

Exploiting Structured Human Interactions to Enhance Estimation Accuracy in Cyber-physical Systems

Yunlong Gao¹, Shaohan Hu¹, Renato Mancuso¹, Hongwei Wang¹, Minje Kim¹,
PoLiang Wu¹, Lu Su², Lui Sha¹, and Tarek Abdelzaher¹

¹University of Illinois at Urbana-Champaign, ²State University of New York at Buffalo

ABSTRACT

In this paper, we describe a general methodology for enhancing measurement accuracy in cyber-physical systems that involve *structured human interactions* with a noisy physical environment. We define structured human interactions as those that follow a domain-specific workflow. The idea of the paper is simple: we exploit knowledge of the workflow to correct unreliable sensor data. The intellectual contribution lies in an algorithm for joint estimation of the current state of the workflow together with correction of noisy sensor measurements, given only the noisy measurements and an overall workflow description. We demonstrate through simulations and a physical implementation the degree to which knowledge of workflow can increase sensing accuracy. As a specific instantiation of this idea, we present a novel situation-awareness tool called the *Emergency Transcriber* designed to automatically document operational procedures followed by teams of first responders in emergency-response scenarios. Evaluation shows that our system provides a significant fidelity enhancement over the state of the art, effectively coping with the noisy environment of emergency teams.

Categories and Subject Descriptors

G.3 [Probability and Statistics]: Markov processes; J.3 [Life and Medical Sciences]: Medical information systems

General Terms

Algorithms, Design

Keywords

workflow, unreliable sensor data, emergency

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from Permissions@acm.org.

ICCPs'15, April 14 - 16, 2015, Seattle, WA, USA.
Copyright 2015 ACM 978-1-4503-3455-6/15/04 ...\$15.00
<http://dx.doi.org/10.1145/2735960.2735965>.

1. INTRODUCTION

Our work is motivated by enhancing situation awareness in cyber-physical systems that involve structured interactions between human teams and noisy physical environments. Situation awareness is fundamental in domains, where understanding the state of the environment is crucial to the execution of the task at hand. We focus on those environments where the task is critical and failure may entail damage or loss of life. Common examples include military command and control [29], industrial plant operation [10], and emergency services [22] such as resuscitation [24] and fire-fighting [13].

In operation domains, where human risk is involved, well-established protocols typically exist that define rules of engagement, operational procedures, and contingency management. These protocols exist for at least two reasons. First, handling dangerous situations is stressful. Having well-established rules on what exactly to do alleviates the cognitive burden on the individuals involved, when they need to make decisions under pressure. Second, since unfavorable outcomes are not uncommon, rules are needed to clearly exonerate individuals from blame as long as a proper process is followed. The paper explores whether the existence of such well-established processes to structure human interactions can help improve quality of sensor measurements.

In this work, we assume that human interactions with the cyber-physical system evolve according to a predefined workflow model. This model can be obtained, for example, from an operations manual. Each state of the workflow is associated with actions that team members are allowed to perform. These actions have sensory signatures. Hence, a different expectation for sensor values exists in different states. It therefore becomes possible to use the sequence of received sensor measurements to jointly estimate both (i) the state transitions experienced by individuals following the workflow, and (ii) the most likely measured values given the obtained noisy measurements and the expected state-specific ground-truth value distribution. We show how the problem can be formulated using an extended Hidden Markov Model (HMM) [20], and investigate the impact of the resulting HMM on reducing noise in sensor measurements.

The above problem is *cyber-physical* in that it offers a novel example of how knowledge of physical processes (in this case, knowledge of structured human interactions with a cyber-physical system) can enhance computational processes (in this case, sensing accuracy). As a concrete application of this idea, we develop a novel situation awareness

tool for teams of first responders, called the *emergency transcriber*. It constitutes an audio interface for reliably recording and disseminating situation progress as extracted from the team’s audio communications. As noted above, such teams typically follow predefined collaborative workflow as dictated by the relevant engagement protocols, specifying their roles and communications. Given the critical nature of the situation, the vocabulary used is often constrained and dependent on the current stage of the workflow being executed. The emergency transcriber documents the sequence of procedural steps executed by the team, as well as their parameters, if any (e.g., dosage of medications administered). The service can help with (i) early detection of procedural mistakes, (ii) accurate logging of events for future reference, or (iii) improved interactions among emergency team members thanks to a more consistent joint understanding of followed steps in the procedural workflow.

We first explore the advantages and limitations of workflow-based enhancement of sensor data through simulations, where abstract workflow states are associated with sets of possible measurable values in the physical world, and a noisy sensor is used to determine which values were emitted. In this paper, we focus on discrete values. We also conduct a physical experiment involving a medical scenario based on the adult cardiac arrest workflow [2]. This workflow is followed in emergency rooms when patients with serious trauma arrive. Our evaluation demonstrates that we are able to achieve 80% accuracy in workflow state identification, when relying on an acoustic sensor of only a 40% accuracy to detect emitted values (in this case, spoken instructions by physicians). When the accuracy of the underlying acoustic sensor grows to 77%, our state estimation becomes 100% correct. This result speaks to the utility of the analytic framework described in this paper.

The rest of the paper is organized as follows. In Section 2, we describe our system model and the present terminology used. In Section 3, we offer a formal problem formulation and an optimal solution. We comment on the limitations of our scheme in Section 5, and discuss our case study in Section 6. Related work is covered in Section 7. We conclude this paper in Section 8.

2. PROBLEM DESCRIPTION AND SYSTEM MODEL OVERVIEW

In this section, we give a detailed description of the general problem we target, and introduce our system model.

2.1 Problem Description

Our goal is to develop an algorithm for workflow-assisted data cleaning in cyber-physical systems that feature structured human interactions with a physical environment. Figure 1 offers a simple abstract example. The figure shows a simplified workflow topology of some emergency procedure. (Perhaps it represents a medical procedure or a fire-fighting drill.) The nodes represent states (or stages) of the procedure defined in the workflow. Each state is associated with a probability of emitting certain ground-truth data objects. For example, when inspecting a person’s airway, physicians might utter words such as “air”, “breath”, “lung”, “airway”, “obstructed”, “clear” and “no”. These words correspond to the ground truth objects emitted in the aforementioned state. A noisy sensor is used to detect the values

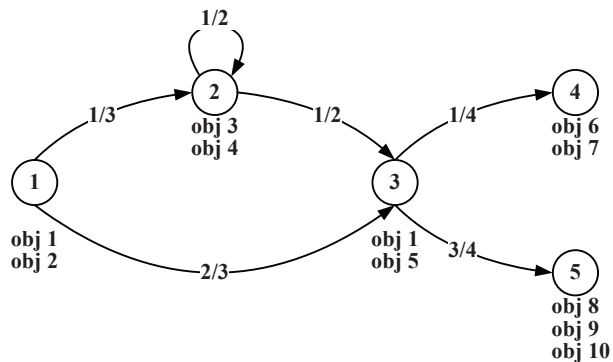


Figure 1: An example workflow

emitted. As the example suggests, we focus on discrete data. Hence, we cast the challenge as a classification problem.

