A Cognitive Assistant System for Context Inference and Decision Making in Emergency Medical Services

A Thesis

Presented to

the faculty of the School of Engineering and Applied Science

University of Virginia

in partial fulfillment
of the requirements for the degree

Master of Science

by

Sile Shu

May 2019

APPROVAL SHEET

This Thesis
is submitted in partial fulfillment of the requirements
for the degree of
Master of Science

Author Signature: Sile Shu

This Thesis has been read and approved by the examining committee:

Advisor: Homa Alemzadeh

Committee Member: Scott T. Acton

Committee Member: John A. Stankovic

Committee Member:

Committee Member:

Committee Member:

Accepted for the School of Engineering and Applied Science:

Craig H. Benson, School of Engineering and Applied Science

May 2019

A Cognitive Assistant System for Context Inference and Decision Making in Emergency Medical Services

Sile Shu

Wuhan, China

B.S., Huazhong University of Science and Technology (2016)

A Dissertation Presented to the Graduate Faculty
of the University of Virginia in Candidacy for the Degree of
Master of Science

Department of Electrical and Computer Engineering

University of Virginia

May, 2019

Abstract

In emergency situations, the first responders need to collect, aggregate, filter and interpret information from different static and real-time sources and provide timely interventions and treatments to victims in a short period of time. Dealing with such a huge information load at the incident scene requires a significant amount of human cognitive effort. This thesis presents a cognitive assistant system for emergency medical services (EMS) that aims at improving situational awareness of the first responders by automated collection and analysis of data from the incident scene and providing suggestions on the most effective response actions 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 United States to evaluate the responsiveness and cognitive ability of the system and assess the safety of the suggestions provided to the responder. 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. We also simulate the streaming speech from emergency scenes to examine the effectiveness of the developed model in providing timely accurate suggestions. The simulation results show that the proposed cognitive assistant is able to achieve an average 70% F1-score in predicting correct interventions with only 45% of the input speech.

Keywords— Cognitive assistant systems, Emergency medical services, Behavior trees, Natural language processing, Machine learning

Acknowledgments

I would first like to thank my thesis advisor Professor Homa Alemzadeh of the Department of Electrical and Computer Engineering at the University of Virginia. The door to Prof. Alemzadeh's office was always open whenever I ran into a trouble spot or had a question about my research or writing. She consistently allowed this thesis to be my own work but steered me in the right direction whenever she thought I needed it.

I would additionally like to thank my collaborators and the experts who were involved in this research project: Professor Homa Alemzadeh, Professor Jack Stankovic, Professor Ronald D. Williams, Sarah Preum, Haydon M. Pitchford, Mustafa Hotaki, and M. Arif Imtiazur Rahman. Without their passionate participation and input, the progress and success we have so far in this research project could not have been achieved.

I would also like to acknowledge Professor Scott T. Acton of the Department of Electrical and Computer Engineering at the University of Virginia as the chair of my thesis defense committee. I am gratefully indebted to him for his very valuable comments on this thesis. I would also like to acknowledge that this work was supported by the award 60NANB17D162 from the U.S. Department of Commerce, National Institute of Standards and Technology (NIST).

Finally I would like to extend my deepest gratitude to my parents without whose love, support and understanding I could never have completed my master degree.

Publications

S. Preum, S. Shu, J. Ting, V. Lin, R. Williams, J. Stankovic, H. Alemzadeh, *"Towards a cognitive assistant system for emergency response."* 2018 ACM/IEEE 9th International Conference on Cyber-Physical Systems (ICCPS). IEEE, 2018.

S. Preum, S. Shu, M. Hotaki, R. Williams, J. Stankovic, H. Alemzadeh, *"CognitiveEMS: A Cognitive Assistant System for Emergency Medical Services."* 7th IEEE Workshop on Medical Cyber-Physical Systems, 2018.

S. Shu, S. Preum, H. Pitchford, R. Williams, J. Stankovic, H. Alemzadeh, *"Behavior Tree Cognitive Assistant System for Emergency Medical Services."* (under submission to IROS 2019)

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Background

Emergency medical responders and firefighters are supposed to initially assess and control the situation at the accident scene and provide basic life support to the victims before transporting them to the hospital. During the rescue operations, they need to quickly process a significant amount of information with different levels of importance and confidence, such as the status and medical history of the victims, the circumstances of the accident scenes and communication with the command center. However, detecting, gathering, filtering and processing these data at the incident scene requires lots of human cognitive effort. Such cognitive efforts can be saved, and consequently, the speed and accuracy of emergency response can be improved if the emergency medical responders and firefighters are aided by a cognitive assistant system for emergency medical services. These improvements, even tiny, can largely affect the rescue results.

Currently, emergency medical technicians make decisions and provide interventions based on their observations in the emergency scene and their training and knowledge of emergency medical service (EMS) protocols. The purpose of EMS protocols is to standardize medical procedures for all emergency medical services and thus achieve excellent, consistent pre-hospital care for patients. They also serve as a framework to help emergency medical technicians (EMTs) make decisions when operating in crises. In our work, we applied Old Dominion EMS Alliance (ODEMSA) protocols [1] which are used by four Virginia Planning Districts as a reference for analyzing the collected data and generating suggestions to the first responders.

Previous research [28, 10, 4] proposed the use of assistive technologies to improve first responders' situational awareness and decision making. For example, using wearable assistive agents for trauma documentation and management [10, 11], developing a portable communication framework for coordinating multiple agents (e.g., medical and communication devices, EMS vehicles) in distributed emergency response [14, 4, 24], simulating dynamic interactions between different human agents and potential digital agents in a hospital emergency environment using state-machine based models [33], real-time information extraction under noise for emergency response [28], and an information visualization agent that presents information gathered based on intent predicted using recent observations collected at emergency scene [23]. However, to the best of our knowledge, none of the existing research focuses on dynamically recommending situation-aware EMS protocol specific interventions for real-time emergency response decision support.

1.2 Challenges and Contributions

This thesis presents a design of a cognitive assistant system, CognitiveEMS, 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 cognitive assistant can be implemented as a wearable virtual assistant, interacting with a team of first responders before, during, and after arrival to incident scene or during EMS training exercises. The overall architecture of the system is described and shown in Fig. 1-1. In this thesis, we mainly focus on developing the perception and cognition capabilities for such a cognitive assistant system. There are several challenges in the 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. To perform a similar task, 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.

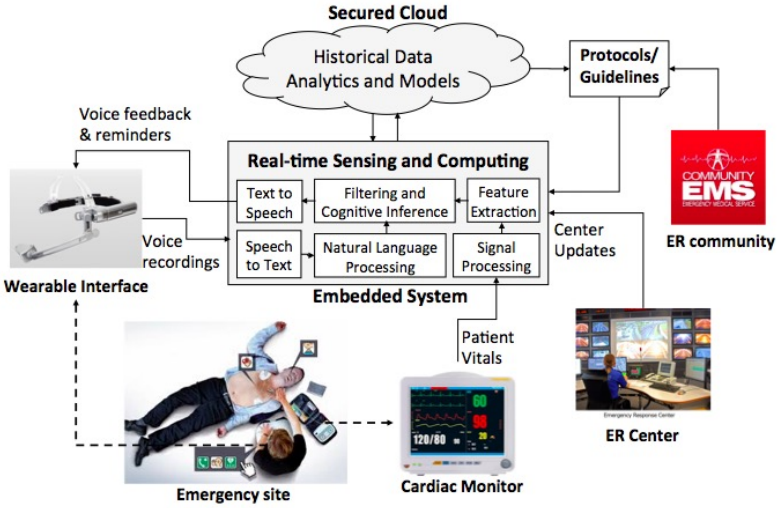


Figure 1-1: The system architecture of CognitiveEMS [28]

To address these challenges, this thesis adopts a Behavior Tree (BT) framework for real-time retrieval of the critical information from the scene and infer 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 of the most appropriate EMS protocols to be followed based on the situations inferred from the scene and the knowledge of the EMS protocol guidelines. Our evaluation using 3657 labeled EMS records indicate that this method achieves an average top-3 accuracy of 89%.

- We present two kinds of methods for prediction and suggestion of the most effective interventions by the cognitive assistant: a 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 records. Our results show that BT methods achieve comparable accuracy (F1 score) in predicting correct interventions but worse convergence rate in dealing with streaming input data compared to the ML models. Meanwhile, the BT method provides more transparency and evidence for the suggested intervention and does not rely on the availability of labeled data.

- We introduce a method for assigning confidence 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 risks of performing incorrect interventions by responders, the suggestions provided by the BT model on average lower risk compared to the best performing supervised ML models using the same setting.

- We simulate the streaming input text data from emergency scenes to evaluate the performance of both the knowledge-driven behavioral model and the supervised data-driven ML model. The simulation results show both models can generate reliable suggestions with only part of the narratives and the supervised data-driven ML model has better performance in intervention prediction.

Chapter 2

Related Work

2.1 Cognitive Assistant Systems

Cognitive assistant systems, or intelligent cognitive assistant (ICA) systems, are defined as a set of intelligent architectures and systems equipped with common sense knowledge and reasoning capabilities [16]. Cognitive assistant systems have been widely applied in smart health applications [17, 36, 11]. Ha's work [17] proposed the architecture and prototype implementation of an assistive system operating on Google glass to perform real-time scene interpretation by combining the first-person image capture and sensing capabilities of Glass with remote processing. In Zhang's work [36], an agent-based computer-aided scheduling (ACAD) system was proposed to organize multiple emergency departments to automatically dispatch appropriate rescue units to handle different types of emergencies. Croatti's paper [11] introduced and discussed traumatic documentation and management of personal medical assistant agent technology based on the Faith-Desire-Intention (BDI) architecture to assist the trauma team to accurately track and generate alerts in real time during trauma resuscitation.

