A Behavior Tree Cognitive Assistant System for Emergency Medical Services

Sile Shu[1], Sarah Preum[2], Haydon M. Pitchford, Ronald D. Williams[1], John Stankovic[2], Homa Alemzadeh[1]

Abstract— **This paper presents a cognitive assistant system for emergency medical services (EMS) that can serve as a rescue robot or virtual assistant, helping with improving situational awareness of the first responders through automated collection and analysis of data from the incident scene and providing suggestions to them. The proposed system relies on a Behavior Tree (BT) framework that combines the knowledge of EMS protocol guidelines with speech recognition, natural language processing, and machine learning methods to (i) extract critical information from responders' conversations and verbalized observations, (ii) infer the incident context, and (iii) decide on safe and effective response interventions to perform. We use a data-set of 8302 real EMS call records from an urban, high volume regional ambulance agency in the U.S. to evaluate the responsiveness and cognitive ability of the system and assess the safety of the suggestions provided to the responders. The experimental results show that the developed cognitive assistant achieves an average top-3 accuracy of 89% in selecting the correct EMS protocols and an average F1-score of 71% in suggesting the protocol specific interventions while providing transparency and evidence for the suggestions.**

I. Introduction

First responders such as EMS personnel and firefighters need to initially assess and control the situation at the accident scene and provide both basic and advanced life support to the victims prior to transporting them to a hospital. However, filtering, processing, and recording information with different levels of importance and confidence requires a significant amount of responders' cognitive effort that could otherwise be utilized for emergency response.

Previous research [1], [2], [3] proposed use of assistive technologies to improve first responders' situational awareness and decision making. Examples include using wearable assistive agents for trauma documentation and management [2], [4], developing portable communication frameworks for coordinating multiple agents (e.g., medical and communication devices, EMS vehicles) in distributed emergency response [5], [3], simulating dynamic interactions between different human agents and potential digital agents in a hospital emergency environment using state-machine based models [6], information visualization agents that present information gathered based on predicted intent and recent observations from the emergency scene [7], and robot-assisted medical reach-back in situations such as urban search and rescue [8]. However, to the best of our knowledge, none of the existing research focuses on dynamically recommending situation-aware interventions for real-time emergency response decision support.

This paper presents a cognitive assistant system that analyzes speech data from the responders' communications and observations at the scene, to infer the incident context, and suggest on the best response actions or interventions to perform based on standard EMS protocols. The proposed system can serve as a *rescue robot* or virtual assistant, interacting with a team of responders before, during, and after arrival to incident scene or during EMS training exercises. In this paper, we focus on developing the perception and cognition capabilities for such a robot. There are several challenges in design of a cognitive assistant for EMS:

- Emergency medical responders make decisions and provide interventions based on their training and knowledge of local EMS protocols. These protocols are developed and approved by a physician medical director to standardize medical care for all the responders and thus achieve excellent, consistent pre-hospital care for patients. To assist responders in their tasks, a cognitive assistant system needs to be trained with the same knowledge and have the ability to process the information from the scene and make decisions in real-time.
- Despite limited availability of pre-collected EMS scenario datasets, most of this data is not properly labeled according to the EMS guidelines. Significant amount of manual effort and domain expertise are needed for labeling such data.
- At an incident scene, the speech data might be noisy or missing critical information needed for inference, which might affect the quality of decision making and intervention suggestion by the cognitive assistant.
- Many of the EMS protocol specific interventions are safety critical in nature (e.g., Fentanyl in pain management protocols or endotracheal intubation in respiratory distress protocols) and might cause serious consequences for the patient if mistakenly suggested by the system and performed by the responder.

To address these challenges, this paper adopts a Behavior Tree (BT) framework for real-time retrieval of the critical information from the scene and inference of the correct EMS protocol specific interventions based on the retrieved information. The main contributions of the proposed framework can be summarized as follows:

- We develop a weakly supervised method for selection

*This work was supported by the award 60NANB17D162 from the U.S. Department of Commerce, National Institute of Standards and Technology (NIST).

*Authors are with the Department of Electrical and Computer Engineering[1] and the Department of Computer Science[2] at the University of Virginia, Charlottesville, VA 22904, USA {ss5de, preum, hmp4yf, rdw, jas9f, ha4d}@virginia.edu

of the most appropriate EMS protocols based on the situations inferred from the scene and the knowledge of the EMS protocol guidelines. Our evaluation using a subset of 3657 labeled EMS records indicate that this method achieves an average top-3 accuracy of 89%.

- We present two kinds of methods for suggesting the most effective interventions by the cognitive assistant: a weakly supervised knowledge-driven method based on developing executable behavioral models of the EMS protocols using BTs and a supervised data-driven ML method based on learning models from historical EMS data. Our results show that ML and BT methods achieve comparable accuracy in predicting correct interventions. However, the BT method provides more transparency and evidence for the suggested interventions and does not rely on the availability of labeled data.
- We develop a method to assign confidence scores for the protocol and intervention suggestions made by the BT model to reduce the risk of performing safety-critical interventions and prevent harm to patients. When considering the potential risk of performing incorrect interventions by responders, suggestions provided by the BT model on average have at least 22% lower risk compared to the best performing supervised ML models.