To improve the quality of classification we utilize knowledge of workflow topology. Specifically, the directed edges in the workflow indicate transition probabilities among states. For instance, at *State1*, *Object1* and *Object2* are the potential measurements for the sensor(s). Upon completion of *State1*, the workflow transitions to *State2* or *State3*, with probability $1/3$ or $2/3$, respectively. Our goal is to correctly classify emitted data objects, and identify the sequence of states traversed, given the string of noisy measurements and the topology of the underlying workflow.

2.2 System Model Overview

We describe our workflow as a directed graph with a set of states, $\mathbf{s} = \{s_1, \dots, s_N\}$, where N is the total number of states. State transitions take place in accordance with a Hidden Markov Model [20]. HMMs have been widely used in modeling workflows and the emission procedures in machine-aided human translation [7], cryptanalysis [12], and time series [28]. Given the workflow, the transition probability is $t_{i,j} = p(z_{t+1} = s_j | z_t = s_i)$. We use $\mathbf{z} = (z_1, \dots, z_T)$ to denote the state transition sequence for the time $t = 1, \dots, T$, as the circles in Fig. 2 show.

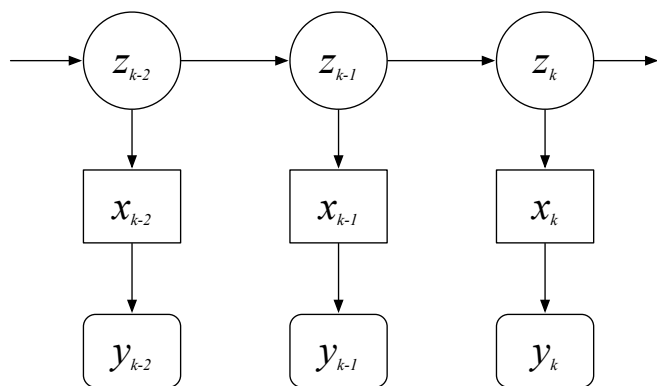


Figure 2: System Model

Note that, in practice, most states will have self-loops, since the team might take a long time executing one step of the original workflow. During that time, multiple measurements will be emitted. The probability associated with self-loops therefore will depend on the expected length of residence in the same state.

Note also that, in HMM terminology, we call our workflow states hidden, because state transitions cannot be directly observed, even though the actual state transition diagram and its parameters may be known. Instead, transitions are inferred indirectly from noisy sensor measurements. At each state, an observation of the object associated with that specific state is acquired, denoted as y_i , where $i = 1, 2, \dots, T$, as represented by the rounded rectangles in the figure. Note that, due to noise, those observations may not correctly represent the true object values. The relationship between the true object values and sensor observations is captured by a confusion matrix, which we explain in detail in the next section. The total number of possible true objects is limited, and the true object space is denoted by $\mathbf{x} = \{x_1, \dots, x_T\}$. The probability that a true object x_t is observed at state s_j is called emission probability, denoted by $e_{j,t} = p(x_t|z_t = s_j)$. Note that, to account for ground truth objects not included in the above limited set, one can add an object called “other” to indicate “everything else”. Clearly, recognizing a measurement as “other” merely says that it has not really been recognized.

We use confusion probability, denoted $c_{i,j} = p(y_i|x_j)$, to model the correspondence between sensory observations and the true object set. The confusion probability $c_{i,j}$ determines the precision of the sensor, which is defined as the ratio $\frac{\sum \text{True Positive}}{\sum \text{True Positive} + \sum \text{False Positive}}$.

As we assume the state transitions to follow the Markov model, the current state only depends upon the most recent previous state, thus we have:

$$p(z_{t+1}|z_t, z_{t-1}, \dots, z_1) = p(z_{t+1}|z_t) \quad (1)$$

and the emission probability satisfies:

$$p(x_t|z_t, z_{t-1}, \dots, z_1) = p(x_t|z_t) \quad (2)$$

3. MATHEMATICAL FORMULATION

The solution to the problem mentioned in the previous section lies in finding the sequence of states (represented by the state vector \mathbf{z}) and the sequence of objects (represented by the object set \mathbf{x}) that maximize the posterior probability $p(\mathbf{z}\mathbf{x}|\mathbf{y})$, based on inaccurate measurement of \mathbf{y} . Mathematically, we write this as follows:

$$\widehat{\mathbf{z}\mathbf{x}} = \arg \max_{\mathbf{z}\mathbf{x}} p(\mathbf{z}\mathbf{x}|\mathbf{y}) \quad (3)$$

In general, solving this equation would involve an exhaustive search for all possible state sequences and object sets, which would rapidly become intractable due to exponential complexity. Thus, we propose the following method to reduce computational effort. Based on Bayes’ theorem, the posterior probability could be expressed as:

$$p(\mathbf{z}\mathbf{x}|\mathbf{y}) = \frac{p(\mathbf{z}\mathbf{x}\mathbf{y})}{p(\mathbf{y})} \quad (4)$$

$$= \frac{p(\mathbf{y}|\mathbf{z}\mathbf{x})p(\mathbf{z}\mathbf{x})}{p(\mathbf{y})} \quad (5)$$

$$= \frac{p(\mathbf{y}|\mathbf{x})p(\mathbf{x}|\mathbf{z})p(\mathbf{z})}{p(\mathbf{y})} \quad (6)$$

The emission probability, which, by using the independence characteristics of the object set is written as follows:

$$p(\mathbf{x}|\mathbf{z}) = p(x_1, \dots, x_T|z_1, \dots, z_T) \quad (7)$$

$$= p(x_1|z_1)p(x_2|z_2)\dots p(x_T|z_T) \quad (8)$$

$$= \prod_{i=1}^T p(x_i|z_i) \quad (9)$$

Similarly, the probability of sequence of state transition \mathbf{z} is given by considering the Markov property, shown as below:

$$p(\mathbf{z}) = p(z_1, z_2, \dots, z_T) \quad (10)$$

$$= p(z_T|z_{T-1}, \dots, z_1)\dots p(z_1|z_0)p(z_0) \quad (11)$$

$$= \prod_{i=1}^T p(z_i|z_{i-1})p(z_0) \quad (12)$$

where $p(z_0)$ is the initial probability given as prior knowledge.