In the context of emergency medical services, existing systems attempt to reduce the cognitive load of responders by providing new interfaces for on-scene reporting of electronic events [20, 15]. ImageTrend [20] provides a virtual data input interface for EMS responders. However, an important part of the scene report is still a narrative written in free-form text that describes the observations and actions made by the first responder at the scene of the emergency. In [15], the authors developed a mobile portal solution that helps collect data by dynamically customizing data fields. But these systems still rely on touch screens and messaging interfaces that are difficult to operate in an emergency scene. Hence this thesis aims at automatically extracting data from the responders' speech to reduce the cognitive burden of the first responders.

2.2 Clinical Decision Support Systems

Clinical decision support (CDS) is formally defined as the *process that provides clinicians, staff, patients, or other individuals with knowledge and person-specific information, intelligently filtered or presented at appropriate times, to enhance health and health care* [25]. The goal of CDS systems is to improve the health care decision outcomes by providing the relevant knowledge and analyses to the decision makers, i.e., clinicians, patients, and health care organizations. CDS systems have been used since as early as 1970, such as, Dombal's system for diagnosis of abdominal pain [12] using Bayes's theorem and Shortliffe's MYCIN rule based system for selection of antibiotic therapy [32] for patients with infections. Kuperman et al. demonstrated HELP, a CDS system integrated with hospital information system that generates automated alerts upon detection of abnormalities in the patient records [30]. In the Next section, we will provide a summary of the finite state machine and behavior tree modeling techniques and their applications in decision-making tasks.

2.3 Behavior Selection Models

In artificial intelligence, a behavior selection model or an action selection model is a model that selects an appropriate behavior or action for one or more intelligent agents. Typical behavior selection models include finite state machines (hierarchical finite state machines), decision trees, behavior trees, hierarchical task networks, hierarchical control systems, utility system, and dialogue tree.

2.3.1 Finite State Machine

Finite State Machines (FSM) is one of the most basic mathematical models for behavior selection tasks [6, 22, 37]. The FSM consists of a set of states, transactions, and events. FSM are widely used because it has a ubiquitous structure; it is intuitive and easy to understand and implement. In Chung's work [6], the authors proposed a novel heart rate estimation and verification algorithm based on FSM framework, which used the crest factor in the periodogram obtained after motion artifact removal and the estimated heart rate change in the continuous window as the estimation accuracy index. In Li's work [22], the authors tried to address the problem of changing the behavior sequences of the robots in manipulation tasks with human intervention by proposing a framework based on the FSM to model the robot action sequences in manipulation tasks. The framework showed how the robot slides its autonomy level by considering the results of the transition action and the input from the human operator. The framework could be easily extended to include higher levels of autonomy. In Zhang's work [37], the authors developed an automatic road driving controller based on an FSM framework to describe the controller through feedback control laws under each state and state transition condition. The authors also tested the controller in the traffic simulator and evaluated its performance.

However, when the complexity of system modeling and the number of states increase, the flaws of FSM can cause problems. To build a more reactive system using FSM, more transitions are needed and this consequently reduces the maintainability, scalability, and reusability of the system [9].

2.3.2 Behavior Tree

Behavior trees (BTs) is a behavior modeling framework emerged from game industry and it is extended to intelligent agents and robotics applications. BTs encode the control logic as a tree structure, with each sub-task modeled as leafs and combining the sub-tasks into behaviors through nodes in different positions in the tree. Bojic et.al. [5] developed a functional prototype of foundation for intelligent physical agents (FIPA)-Request Interaction Protocol by using their JBehaviorTrees framework which extends Java agent development framework (JADE) behaviors via BTs model. They showed both FSM implementation and the implementation via BTs can be used interchangeably, while the BTs model provides better modularity and code reusability. Hu et.al. [19] modeled and implemented the semi-autonomous surgical tasks via BTs framework. The modeled surgical procedure of brain tumor ablation was presented by RAVEN surgical robot and stereo visual feedback. It was demonstrated that the system could correctly detect the tumor and automatically generate ablation plans. Pereira et.al. [27] proposed a framework based on BTs that applied reinforcement learning nodes to add learning capabilities in current behavior-based intelligent agents. They showed how the learning framework works and validated that the choice of learning node would converge after iterations and the learning process would not affect the execution of the other nodes in the tree. Dometios et.al. [13] developed a novel real-time speech based perception system integrating all subsystems as modules and hierarchical decision architecture, which were organized as a behavior tree. A speech recognition system and a depth camera were used to achieve robust human-system communication and end-effector motion planning. With a set of spoken commands, the system could satisfy user's needs and preferences. Paxton et.al. [26] proposed a BT-based task editor integrating high-level information from known object and pose estimation with spatial reasoning and robot actions to create robust task planning. Evaluations and implementations of this system in various industrial robots and cases were performed in this paper. Hannaford et.al. [18] proposed the construction of BTs for example set of medical procedures and showed examples of functional medical algorithms implemented using BTs framework.

Stochastic behavior tree (STB) is also introduced in [9] to make the probabilistic descriptions of the actions and conditions in a behavior tree. STB describes the interaction of a BT node with its children in terms of a Discrete Time Markov Chain (DTMC). This enables the BT to propagate performance estimates from one level in the tree to the next. Applying the scheme in a recursive fashion then makes it possible to compute the properties of an arbitrarily complex BT composition of actions. [8]

Based on the review of the related work and the application scenarios mentioned above, we can summarize the unique features of the BTs as follows:

- Each sub-tree of a BT can be seen as a module. Thus, BTs are modular in all scales ranging from the whole tree to all the leaves of the tree [9]. This feature results in that the projects with BTs framework are easier to read, design, and maintain.

- BTs enable reusable code. That makes it easier to build a large, complex, long-term project using BTs framework.

- BTs are highly responsive because of their continuously-ticking mechanism, which makes it possible for specific actions to be executed or aborted according to the leaf nodes' return status at every ticking. These returned statuses are tightly connected with the environment and ensure that the BT models can react to the changes in the situation quickly and efficiently [9].

- A BT needs to traverse all the conditions in the tree to implement the closed-loop task execution [9]. In some applications, this all-condition-checking is too expensive.

- Although learning the ability of BTs is still an open problem, the work of Pereira et al. provides the formalization of adaptive and constrained behavior-based agents using BTs and reinforcement learning [27]. More recent and ongoing work in this area is expected to improve this framework further and allow the agents to work in more complex environments.

- The tools for developing BTs are much less mature than other modeling frameworks such as logic programming and FSMs.

Chapter 3

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 (Fig. 4-1b) indicates when the first responders observe that a patient is suffering from hypoxemia (i.e., patient's SpO_2 level is lower than the normal range), they need to provide supplemental oxygen to the patient. Emergency medical technicians need to gather, record and analyze these information and then make interventions based on some specific emergency medical service protocols. To assist the responders to analysis situations and take interventions, our system records speech data from first responders, which are typically oral report to describe the situations in the accident scene and make inference about what intervention should be taken based on emergency medical service protocols.

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 separate it into three consecutive sub-tasks:

1. Extract a subset of C to represent the situation for an arbitrary time t from the speech data S_t;

2. 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;

3. Find the intervention subset I_j based on the knowledge of the selected protocol subset P_i.

In the following sections we present the methodologies to perform these three sub-tasks.