II. Problem Formulation

Our goal is to design a cognitive assistant system that can infer critical information about the situations at the accident scene, including physical status and medical history of the patients, from responders' conversations and verbalized observations. This information are represented in the form of medical or EMS semantic concepts and are mapped into the standard EMS protocols to provide suggestions on the best interventions to perform. For example, the opioid overdose protocol (Figure 1b) indicates when the first responders observe that a patient is suffering from hypoxemia (i.e., patient's SpO2 level is lower than the normal range), they need to provide supplemental oxygen to the patient.

Formally, we consider a set of standard emergency medical service protocols P. For each protocol P_i, we use a set of critical concepts (e.g., signs, symptoms, and medical history of patient) to model the conditions for which the protocol should be selected by the first responder to manage the emergency situation. We define C as the set of all the concepts describing the protocol set P. We define I as the set of all possible interventions recommended by the protocols in P. At any time t, we assume all the information verbalized by the first responder so far are included in a segment of speech data denoted as S_t. Then, we can summarize the problem as follows. At an arbitrary time t, the cognitive assistant needs to find the appropriate subset I_j in the intervention set I based on the knowledge of P_i in the EMS protocol set P according to a subset of C extracted from the speech data S_t. To solve this problem, we divide it into three consecutive sub-tasks:

(i) Extract a subset of C to represent the situation for an arbitrary time t from the speech data S_t;

(ii) Rank the EMS protocols in P and find a subset of EMS protocol P_i in P whose usage scenario is closest to what is described by the speech data S_t;

(iii) Find the intervention subset I_j

III. Approach

We propose a BT framework for implementing the natural language processing and cognitive inference by the cognitive assistant as illustrated in Figure 1a. Figure 1b shows an example of Overdose Opioid protocol sub-tree in the BT. The details are presented below.

A. Overall BT Framework

Behavior Trees are a mathematical model of plan execution used in robotics and intelligent agents, which first emerged from video game industry. Recent work has shown the potential of BTs as a flexible and interpretable data structure for representing medical processes and clinical practice guidelines in AI systems [9]. BTs can model the behavior of an intelligent agent as a directed rooted tree, presenting each sub-task as a leaf, and combine them into behaviors through nodes in a specific order [10]. A BT root generates a signal, called *tick*, periodically following a frequency *F*. Every node receiving the tick from its parent, starts execution and returns its status on achieving its goal as *success* or *failure*. There are two types of execution nodes: *Action* nodes that return success upon completion of certain action and *Condition* nodes that return success if a specific condition is met [11].

We choose BTs as an executable behavioral modeling framework for the proposed cognitive assistant due to its modularity, high responsiveness, and the ability to learn and adapt using reinforcement learning. As shown in Figure 1a, in every tick, the sequential node "Root" ticks the execution of the different nodes of the cognitive assistance pipeline, to perform conversion of text from speech, gathering important concepts from the text, transforming the concepts into vector space, protocol selection, and protocol execution/intervention suggestion. The protocol execution and intervention suggestion is implemented as a parallel node with multiple children, concurrently executing multiple applicable protocols. Every protocol node is a sequential node, which sequentially ticks the *condition* and *action* nodes, respectively, implementing the conditions to satisfy for executing the protocol and the sub-tree of the protocol logic as defined by the EMS protocols. The details of the BT nodes implementing different components of the cognitive assistant are provided next.

B. Speech to Text Conversion

The purpose of this component is transferring the input speech data S_t from first responder to text T. We apply the Google Speech API to perform speech to text conversion on the audio streams collected from the accident scene. Our previous experiments have shown that Google Speech API provides the best results among other state-of-the-art speech recognition tools [12]. As shown in Figure 1a, at every tick of the behavior tree, first the sub-task *Speech to Text*

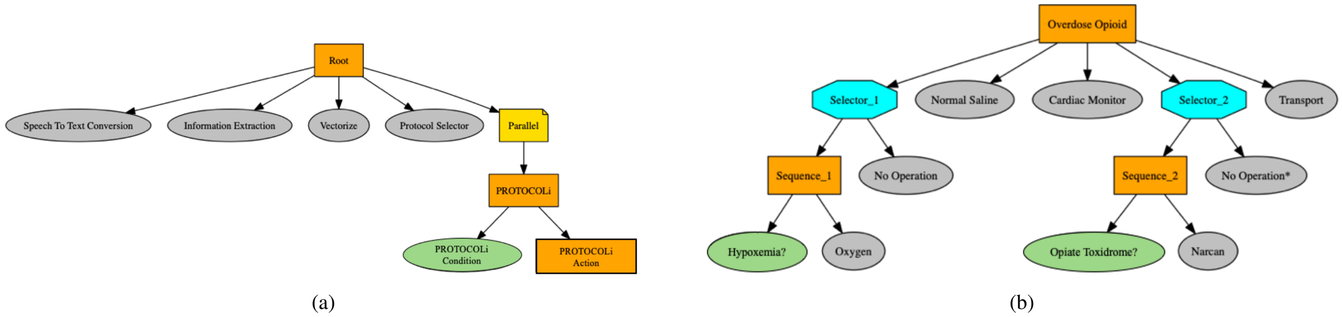


Fig. 1: (a) Overall BT framework, (b) Opioid Overdose Protocol Action Subtree

Conversion is executed to get the generated text from the incoming audio stream. Then the collected text is passed to the following components via a blackboard, a typical component in BTs to store and transport data between the sub-trees and nodes. Upon completion of these steps, the *Speech to Text Conversion* sub-task will return success to its parent node.

