A Deep Learning Methodology for Semantic Utterance Classification in Domain-Specific Dialogue Systems

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

Debajyoti Datta

May

2016

APPROVAL SHEET

The thesis

is submitted in partial fulfillment of the requirements

for the degree of

Master of Science

Debajyoti Datta

AUTHOR *signature*

The thesis has been read and approved by the examining committee:

Please insert committee member names below:

Dr. Laura E. Barnes

Advisor

Dr. Matthew Gerber

Dr. Peter A. Beling

Accepted for the School of Engineering and Applied Science:

Craig H. Benson, Dean, School of Engineering and Applied Science

Month degree is awarded May

Year 2016

Print Form

A Deep Learning Methodology for Semantic Utterance Classification in Domain-Specific Dialogue Systems

by

Debajyoti Datta

A thesis submitted to The faculty of the Graduate School of Engineering and Applied Science, University of Virginia in partial fulfillment of the requirements for the degree of Master of Science.

Charlottesville, Virginia

May, 2016

Abstract

Recent advances in deep learning approaches have transformed fields such as natural language and image processing. In particular, these new advances have the potential to transform dialogue systems which are traditionally implemented using language model based or boosting approaches. In this work, we have proposed a deep learning framework to perform semantic utterance classification (SUC) for use in domain-specific dialogue systems. Deep learning has only recently been used for SUC but has not been used with domain-specific word embeddings or dialogue systems. Semantic classifiers need to account for a variety of instances where the utterance for the semantic domain class varies. In order to capture the candidate relationships between the semantic class and the word sequence in an utterance, we have proposed a shallow convolutional neural network (CNN) that uses domain-specific word embeddings, that has been initialized using word2vec for determining semantic similarity of words. These embeddings can remain static, be updated during training or can even be created from scratch for the particular intent determination task at hand.

ABSTRACT

Finally, these methodologies have been integrated into a library for easy deployment into existing platforms with dialogue systems. Experimental results obtained on two different use cases demonstrate the effectiveness of shallow neural networks for SUC. The methods produce superior classification accuracy comparable to existing benchmarks. We also demonstrate our framework in a real-world medical training system.

Acknowledgments

I am extremely thankful to Dr. Laura Barnes for her invaluable contribution towards my thesis and her suggestions at every step of it. This would have been impossible without her help. I am also extremely thankful to my friends Trishala Neeraj and William Cosby for their invaluable help in proof reading, structuring and organization of the thesis.

Dedication

This thesis is dedicated to my sister Noel.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

Intelligent Virtual Agents (IVA) or Embodied Conversational Agents have gained popularity since enabling human like conversation in a variety of scenarios like virtual patients [3] [4], pedagogical agents [5] [6] and military training [7] [8] have shown effectiveness in training, development of interpersonal skills, medical education and entertainment.

One of the key components of a system using automated virtual agents is the dialogue system. The purpose of the dialogue systems is to automatically identify the domain and intent of the user as expressed in natural language and to extract associated arguments and slots to achieve a specific goal that is generally task specific. The semantic parsing of input utterances in spoken language understanding (SLU) typically consists of three tasks: Domain Detection, Intent Determination and Slot filling.

For a ticket booking system, Domain detection part would be *airline-ticket*, the intent determination part could be *book-ticket* or *cancel-ticket* and the slot filling sections would be *to-place, from-place,date,time* and so on.

In this work, a novel architecture for intent determination in the context of dialogue systems is presented. For example, in the context of dialogue systems for medical training for a particular chronic disease, the most important aspect is intent determination. So in the context of Chronic Obstructive Pulmonary Disorder (COPD), the dialogue system should be accurately able to classify the intent of the conversation. So for the specific scenario COPD, when the nurse asked *"Do you want me to increase the Oxygen?"*, can we accurately classify the intent, i.e *intent:improve-condition*. This task is essentially a fine-grained multi-class classification problem in the specific domain of the training environment.

In this work, we have proposed novel deep learning architectures that improve upon existing state-of-the art methods for question classification. These architectures can easily be integrated and used to improve context-specific dialogue systems since most of these architectures do not require any task specific feature engineering unlike existing approaches [9].

1.1 Motivation

A key component of the embodied conversational agent is to enable human like conversations of the virtual agents. However, in various scenarios like health care [3] [4], the effectiveness of a training simulation through the usage of a conversational agent is natural language understanding component. For instance, in a virtual medical scenario the most common form of training method for training of nurses and doctors is using a standardized patient [10–14]. The problem with this approach is that training using standardized patients is generally very costly, requires a very high level of human involvement and it lacks standardization.