Chapter 4

Approach

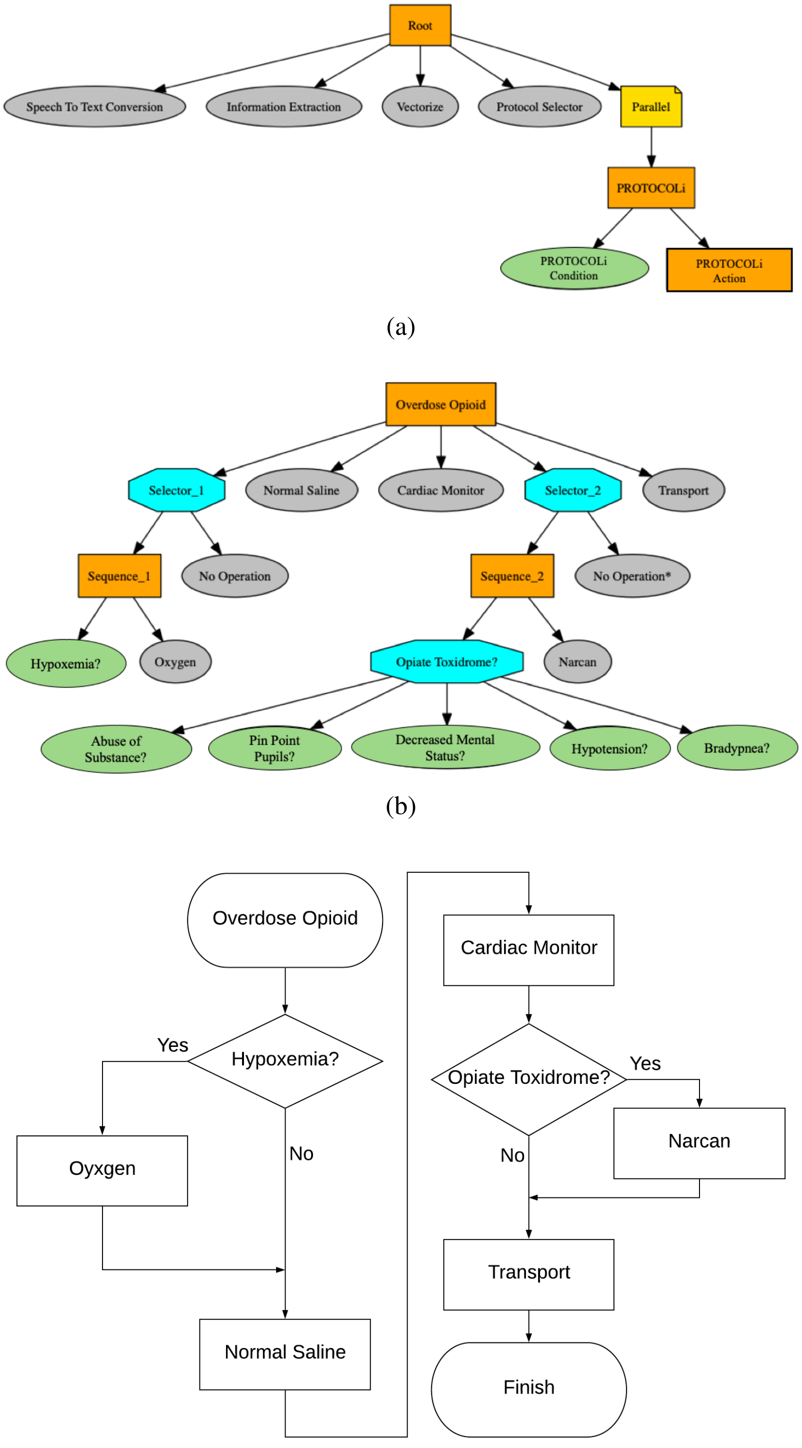
4.1 BT Framework

We propose a BT framework for implementing the natural language processing and cognitive inference by the cognitive assistant as illustrated in Fig. 4-1a. Details about the components in this pipeline are discussed in the Section 4.1.1.

4.1.1 Behavior Trees

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 [18]. 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 [9]. 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 [8]. The control of the execution nodes in BT is achieved by using the composite nodes, whose return status depends on its children. Three typical composite nodes are used in our work: Sequence, Selector and Parallel. The notation of these three kinds of nodes is shown in Fig. 4-2

We choose BTs as an executable behavioral modeling framework for design of our cognitive

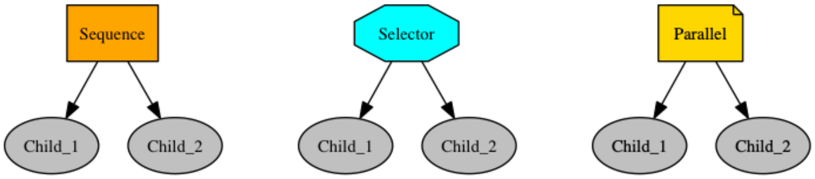


(a)

(b)

(c)

Figure 4-1: (a) Overall BT framework, (b) Opioid Overdose Protocol Action Subtree, (c) Flowchart of Opioid Overdose Protocol



(a) Sequence Node (b) Selector Node (c) Parallel Node

Figure 4-2: Three Typical Nodes in BTs

system due to its modularity, high responsiveness, and the ability to learn and adapt using reinforcement learning. As shown in Fig. 4-1a, 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 results from each component are stored and passed to the other components to be processed via blackboard, a typical component in behavior tree models to store and transport data between the sub-trees and nodes. As a fairly standard structure in most behavior tree implementations, it has the following features:

- No sharing between behavior tree instances;
- No locking for reading/writing;
- Global scope, i.e. any behavior/node in the behavior tree can access any variable stored in the blackboard;
- No external communications (e.g. to a database)

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. Fig. 4-1b shows an example of Overdose Opioid protocol action sub-tree in the BT. The details about the mechanism of these sub-trees are presented in the Section 4.1.6. Here we show an example of the action node of the Overdose Opioid protocol. The logic for this protocols is shown as a flowchart in Fig. 4-1c. We transfer this logic into a behavior tree model as shown in Fig. 4-1b. Note that the condition "Opiate Toxidrome? " in this case is a composite condition containing several sub-conditions to be checked. We will further discuss how we model the composite conditions into the BT model in Section 4.1.6.

The details of the BT nodes in Fig. 4-1a implementing different components of the cognitive assistant pipeline are provided next.

4.1.2 Speech to Text Conversion

The purpose of this component is transferring the input speech data S_t from the first responders to text T. In our previous works [29], We applied data collected from real emergency scenarios to quantitatively compare the performance of the four off-the-shelf, state-of-the-art speech-to-text conversion tools under noise in terms of word error rate and computation time. These tools included the Google Cloud API, the Microsoft Voice API, the PocketSphinx and the IBM BlueMix API. The results have shown that the Google Speech API provides the best results among other state-of-the-art speech recognition (shown in Table 4.1). Thus, we apply the Google Speech API to perform speech to text conversion on the audio streams collected from the accident scene.

Scenario	Metrics	PocketSphinx	Google	Microsoft	IBM
Noise-free	WER	0.80	**0.19**	0.24	0.45
	Runtime	2.48	**2.72**	3.42	5.34
Noisy	WER	1.05	**0.39**	0.62	0.89
	Runtime	3.41	**3.00**	3.38	9.84

Table 4.1: Comparing different speech-to-text conversion APIs in terms of word error rate (WER) and runtime [29]

As shown in Fig. 4-1a, at every tick of the behavior tree, first the sub-task *Speech to Text Conversion* is executed to get the generated text from the incoming audio stream. Then the collected text is passed to the following components via 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.

4.1.3 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 emergency medical services 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 steps (Shown in Fig. 4-3).

UMLS Concept Extraction

At this step, we apply MetaMap, a widely available tool for mapping biomedical text into the concepts in the Unified Medical Language System (UMLS) Metathesaurus [2], to 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).

Concept Filtering

Metamap is a highly configurable tool to map biomedical text to the concepts in the UMLS Metathesaurus, and it returns a list of UMLS concepts when a piece of the input text is given. However, not all of the output concepts yielded by MetaMap are required by the EMS protocols. Hence, we compiled a set of EMS protocol specific concepts that are required by the EMS protocols or are frequently used by the medical responders. The way we applied to filter concept is using concept-unique-identifiers (CUI) of the target concepts. 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 were 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.

Value Retrieval

We find additional information related to the concepts identified in the text, e.g., for the extracted concept *pulse rate*, we are interested to also extract the value of the pulse rate. 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).

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

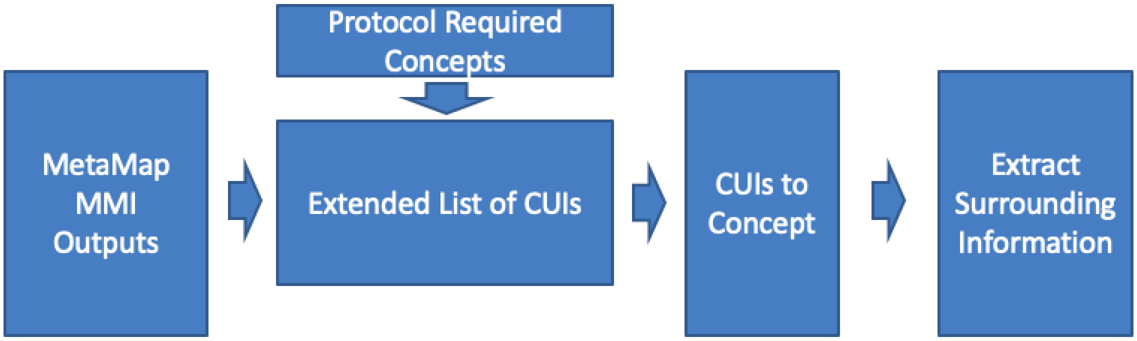


Figure 4-3: Information Extraction Pipeline

and concept annotation components. In our confidence calculation and assignment, we consider the confidence score for the recognized words by the Google Speech API [7] and the similarity score provided by the MetaMap API indicating the level of confidence in mapping between the input text and UMLS concepts [3]. 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:

$$Conf(C_i) = Conf_G(C_i) \cdot mmScore(C_i) \tag{4.1}$$

An example piece of text along with the corresponding dictionary elements extracted by the *Information Extraction* phase are shown in Fig 4-6a. These outputs are formatted in a unified structure and then fed to the next stage for protocol selection and execution and intervention suggestion.

4.1.4 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 and ranked by real first responders participating in our project. Formally, these two vectors can be represented as follows:

$$\vec{V_T} = \{Conf(C_i) | \forall C_i \in C\} \tag{4.2}$$

$$\begin{aligned} \vec{V_{P_i}} = & \{Weight(C_j) | \forall C_j \in C \\ & \wedge Weight(C_j) = Softmax(Pri_{i_j}) \\ & \wedge Pri_{i_j} \in \{0, 1, 2, 3\}\} \end{aligned} \tag{4.3}$$

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.

4.1.5 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. Cosine similarity, as a commonly used metric in information retrieval and question answering systems is used. We calculate the similarity and relevance between the text ($\vec{V_T}$) and protocol ($\vec{V_{P_i}}$) vectors, as follows:

$$S_i = \frac{\vec{V_T} \cdot \vec{V_{P_i}}}{\|V_T\| \cdot \|V_{P_i}\|} \tag{4.4}$$

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 protocol to be executed by the cognitive assistant system. The whole protocol ranking procedure is shown in Fig. 4-4. 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:

$$Conf(P_i) = \begin{cases} \frac{S_i}{\sum S_j}, & \forall P_j \in Candidate \\ 0, & \text{otherwise} \end{cases} \tag{4.5}$$

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.