C. Information Extraction

After retrieving text from the speech recognition component, the collected text is fed into the *Information Extraction* component. In this component, input text T is represented by a concept set E, which is a subset of the whole concept set C, and consequently essential information for EMS can be extracted, including patient's physical condition and medical history, situations of the accident scene, and treatments performed by the first responders. The information extraction process consists of the following four steps.

1) UMLS Concept Extraction: At this step, we apply MetaMap, a widely used tool for mapping biomedical text into the concepts in the Unified Medical Language System (UMLS) Metathesaurus [13]. Using MetaMap we extract the biomedical concepts from the text along with their negation condition, semantic type, and position information. Every single concept is assigned with a unique identifier in the UMLS, called Clinical Unique Identifier (CUI).

2) Concept Filtering: Our previous analysis on MetaMap, showed that not all of the extracted concepts are useful for EMS decision support [1]. Thus, we compiled a set of EMS protocol specific concepts that are required by the EMS protocols or are frequently used by the medical responders. Each concept was then extended to an additional set of terms that share the same or similar meaning with the original concept and these terms are mapped into unique UMLS CUIs. The list of CUIs was generated by sending the original text as queries to UMLS online API and selecting the 25 most related CUIs (i.e., top 25 ranked in order of relevance). At the concept filtering stage, this extended list of CUIs (C) is used to filter the results from MetaMap and keep the concepts most relevant to the EMS protocols.

3) Value Retrieval: Next, additional information related to the concepts are extracted from the text, e.g., for the extracted concept *pulse rate*, the value of pulse rate is also extracted. For retrieving the corresponding numeric values of specific concepts such as vitals (e.g., pulse rate, blood pressure, SpO2) we find the closest number to the concept as their value via regular expression matching. We directly use the preferred names as the value of the abstract concepts (e.g. history of symptoms, quality of pain, past illness).

4) Confidence Assignment: We assign a confidence score to the extracted concepts from text to indicate the notion of uncertainty in our detected evidence from the scene due to non-perfect quality of speech recognition and concept extraction components. For confidence calculation and assignment, we multiply (i) the confidence score for the recognized words by the Google Speech API [14] and (ii) the similarity score provided by the MetaMap API indicating the level of confidence in mapping between the input text fragment and the UMLS concepts [13]. By combining these two different confidence scores, we can have a score representing the overall confidence in the information collected from the conversations of emergency responders at the scene. Incorporating other factors contributing to uncertainty and lack of confidence (e.g., missing information, noisy speech) is the subject of future work.

The collected concept set E is modeled as a dictionary with each element defined using the following unified format:

$$(C_i : P_{i,t}, V_{i,t}, T_{i,t}, Conf(C_i, t), t)$$

where C_i refers to the ith concept in the dictionary, which also serves as a key in the dictionary; $P_{i,t}$ is a boolean variable representing the presence or absence of C_i in the text at tick t; $V_{i,t}$ is a number representing the value of C_i at tick t; $T_{i,t}$ is the normalized original trigger text of C_i at tick t, and $Conf(C_i, t)$ indicates the confidence of the concept C_i at tick t. Assuming the text from which the concept C_i was detected has a speech-to-text confidence score $Conf_{G(C_i)}$ provided by Google Speech API and its CUI detected by MetaMap has a similarity score $mmScore(C_i)$, we calculate the confidence score $Conf(C_i)$ for every C_i in C as follows:

$$Conf(C_i) = Conf_G(C_i) \cdot mmScore(C_i) \tag{1}$$

An example piece of text along with the corresponding dictionary elements extracted by the *Information Extraction* phase are shown in Figure 3a. For example, the occurrence of the term GCS in the input text and its low value (i.e., 3) trigger the concept *decreased mental status* which is identified with a confidence score of 1000 by our information extraction module at tick 5.

D. Vectorizer

Once we get the concept set E representing the input text by EMS related concepts, similar to text vectorization, we can transfer the concept set E as a vector V_T, whose size equals the length of the concept set C and values are the confidence scores $Conf(C_i, t)$ for each extracted concept C_i in E. Each item in the input text vector indicates if the concept has appeared in the input text and how much confidence we have for its mapping (mapping textual contents to concepts). Thus, if any concept C_i is detected at tick t, the corresponding item in the text vector will be encoded with a value of $Conf(C_i, t)$.

We also use a set of vectors V_P to represent the concepts related to signs and symptoms that are required for the execution of a specific EMS protocol. Each protocol in protocol set P is represented as a vector V_{P_i}, whose size also equals the length of the concept set C but values are assigned with different weights based on the importance of these concepts in selecting the protocol. These weights are assigned by real first responders participating in our project. Formally, these two vectors can be represented as follows:

$$\vec{V_T} = \{Conf(C_i) | \forall C_i \in C\} \tag{2}$$

$$\begin{aligned} \vec{V_{P_i}} = &\{Weight(C_j) | \forall C_j \in C \\ &\wedge Weight(C_j) = Softmax(Pri_{i_j}) \\ &\wedge Pri_{i_j} \in \{0, 1, 2, 3\}\} \end{aligned} \tag{3}$$

where Pri_j is a priority score assigned based on the relevance between the protocol P_i and concept C_j (with 3 representing most relevance and 0 representing no relevance). We apply softmax function to normalize these scores into weights and make them add up to 1.