One alternative to using a standardized patient is to use a virtual patient with support for various conversational elements and interactive on screen elements to make the training procedure robust and effective [3] [4]. However despite key improvements in various areas of virtual patient provider platforms in graphics, animations and sounds, they are still unable to be used without significant involvement. A dialogue system [15] is a natural language understanding component that gives realistic response to questions specific to a training environment. Since the effectiveness of the training system completely depends on the accuracy of the dialogue system, the dialogue system is a critical component of virtual training environments. One of the key components of the dialogue system is the Semantic Utterance Classification part that aims to categorize to various intents within a conversation [16] which has been the focus of this work.

1.2 Problem Definition

The deep learning approaches highlighted in this work propose a variety of architectures that can make the process of intent determination in a dialogue system robust and task specific. For example, in the case of a patient suffering from Chronic Obstructive Pulmonary Disorder, who is struggling to breathe and having severe chest pains, the two sentences, “How are you doing?” and “How are you feeling?” mean the same thing.

The proposed methods have been validated using the popular TREC dataset [17] where the new methods produce superior classification accuracy comparable to existing benchmarks that were obtained through Deep Learning. Previous approaches using the TREC dataset [17], used numerous hand-coded features to achieve the state-of-the-art accuracy [9]. The approaches highlighted in this work essentially involve training the word embeddings for task-specific dialogue systems and then having a combination of neural network architectures that use the word embeddings as input for the task of intent determination and classification. This work can also be used for domain identification (not the focus of this work) of the dialogue along with intent determination. Future extensions of this work will include various slot filling techniques.

1.3 Contributions

We explore various deep learning models to solve the proposed semantic utterance classification problem also known as the intent determination problem. The proposed models for intent determination using deep learning architectures without any feature engineering have three key components which can be broadly divided as the initial word embeddings (the vector representation of words), the network architectures that exploit these embeddings and finally an approach to building an end-to-end semantic utterance classification unit. The developed network architectures give comparable results to the existing state-of-the-art results in the TREC question answer [17] database using deep learning approaches. Further, we demonstrate the utility of the proposed methods by integrating the dialogue system into a virtual human medical training simulation for interprofessional education (IPE). In the domain specific context of medical interprofessional education, our models achieve an accuracy of 98.33% in semantic utterance classification. The contribution of this thesis is both the method for intent determination for domain-specific dialogue systems as well as the extensible library for integration into domain-specific virtual human training systems.

1.4 Thesis Organization

Section 2 introduces the recent research in NLP and dialogue systems. Section 3 introduces the proposed word embedding approaches and the deep learning architectures as well as a modified architecture for semantic utterance classification. Section 4 presents the validation of our methods using the TREC [17] benchmark dataset and in a domain specific medical dataset for IPE. Section 5 presents the conclusions, limitations, and future work.

Chapter 2

Natural Language Processing and Dialogue Systems

Neural Networks are powerful learning models. Broadly they can be divided into two categories: feedforward and recurrent neural networks. There are various types of feedforward neural networks such as Convolutional Neural Networks (CNNs) with pooling layers [1]. Recurrent Neural Networks (RNNs) also have variations such as Long Short-Term Memory (LSTM) [18] and Gated Recurrent Units (GRU) [19] to name a few. These neural network models are broadly part of an area of machine learning known as deep learning. Deep learning is the stacking of multiple hidden layers or even multiple neural networks to accomplish the task of learning complex features. These network architectures thus vary based on complexity of the tasks. Deep Learning has been used for tasks such as modeling sentences [20] and sentiment analysis [1].

2.1 Dialogue Systems

Dialogue systems aim automatically identify the intent of the user as expressed in natural language and then perform the corresponding task specific to the domain. Historically the task of intent determination has emerged from the call classification systems at AT&T [21] after the success of early commercial interactive voice response applications. Another key component of dialogue systems is the slot filling which originated from non-commercial research projects from DARPA (Defense Advanced Research Program Agency) for the airline travel information systems, ATIS [22].

Majority of the work in dialogue systems rely on semantic utterance classification for the evaluation of natural language query into a particular category and then extract related parameters from that [23]. Typically, these systems use supervised classification methods like boosting [16], support vector machine approaches [9] or maximum entropy models [24]. In this work, we propose techniques for automated feature engineering using deep learning and task specific word embeddings. Feature engineering is often task specific and not generalizable to different dialogue systems and conversational agents, thus limiting the reuse of existing systems.