4.1.6 Protocol Execution - Intervention Suggestion

Typically, each EMS protocol describes some specific rules to perform interventions in an emergency scene, and the conditions in these rules are signs, symptoms or medical history of the patient, which we extracted and represented as concepts in the previous components (see example in Fig. 4-1b). Therefore, 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 makes it possible for all the selected candidate protocols to be executed concurrently at the same time and suggest most relevant interventions with highest confidence score to the responder. Due

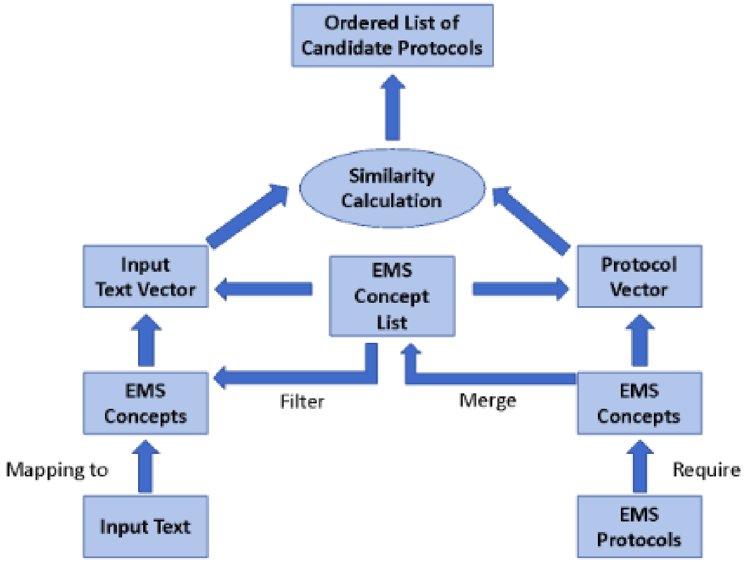


Figure 4-4: Pipeline to Generate Text and Protocol Vectors

to the modularity of the BTs structure, we can easily replace or extend the set of EMS protocols by merely replacing or adding to the sub-trees under the parent node.

Each EMS protocol connected to the parallel node is modeled as a sequential composite, whose children including a protocol condition node and a protocol action node (shown in Fig. 4-1a). That makes the protocol condition node executes prior to the protocol action node to check if the protocol selector selects the protocol. If it is selected, the protocol condition node will return success and allow the parent node to tick the protocol action node. The protocol action leaf is a sub-tree that keeps track of the execution of the actions in the EMS protocols. An example among these action sub-tasks is shown in Fig. 4-1b. Note that the children nodes in green (e.g., the leaf "Hypoxemia?") are condition nodes. If the condition is satisfied, it will return success to its parent node. Otherwise, it returns failure. At each tick of the BTs, the protocol execution component tracks the action of the medical responders based on the input text and suggests the next step.

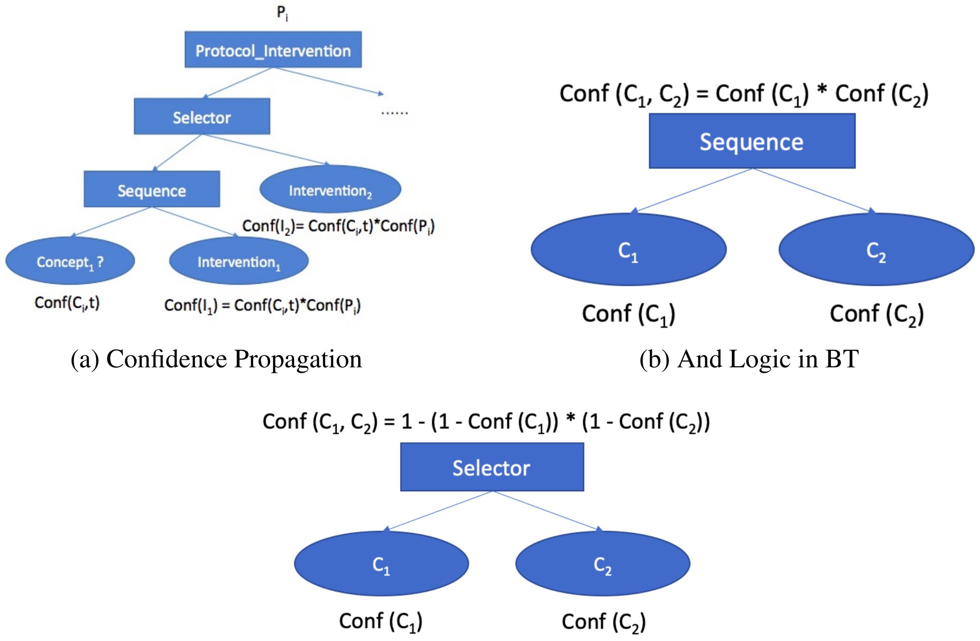
Although stochastic BT is mentioned in the Section 2.3, it is built based on the execution times and pre-defined success probabilities of each action in the behavior tree, which we cannot define or measure in our system. Thus, we developed a mechanism to calculate and assign confidence scores to the interventions based on the confidence score we assigned to each extracted concepts in Section 4.1.3 and the confidence scores for the selected EMS protocols calculated by Equation 4.5. The mechanism which is used to calculate the confidence scores for the interventions in the EMS protocols is developed based on two assumptions:

- The confidences of the concepts extracted by the information extraction component are independent of each other;
- The confidences of the selected protocols and the confidences of the concepts extracted by the information extraction components are independent.

Based on these two assumptions, the confidence of the interventions can be calculated as follows:

$$Conf(I_i) = \sum Conf(C_1, C_2, ..., C_k) \cdot Conf(P_i), \forall P_i \ni I_i \tag{4.6}$$

Where $Conf(C_1, C_2, ..., C_k)$ refers to the combined confidence score from the confidence scores of all of the conditions $C_1, C_2, ..., C_k$ of the intervention I_i described in EMS protocol P_i. The combination of the confidence scores from different conditions are calculated based on their logic relationship. For an example, if the condition for intervention I is $C_1 \wedge C_2$, then the combined confidence score $Conf(C_1, C_2)$ will be calculated as $Conf(C_1) \cdot Conf(C_2)$; If the condition is $C_1 \vee C_2$, then the combined confidence score $Conf(C_1, C_2)$ will be calculated as $1 - (1 - Conf(C_1)) \cdot (1 - Conf(C_2))$. These basic logic combinations of the conditions can be implemented as composite condition nodes in BT framework (shown in Fig. 4-5b and Fig. 4-5c).



(a) Confidence Propagation

(b) And Logic in BT

(c) Or Logic in BT

Figure 4-5: Confidence Calculation and Propagation in a Protocol Action Sub-tree and Composite Condition Nodes

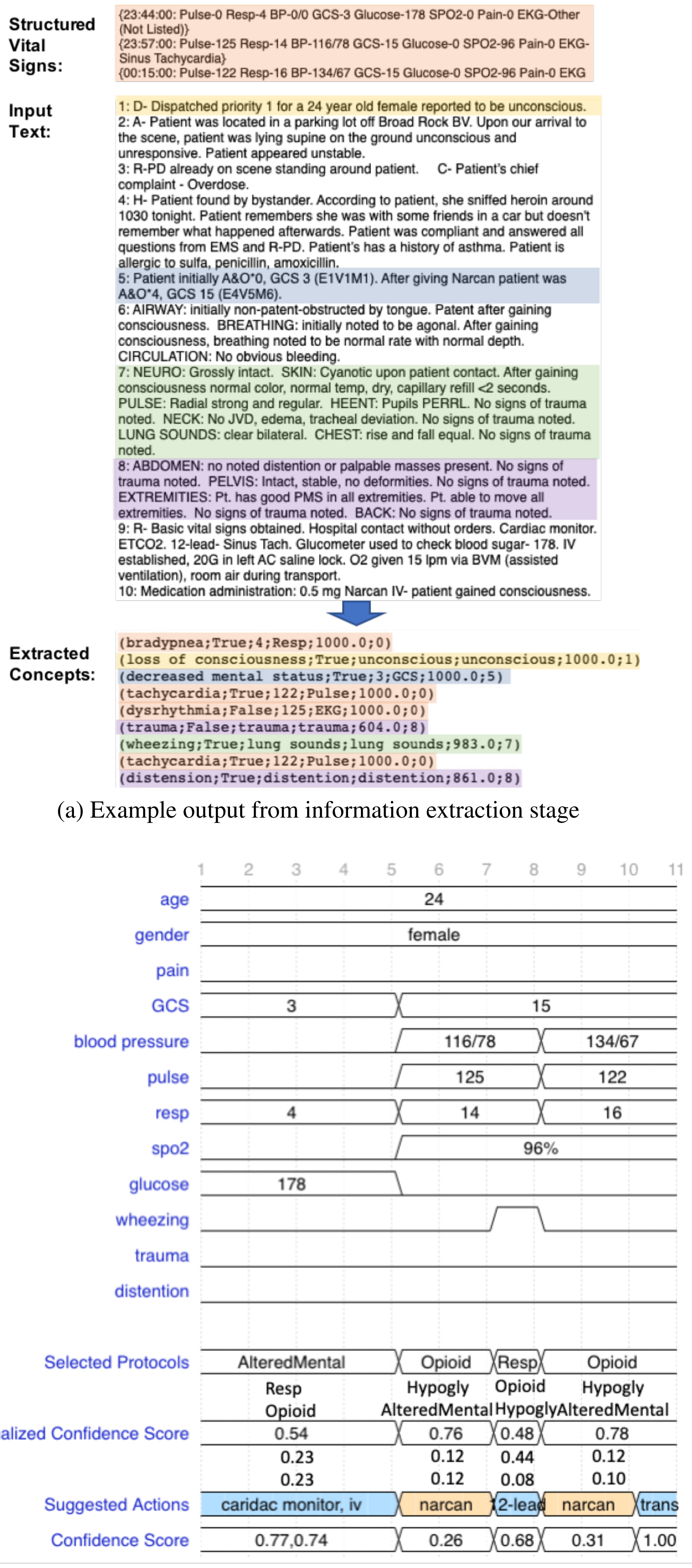
There is an obvious risk to execute protocols connected to the parallel node concurrently in this system. In most cases, extra protocols will be executed, and consequently, inappropriate or even safety-critical feedback might be suggested to the responder. To avoid such risks, we have extended the BT 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. Fig. 4-5a 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 inappropriate interventions tend to have a lower confidence scores. Because:

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

- 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.