E. Protocol Selection

Given the input text vector V_T, representing the information gathered from the scene at tick t and the protocol vector set V_P, representing the required concepts (conditions, signs and symptoms) for executing a specific EMS protocol, we take a weakly supervised approach to determine the relevance between the current situation at the scene and each EMS protocol in P by calculating the similarity between their vectors. For this, cosine similarity, as a commonly used metric in information retrieval and question answering systems is used. Thus, the similarity or relevance between the text ($\vec{V_T}$) and protocol ($\vec{V_{P_i}}$) vectors is calculated as follows:

$$S_i = \frac{\vec{V_T} \cdot \vec{V_{P_i}}}{\|V_T\| \cdot \|V_{P_i}\|} \tag{4}$$

After calculating the cosine similarity between a given text vector and all the protocol vectors in our library of EMS protocols, we rank the protocols based on their similarity to the input text and select the ones with highest scores as the appropriate protocols to be executed by the cognitive assistant system. If multiple protocols have a high relevance score, an ordered list of candidate protocols will be selected and used for the feedback generation. We assign the cosine similarity index calculated for each protocol as a confidence score for its selection and normalize the confidence scores such that the sum of all scores in the final list is equal to 1. For a subset of protocols from P, called $Candidate$, containing top N protocols based on their cosine similarity scores, the normalized confidence score of each candidate protocol, $Conf(P_i)$, is calculated as follows:

$$Conf(P_i) = \begin{cases} \frac{S_i}{\sum S_j}, & \forall P_j \in Candidate \\ 0, & \text{otherwise} \end{cases} \tag{5}$$

This normalization of the confidence scores provides a frame of reference to the responders for comparing the scores and potentially considering the protocols with the higher scores. It also enables confidence propagation and assignment to the interventions suggested by the BT framework.

F. Protocol Execution - Intervention Suggestion

Typically, each EMS protocol describes some specific rules to perform interventions in an emergency scene. The conditions in these rules are signs, symptoms and medical history of the patient, that are extracted and represented as concepts in the previous components (see example in Figure 1b). So, we model the execution logic for each EMS protocol as a separate sub-tree in the BT whose children implement the conditions to be checked and actions or interventions to be taken as part of the protocol. All of the protocol nodes are connected to a parallel parent node, which enables all the selected candidate protocols to be executed concurrently at the same time and suggest most relevant interventions with the highest confidence score to the responder. Due to the modularity of the BTs structure, the set of EMS protocols can be easily replaced or extended by merely replacing or adding to the sub-trees under the parent node.

There is an obvious risk to execute protocols concurrently in this system. In most cases, extra protocols will be executed, and consequently, inappropriate or even safety-critical feedback might be suggested to the responder. To avoid such risks, we have extended the BT framework with a new capability for assigning confidence values to the nodes and propagating them through the execution path on the BT. This enables us to provide a confidence for each final feedback generated by the selected protocols and let the responder consider different interventions with different confidence levels. When calculating the propagation of the confidence scores on the paths of the BT, we assume that the appearance of the concepts in the protocols are independent events from each other and they are also independent from the event that a protocol is selected. Thus, we assign a confidence score to every final feedback node (leaf action or intervention node in the protocol subtree) by multiplication of confidence scores assigned to previous nodes in the path to that node, including the concepts and conditions observed in the input text and the the protocols selected. Figure 2 shows an example of the propagation of confidence scores from a selected protocol and an observed concept in text into an action node on the BT. Finally, The interventions with a confidence score of less than 0.1 are filtered out from the final list of suggestions presented to the responder. As a result of applying the above-mentioned mechanism, the safety-critical and potentially incorrect interventions tend to have lower confidence scores. Because:

(i) The initial confidence assigned to each protocol is based on the similarity between the text vector and protocol vectors, which means the interventions within the less relevant protocols will be assigned with lower confidence scores.

(ii) Even if an irrelevant protocol is selected by the model, some of the interventions suggested by these protocols are less likely to be suggested because the relevant observations are not extracted from the scene and required conditions for those interventions are almost impossible to be satisfied. For example, if chest pain protocol is triggered in a abdominal pain case, "STEMI" is less likely to appear in this case and corresponding interventions will not be suggested.

(iii) In EMS protocols, the safety-critical medications/interventions typically have more conditions/prerequisites to be satisfied and some of them can only be performed when other less safety-critical interventions were not effective (e.g., Fentanyl will only be administered when pain persists after giving Nitroglycerin in Chest Pain protocol). Thus, these interventions tend to have lower confidence scores and more likely to be filtered by the confidence threshold.