2.2 Deep Learning in NLP

Fully connected feedforward neural networks can be used in classification problems, or even in more complex prediction problems. Superior accuracy can be achieved given the non-linearity of the network and the ability to easily integrate pre-trained word embeddings. Using a fully connected feedforward network instead of a linear network, in addition to using pre-trained word vectors, has resulted in drastic improvements in syntactic parsing. Multilayer feedforward networks provide superior results on sentiment classification and factoid question answering, evident from their performance in language modeling [25]. Convolutional and pooling architectures [1, 20] allow the model to learn to find local indicators, regardless of their position, and hence networks using such architectures are good at complex tasks like sentiment classification, short-text categorization, relation type classification between entities, event detection, paraphrase identification semantic role labeling and question answering. While convolutional and pooling architectures allow us to encode arbitrarily large items as fixed size vectors capturing their most salient features, they fail to preserve most of the structural information present in natural language. This drawback is not seen in recurrent neural networks which are designed to model sequences and recursive networks which handle trees [25].

2.3 Feature Representation in NLP

In deep learning, there are two key approaches for representing input features. The first and the most frequently used approach is dense vectors [26,27], and the second is one-hot representations. The length of one-hot representations or one-hot vectors [28] is equal to the number of distinct features and even though there is no evidence as to which is a better approach, the performance of sparse or dense vectors.

One-hot representations are generally memory consuming, and unless there are very few unique possibilities of the predictor variables, one-hot encoding is not frequently used in natural language processing since the input vector grows exponentially with respect to the number of unique categories per variable [29]. The second approach and more common approach is dense representations. In a dense representation, each feature in a d-dimensional space will be a vector of size d. Similar features will have similar vectors and thus these low dimensional dense vectors are better than high dimensional sparse one-hot encoded vectors [26, 27].

2.4 Common Non-Linearities in Neural Networks

Non-linearities (denoted as g) take a single input and perform a fixed mathematical operation on it based on the definition of the activation function. A particular non-linearity can be more appropriate for a certain NLP task than another non-linearity [29]. Some common non-linearities are shown in equations 2.1, 2.2, 2.3 and 2.4:

Sigmoid Transformation : The Sigmoid function transforms each input into a range of $[0, 1]$. This function is defined as:

$$\sigma = \frac{1}{1 + e^{-x}} \tag{2.1}$$

Hyperbolic Tangent : The hyperbolic tangent transforms each input into the range $[-1, 1]$. This function is defined as:

$$tanh(x) = \frac{e^{2x} - 1}{e^{2x} + 1} \tag{2.2}$$

Hard Hyperbolic Tangent : The hard hyperbolic tangent function is faster to computer than the standard hyperbolic tangent and takes derivatives of:

$$hardtanh(x) = \begin{cases} -1 \text{ when } x < 1 \\ 1 \text{ when } x > 1 \\ x \text{ otherwise} \end{cases} \tag{2.3}$$

Rectifier Linear Unit : This transformation clips negative inputs to 0. ReLUs do not saturate gradients and instead act to diminish and eliminate them. This behavior is fundamentally different from sigmoid and tanh functions and is thus more suitable for Deep Neural Networks. ReLU is defined as:

$$ReLU(x) = \begin{cases} 0 \text{ when } x < 0 \\ x \text{ otherwise} \end{cases} \tag{2.4}$$

Generally, ReLU performs better than tanh functions because ReLU systems do not saturate the gradient. However tanh functions perform better than sigmoid functions because their outputs focus around zero.

2.5 Output Transformations in Neural Networks

The output layer vector can also be transformed, most commonly using the Softmax function, producing a vector of non-negative real numbers that sum to one, i.e., a discrete probability distribution over k possible outcomes. This transformation however is used only when modeling a probability distribution over the possible output class. For this method to be effective, it is used in conjunction with a probabilistic training objective like cross-entropy [29] .

2.6 Loss Functions in Neural Networks

While training a neural network, the objective is to maximize the agreement between the predicted output y' and the true output y. A loss function $L(y', y)$ is defined to quantify this agreement, in the form of a numerical score. The parameters of the network, including matrices, biases and embeddings are, hence, modified so as to minimize the loss across the training examples. Common forms of loss functions that have been studied in the deep learning literature by LeCun et al [30, 31]

Hinge Losses : Hinge losses, also known as margin losses, are used with a linear output layer. They are most useful when a hard decision rule is required and class membership probabilities are not needed. These functions are described as:

$$L_{\text{hinge(binary)}}(y', y) = max(0, 1 - y * y') \tag{2.5}$$

Binary hinge loss ensures that the binary classification is correct, by a minimum margin of 1. As per hinge loss, the loss is 0 when the predicted and true output vectors share the same sign and $|y| >= 1$. Otherwise, the loss is linear.