- 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.



Structured Vital Signs:

```
{23:44:00: Pulse-0 Resp-4 BP-0/0 GCS-3 Glucose-178 SPO2-0 Pain-0 EKG-Other
(Not Listed)}
{23:57:00: Pulse-125 Resp-14 BP-116/78 GCS-15 Glucose-0 SPO2-96 Pain-0 EKG-
Sinus Tachycardia}
{00:15:00: Pulse-122 Resp-16 BP-134/67 GCS-15 Glucose-0 SPO2-96 Pain-0 EKG
```

Input Text:

1: D- Dispatched priority 1 for a 24 year old female reported to be unconscious.

2: A- Patient was located in a parking lot off Broad Rock BV. Upon our arrival to the scene, patient was lying supine on the ground unconscious and unresponsive. Patient appeared unstable.

3: R-PD already on scene standing around patient.   C- Patient's chief complaint - Overdose.

4: H- Patient found by bystander. According to patient, she sniffed heroin around 1030 tonight. Patient remembers she was with some friends in a car but doesn't remember what happened afterwards. Patient was compliant and answered all questions from EMS and R-PD. Patient's has a history of asthma. Patient is allergic to sulfa, penicillin, amoxicillin.

5: Patient initially A&O*0, GCS 3 (E1V1M1). After giving Narcan patient was A&O*4, GCS 15 (E4V5M6).

6: AIRWAY: initially non-patent-obstructed by tongue. Patent after gaining consciousness. BREATHING: initially noted to be agonal. After gaining consciousness, breathing noted to be normal rate with normal depth. CIRCULATION: No obvious bleeding.

7: NEURO: Grossly intact. SKIN: Cyanotic upon patient contact. After gaining consciousness normal color, normal temp, dry, capillary refill <2 seconds. PULSE: Radial strong and regular. HEENT: Pupils PERRL. No signs of trauma noted. NECK: No JVD, edema, tracheal deviation. No signs of trauma noted. LUNG SOUNDS: clear bilateral. CHEST: rise and fall equal. No signs of trauma noted.

8: ABDOMEN: no noted distention or palpable masses present. No signs of trauma noted. PELVIS: Intact, stable, no deformities. No signs of trauma noted. EXTREMITIES: Pt. has good PMS in all extremities. Pt. able to move all extremities. No signs of trauma noted. BACK: No signs of trauma noted.

9: R- Basic vital signs obtained. Hospital contact without orders. Cardiac monitor. ETCO2. 12-lead- Sinus Tach. Glucometer used to check blood sugar- 178. IV established, 20G in left AC saline lock. O2 given 15 lpm via BVM (assisted ventilation), room air during transport.

10: Medication administration: 0.5 mg Narcan IV- patient gained consciousness.

Extracted Concepts:

```
(bradypnea;True;4;Resp;1000.0;0)
(loss of consciousness;True;unconscious;unconscious;1000.0;1)
(decreased mental status;True;3;GCS;1000.0;5)
(tachycardia;True;122;Pulse;1000.0;0)
(dysrhythmia;False;125;EKG;1000.0;0)
(trauma;False;trauma;trauma;604.0;8)
(wheezing;True;lung sounds;lung sounds;983.0;7)
(tachycardia;True;122;Pulse;1000.0;0)
(distension;True;distention;distention;861.0;8)
```

(a) Example output from information extraction stage

(b) Example output from protocol execution & intervention prediction

Figure 4-6: An example of the results from the proposed BT Cognitive Assistant System

4.2 Supervised Machine Learning Classifiers

Other than the proposed BT framework, we also tried to resolve the intervention suggestion based on an end-to-end method: directly predict the intervention based on the input text. Given this idea, we modeled each historical EMS cases and queries using vector space models and treated the recorded interventions as the ground-truth of the corresponding EMS case. Thus, we can define this end-to-end method formally as follows:

Given an EMS narrative/document D_j represented as a vector:

$$D_j = (w_{1j}, w_{2j}, ..., w_{tj})$$

Each dimension w_{ij} corresponds to a separate term. These terms could be single words, phrases or keywords depending on the way we build the vector space model. If the term occurs in the document, its value in the vector is non-zero.

Meanwhile, the recorded/predicted interventions I_j can be encoded as a vector as well:

$$I_j = (i_{1j}, i_{2j}, ..., i_{nj})$$

Each dimension i_{ij} corresponds to a specific intervention in the intervention set I. If the intervention is recorded/ predicted in the EMS report, its value in the vector is 1. Otherwise, it is 0.

Therefore, we can train a multi-class and multi-output classifier by a set of historical EMS reports along with recorded interventions to perform this end-to-end intervention prediction task.

4.2.1 Vector Space Model

Before building the vector space model representing the text documents, we need to pre-process the input text. Here we applied the basic text processing techniques to perform the preprocessing, including tokenization, normalization, stemming, and stopwords removal, which extracted essential features from texts and reduced the dimension of the text vector. The overall pipeline is shown in Fig. 4-7

We first tried two kinds of n-gram settings in the feature extraction step: unigrams and technical n-grams. Uni-grams are typically single terms extracted from text after pre-processing. The technical n-grams here refer to a controlled vocabulary containing a set of contiguous sequence of n words that follow a specific Parts-of-Speech (POS) pattern. For example, we extracted n-grams

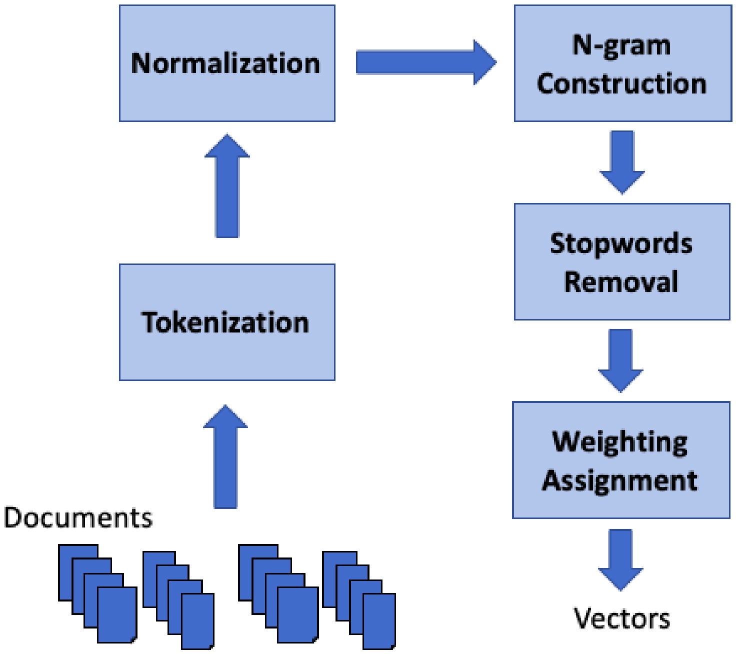


Figure 4-7: Text Processing Pipeline in Vector Space Model

(1 to 4 grams) representing phrases that are constructed only of nouns and adjectives. The weights w_{ij} assigned to the vectors are calculated based on the TF-IDF [35] of each unigram/ technical n-gram in the training dataset. The TF-IDF weighting for each term in the vector can be calculated as follows:

$$tf - idf(t, d) = tf(t, d) \times idf(t) \tag{4.7}$$

$$idf(t) = \log \frac{1+n}{1+df(t)} + 1 \tag{4.8}$$

where tf means term-frequency ($tf(t, d)$ means the frequency of term t in document d) while $tf-idf$ means term-frequency times inverse document-frequency. In the idf calcuation, n is the total number of documents in the document set, and $df(t)$ is the number of documents in the document set that contain term t.

In addition to these two typical methods for the vector space representation of the documents, we also applied the concept vector mentioned in Section 4.1.4 that uses EMS protocol specific concepts and corresponding confidence scores to encode the given texts.

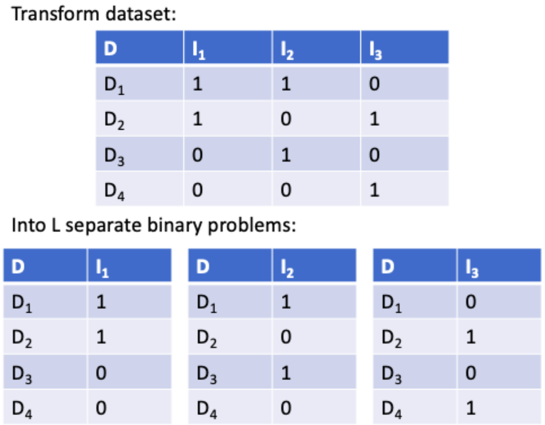
4.2.2 Multi-label Classification

Once we have the input texts represented as text vectors, classifiers can be trained with these text vectors and their corresponding interventions. Because there are multiple interventions recorded in most EMS reports, we treat this intervention prediction problem as a multi-label classification task. We picked several different machine learning algorithms and strategies to perform this multi-label classification task.