Furthermore, based on the confusion probability discussed in Section 2, the conditional probability for the objects that are measured is:

$$p(\mathbf{y}|\mathbf{x}) = p(y_1, y_2, \dots, y_T|x_1, x_2, \dots, x_T) \quad (13)$$

$$= p(y_1|x_1, \dots, x_T)\dots p(y_T|x_1, \dots, x_T) \quad (14)$$

$$= \prod_{i=1}^T p(y_i|x_1, \dots, x_T) \quad (15)$$

With the above in mind, and given the fact that the posterior probability is proportional to its numerator, $\widehat{\mathbf{z}\mathbf{x}}$ in Eqn. (3) could be written as follows:

$$\arg \max_{\mathbf{z}\mathbf{x}} \left[\prod_{i=1}^T p(x_i|z_i) \prod_{i=1}^T p(z_i|z_{i-1}) \prod_{i=1}^T p(y_i|x_1, \dots, x_T)p(z_0) \right]$$

We denote the final result of the above equation as $\mu_T(z_T, x_T)$, thus we can rewrite it as:

$$\begin{aligned} \mu_T(z_T, x_T) &= \arg \max_{z_{1:T}, x_{1:T}} \left[\prod_{i=1}^T p(x_i|z_i)p(z_i|z_{i-1})p(y_i|\mathbf{x})p(z_0) \right] \\ &= \arg \max_{z_{1:T}, x_{1:T}} \left[\prod_{i=1}^{T-1} p(x_i|z_i)p(z_i|z_{i-1})p(y_i|\mathbf{x})p(z_0) \right. \\ &\quad \left. p(x_T|z_T)p(z_T|z_{T-1})p(y_T|\mathbf{x}) \right] \\ &= \arg \max_{z_{1:T}, x_{1:T}} [\mu_{T-1}(z_{T-1}, x_{T-1})p(x_T|z_T) \\ &\quad p(z_T|z_{T-1})p(y_T|\mathbf{x})] \end{aligned}$$

Based on the above analysis, we propose Algorithm 1 to find the optimal state transition sequence as well as the objects that maximize the posterior probability. The algorithm takes the observations of objects **obs**, the state space **s**, the object space **o**, the workflow information containing transition probability matrix **T** and the emission probability matrix **E**, as well as the confusion matrix **C** of the sensor as input, where $t_{i,j}$, $e_{i,j}$ and $c_{i,j}$ are the transition probability, emission probability and confusion probability, respectively. It also needs the start probability of each state **start** based on domain knowledge for initialization. The output of the

algorithm is the optimal state-transition sequence \mathbf{z} and the optimal emitted objects \mathbf{x} . In the following pseudo codes, $V[t][i]$ stores the maximum cumulative probability of reaching state i at time t . $B[t][i]$ stores the previous state that transmits to the current state i , and $X[t][i]$ stores the corresponding object emitted at state i .

Algorithm 1 find optimal \mathbf{z} and \mathbf{x}

RecoverNTrack(**obs**, **s**, **o**, **start**, **T**, **E**, **C**):(**z** **x**)

```

1: for  $s_i \in \mathbf{s}$  do
2:   initialize  $V[1][i]$ ,  $B[1][i]$ ,  $X[1][i]$ 
3: end for
4: for  $obs_t \in \mathbf{obs}$ ,  $t \leftarrow 2, 3, \dots, T$  do
5:   for  $s_i \in \mathbf{s}$  do
6:     initialize  $p_{max}$ ,  $s_{max}$ ,  $v_{obj}$ 
7:     for  $o_m \in \mathbf{o}$  do
8:       for  $s_j \in \mathbf{s}$  do
9:          $p \leftarrow V[t-1][j] \times t_{i,j} \times e_{m,i} \times c_{t,m}$ 
10:        if  $p > p_{max}$  then
11:           $p_{max} \leftarrow p$ 
12:           $s_{max} \leftarrow j$ 
13:           $v_{obj} \leftarrow o_m$ 
14:        end if
15:      end for
16:    end for
17:     $V[t][i] \leftarrow p_{max}$ 
18:     $B[t][i] \leftarrow s_{max}$ 
19:     $X[t][i] \leftarrow v_{obj}$ 
20:  end for
21: end for
22:  $p_{max} \leftarrow \arg \max_{s_i \in \mathbf{s}} V[T][i]$ 
23:  $s_n \leftarrow n$  such that  $V[T][n] = p_{max}$ 
24:  $z[T] \leftarrow s_n$ 
25:  $x[T] \leftarrow X[T][n]$ 
26: for  $i \leftarrow T-1, \dots, 1$  do
27:    $z[i] \leftarrow B[i+1][z[i+1]]$ 
28:    $x[i] \leftarrow X[i][z[i]]$ 
29: end for

```

The algorithm first initializes $V[t][i]$, $B[t][i]$ and $X[t][i]$ based on the start probability **start**, confusion probability matrix **C** and the actual observation of objects **obs**. Line 4-21 is our main loop for keeping record of the maximum cumulative probability of reaching each state and its corresponding emitted object. The outer-most *for* loop iterates through each observation of objects. Starting from the loop at Line 5, we start iterating each state to find the optimal state-transition sequence. Notice that T here stands for the total number of objects observed. We set three intermediate variables, namely p_{max} , s_{max} and v_{obj} , to keep the temporary maximum cumulative probability, temporary previous state and temporary corresponding object. At Line 7, the function iterates each object in the space of **o** to get the confusion probability $c_{t,m}$. And finally, in the inner loop from Line 8 to Line 15, the function loops through each previous state to lookup for transition probability $t_{i,j}$ and compute the joint probability p . Via the inner two loops, it finds the maximum cumulative joint probability and update the corresponding entries in the matrices of V , B and X . Line 22-29 find the optimal workflow path and object sequence in a backward way. That is to say, it first finds the maximum joint probability at time T among all the states, as well as its corresponding state and object. It then starts from the

second last entry of matrix B and X to look up the overall sequences of states and objects.

We now analyse the computational complexity of Algorithm 1. We use N to denote the size of the overall state space, T the number of objects observed, and N_{obj} the size of the object space. Therefore, it is quite obvious that the space complexity is $O(N \times T)$, and the time complexity is $O(N^2 \times T \times N_{obj})$, as dominated by the main *for* loop in the algorithm. Note that this is significantly better than exhaustive search which takes $O((N \times N_{obj})^T)$ time.

4. PRACTICAL ISSUES

In this section, we discuss some practical issues that may arise when using our algorithm in real application scenarios, as well as extensions to address them.

4.1 Missing Measurements

In applications where some form of event-driven sensing is used, a common problem might be missing measurements. For example, a microphone might not register some words due to a low signal-to-noise ratio. The algorithm, therefore, must account for the possibility that missing items exist in the noisy measurement string.

Most of the time, missing measurements will not affect the algorithm, although such measurements will remain missing from the final output. This is because, as we noted earlier, in most practical workflows, states will have self-loops. That is to say, most of the time, human teams will emit multiple objects in the same state. Given that this a Markov model, the odds of another self-loop do not change with the number of times the loop was previously taken. Hence, missing one object (and one loop) does not affect future predictions.