In the case of multi-class scenarios the loss function is expressed as:

$$L_{\text{hinge(binary)}}(y', y) = max(0, 1 - (y'_t - y'_k)) \tag{2.6}$$

Let $y = y_1, ...y_n$ be the output vector, and y be the one-hot vector for the correct output class. Let $t = argmax_i y_i$ be the correct class, and $k = argmax_{i \neq t} y_i$ be the highest scoring class such that $k \neq t$. Since the classification rule is defined as selecting the class with the highest score, multi-class hinge loss tries to score the correct class above all other classes with a minimum margin of 1.

Log Loss : The log loss variation of hinge loss has an infinte margin and is defined as:

$$L_{log}(y', y) = log(1 + exp(-(y'_t - y'_k)) \tag{2.7}$$

Categorical Cross-Entropy Loss : Categorical cross-entropy loss, also known as negative log likelihood, is defined as

$$L_{cross-entropy}(y', y) = -\sum_i y_i log(y_i') \tag{2.8}$$

This loss function measures the dissimilarity between y(true label distribution) and y'(predicted label distribution, usually assumed to have been transformed by the Softmax activation function), and is useful when we want a probabilistic interpretation of the scores. Class membership conditional distribution is defined as: $y_i = P(y = i|x)$.

Ranking Loss : The margin-based ranking loss is defined for a pair of correct and incorrect examples. It attempts to score (rank) correct inputs over incorrect ones with a minimum margin of 1. This loss function is defined as:

$$L_{ranking(margin)}(x, x') = max(0, 1 - (NN(x) - NN(x'))) \tag{2.9}$$

Here, $NN(x)$ is the score assigned by the network for input vector x.

2.7 Neural Network Training

Neural network training is done by primarily minimizing a loss function over a training set. A gradient-based method is used for training the weights of the neural network. The various training models differ in method of computation of error estimate, and in setting the parameters in the direction of the gradient.

The most common Neural Network training procedure is Stochastic Gradient Training or SGD [32]. SGD is a general optimization algorithm which attempts to set the parameters θ of the loss function $f(\theta)$, such that total loss of f is minimum over the training examples while training a neural network. SGD broadly samples a training example, computes the gradient of error on the example with respect to θ (assuming input and expected output as constants), and finally updates θ in the direction of the gradient, scaled by a learning rate η_k.

One of the methods to compute the gradients of the network's error with respect to the parameters is the backpropagation algorithm, which is essentially computing the derivatives of a complex function using the chain rule.

Chapter 3

Proposed Approach

3.1 Word Embeddings Approach

3.1.1 Word Embedding Initialization

Word Embeddings are a key component in neural networks in natural language processing. Essentially, in word embeddings, each feature is represented as a vector in low dimensional space. Currently for pre-initialization GloVe [33] and Word2Vec [34] approaches are used for word embeddings.

Word2Vec: Word2Vec [34] is one of the most commonly used word embedding technique for treating words as a feature vector for the neural network. The word2vec method essentially treats words as atomic units where they train on extremely large datasets, such as the google news dataset, which has billions of words. This method produces effective high dimensional word vectors and eventually associates words with points in space. The spatial distances show syntactic similarities (tall: taller::short: shorter), as well as they show interesting relationships (vector(King) - vector(Man) + vector(Woman) = vector(Queen)).

The fundamental idea behind word2vec is the distributional hypothesis, i.e words are characterized by the company that they keep. CBOW and Skipgram [34] are the approaches for the learning the word embeddings. CBOW or Continuous Bag of Words predict the current word w given only the context C. Skipgram on the other hand predicts words from context C given word w. Skipgram produces better word vectors for infrequent words. CBOW is faster by a factor of window size, and generally finds better word vectors for large corpuses.

GloVe : [33] The statistics of word occurrences in a corpus is key to any unsupervised methods for learning word representations. The two classes of methods for learning distributional word representations: count-based and prediction-based, both explore the word-word co-occurrence statistics of the corpus. Count-based method, however, captures the global statistics more efficiently. GloVe [33], for Global Vectors, utilizes the advantage of count data while simultaneously capturing the meaningful linear substructures prevalent in recent log-bilinear prediction-based methods.

This global log-bilinear regression model combines the advantages of the two major model families in the literature: global matrix factorization similar to latent semantic analysis (LSA) and local context window similar to skip-gram [34] model. Methods like LSA efficiently leverage statistical information but do not perform well on the word analogy task, indicating a sub-optimal vector space structure. On the other hand, methods like skip-gram may do better on the analogy task, but they are unable to utilize the statistics of the corpus well since they train on separate local context windows instead of on global co-occurrence counts. GloVe efficiently leverages statistical information by training only on the non-zero elements in a word-word co-occurrence matrix, rather than on the entire sparse matrix or on individual context windows in a large corpus.