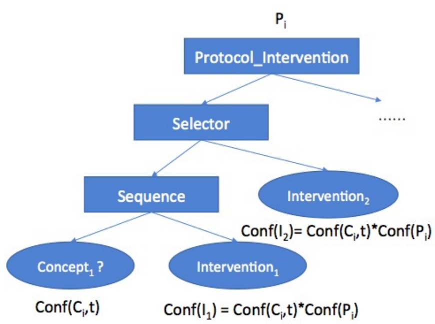


Fig. 2: Confidence propagation in a protocol action sub-tree

IV. Evaluation

Two sets of experiments were performed to evaluate the BT framework. First, we assessed the accuracy of selected protocols by the automated protocol selection procedure. Second, we executed the top three selected candidate protocols in parallel on the BT framework and compared the suggested interventions by the system with the actual interventions performed by the first responders as logged in the data. We also compared the performance of BT framework with supervised ML methods trained on historical EMS data.

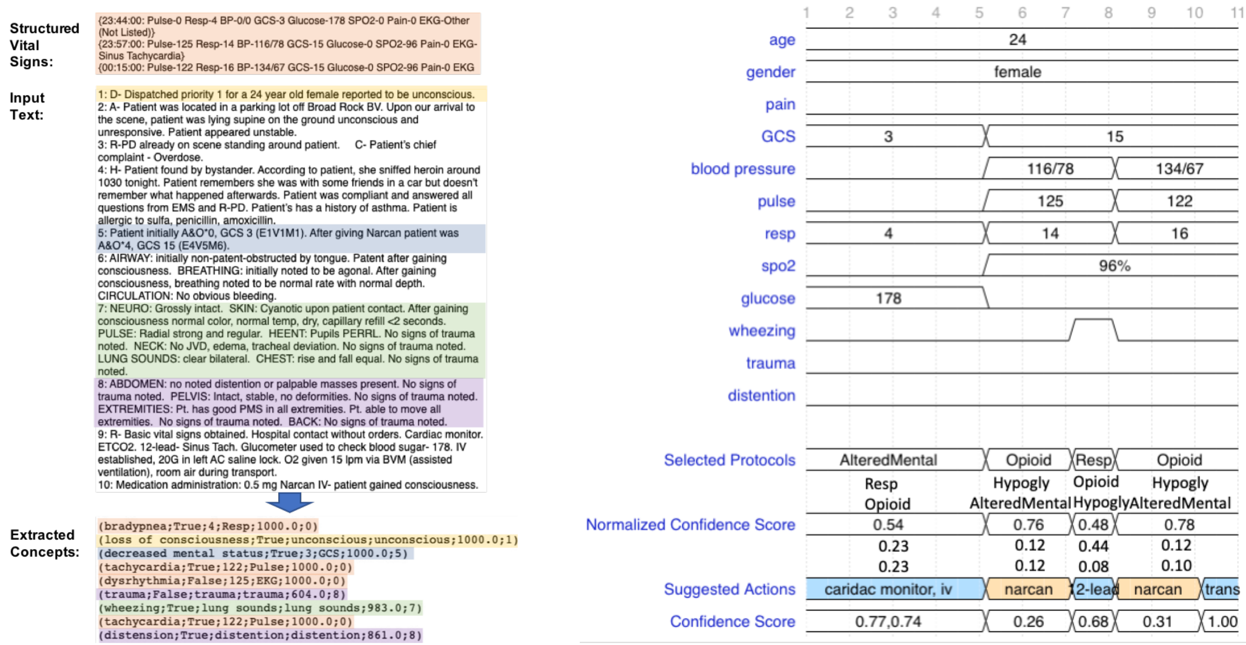
A. Experimental Setup

For these two experiments, we considered 8 commonly used EMS protocols from a regional set of protocol guidelines and a dataset of 8302 pre-hospital call sheets from a regional ambulance authority (RAA). The information inside these reports are originally organized into several categories including the type of the call, priority of the dispatch, chief complaint from the patient, first and second impressions from the first responders, vital signs recorded in the emergency scenes, interventions taken by the first responders, outcome after the interventions and the narratives describing the emergency situations. Narratives and vital signs were fed to our model as inputs because the narratives from the first responders and the vital signs are the only information that we can directly obtain from verbal conversations in the emergency scenes. In these experiments, we used the textual narratives and vitals transcribed by the responders as they are directly available in the data and do not require speech to text conversion. So the $Conf_G(C_i)$ score in Equation 1 was always set as 1. The results of evaluating the speech to text conversion step for both noise-free and noisy realistic audio data from incident scenes were presented in [1].

For a subset of 3657 records, the actual protocols used by the responders were labeled by an EMS responder participating in our project with advanced life support training. This labeling was used as ground truth for assessing the accuracy of the protocol selection component. This was done by developing a set of rules unique to each of the pre-selected protocols to filter out cases that were either ambiguous or fell into another protocol. For example, in order for a case to be labeled as an opioid overdose, the medication Naloxone must have been administered and the documented field impression must indicate that the original responder believed the patient's presentation was due to an overdose. Thus, we marked the cases that the medication Naloxone was given and the impressions included opioid overdoses. First responders' interventions recorded in these reports served as the ground truth to evaluate the suggestions generated by our model and the quality of the feedback to responders.

It should be noted that in actual deployment, the narratives and vital signs will be extracted from streaming speech data and fed into the system, as they become available. So, the cognitive assistant system has to perform context inference and provide suggestions on protocols and interventions in real time using partial information and evidence. However, due to unavailability of such data, in these experiments we use the final complete narrative transcribed by the responders for evaluation. Evaluating the performance of system using streaming data and assessing the impact of partial information on the accuracy of results is the subject of future work.