Occasionally, however, the existence of missing measurements impacts transition probabilities between different states of the workflow. Take Figure 1 as an example, in which *State1* transits to *State3* with probability 50%, but *State1* cannot transit to *State4* directly. Suppose it was known that the previous state was *State1*, and *Object1* was emitted and recognized. Suppose the operation moved to *State3*, and an object has been emitted (e.g., *object5*) but missed by the sensor. The workflow then reaches *State4*, where one of the objects, say *Object6*, is emitted and classified correctly by the sensor. Therefore, the overall output from sensor is: *Object1* followed by *Object6*; implying that *State1* transits to *State4* directly, which is impossible according to the pre-defined workflow. If we use the basic algorithm alone, it considers measurement of *Object6* to be an error and tries to match it to objects in *State3*, according to the confusion matrix, thereby giving an erroneous classification result. Hence, we augment our solution by preprocessing the transition probability before running the basic algorithm to include the probability of missing objects.

Suppose the probability that a word is missing from the measured workflow is p . Assume that p is small enough that higher powers can be ignored. The transition probability can thus be modified according to the following equation:

$$t'_{i,j} = t_{i,j} - t_{i,j}p + \sum_{m \in M} t_{i,m}pt_{m,j} \quad (16)$$

where $t'_{i,j}$ represents the updated transition probability, and M stands for the set of source nodes of all the incoming edges of node j . $t_{i,j}$ indicates the traditional transition probability

from state i to j based on the original protocol, which is prior knowledge. $t_{i,j,p}$ indicates the probability that the workflow transits from i to j without receiving a word associated with j . The last term indicates the probability that i transits to m but the word associated m is missing at the previous round, and then m transits to j at the current round, where m has direct transition link to j . After updating the transition probability matrix as Eqn. (16), we combine it with our basic algorithm to handle situations where occasional missing measurements occur.

It remains to show how to handle situations of multiple consequent missing measurements. Here, a few issues are worth noting. First, since most states have self-loops, the odds that missing measurement result in skipping two states are low. It is likely that only one transition among different states will be missed and this case has already been handled above. If two such transitions were missed, it usually means that something unexpected has happened. We handle this situation by employing a timeout. If no input is recorded and the timeout expires, the history of previous states is ignored. The algorithm starts over, with no assumption on current state (or, equivalently some default prior distribution of probabilities of states).

4.2 Deviations from Workflow

Knowledge of human workflow helps only if team members generally adhere to the workflow. If they do not, then this knowledge is more of a hindrance since it may lead the system to disbelieve a true measurement because it violates what it knows of the workflow. In reality, the measurement could be correct albeit not compliant with the workflow. Such deviations can be accommodated by adding low-probability transitions between pairs of states where transitions are disallowed. The probability of these transitions will be equal to the probability of deviations from the original workflow. This probability represents a trade-off between optimality and robustness. A high probability of deviation will cause the original workflow topology to become less useful in correcting measurement errors. This is because it offers an alternative explanation, where unexpected measurements are attributed to deviations from the workflow and not to measurement errors. Sensor performance will be suboptimal, but the system will be robust to actual deviations.

On the other hand, a low probability of workflow deviation will maximize the benefit gained from workflow topology in improving accuracy of sensing. However, should actual deviations occur, they may cause error chains as correct measurements are repeatedly viewed as misclassifications and replaced with others according to the confusion matrix, causing an incorrect interpretation of current state, and hence, an incorrect interpretation of future input. As a safety check, when the system detects many misclassifications and low-probability transitions in a row, it assumes that its current knowledge of state may have grown inaccurate and resets by ignoring the history of previous states. Estimation starts over, with no assumption on current state.

5. SIMULATION

In this section, we evaluate the performance of the schemes proposed in the paper. We first introduce our experiment settings, and then present the results.

5.1 Simulation Settings

We consider two types of workflow in our simulation study, namely, random directed graph and directed tree, as they represent two distinct types of graphs with very different characteristics. A random graph represents a workflow where loops could be possible. In medical applications, two examples are the intubation and airway management workflow [3], and the adult cardiac arrest workflow [2]. Some workflows, however, could be abstracted as directed trees, such as the workflow for ventricular failure with cardiogenic shock [8], and the bradycardia workflow [8].

In our study, we randomly generate these two types of workflow topologies. We simulate the behaviours of sensors, which are represented by confusion matrices. By controlling the *recall* and the *intra-state object similarity*, which will be introduced in the later paragraph, we change the behaviors of the simulated sensors. For the convenience of description, ‘sensor(s)’ is short for ‘simulated sensor(s)’ in the following paragraphs. Each state of a workflow, as represented by a node in the graph, is associated with some objects, whose values were sensed by sensors.

For each topology of the workflow, a path with associated objects is randomly selected as the ground truth workflow path. The ground truth objects are randomly selected along the ground truth path. As mentioned in the above section, the confusion matrix of sensors captures the relationship between the observations and true objects. In the simulation, the confusion matrix is generated in the following way. Suppose the *recall*, which is defined as the ratio,

$$\frac{\sum \text{True Positive}}{\sum \text{True Positive} + \sum \text{False Negative}},$$

of each object is r . We assume that, for a single simulation setting, the recall is the same. Therefore, the diagonal of the confusion matrix is identical, and it represents the accuracy of the sensor. The remaining entries of the same row of the matrix are controlled by another factor called *intra-state object similarity*. It captures the total probability that an object, say *Object1* is classified as another object, say *Object2* that is associated with the *same* state as *Object1* does. For instance, one sensor has the recall of 0.6, and the intra-state object similarity is 0.3. Therefore, the total probability *Object1* could be classified by another *Object2*, which is not within the same state of *Object1* is 0.1. For a given ground truth object, the simulated sensor returns the classified result in accordance with its confusion matrix. The result is fed into our scheme, and the corrected objects and state-tracking results are computed.

To evaluate our scheme, we compute *fidelity*, which indicates the percentage of correctly classified objects, and can be expressed as $\frac{\sum \text{Correctly Classified Objects}}{\sum \text{Total Objects}}$. We plot

three quantities in the following figures. They represent fidelity of *Raw measurements*, *Workflow-aware estimates of objects*, and *Workflow-aware estimates of states*. The fidelity of *Raw measurements* represents the native sensor fidelity prior to using our correction scheme. *Workflow-aware estimates of objects* represent the fidelity of object classification when our scheme is applied. *Workflow-aware estimates of states* represents the state classification fidelity of our scheme. Usually, it goes hand-in-hand with the workflow-aware object fidelity.

Unless otherwise stated, the default parameters are set as follows. The random graph has 30 nodes for the workflow,

and average incoming and outgoing degree of the graph is 3, the object per node is 5, the path length is 8, and the recall of the confusion matrix is 0.6. For the directed tree, the default parameters are set as follows. The order of the tree is 3, the height of the tree is 5, the object per node is 5, and the path length is 6, and the recall is 0.6. We make the entries other than the diagonal of the confusion matrix have uniform distribution with the probability of $\frac{1-r}{N_{obj}-1}$, where N_{obj} is the total number of objects. The detailed simulation results is presented in the following section.