3.1.2 Word Embedding Training

The pre-trained vectors can either be treated like a static fixed vectors that are not updated during training or can also be updated along with the entire network in which case the vectors will get reoriented based upon the task at hand even with some form of pre-initialization. In the current implementation of the code, we have three possibilities as has been described below.

Random Initialization Model : In this method the embedding vectors are initialized in a random fashion and like other parameters of the neural networks, such as the weight and learning parameters, the word embeddings are learned during training. The approach that followed was to initialize between $[\frac{-1}{2d}, \frac{1}{2d}]$, where d is the number of dimensions.This process enables having task specific initializations but does not necessarily perform optimally. Further this approach can not generalize well for new words that have not been seen in the training example.

Static Word Vectors : In this method the word vectors are static from the pre-trained word embedding models word2vec or glove. In the case of static embeddings, all the words including the unknown ones were initialized and are not updated during training.

Non-Static Word Vectors : In this method the word vectors are updated after the initialization with the pre-trained word embeddings from word2vec or glove. In this case, the embeddings of all the words including the unknown ones were initialized with the pre-trained word vectors, and they get updated during the training. This is especially well suited for task specific word embeddings.

3.2 Traditional Deep Learning Architectures

3.2.1 Convolutional Neural Networks

Single Layer CNN Implementation: One of the most common requirements in deep learning in Natural Language processing is to make predictions on ordered sets of items, which can involve words in a sentence or sentences in a document. From sentences one may need to predict the sentiment (positive, negative or neutral) of a sentence, where some words convey more meaning than the others where not only the order of words is important but so is the position. For example the two sentences, "It was not good, it was actually quite bad" and "It was not bad, it was actually quite good", have the same words, but different ordering and completely opposite intents. In cases like this, bag of words, or n-grams will not work very effectively or will result in huge and sparse embedding matrices. In this case where convolutional neural network architectures work particularly well and are a robust and elegant solution to the problem. Convolutional Neural Networks utilize layers with convolving filters that are applied to local features. This technique was popularized by LeCun [35] and have since then been used in a wide variety of NLP tasks like semantic parsing [36], search query retrieval [37], sentence modeling [20], and other traditional NLP tasks in this paper by Collobert [26].

The proposed implementation of CNN, there is one layer of convolution applied on top of the word vectors as shown in Figure 3.1 . There are three possible word vectors that can be applied. The first is a randomly initialized word vector which then trains as the network updates along with the weights. The second is a pre-trained word vector (like the word2vec model trained via word2vec approach or the Glove vector trained on the twitter corpus) where the word vector does not update, that is remains static. In the third and the final implementation, the pre-trained vectors described above update and the word embeddings are oriented in the d-dimensional vector space that are task-specific. This implementation also allows for the possibility of having both pre-trained and task-specific vectors by having multiple channels [1] for classification tasks. This is similar to Razavian's approach [38] that used feature extractors from a different model; it performed quite well on an image classification task even when the classification task was very different from the original task for which the feature extractors were trained.

The convolution operation involves multiple filters which is applied to a window of words which produces a new feature. Each filter is applied to each possible window of words in the sentence to produce a feature map and then max pooling over time operation is applied over the feature map to take the maximum value as the feature, which essentially corresponds to capturing the

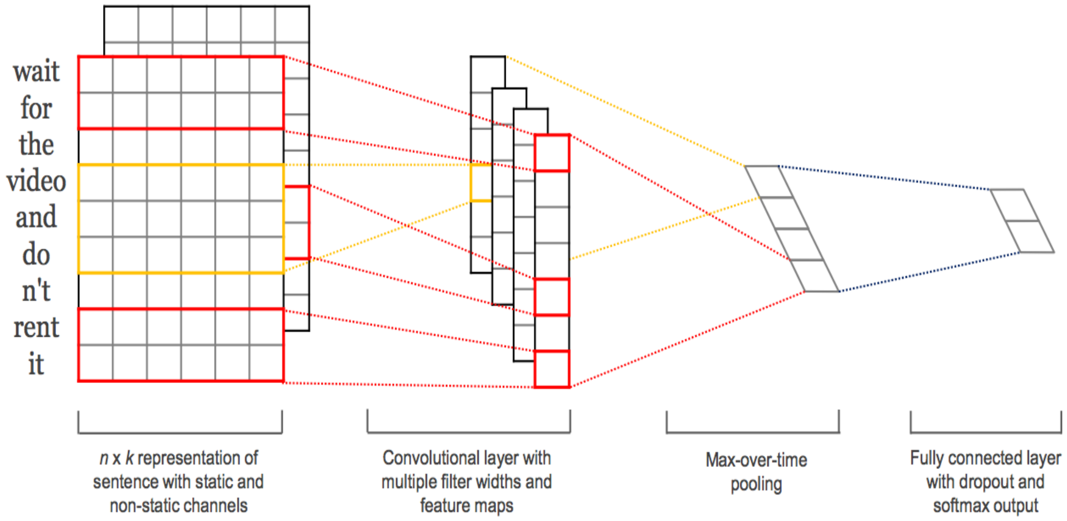

wait
for
the
video
and
do
n't
rent
it
n x *k* representation of sentence with static and non-static channels
Convolutional layer with multiple filter widths and feature maps
Max-over-time pooling
Fully connected layer with dropout and softmax output