We developed multiple machine learning (ML) models with several variations of hyper-parameters that were trained on the RAA data to perform intervention prediction. These models were used to compare the performance of the intervention prediction by our proposed weakly supervised knowledge-driven BT method. We specifically compared the following three supervised ML models for intervention prediction to our BT method: Support Vector Machines (SVM), Random Forest (RF), and Decision Trees (DT). To train these supervised ML models, we used the intervention column in the RAA reports as the ground truth and the narrative text as input. For feature extraction, each narrative was represented in vector space after pre-processing (i.e., tokenization, normalization, stemming, stopwords removal). Three different kinds of feature vectors, namely, n-grams,



(a) Example output from information extraction stage

(b) Example output from protocol execution & intervention prediction

Fig. 3: An example of the results from the proposed BT Cognitive Assistant System

uni-grams, and signs and symptoms were coonstructed from the narratives and used for ML models. The test dataset for intervention prediction included 1000 RAA EMS reports. The remaining 7302 EMS reports were used to train the ML models. We applied 5-fold cross-validation by splitting the training data consisting of 7302 reports into 5841 training cases and 1461 validation cases and trained multi-class classifiers (for 94 intervention classes) using the three supervised ML models. To achieve a fair comparison between the risk-aware, knowledge-driven, weakly supervised BT method and the data driven, supervised ML models, we also added the following two settings to the ML models: (i) Training the ML models using a **class weighting** approach where intervention classes with higher risk scores were assigned lower weights to direct the ML models towards selecting less safety-critical interventions with lower risk factors; (ii) Applying a similar **confidence score filtering** implemented for the BT model (in Section III.F) to filter the interventions with low confidence scores from the list of suggestions by the ML models.

B. Experimental Results

Protocol Selection. To evaluate the automated protocol selection procedure, we compared the ranked list of protocols selected by the cognitive assistant with the ground truth protocol labels in the test data annotated by the first responders participating in our project. Since the protocol selection component generates a ranked list of top 3 protocols with their confidence scores, we applied a top-3 accuracy metric to evaluate if the target label by the responder is one of the top 3 predictions by the cognitive assistant. Our experiments with a set of 3657 test cases showed an **average top-3 accuracy** of **89.0%**.

We identified the following reasons for sub-optimal performance of our protocol selection method based on our review of the cases where the protocol selection method resulted in incorrect predictions (i.e., it's predictions did not match the labels provided by the EMS responders).

- Errors occurred in mapping between input text and standard concepts in our Information Extraction component, which led to generation of inaccurate text vectors and consequent generation of wrong ranking for the selected protocols. These errors were due to: (1) MetaMap not recognizing the required concepts as CUIs; (2) Some identified CUIs by MetaMap not appearing in the mapping between CUIs and standard concepts in our model; (3) CUIs and concepts not precisely matching (e.g., We get the CUI "Respiratory Sound" from UMLS mapping to the required concept "Wheezing." However, they are not the same since wheezing is one kind of respiratory sound. Thus, some other respiratory sounds will be mapped to wheezing, which leads to mapping errors.); (4) MetaMap not producing the correct negation detection results, leading to failure in identifying the presence of some concepts in the input text.
- Protocol vectors were manually developed based on the descriptions of concepts representing signs and symptoms in the set of protocols. The value of each concept in the protocol vector was assigned with different weights based on their importance, as reviewed and ranked by one of the participating EMS responders in our project. Some of the manually assigned weights in the protocol vectors caused errors.
- In some cases, missing critical information (e.g., incomplete vital signs) affected the correctness of the text vector representing the EMS narration.

Intervention Suggestion. In this experiment, we evaluated the performance of the intervention suggestion module of our cognitive assistant by formulating it as a multi-class classification problem. We used the list of interventions performed by the responders in the dataset as ground truth and compared it with the list of interventions suggested by the cognitive assistant system. We define the predictions which appear in the ground truth as true positives (TP), while the ones which are not included in the ground truth as false positives (FP). We define the interventions in the ground truth that are missed by our cognitive assistant as false negatives (FN). By calculating the TPs, FNs and FPs for each RAA case, we use both weighted and micro average precision, recall and F1-score to evaluate the performance of the intervention prediction. The weighted metrics calculate precision, recall, and F1-score for each class, and find their weighted average based on the number of instances for each class to take the class-imbalance into account.

In addition to traditional methods for evaluation of multi-output prediction results, we also consulted with first responders about the FN and FP intervention predictions because some of the suggested interventions although reasonable, might not be performed by the first responders and some of the suggestions are too risky to be performed at the scene. EMS protocols are written in terms of escalating clinical care, therefore even if an intervention is indicated under a certain protocol the responder may not perform it due to time or resource limitations. Further, EMS protocols prioritize life and limb saving interventions over comfort measures, and simple interventions are preferred over the complex ones whenever possible. Under this consideration, we used another metric to evaluate our intervention suggestion method in terms of the **risk incurred by the interventions**.