Binary Relevance Method

One straight forward strategy to resolve a multi-label classification problem is to transform this problem into multiple binary classification problems (shown in Fig. 4-8), which is also referred to as the binary relevance method [31]. This approach amounts to independently training one binary classifier for each label. Given a test sample, the combined model then predicts all labels for this sample for which the respective classifiers predict a positive result.

One-vs-the-rest (OvR) strategy is a typical binary relevance method [21]. This strategy consists of training one classifier per class. For each classifier, the class is fitted against all the other classes. One advantage of this approach is its interpretability since each class is represented by one and only one classifier. Thus, it is possible to gain knowledge about the class by inspecting its corresponding classifier.



Transform dataset:

D	l_1	l_2	l_3
D_1	1	1	0
D_2	1	0	1
D_3	0	1	0
D_4	0	0	1

Into L separate binary problems:

D	l_1
D_1	1
D_2	1
D_3	0
D_4	0

D	l_2
D_1	1
D_2	0
D_3	1
D_4	0

D	l_3
D_1	0
D_2	1
D_3	0
D_4	1

Figure 4-8: Binary Relevance Transformation

Once we transfer the multi-label classification problem into multiple separate binary classification problems (one for each label), we can train the model with any off-the-shelf binary classifier. We chose the support vector machine as the binary classifier in our implementations.

Adapted Classification Algorithms

There are also classification algorithms/models that have been adapted to the multi-label task, without requiring problem transformations. There are several advantages to use a single model: The single model can consider the dependencies, and it is more scalable than the model combining multiple binary classifiers. However, the performance of the single classifier largely depends on the problem domain. We used decision trees and random forests to perform the multi-label classification task.

Chapter 5

Evaluation

Three sets of experiments were performed to evaluate the proposed BT framework and trained supervised machine learning models. 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. Last, we evaluate the performance of the models under the simulated streaming data by splitting the input text into chunks.

5.1 Experimental Setup

For these three 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. Note that in these experiments, we directly used the narratives and vitals transcribed by the responders and did not perform the speech to text conversion, so the $Conf_G(C_i)$ score in Equation 4.1 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 [28].

For a subset of 3657 records, the actual protocol used by the responder was labeled by one of the advanced life support trained responders in our project and was used as grand truth for assessing the accuracy of automated protocol selection component. This was done by developing a set of rules unique to each of the pre-selected protocols in order to filter out cases that were either ambiguous or fell into another treatment 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 as overdose opioid protocol. 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.

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 evaluate the performance of the intervention suggestion by our proposed knowledge-driven BT method. Specifically, we applied the following three supervised data driven ML models to perform the intervention prediction: Support Vector Machines (SVM), Random Forest (RF), and Decision Trees (DT). We applied the intervention column in the RAA reports as the ground truth and the narrative column as inputs to train the these models. Each narrative was represented in three vector spaces using technical n-grams ,uni-grams and protocol concepts of the narratives. Our test dataset for intervention prediction included 1000 RAA EMS reports for both BT and ML models. 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 algorithms. To achieve a fair comparison between the confidence-aware, knowledge driven unsupervised method based on BTs 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 in which intervention classes with higher risk scores were assigned lower weights to direct the supervised ML models towards selecting less safety-critical interventions with lower risk factors; (ii) Applying the similar confidence score filter implemented in the BT model to filter the interventions with low confidence.

Furthermore, to simulate the speech data streaming in emergency scenes, we conducted another set of experiments in which we gradually fed the the input text into the proposed BT model and the supervised ML models which had the best performance in our previous evaluations, to evaluate their performance for the streaming data. We did this by splitting the input narratives into chunks of text. The performance of the models at each chunk was evaluated by comparing their output with the ground truth and their final outputs (the outputs from the models when provided with the whole narrative). Note that we need to provide the BT model with text chunks since the model itself stores the extracted EMS concepts, but the ML models need to be fed with accumulated text chunks to perform the intervention prediction. Here we split each EMS reports into 11 chunks in the 1000 test reports and compares the performances in terms of micro-averaging precision, recall, F1 score, and convergence rate. Note that here we chose the number of text chunks as 11 based on the average length of the EMS reports in our dataset and an empirical minimum length of the finalized text generated by the Google Cloud Speech API.

The convergence rate here is defined as the rate at which a model converges to its final result. To be more specific, if we define the temporary intervention prediction result for the kth text chunk as $Pred(k)$ the convergence rate can be calculated as:

$$ConvergenceRate(k) = 1 - \frac{\sum_{i=1}^{I} |Pred(k)_i - Pred(n)_i|}{I} \tag{5.1}$$

where n is the total number of the text chunks and I is the length of the intervention vector. Note that the element in this vector would be 1 if the corresponding intervention is predicted in the results and otherwise it would be 0. We introduce this evaluation metric because of our concern about the performance of intervention prediction, when the model is given only partial information or evidence from the emergency scene, and to evaluate the reliability of intermediate interventions suggestions provided to the first responders.

5.2 Experimental Results

5.2.1 Protocol Selection

To evaluate the automated protocol selection procedure, we compared the ranked list of selected protocols by the cognitive assistant with the grand-truth protocols labeled by the participating first responders 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 the 3657 test cases from RAA dataset showed an average top-3 accuracy of 89.0%. By reviewing the cases for which the cognitive assistant predictions did not match the labels provided by the responders, we identified the following reasons for sub-optimal performance of our protocol selection method:

- 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 a mapping errors.); 4) MetaMap not producing the correct negation detection results, leading to missing the presence of some concepts from input text.

- Protocol vectors were manually developed based on the descriptions of signs and symptoms in the set of protocols and 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 responders in our project. Some of the manually assigned weights in the protocol vectors caused errors.

- Missing critical information (e.g., incomplete vital signs) in some of the EMS cases also affected the correctness of the text vector.

5.2.2 Intervention Suggestion

To evaluate the performance of the intervention suggestion, 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 also define the interventions in the ground truth that failed to be predicted by our system as false negatives (FN). By calculating the TPs, FNs and FPs for each RAA case, we applied both weighted and micro average precision, recall and F1-score to evaluate the performance of the intervention predictions. Compared to the macro-averaging method which computes the metrics independently for each class and then takes the average (treating all classes equally), the micro-averaging method aggregates the contributions of all classes to compute the average metric [34]. The weighted averaging method calculates 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 imbalance among classes into account.

$$P_{micro} = \frac{\sum TP_i}{\sum TP_i + \sum FP_i} \tag{5.2}$$

$$R_{micro} = \frac{\sum TP_i}{\sum TP_i + \sum FN_i} \tag{5.3}$$

$$F_{micro} = \frac{2 \cdot P_{micro} \cdot R_{micro}}{P_{micro} + R_{micro}} \tag{5.4}$$

$$P_{weighted} = \frac{\sum P_{C_i} \cdot C_i}{\sum C_i} \tag{5.5}$$

$$R_{weighted} = \frac{\sum R_{C_i} \cdot C_i}{\sum C_i} \tag{5.6}$$

$$F_{weighted} = \frac{2 \cdot P_{weighted} \cdot R_{weighted}}{P_{weighted} + R_{weighted}} \tag{5.7}$$

where C_i indicates the number of instances in intervention class i; P_{C_i} and R_{C_i} indicates the precision and recall of class i.

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,

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	ngram	0.83	0.89	0.77	0.77	0.78	0.83	0.81	0.32
	ngram, filtering	0.83	0.89	0.77	0.77	0.78	0.83	0.81	0.31
	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**
	feature vector	0.62	0.81	0.63	0.64	0.61	**0.72**	0.73	**0.44**
Random Forest (RF)	ngram	0.77	0.88	0.67	0.66	0.68	0.76	0.75	0.46
	ngram, weighted	0.77	0.87	0.61	0.62	0.63	0.72	0.72	0.46
	ngram, filtering	0.80	0.89	0.66	0.67	0.68	0.76	0.75	0.43
	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
	feature vector	0.65	0.76	0.60	0.60	**0.66**	0.67	0.72	**0.60**
Decision Trees (DT)	ngram	0.73	0.77	0.73	0.75	0.72	0.76	0.74	0.45
	ngram, weighted	0.71	0.75	0.70	0.73	0.70	0.74	0.73	0.54
	ngram, filtering	0.73	0.78	0.72	0.75	0.72	0.76	0.74	0.46
	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
	feature vector	0.62	0.64	0.60	0.61	0.60	0.63	0.67	**0.82**

Table 5.1: Intervention suggestion provided by the unsupervised BT model vs. the supervised linear support vector machine (SVM), random-forest (RF), and decision tree (DT) models. Weighted supervised models represent the models trained with inverse intervention risk scores while filtered models represent models with confidence score filtering that remove interventions with low confidence from the list of suggestions.

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 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 grand truth interventions for each case. The average normalized risk to evaluate the performance of model over n test cases was calculated as follows:

$$Avg.\ Normalized\ Risk = \frac{1}{n} \frac{\sum Conf(I_i) \cdot Risk(I_i)}{|I|} \tag{5.8}$$

Note that since the BT model cannot calculate the confidence score for the interventions in the protocols that are not selected, we set the $Conf(I_i)$ of the FN interventions predicted by the BT model as 1.

The evaluation results using the metrics mentioned above are shown in Table 5.1. Our results show that the knowledge-driven unsupervised BT method performs worse than supervised ML methods when evaluated using the traditional evaluation metrics (precision, recall, and F1) used for a multi-class classification task. However, there is a limitation in the dataset that we use in our evaluation; some of the interventions appeared in the narratives and were encoded into uni-gram and technical n-gram vectors, which might lead to bias in the ML models. This limitation can be partly reflected by the fact that the top 30 important features among all the features in the decision tree models were found to be interventions or part of interventions. This flaw of the training dataset is also further revealed and discussed in Section 5.2.3. Thus, we mainly compare the performances between the BT model and the ML models trained with concept vectors, which are described in Section 4.1.4.