Figure 3.1: Convolutional Neural Network, Source: Yoon Kim [1]

most important feature of the sentence. Thus, because of padding and then taking the max pool operation, it can very easily deal with sentences of any length. In the given library the filter lengths can be varied along with dropout probabilities, and the training is done with stochastic gradient descent over shuffled mini batches with the adadelta update rule.

3.2.2 Recurrent Networks

In natural language it is quite common to deal with sequences of units like words in case of sentences, sentences in case of documents and so on. The previous network architectures, feed-forward architectures and convolutional neural networks suffer from fixed-length inputs and despite padding sentences and documents perform and give sub-optimal results. RNNs or Recurrent Neural Networks [25] allow representing arbitrarily sized structured inputs in a fixed-size vector, while paying some attention to the sequence in which the input was observed.

RNN Architecture : RNNs are good with sequences as was shown by Ling et al [39]. For instance, if one wants to predict the next word in the sequence xi,....,xj, a RNN will take an ordered list of inputs, that is the words that just came before it and try to predict the next word. Essentially RNNs as shown in Figure 3.2 are recurrent in the sense that they perform the same task for every element of the sequence. An alternative way to think about this is that RNNs have memory which captures information about the calculation from the previous stages. Thus in theory RNNs can model long sequences, but in practice simple RNNs do not perform well beyond a few steps.

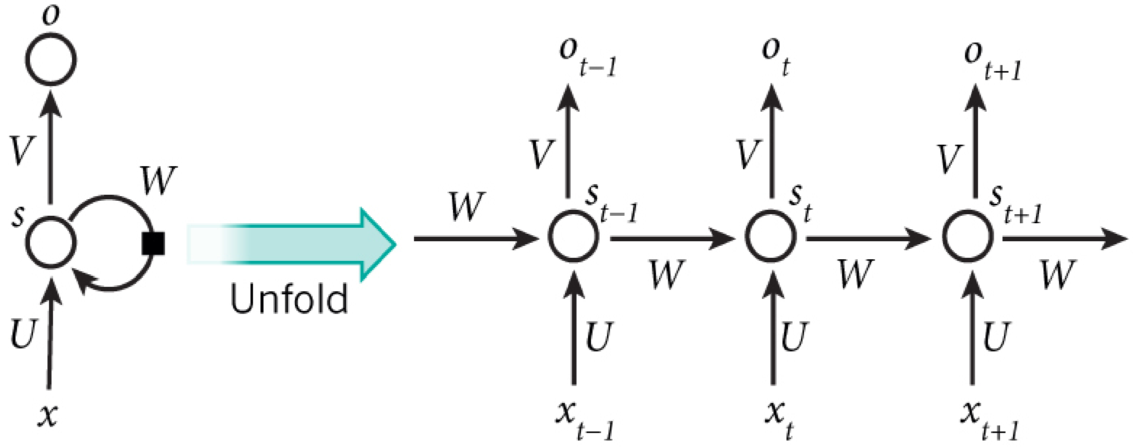


Figure 3.2: RNN sequence folded and unfolded, Source: Nature

RNN Training : The training is slightly different from training feedforward networks and convolutional neural networks in that, the same parameters are shared across many parts of the computation. To train an RNN, thus one needs to unroll the sequence, that is in order to calculate the gradient at t=3, one would need to backpropagate 2 steps and sum up the gradients. This is known as the BPTT or Back Propagation Through Time [40]. Now despite the RNN theory, RNN training does not work well beyond a few steps because of two key problems known as vanishing and exploding gradient problems. Now modifications of RNN, specifically LSTM [41] (Long Short Term Memory) and GRU [42] (Gated Recurrent Units) were designed to mitigate the vanishing gradient problem.

