining for industrial knowledge management and text classification: A business oriented approach. *Expert Systems with Applications*, 39(5):4729–4739, 2012.

[76] Wil Van Der Aalst. *Process mining: discovery, conformance and enhancement of business processes*. Springer Science & Business Media, 2011.

[77] Wil Van der Aalst, Arya Adriansyah, and Boudewijn van Dongen. Replaying history on process models for conformance checking and performance analysis. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, 2(2):182–192, 2012.

[78] Wil MP Van der Aalst. Getting the data. In *Process Mining*, pages 95–123. Springer, 2011.

[79] Wil MP Van der Aalst. Using process mining to bridge the gap between bi and bpm. *IEEE Computer*, 44(12):77–80, 2011.

[80] Wil MP Van der Aalst, M Helen Schonenberg, and Minseok Song. Time prediction based on process mining. *Information Systems*, 36(2):450–475, 2011.

[81] Kush R Varshney, Vijil Chenthamarakshan, Scott W Fancher, Jun Wang, Dongping Fang, and Aleksandra Mojsilović. Predicting employee expertise for talent management in the enterprise. In *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 1729–1738. ACM, 2014.

[82] Weiquan Wang and Izak Benbasat. Attributions of trust in decision support technologies: A study of recommendation agents for e-commerce. *Journal of Management Information Systems*, 24(4):249–273, 2008.

[83] Yingxu Wang and Vincent Chiew. On the cognitive process of human problem solving. *Cognitive Systems Research*, 11(1):81–92, 2010.

[84] WebMD. `http://www.webmd.com/`.

[85] Anita Williams Woolley, Christopher F Chabris, Alex Pentland, Nada Hashmi, and Thomas W Malone. Evidence for a collective intelligence factor in the performance of human groups. *science*, 330(6004):686–688, 2010.

[86] Shaomei Wu, Jake M Hofman, Winter A Mason, and Duncan J Watts. Who says what to whom on twitter. In *Proceedings of the 20th international conference on World wide web*, pages 705–714. ACM, 2011.

[87] Jian Xu, Liang Tang, and Tao Li. System situation ticket identification using svms ensemble. *Expert Systems with Applications*, 60:130–140, 2016.

[88] Zhe Xu and Jay Ramanathan. Thread-based probabilistic models for expert finding in enterprise microblogs. *Expert Systems with Applications*, 43:286–297, 2016.

[89] Lean Yu, Shouyang Wang, and Kin Keung Lai. An intelligent agent-based fuzzy group decision making model for financial multicriteria decision support: The case of credit scoring. *European Journal of Operational Research*, 195(3):942–959, 2009.

[90] Chunqiu Zeng, Wubai Zhou, Tao Li, Larisa Shwartz, and Genady Y Grabarnik. Knowledge guided hierarchical multi-label classification over ticket data. *IEEE Transactions on Network and Service Management*, 2017.

[91] Liang-Jie Zhang, Jia Zhang, and Hong Cai. Service-oriented architecture. *Services Computing*, pages 89–113, 2007.

[92] Wubai Zhou, Liang Tang, Tao Li, Larisa Shwartz, and Genady Ya Grabarnik. Resolution recommendation for event tickets in service management. In *IFIP/IEEE International Symposium on Integrated Network Management (IM)*, pages 287–295. IEEE, 2015.

[93] Wubai Zhou, Liang Tang, Chunqiu Zeng, Tao Li, Larisa Shwartz, and Genady Ya Grabarnik. Resolution recommendation for event tickets in service management. *IEEE Transactions on Network and Service Management*, 13(4):954–967, 2016.

[94] Yanhong Zhou, Gao Cong, Bin Cui, Christian S Jensen, and Junjie Yao. Routing questions to the right users in online communities. In *Data Engineering, 2009. ICDE'09. IEEE 25th International Conference on*, pages 700–711. IEEE, 2009.