By comparing the models using the same feature vectors, we can see that BT model has comparable performance in terms of F1 score and much better performance in terms of average risk factor (shown in Fig. 5-1). This means that we can effectively avoid safety-critical suggestions using the confidence propagation and filtering mechanisms of the BT model.

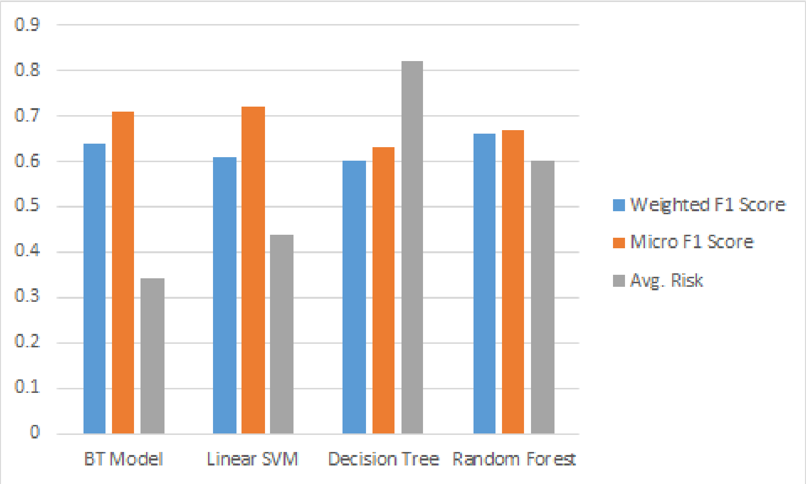
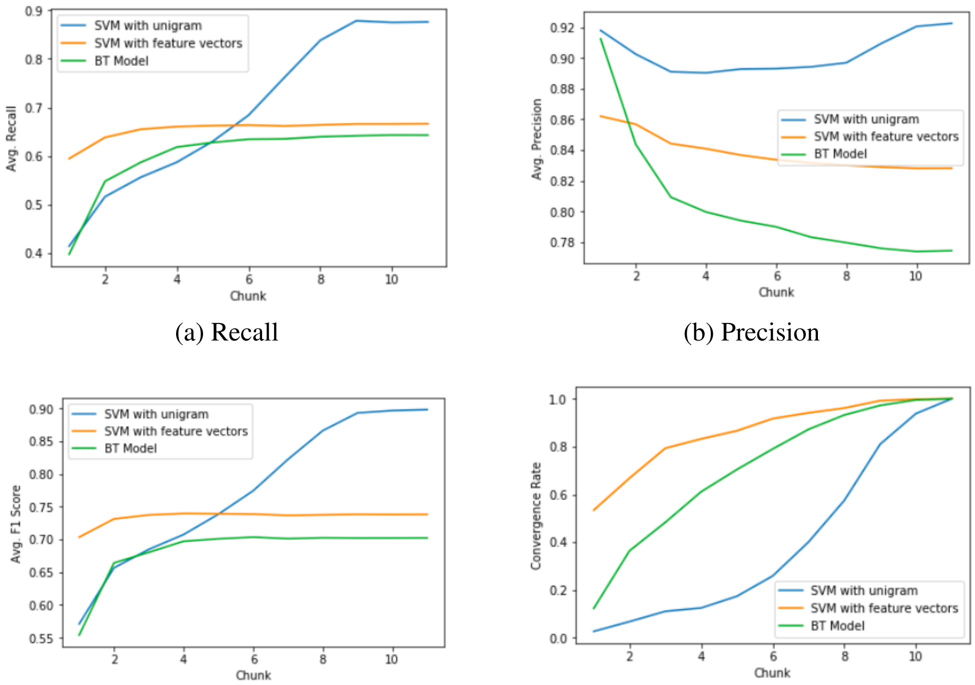


Figure 5-1: Performance of Models Based On Concept Vector

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 as well. Based on the formula of the risk factor in our evaluation, the reason for these results might be the extra FN and FP predictions brought by weight assignments for the ML models. On the other hand, the ML models with filters, which get rid of predictions with confidence scores lower than a threshold, slightly reduced the risk factor compared to the original models.

5.2.3 Performance with Streaming Data

In this section, we picked linear SVM trained by uni-gram vectors and linear SVM trained by feature vectors, which are the models with the best performances trained by text vectors and concept vectors, to make the comparison with BT model in terms of their performance with streaming input text data.



(a) Recall

(b) Precision

(c) F1 Score

(d) Convergence Rate

Figure 5-2: Model Performance with Streaming Input Text

The evaluation results are shown in Fig.5-2. By comparing the models based on the feature vectors (BT model and linear SVM) and the model based on the uni-gram vectors, we can conclude that the models developed by feature vectors converge to its final result faster than the model developed by the uni-gram vector. That means, the models trained with feature vectors need less information in the latter part of the reports and have better performance with streaming data than the models trained with text vectors. Especially, as mentioned in Section 5.2.2, the limitation brought by the dataset can be further revealed: by checking the EMS reports, we found that most of the signs and symptoms of the patient were described in the former part of each report while most of the interventions taken by the first responders were recorded in the latter part, which is consistent with the fact that the ML models tend to quickly converge after the fifth chunk of input data and the models based on feature vectors almost converged at the first several text chunks.

On the other hand, by comparing the performance of the BT model and the linear SVM model trained by feature vectors, we can find that the performance of the linear SVM model is better than BT model. That means ML models perform better than the BT model in the streaming text data situation.

Another consideration we have here is the incorrect intermediate intervention suggestion generated by the models when they receive the streaming data. Even the model with the best convergence performance only has a 53.4% and 66.8% convergence ratio after the first two chunks of texts, which means the model cannot give correct intervention suggestions with small amount of information.

Chapter 6

Discussion

6.1 Discussion

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 reason for missing important concepts from the input text. Also, as the number of the target EMS protocols grows, the cost and amount of effort needed for modeling the protocols and manually extending the concept list significantly increases. Thus, we plan to find an automatic way of obtaining a more complete and accurate concept list in the future. We are now investigating the vector space models and unsupervised machine learning techniques to automatically expand the limited set of manually identified concepts to a larger database of EMS relevant terms to be extracted from the text.

- 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.

- 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 extending the behavior trees based on EMS data or protocols, by adding learning capability to the nodes.

- Interventions performed by a first responder are necessarily limited by the underlying context such as transport time, the 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.

- The machine learning models trained by the uni-gram and n-gram text vectors were found to suffer from generalization problem since some of the target interventions were mentioned in the input narrative. The method we applied in this thesis to avoid this problem is using the vectorizer mechanism in the BT model to extract the signs and symptom features in the narrative. This method made the comparison between the BT model and ML models fair since they are using the same input vectors. However, the concept vectors lost some information that may not be required by EMS protocols from the narratives. Thus, we need to find a way to filter the target interventions mentioned in the narratives to develop the vector space models without bias.

- We simulated the streaming speech input in the evaluation section by merely splitting the EMS reports into chunks. This evaluation would be much more accurate if we can have a data set of actual streaming speech data from the scene along with the corresponding time marks.

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

- The evaluation results shown in Sections 5.2.2 and 5.2.3 indicate that the proposed BT model can achieve a comparable intervention prediction performance in terms of precision, recall, and F1 score compared to the performance of the supervised ML models, and meanwhile, it can significantly reduce the risk factor of the intervention suggestions.

- 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 sub-trees. However, when it comes to supervised data-driven ML methods, re-collection and labeling of data and re-training the whole model is required.

- The proposed BT framework is weakly supervised and knowledge-driven, which means that it 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.

- In contrary to ML methods which are black box end-to-end solutions from input text to interventions, the BT framework is transparent and can provide an explanation (i.e., indicate which EMS protocols are selected) for the decisions made and suggestions provided to the responders.

6.2 Conclusion

This thesis presented a Behavior Tree cognitive assistant system for emergency response which can be implemented as a portable 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 slightly outperform the knowledge-driven BT method when compared using traditional accuracy metrics with the streaming input speech data. 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 simulation training experiments for preparing responders with the knowledge of protocol guidelines and scoring their performance in executing the protocols.

6.3 Future Work

Given the results and the conclusions presented in the previous parts of this thesis, our preliminary works have already shown some achievements and limitations in developing the cognitive assistant system for emergency medical services. This work could be improved and further explored in the future from the following aspects:

- Accurate and domain-specific information retrieval from EMS data: In this thesis, MetaMap and corresponding filtering mechanism are applied to extract EMS related concept/information from the EMS reports, which is still not mature. On the other hand, the accuracy of the results from the following component largely depends on the information retrieval results. Thus, the improvement of accuracy and reliability could largely enhance the performance of the whole cognitive assistant system.

- Learning and adaptive ability of the BT model/ ML model: In this thesis, the proposed BT model need manual effort to pick the feature concepts and encode the protocol rules, and the ML models need to be pre-trained with EMS records. The learning and adaptive ability could reduce the effort to develop the BT model and train the ML model manually.
- Better dataset/pre-processing on the dataset: The dataset applied in this thesis brought some limitation to the model training and evaluation, such as the inaccurate label from the first responders, the interventions appearing in the narratives, and the ambiguity from some of the abbreviations, etc. Modified dataset or pre-processing methods could avoid part of these limitations and yield more accurate models and evaluations.
- Further analysis with noisy/missing data: At the start of this thesis, we claimed the issue that the speech data might be noisy or missing critical information needed for inference in the incident scene and we resolve this issue by taking the confidence score from speech to text conversion into account. However, we didn't further evaluate this solution in the evaluation part because of lack of access to large enough data sets of noisy and corrupted speech. It is essential to assess the performance of the proposed cognitive assistant system pipeline with noisy and corrupted speech data in the future.
- Incorrect intermediate intervention suggestions in the streaming data: In our evaluations, we found that our ML and BT models have limited performance with the small amount of input speech data because the models are purely relying on information collected from verbalized observations to infer context. To avoid the influence of the limited speech data, our model needs to have the ability to merge more information sources (e.g., vision from the environment, vital measurements from the patients, etc.) to provide better suggestions.
- More potential application scenarios: This thesis only focused on the application of intervention suggestion to first responders in the emergency scene. In the future, the proposed BT framework can be easily extended to other applications due to its modularity. For an example, we can make use of the retrieved concepts from the Information Extraction component to develop a new component that can automatically fill in the EMS report forms or make use of the intervention predictions as references to train the emergency medical technicians and score their performance.