Bidirectional RNNs or Deep Bidirectional RNNs : This type of network is equivalent to an RNN but in this case the output at time t, may not only depend on the previous elements in the sequence but also on the future time steps. One particular example of the bidirectional RNN [43] would be to predict the missing intent in an intent classification task, or a missing word in a sequence by looking at either side of the word. So even though the two RNNs, which is the forward computation and the backward computation flow independently of each other, the error gradient at position i will flow both forward and backward through the two RNNs as shown in Figure 3.3

Long Short Term Memory Networks : LSTMs (Long short term memories) [41] are a special kind of RNN that is capable of learning long term de-

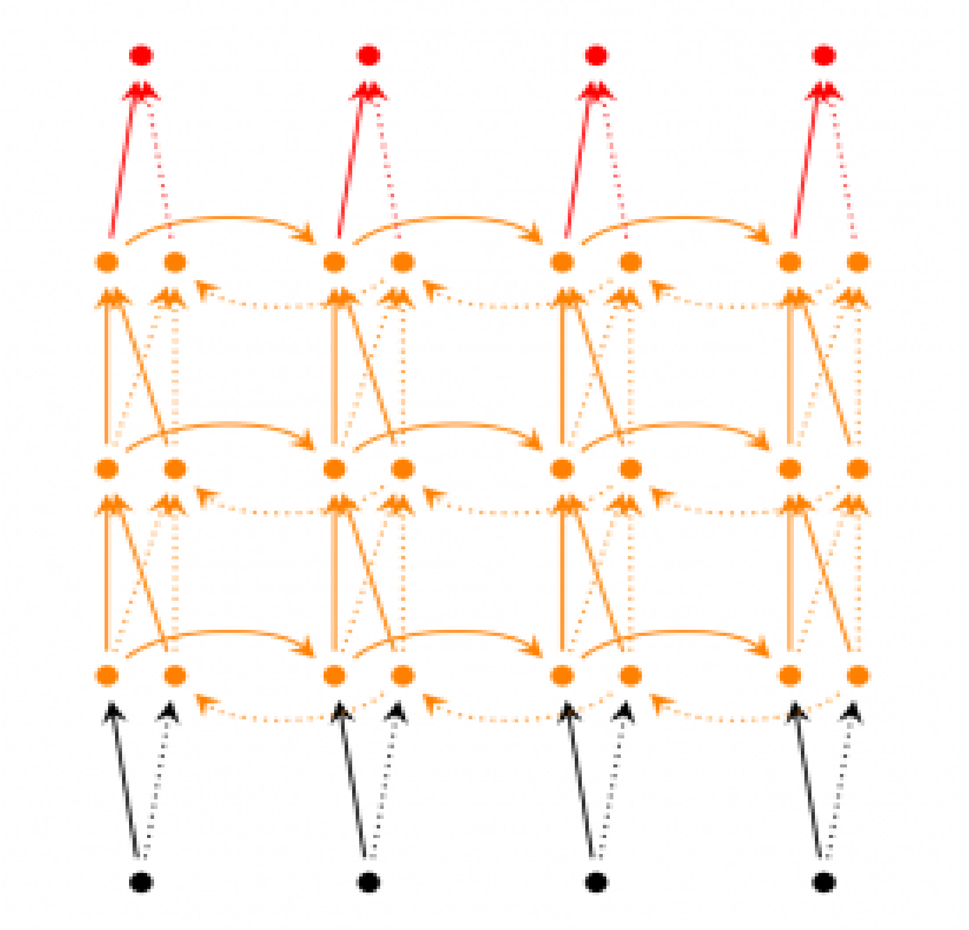


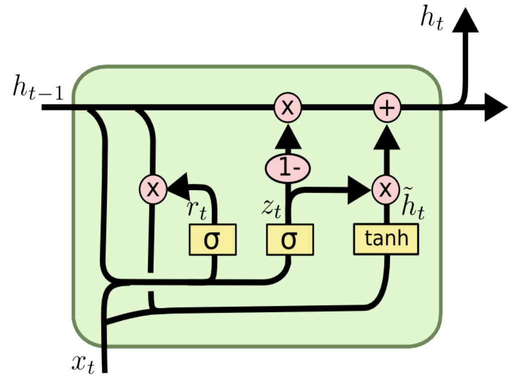
Figure 3.3: Deep bidirectional RNN, source: Wildml.com

pendencies without suffering from the vanishing gradient problem. The main idea behind LSTM networks is that they have a vector, also known as memory cells that can preserve gradients across time. Access to the memory cell is controlled by Gates that are a way of optimally letting information through. They are composed out of sigmoid neural net layer and a pointwise multiplication operation. Thus, the sigmoid layer outputs a number between zero and one to control the amount of information that should flow through the gate. There are three gates, the input, forget and the output gate. The gate values are computed based on linear combinations of the current input x_j and the previous state h_{j1}, passed through a sigmoid activation function. An update candidate g is computed as a linear combination of x_j and h_{j1}, passed through a $tanh$ activation function. The memory c_j is then updated: the forget gate controls how much of the previous memory to keep (c_{j1}, f), and the input gate controls how much of the proposed update to keep (g_i). Finally, the value of h_j (which is also the output y_j) is determined based on the content of the memory c_j , passed through a $tanh$ non-linearity and controlled by the output gate. The gating mechanisms allow for gradients related to the memory part, c_j to stay high across very long time ranges.