5.2 Simulation Results

Random Graph based Workflows: We first study the fidelity result for random graphs. We study the system performance from four different aspects: impact of raw sensor classification reliability, impact of traversed path length in the task workflow, impact of the average node degree in the graph and the impact of intra-state object similarity.

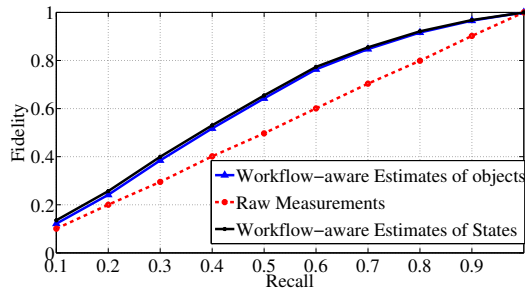


Figure 3: Impact of sensor classification reliability on system performance under graph based workflows

First, we take a look at how the reliability of raw sensors affects system performance, where we use the classification recall as a measure of raw sensor reliability. The results are shown in Fig 3. First and foremost, more reliable sensors lead to better system performance, as expected. We also observe that our proposed approach of taking advantage of workflow information brings consistent improvement across all sensor reliability settings. This improvement is up to 20%. Evidently, our approach would have a hard time improving system performance when the raw sensor reliability is too low (e.g., with only about 10% recall) or too high (e.g., sensors are perfectly reliable). We do, however, argue that in reality, “perfect” sensors rarely exist, and that system practitioners would normally not utilize completely unreliable sensors either, when building systems. Therefore, our proposed method will bring meaningful benefits to sensing systems in practice.

Next, we study how system behaves when we vary the node degree in the graph. The results are shown in Fig 4. Clearly, the performance in the baseline sensor remains the same around 60% (it is unaffected by how the workflow structure changes because it does not use the workflow). On the other hand, our proposed workflow-aware method achieves a good advantage. We do see that, as the node degree increases, the effectiveness of our method drops. This is understandable because a high node degree means higher connectivity in the graph, which in turn leads to less task progression constraints. This means that knowing the workflow gives less information. An extreme case would be a fully connected graph, which effectively provides no meaningful

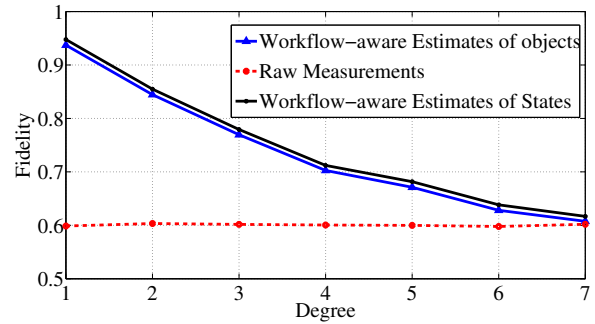


Figure 4: Impact of graph node degree on system performance under graph based workflows

information to assist with sensor data correction.

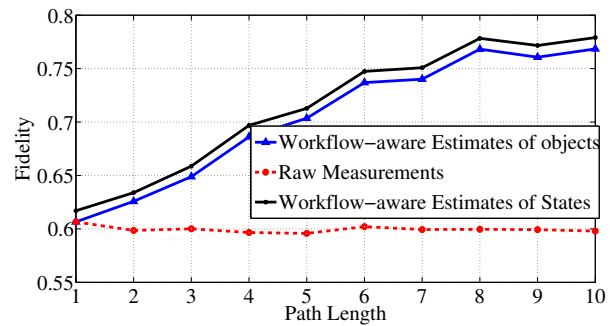


Figure 5: Impact of task path length on system performance under graph based workflows

We also look at how our method performs when varying the traversed path length under the same workflow topology. The results are shown in Fig 5. We observe that our scheme performs better than the baseline, as expected. Tasks corresponding to longer paths in the workflow topology tend to benefit more from our approach. This is because more confidence is developed over time in current state estimates, when tasks last longer under a workflow, leading to a higher error correction power from our method.

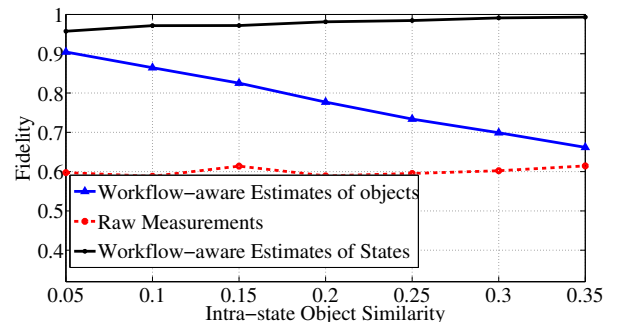


Figure 6: Impact of intra-state object similarity on system performance under graph based workflows

To test the impact of intra-state object similarity on system performance, we vary this parameter when generating the confusion matrix. The results are shown in Fig 6. As intra-state object similarity increases, it is much easier

to confuse different objects within the same state. Hence, workflow-aware estimates of objects become less accurate. Another interesting point to mention here is that the workflow-aware estimates of states increase slightly as the intra-state object similarity goes up. This is understandable because when objects have an increasing chance of being misclassified for other objects in the same state, the probability that object misclassification results in state misclassification decreases.

Tree based Workflows: We also study the sensing fidelity results for workflows with tree topologies. We investigate the impact of sensor reliability, the path length, the number of objects per node as well as the intra-state object similarity.

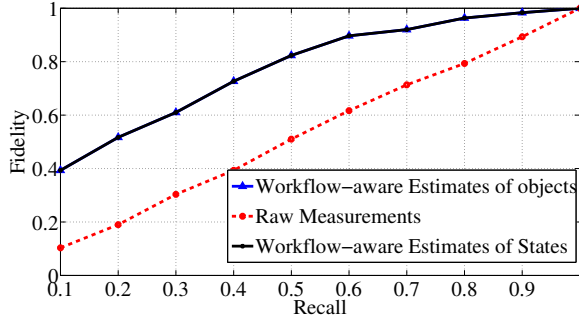


Figure 7: Impact of sensor classification reliability on system performance under tree based workflows

We first look at how the reliability of raw sensors affects system performance. The results are shown in Fig 7. The general trends of system behavior are similar to the previous graph-based experiment, as shown in Fig 3. Note that our scheme performs better in tree topologies, which is expected since there are no cycles in a tree structure, and the workflow naturally prunes impractical transitions, thus increases the fidelity.

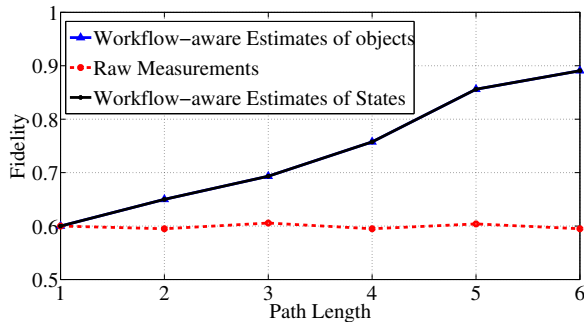


Figure 8: Impact of task path length on system performance under tree based workflows

Fig. 8 shows the results of how traversed path lengths affects system performance. We see that Fig. 8 is quite similar to the previously discussed graph-based experiment (Fig. 5), but with larger improvement.