Bibliography

[1] Old Dominion EMS Alliance. 2018 odemsa prehospital patient care protocols. http://odemsa.vaems.org/index.php?option=com_content&view=article&id=5&Itemid=50.

[2] Alan R Aronson. Effective mapping of biomedical text to the umls metathesaurus: the metamap program. In *Proceedings of the AMIA Symposium*, page 17. American Medical Informatics Association, 2001.

[3] Alan R Aronson. Metamap evaluation. 2001.

[4] Federico Bergenti and Agostino Poggi. Developing smart emergency applications with multi-agent systems. *International Journal of E-Health and Medical Communications (IJEHMC)*, 1(4):1–13, 2010.

[5] Iva Bojic, Tomislav Lipic, Mario Kusek, and Gordan Jezic. Extending the jade agent behaviour model with jbehaviourtrees framework. In *KES International Symposium on Agent and Multi-Agent Systems: Technologies and Applications*, pages 159–168. Springer, 2011.

[6] Heewon Chung, Hooseok Lee, and Jinseok Lee. Finite state machine framework for instantaneous heart rate validation using wearable photoplethysmography during intensive exercise. *IEEE journal of biomedical and health informatics*, 2018.

[7] Google Cloud. Confidence values of google speech-to-text api. https://cloud.google.com/speech-to-text/docs/basics#confidence-values.

[8] Michele Colledanchise, Alejandro Marzinotto, and Petter Ögren. Performance analysis of stochastic behavior trees. In *2014 IEEE International Conference on Robotics and Automation (ICRA)*, pages 3265–3272. IEEE, 2014.

[9] Michele Colledanchise and Petter Ögren. Behavior trees in robotics and ai, an introduction. *arXiv preprint arXiv:1709.00084*, 2017.

[10] Angelo Croatti, Sara Montagna, and Alessandro Ricci. A personal medical digital assistant agent for supporting human operators in emergency scenarios. In *Agents and multi-agent systems for health care*, pages 59–75. Springer, 2017.

[11] Angelo Croatti, Sara Montagna, Alessandro Ricci, Emiliano Gamberini, Vittorio Albarello, and Vanni Agnoletti. Bdi personal medical assistant agents: The case of trauma tracking and alerting. *Artificial intelligence in medicine*, 2018.

[12] FT De Dombal, DJ Leaper, John R Staniland, AP McCann, and Jane C Horrocks. Computer-aided diagnosis of acute abdominal pain. *Br Med J*, 2(5804):9–13, 1972.

[13] Athanasios C Dometios, Antigoni Tsiami, Antonis Arvanitakis, Panagiotis Giannoulis, Xanthi S Papageorgiou, Costas S Tzafestas, and Petros Maragos. Integrated speech-based perception system for user adaptive robot motion planning in assistive bath scenarios.

[14] Elton Domnori, Giacomo Cabri, and Letizia Leonardi. Ubimedic2: An agent-based approach in territorial emergency management. In *2011 5th International Conference on Pervasive Computing Technologies for Healthcare (PervasiveHealth) and Workshops*, pages 176–183. IEEE, 2011.

[15] MA Fleshman, IJ Argueta, CA Austin, HH Lee, EJ Moyer, and GJ Gerling. Facilitating the collection and dissemination of patient care information for emergency medical personnel. In *Systems and Information Engineering Design Symposium (SIEDS), 2016 IEEE*, pages 239–244, 2016.

[16] National Science Foundation. Intelligent cognitive assistant: Workshop summary and recommendations, 2016. https://www.nsf.gov/crssprgm/nano/reports/2016-1003_ICA_Workshop_Final_Report_2016.pdf.

[17] Kiryong Ha, Zhuo Chen, Wenlu Hu, Wolfgang Richter, Padmanabhan Pillai, and Mahadev Satyanarayanan. Towards wearable cognitive assistance. In *Proceedings of the 12th annual international conference on Mobile systems, applications, and services*, pages 68–81. ACM, 2014.

[18] Blake Hannaford. Behavior trees as a representation for medical procedures. *arXiv preprint arXiv:1801.07864*, 2018.

[19] Danying Hu, Yuanzheng Gong, Blake Hannaford, and Eric J Seibel. Semi-autonomous simulated brain tumor ablation with ravenii surgical robot using behavior tree. In *Robotics and Automation (ICRA), 2015 IEEE International Conference on*, pages 3868–3875. IEEE, 2015.

[20] ImageTrend. Critical Care Solutions, ImageTrend. http://www.imagetrend.com/solutions-ems-critical-care/, 2017. [Online; accessed 10-Jan-2018].

[21] Scikit Learn. One-vs-the-rest (ovr) multiclass/multilabel strategy. https://scikit-learn.org/stable/modules/generated/sklearn.multiclass.OneVsRestClassifier.html.

[22] Jiajun Li, Xu Xu, Jianguo Tao, Liang Ding, Haibo Gao, and Zongquan Deng. Interact with robot: An efficient approach based on finite state machine and mouse gesture recognition. In *2016 9th International Conference on Human System Interactions (HSI)*, pages 203–208. IEEE, 2016.

[23] Felipe Meneguzzi, Jean Oh, Nilanjan Chakraborty, Katia Sycara, Siddharth Mehrotra, James Tittle, and Michael Lewis. A cognitive architecture for emergency response. In *Proceedings of the 11th International Conference on Autonomous Agents and Multiagent Systems-Volume 3*, pages 1161–1162. International Foundation for Autonomous Agents and Multiagent Systems, 2012.

[24] Álvaro Monares, Sergio F Ochoa, José A Pino, Valeria Herskovic, Juan Rodriguez-Covili, and Andrés Neyem. Mobile computing in urban emergency situations: Improving the support to firefighters in the field. *Expert systems with applications*, 38(2):1255–1267, 2011.

[25] Mark A. Musen, Blackford Middleton, and Robert A. Greenes. *Clinical Decision-Support Systems*, pages 643–674. Springer London, London, 2014.

[26] Chris Paxton, Andrew Hundt, Felix Jonathan, Kelleher Guerin, and Gregory D Hager. Costar: Instructing collaborative robots with behavior trees and vision. In *Robotics and Automation (ICRA), 2017 IEEE International Conference on*, pages 564–571. IEEE, 2017.

[27] Renato de Pontes Pereira and Paulo Martins Engel. A framework for constrained and adaptive behavior-based agents. *arXiv preprint arXiv:1506.02312*, 2015.

[28] Sarah Preum, Sile Shu, Mustafa Hotaki, Ronald Williams, John Stankovic, and Homa Alemzadeh. Cognitiveems: A cognitive assistant system for emergency medical services. In *Special Issue on Medical Cyber Physical Systems Workshop (CPS-Week 2018)*. ACM SIGBED Review, 2018.

[29] Sarah Masud Preum, Sile Shu, Jonathan Ting, Vincent Lin, Ron Williams, John A Stankovic, and Homa Alemzadeh. Towards a cognitive assistant system for emergency response. In *In the poster session of the 9th ACM/IEEE International Conference on Cyber-Physical Systems (ICCPS)*. IEEE, 2018.

[30] T Allan Pryor, Reed M Gardner, Paul D Clayton, and Homer R Warner. The help system. *Journal of medical systems*, 7(2):87–102, 1983.

[31] Jesse Read, Bernhard Pfahringer, Geoff Holmes, and Eibe Frank. Classifier chains for multi-label classification. *Machine learning*, 85(3):333, 2011.

[32] Edward H Shortliffe. A rule-based computer program for advising physicians regarding antimicrobial therapy selection. In *Proceedings of the 1974 annual ACM conference-Volume 2*, pages 739–739. ACM, 1974.

[33] Manel Taboada, Eduardo Cabrera, Ma Luisa Iglesias, Francisco Epelde, and Emilio Luque. An agent-based decision support system for hospitals emergency departments. *Procedia Computer Science*, 4:1870–1879, 2011.

[34] Vincent Van Asch. Macro-and micro-averaged evaluation measures [[basic draft]]. *Belgium: CLiPS*, pages 1–27, 2013.

[35] Wikipedia. term frequency–inverse document frequency. https://en.wikipedia.org/wiki/Tf-idf.

[36] Jihang Zhang, Minjie Zhang, Fenghui Ren, Weicheng Yin, Aden Prior, Claudio Villella, and Chun-Yu Chan. Enable automated emergency responses through an agent-based computer-aided dispatch system. In *Proceedings of the 17th International Conference on Autonomous Agents and MultiAgent Systems*, pages 1844–1846. International Foundation for Autonomous Agents and Multiagent Systems, 2018.

[37] Mengxuan Zhang, Nan Li, Anouck Girard, and Ilya Kolmanovsky. A finite state machine based automated driving controller and its stochastic optimization. In *ASME 2017 Dynamic Systems and Control Conference*, pages V002T07A002–V002T07A002. American Society of Mechanical Engineers, 2017.