All the suggested interventions were classified into four distinct classes of red, orange, yellow and green. This is according to the severity of the condition that the intervention addresses as well as possible side effects they might have for patients when incorrectly suggested (FPs) or not suggested (FNs). These severity levels were then encoded as different risk scores $Risk(I_i)$ from 1 to 4 assigned to each intervention class. Larger scores indicate a higher risk if an incorrect intervention is suggested to the responder. Then for each EMs test case with a set of I interventions, we calculated the average risk factor of the suggested interventions by summing the products of the risk scores $Risk(I_i)$ and confidence scores $Conf(I_i)$ of the incorrect interventions (FP or FN) provided by the model and normalized it by dividing by the number of ground truth interventions for each case. The average normalized risk to evaluate the performance of model over n test cases was calculated as follows:

$$Avg.\ Normalized\ Risk = \frac{1}{n} \frac{\sum Conf(I_i) \cdot Risk(I_i)}{|I|} \quad (6)$$

The evaluation results using the metrics mentioned above are shown in Table I. Our results show that the knowledge driven weakly supervised BT method has a comparable performance to the supervised ML methods that use signs and symptoms as their feature set when evaluated using the traditional evaluation metrics (precision, recall, and F1) used for multi-class classification task. However, the average normalized risk factor of the interventions suggested by the BT model is at least 22% (0.34 vs. 0.44) lower than the best performing supervised ML model that uses signs-symptoms features (i.e., Linear SVM model). This means that we can effectively avoid suggesting safety-critical interventions using the confidence propagation and filtering mechanisms of the BT model.

It should be noted that ML models using unigram features outperformed the models using ngrams so we only report the results for unigram features in Table I. Also, in case of the ML models using unigram features, the unigrams were extracted from the semi-structured narratives that often contain the interventions performed by the responders at the scene. Thus the ML models using this feature set performed significantly better than the ML models using signs-symptoms features and the BT model because of having access to the actual ground truth or prediction labels. So, for fairness of comparison we mainly focus on the ML models that use signs and symptoms as their feature set.

The supervised ML methods trained with class weighting perform worse in terms of precision, recall, and F1 than the models with no knowledge of risk scores, and they also yield higher average risk factors. One reason for these results might be the extra FN and FP predictions brought by the weight assignments for the ML models. On the other hand, the ML models with filtering, which get rid of predictions with confidence scores lower than a threshold, only slightly reduced the risk factor compared to the original models (0.24 to 0.23 for SVM-unigram, 0.42 to 0.39 for RF-unigram).

Furthermore, the proposed BT method has the following advantages compared to supervised ML models:

- The BT model has high modularity, which means when we need to edit/add/remove any EMS protocols in the model, what we need to do is only substituting/inserting/deleting the corresponding protocol subtrees. However, when it comes to supervised data-driven ML methods, re-collection and labeling of data and retraining the whole model is required.
- The BT method is weakly supervised and knowledge-driven and does not rely on the availability of training data and correctness of labels. Whereas the performance of supervised ML methods greatly relies on the quantity and quality of the training data and labels.
- Contrary to ML methods which are black box end-to-end solutions from input text to intervention suggestions, the BT framework is transparent and can provide explanation and evidence for the decisions made and suggestions provided to the responders.

V. Discussion and Future Work

From the evaluations conducted in the previous section, the following major challenges were identified:

- The concept list used in the information gathering phase is currently manually created and is limited to the knowledge of protocols and, thus, it might be a possible

Model		**Weighted Precision**	**Micro Precision**	**Weighted Recall**	**Micro Recall**	**Weighted F1 Score**	**Micro F1 Score**	**Cross-Validation Micro F1 Score**	**Avg. Risk**
Behavior Tree		0.65	0.76	0.66	0.66	**0.64**	**0.71**	NA	**0.34**
Linear SVM	unigram	0.88	0.92	0.86	0.88	**0.87**	**0.90**	0.88	0.24
	unigram, filtering	0.88	0.92	0.86	0.88	0.87	0.90	0.88	**0.23**
	signs-symptoms	0.62	0.81	0.63	0.64	**0.61**	**0.72**	0.73	**0.44**
Random Forest (RF)	unigram	0.84	0.91	0.72	0.71	0.74	0.80	0.78	0.42
	unigram, weighted	0.76	0.88	0.63	0.65	0.65	0.74	0.74	0.49
	unigram, filtering	0.84	0.91	0.71	0.71	0.74	0.80	0.78	0.39
	signs-symptoms	0.65	0.76	0.60	0.60	**0.66**	**0.67**	0.72	**0.60**
Decision Trees (DT)	unigram	0.82	0.84	0.80	0.82	0.81	0.82	0.83	0.28
	unigram, weighted	0.77	0.80	0.79	0.79	0.77	0.80	0.79	0.39
	unigram, filtering	0.80	0.84	0.80	0.81	0.80	0.83	0.81	0.28
	signs-symptoms	0.62	0.64	0.60	0.61	**0.60**	**0.63**	0.67	**0.82**

TABLE I: Intervention suggestion provided by the weakly supervised BT model vs. the supervised ML models, Support Vector Machine (SVM), Random-Forest (RF), and Decision Tree (DT). *Weighted* supervised models are trained with classes weighted using inverse intervention risk scores. Models with *filtering* use confidence score filtering to remove interventions with low confidence scores from the final list of suggestions. The ML models using unigram feature set outperform the BT model since they contain the ground truth prediction labels (interventions) that are embedded in the input text. So for a fair comparison, we consider the signs-symptoms feature set. Among the models using signs-symptoms as the feature set, the SVM model outperforms both the RF and the DT models with a micro F1 score of 0.72 and the BT model performs similarly to SVM with a micro F1 score of 0.71. However, the BT model results in a 22% reduction in normalized average risk factor (0.34 vs. 0.44 for the SVM model).

reason for missing important concepts from the input text. Also, as the number of the target EMS protocols grows, the amount of effort needed for modeling the protocols and manually extending the concept list significantly increases. Thus, we plan to develop methods for automated creation of more complete and accurate concept lists in the future. We are investigating the vector space models and weakly supervised techniques to expand the limited set of manually identified concepts to a larger lexicon of EMS relevant terms.