Fig. 9 shows the results of how the number of objects per tree node affects system performance. We see that Fig. 9 is quite different from the previously discussed graph based experiments (Fig. 5). One important conclusion we can draw from the simulation result is that the number of objects per

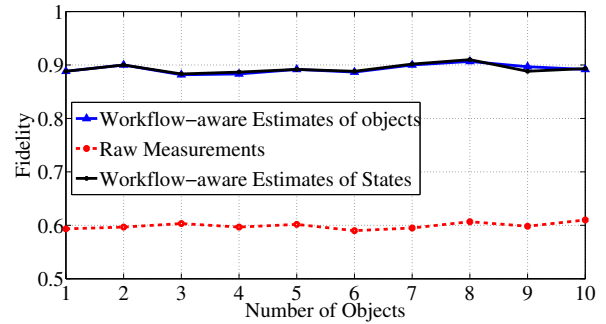


Figure 9: Impact of task object per node on system performance under tree based workflows

tree node does not have an important impact on sensing performance. Moreover, it is quite apparent that our scheme constantly outperforms the standalone sensors. Therefore, there are always a difference between the ‘Workflow-aware estimation fidelity’ and the ‘raw measurements fidelity’.

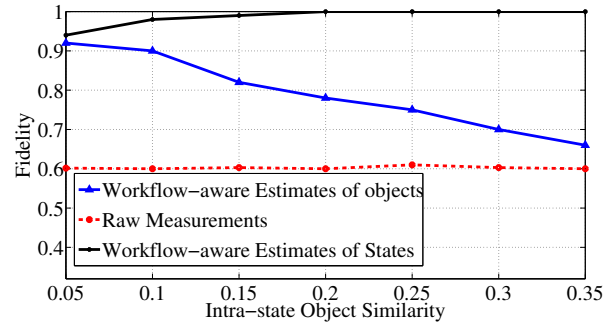


Figure 10: Impact of intra-state object similarity on system performance under tree based workflows

We also test the impact of intra-state object similarity on tree topologies. The results are shown in Fig 10. It is similar to Fig 6 that as intra-state object similarity increases, the workflow-aware estimates of objects become less accurate and the workflow-aware estimates of states increases slightly.

6. CASE STUDY EVALUATION

In this section, we apply our proposed scheme to develop a situation awareness tool, called the *Emergency Transcriber*, for teams of first responders and emergency personnel. The emergency transcriber is a workflow recording device for teams operating in emergency scenarios. It uses a simple microphone to record words uttered by team members and exploits knowledge of operation workflow to jointly improve word classification accuracy as well as keeping track of the operation state flow. Below, we evaluate the accuracy of this tool.

6.1 Experimental Settings

Workflow Information: We chose adult cardiac arrest [19] as our case of study. It strictly follows the emergency reaction algorithm shown in the Fig. 11. In realistic settings, when a patient is subject to cardiac arrest, multiple physicians and nurses operate around the patient at the same time, and medical orders are vocally communicated. The

entire environment is noisy and chaotic. General-purpose standalone voice recognition software often performs poorly in such an environment.

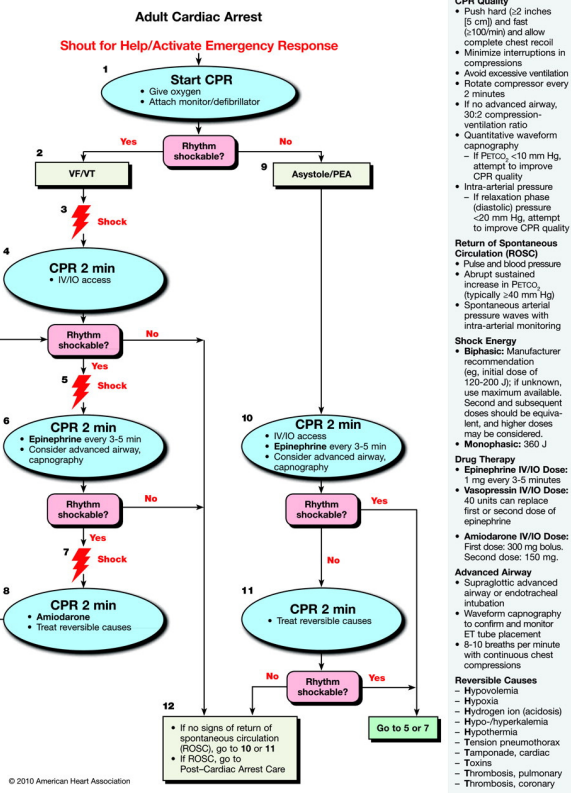


Figure 11: This illustration, from [27], demonstrates adult cardiac arrest algorithm used for resuscitation.

System implementation: Our system consists of two major components. The first component is a standalone automatic speech recognizer (ASR), which we implemented using Google Speech API [4]. It acts as an audio interface for carrying out the initial recognition of medical team’s audio communications, as indicated by $R1$ in Fig. 12. Since the ASR does not have workflow information, it tries to use a general language model and an acoustic model to match the signal it hears, which leads to errors in recognition in noisy environments. $R1$ is then fed to our emergency transcriber, which consists of two modules; a *keyword matching* module and a *word recovery and state tracking* module.

The keyword matching module first applies keyword matching to match ASR output to the most similarly sounding keywords in our workflow. This is equivalent to finding the keyword that has the maximum number of overlapping phoneme characters with the sentence transcribed by the ASR. This is a *convolution* operation. Since both $R1$ and the keywords are in the form of English text, we convert $R1$ and all the keywords into their phoneme representations using a text synthesis software [14], and then calculate the convolution using Algorithm 2.

Next, $R2$ is fed to the *word recovery and state tracking* module, where state-aware correction takes place, as described in Section 3. As an approximation, instead of training the ASR and getting it the classification confusion matrix (which is a very lengthy process), we apply Eqn. (17)

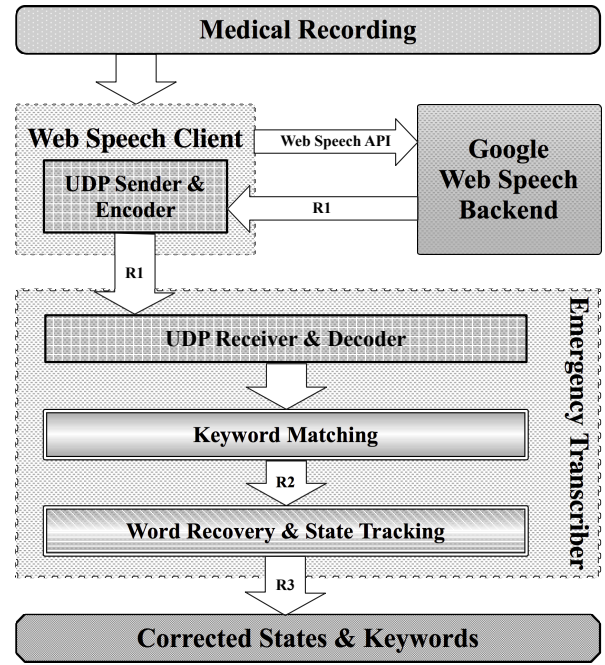


Figure 12: This illustration demonstrates the system component consisting of three major modules.

to calculate each element in the confusion matrix:

$$sim(y_i|x_j) = (1 - \frac{LD_{i,j}}{\max(\text{length}(y_i), \text{length}(x_j))}) conv(y_i, x_j) \quad (17)$$

where $sim(y_i|x_j)$ represents the similarity between the observation word y_j and the true keyword x_i . Note that, $LD_{i,j}$ represents the Levenstein distance [18] between the phoneme representation of y_i and x_j . $\text{length}(y_i)$ and $\text{length}(x_j)$ represent their phoneme length, respectively. $conv(y_i, x_j)$ represents the convolution of their phoneme representations, which captures sub-phoneme overlapping between y_i and x_j . It is calculated according to algorithm 2. We then aim to recover the actual words spoken, and reveal the actual states traversed.

6.2 Experimental Results

We invited seven people (of whom 4 are non-native English speakers) to record the script of a medical episode involving simulated emergency treatment of adult cardiac arrest that follows the workflow presented in Fig. 11. This script was designed by medical personnel from our local hospital [1] as part of a demonstration scenario of novel medical technologies. The script contained 21 sentences spoken during the simulated emergency. The script was re-enacted and the resulting audio was fed to the first component of our system.

A screen-shot of our user interface is shown in Fig. 13. Notice that the leftmost text area shows the initial results $R1$ coming out of ASR. The dynamic graph in the middle tracks and visualizes the state-transition sequence in real time, with recovered keywords shown on the right side. The red circle indicates the current state and the blue circles indicate states that have been traversed in the workflow. We then added noise of different amplitudes to the original

Algorithm 2 find maximum number of overlapping character between sentence x and keyword y

```

1:  $lenSrc \leftarrow x.length$ 
2:  $lenDst \leftarrow y.length$ 
3:  $step \leftarrow lenSrc + lenDst - 2$ 
4:  $retval \leftarrow 0$ 
5: for  $i = 0; i < step; i \leftarrow i + 1$  do
6:    $convIndex \leftarrow 0$ 
7:    $startSrc \leftarrow \max(0, lenSrc - 1)$ 
8:    $startDst \leftarrow \max(0, i - lenSrc - 1)$ 
9:   while  $startSrc < lenSrc$  and  $startDst < lenDst$  do
10:    if  $x.substring(startSrc, startSrc + 1).equals(y.substring(startDst, startDst + 1))$  then
11:       $convIndex \leftarrow convIndex + 1$ 
12:    end if
13:     $startSrc \leftarrow startSrc + 1$ 
14:     $startDst \leftarrow startDst + 1$ 
15:  end while
16:  if  $convIndex > retval$  then
17:     $retval \leftarrow convIndex$ 
18:  end if
19: end for

```

audio file, and sent it through the same pipeline. The result with average accuracy and standard deviation is shown in Fig. 14.

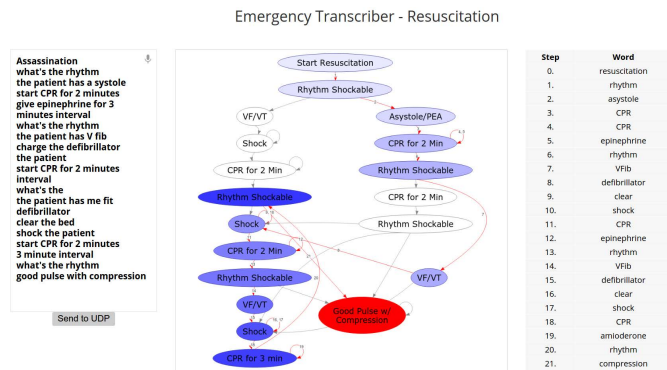


Figure 13: This graph shows the user interface of our system.

As can be seen from the result, when noise-free, the accuracy of the standalone speech recognizer is 76.69%. Keyword matching increases this accuracy by comparing this output to the entire workflow vocabulary at any stage of the workflow. Moreover, with the workflow topology information accounted for, the word recognition accuracy increased to 95.24%, accompanied by 100% state recognition accuracy, which bolsters the claim that workflow knowledge can enhance sensing accuracy. When noise (Gaussian white noise) is added to the original voice signals with a SNR (Signal-to-Noise Ratio) of 40dB, the accuracies of the word recognition of $R1$, $R2$, and $R3$ decrease, but similar trends are observed. The situation is similar when the signal to noise ratio goes up to 33dB. These results show that the emergency transcriber is a useful aid in recording emergency procedures in a range of noisy environments.

7. RELATED WORK

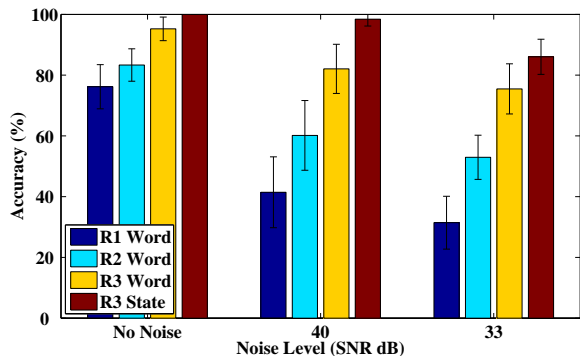


Figure 14: This graph shows the case study results on different noise levels.

Classification techniques in sensing have been widely studied. For instance, [5] studies data classification problem in wireless sensor networks. It proposed a classification approach in combining local classifier to form a global classifier to achieve high accuracy. [25, 26] proposed hierarchical aggregate classification methods to achieve high accuracy in lack of energy and label information, and the authors tested their scheme in the wild to classify bird species. Our work differs from the existing work in the sense that it takes the workflow information into consideration to increase the sensing accuracy in the face of unreliable sensors and environmental noise. What is more important is that it also keeps track of the states that have been traversed in the workflow.

We utilized the Hidden Markov Model (HMM) [20] to model the state transition and the relationship between each sensing object and its associated state. Our scheme is different from traditional HMM because the observations that the sensing system acquired is not accurate. In order to take that into consideration, we combine the confusion matrix of the sensor with the HMM layer and find the optimal sensing object as well as the hidden states as a whole. HMM is also widely used in the area of speech recognition, such as [9, 11, 30]. However, traditional HMM models the state transition between different phonemes as Hidden Markov Process. We took on a different approach where we treat the state transition behind the sensing procedure as Hidden Markov Process. For the case study specifically, the hidden states refer to the stage where physicians have been working on.