- The inaccuracies in detection of presence or absence of the concepts in text largely affect the results of the protocol selection and execution phases. Currently, we rely on the negation detection features of MetaMap to extract the absence of concepts. Future work will focus on developing techniques for more precise detection of concept presence and absence in EMS domain.
- The unified dictionary format for representing and collecting the extracted information, the protocol conditions, and the modularity of behavior tree models enable scalability of the BT framework. We plan to study the possibility of automatically creating and extending the behavior tree models based on EMS data or protocols.
- Interventions performed by a first responder are necessarily limited by the underlying context such as transport time, severity of patient illness and resources available. Systematically accounting for these contexts would improve and better account for both the safety and rate of the intervention suggestion false positives.
- When deployed in the real incident scenes, the cognitive assistant system might need to perform context inference and decision making based on the partial and incomplete information extracted from streaming speech data. Evaluating the performance of system using streaming data is the subject of future work.

VI. Conclusion

This paper presented a Behavior Tree cognitive assistant system for emergency response which can be implemented as a rescue robot or virtual assistant interacting with the responders at the incident scenes to provide them with suggestions on the most appropriate protocols and interventions to execute. Our experimental results show that supervised ML methods trained on historical EMS data might perform similarly or better than the knowledge-driven BT method when compared using traditional accuracy metrics. However, the proposed BT modeling framework provides better guarantees on the safety of interventions suggested to the responder as well as transparency and evidence. The proposed cognitive assistant system has also the potential to be used during simulated training experiments for preparing responders with the knowledge of protocol guidelines and scoring their performance in executing the protocols.

References

[1] S. Preum *et al.*, "CognitiveEMS: A Cognitive Assistant System for Emergency Medical Services," in *Special Issue on Medical Cyber-Physical Systems Workshop (CPS-Week 2018)*, vol. 16, no. 2. ACM SIGBED Review, 2019.

[2] A. Croatti, S. Montagna, and A. Ricci, "A personal medical digital assistant agent for supporting human operators in emergency scenarios," in *Agents and multi-agent systems for health care*. Springer, 2017, pp. 59–75.

[3] F. Bergenti and A. Poggi, "Developing smart emergency applications with multi-agent systems," *International Journal of E-Health and Medical Communications (IJEHMC)*, vol. 1, no. 4, pp. 1–13, 2010.

[4] A. Croatti *et al.*, "BDI personal medical assistant agents: The case of trauma tracking and alerting," *Artificial intelligence in medicine*, 2018.

[5] E. Domnori, G. Cabri, and L. Leonardi, "Ubimedic2: An agent-based approach in territorial emergency management," in *2011 5th International Conference on Pervasive Computing Technologies for Healthcare (PervasiveHealth) and Workshops*. IEEE, 2011, pp. 176–183.

[6] M. Taboada *et al.*, "An agent-based decision support system for hospitals emergency departments," *Procedia Computer Science*, vol. 4, pp. 1870–1879, 2011.

[7] F. Meneguzzi *et al.*, "A cognitive architecture for emergency response," in *Proceedings of the 11th International Conference on Autonomous Agents and Multiagent Systems-Volume 3*. International Foundation for Autonomous Agents and Multiagent Systems, 2012, pp. 1161–1162.

[8] R. R. Murphy, D. Riddle, and E. Rasmussen, "Robot-assisted medical reachback: a survey of how medical personnel expect to interact with rescue robots," in *RO-MAN 2004. 13th IEEE International Workshop on Robot and Human Interactive Communication (IEEE Catalog No. 04TH8759)*. IEEE, 2004, pp. 301–306.

[9] B. Hannaford *et al.*, "Behavior trees as a representation for medical procedures," *arXiv preprint arXiv:1808.08954*, 2018.

[10] M. Colledanchise and P. Ögren, *Behavior Trees in Robotics and AI: An Introduction*. CRC Press, 2018.

[11] M. Colledanchise, A. Marzinotto, and P. Ögren, "Performance analysis of stochastic behavior trees," in *Robotics and Automation (ICRA), 2014 IEEE International Conference on*. IEEE, 2014, pp. 3265–3272.

[12] S. Preum *et al.*, "Towards a Cognitive Assistant System for Emergency Response," in *Poster session of the 9th ACM/IEEE International Conference on Cyber-Physical Systems (ICCPS)*. IEEE, 2018.

[13] A. R. Aronson, "Effective mapping of biomedical text to the UMLS Metathesaurus: the MetaMap program," in *Proceedings of the AMIA Symposium*. American Medical Informatics Association, 2001, p. 17.

[14] G. Cloud, "Confidence values of google speech-to-text api," https://cloud.google.com/speech-to-text/docs/basics#confidence-values.