We apply our scheme in the area of speech recognition [6, 15] under medical environment. There are several commercialized speech recognition software available for medical practice purposes. For example, Nuance Dragon Medical [17] is a speech recognition package specifically for physician practices. [23] is another industrial grade speech recognition system used in the healthcare sector. Our scheme is complementary to the above-mentioned automatic speech recognizers (ASRs) because it includes the effect of workflow when doing the speech recognition. And the workflow information is free from sensor errors and environment noise, which makes our scheme perform better. It can act as a light-weight wrapper outside the ASRs for any specific use case, thus our scheme has the advantage of good compatibility and portability. Topic pertained to dialogue modeling

(DM), which uses different classification models to explore a variety of features of the dialogues, have been researched by the community of speech recognition [16, 21]. However, the main focus of DM is to predict user intention and then generate responses to interact with user, whereas our system uses the external workflow information to increase the accuracy of speech recognition results.

8. CONCLUSION

In this paper, we describe a general methodology for enhancing measurement accuracy in cyber-physical systems that involve structured human interactions with a noisy physical environment. We propose a scheme which can find the optimal sensing measurements and state transition sequence by exploiting workflow information and the confusion property of the sensors. Simulation results show that our scheme can bring performance improvement compared with standalone sensors. We instantiate our idea by carrying out a user study in emergency environment and demonstrate that our scheme can improve the speech recognition accuracy and achieve high state tracking accuracy.

Acknowledgment

This work was funded in part by NSF grant CNS 13-29886 and DTRA grant HDTRA1-1010120.

9. REFERENCES

- [1] Carle foundation hospital. <http://http://www.carle.org/>.
- [2] ACLS. Cardiac arrest algorithm. <http://goo.gl/RRy8PG>.
- [3] Roland Amathieu, Xavier Combes, Widad Abdi, Loutfi El Housseini, Ahmed Rezzoug, Andrei Dinca, Velislav Slavov, Sébastien Bloc, and Gilles Dhonneur. An algorithm for difficult airway management, modified for modern optical devices (airtraq laryngoscope; lma ctrach): a 2-year prospective validation in patients for elective abdominal, gynecologic, and thyroid surgery. *Anesthesiology*, 114(1):25–33, 2011.
- [4] Google Speech API. <http://goo.gl/64VHz5>.
- [5] Xu Cheng, Ji Xu, Jian Pei, and Jiangchuan Liu. Hierarchical distributed data classification in wireless sensor networks. *Computer Communications*, 33(12):1404–1413, 2010.
- [6] Martin Cooke, Phil Green, Ljubomir Josifovski, and Ascension Vizinho. Robust automatic speech recognition with missing and unreliable acoustic data. *Speech communication*, 34(3):267–285, 2001.
- [7] Yonggang Deng and William Byrne. Hmm word and phrase alignment for statistical machine translation. *Audio, Speech, and Language Processing, IEEE Transactions on*, 16(3):494–507, 2008.
- [8] Algorithms for Advanced Cardiac Life Support 2014. <https://www.acls.net/aclsalg.html/>.
- [9] Mark Gales and Steve Young. The application of hidden markov models in speech recognition. *Foundations and Trends in Signal Processing*, 1(3):195–304, 2008.
- [10] InterGraph. Plant operations. <http://goo.gl/poew4p>.
- [11] Biing Hwang Juang and Laurence R Rabiner. Hidden markov models for speech recognition. *Technometrics*, 33(3):251–272, 1991.
- [12] Chris Karlof and David Wagner. *Hidden Markov model cryptanalysis*. Springer, 2003.
- [13] Michael Kastner, Mohamed Wagdy Saleh, Stefan Wagner, Michael Affenzeller, and Witold Jacak. Heuristic methods for searching and clustering hierarchical workflows. In *Computer Aided Systems Theory-EUROCAST 2009*, pages 737–744. Springer, 2009.
- [14] Linux. espeak. <http://espeak.sourceforge.net/>.
- [15] Richard P Lippmann. Speech recognition by machines and humans. *Speech communication*, 22(1):1–15, 1997.
- [16] Joslé Lopes, Maxine Eskenazi, and Isabel Trancoso. Incorporating asr information in spoken dialog system confidence score.
- [17] Dragon Medical. <http://goo.gl/st0rrW>.
- [18] Gonzalo Navarro. A guided tour to approximate string matching. *ACM Comput. Surv.*, 33(1):31–88, March 2001.
- [19] Mary Ann Peberdy, Clifton W Callaway, Robert W Neumar, Romergryko G Geocadin, Janice L Zimmerman, Michael Donnino, Andrea Gabrielli, Scott M Silvers, Arno L Zaritsky, Raina Merchant, et al. Part 9: Post-cardiac arrest care 2010 american heart association guidelines for cardiopulmonary resuscitation and emergency cardiovascular care. *Circulation*, 122(18 suppl 3):S768–S786, 2010.
- [20] Lawrence Rabiner. A tutorial on hidden markov models and selected applications in speech recognition. *Proceedings of the IEEE*, 77(2):257–286, 1989.
- [21] Er I. Rudnicky. Predicting tasks in goal-oriented spoken dialog systems using semantic knowledge bases.
- [22] Christian Sell and Iris Braun. Using a workflow management system to manage emergency plans. In *Proceedings of the 6th International ISCRAM Conference*, volume 41. Citeseer, 2009.
- [23] SpeechMagic. <http://goo.gl/d9a5E6>.
- [24] Stanley. healthcare workflow. <http://goo.gl/8te7Qo>.
- [25] Lu Su, Jing Gao, Yong Yang, Tarek F Abdelzaher, Bolin Ding, and Jiawei Han. Hierarchical aggregate classification with limited supervision for data reduction in wireless sensor networks. In *SenSys*. ACM, 2011.
- [26] Lu Su, Shaohan Hu, Shen Li, Feng Liang, Jing Gao, Tarek F. Abdelzaher, and Jiawei Han. Quality of information based data selection and transmission in wireless sensor networks. In *Proceedings of the 2012 IEEE 33rd Real-Time Systems Symposium*, RTSS '12, 2012.
- [27] uwhealth.org. acls algorithm. <http://goo.gl/r6p1c0>.
- [28] Ingmar Visser, Maartje E. J. Raijmakers, and Han L. J. van der Maas. *Hidden Markov Models for Individual Time Series*. Springer US, 2009.
- [29] Wei Wang, Haining Luo, and Huiyu Deng. Research on data and workflow security of electronic military systems. In *ICICIP*, pages 705–709. IEEE, 2013.
- [30] Steve Young. Hmms and related speech recognition technologies. In *Springer Handbook of Speech Processing*, pages 539–558. Springer, 2008.