Gated Recurrent Unit : A slight variant of the LSTM is the gated recurrent unit or GRU [42]. The GRU combines the forget and the input gates into a single update gate. The resulting model looks like Figure 3.4 and has been growing increasingly popular. There are other variations of LSTMs but they are quite similar [44].

3.2.3 Recurrent Convolutional Neural Networks

Recurrent networks have the ability to capture contextual information. Previously mentioned, RNNs capture later words better than words that appear earlier. Now this reduces the effectiveness of modeling the semantics of the



$$z_t = \sigma\left(W_z \cdot [h_{t-1}, x_t]\right)$$
$$r_t = \sigma\left(W_r \cdot [h_{t-1}, x_t]\right)$$
$$\tilde{h}_t = \tanh\left(W \cdot [r_t * h_{t-1}, x_t]\right)$$
$$h_t = (1 - z_t) * h_{t-1} + z_t * \tilde{h}_t$$

Figure 3.4: Gated recurrent unit

entire sentences since the key components could technically appear anywhere in the sentence instead of appearing towards the end of the sentence. C onvolutional neural networks are more easily able to determine discriminative phrases in a text with the max pooling layer. The issue with convolutional neural networks is that it is difficult to determine the window size, and that may result in the loss of critical information if the window size is small, and alternatively having excessive information if the window size is large. Thus, to address the aforementioned limitation Recurrent Convolutional Networks were created [2]. Essentially a recurrent convolutional network is a bi-directional RNN, that introduces considerably less noise compared to a traditional window based network to capture the contextual information. A max pooling layer that automatically judges which features play a key role in the classification task is then applied to capture the key component of the text. This approach outperforms previous state-of-the-art in 3 text classification tasks, the 20News group dataset, the ACL anthology network dataset and the Stanford Sentiment treebank dataset [2]. The structure of a recurrent convolutional network as was shown in the paper [2] has been shown in Figure 3.5 This figure is a partial example of the sentence A sunset stroll along the South Bank affords an array of stunning vantage points, and the subscript denotes the position of the corresponding word in the original sentence.

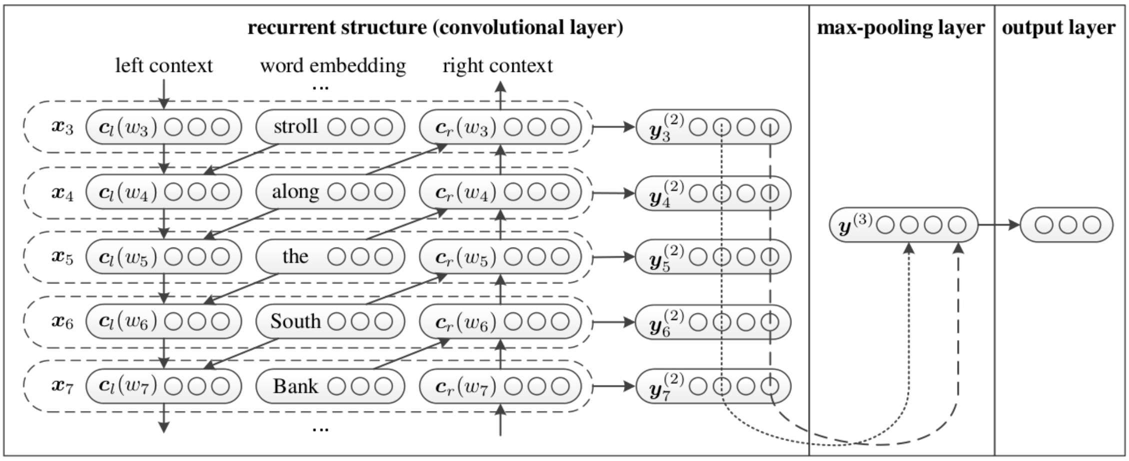

recurrent structure (convolutional layer)
max-pooling layer
output layer
left context
word embedding
right context
x_3 $c_l(w_3)$ stroll $c_r(w_3)$ $y_3^{(2)}$
x_4 $c_l(w_4)$ along $c_r(w_4)$ $y_4^{(2)}$
x_5 $c_l(w_5)$ the $c_r(w_5)$ $y_5^{(2)}$
x_6 $c_l(w_6)$ South $c_r(w_6)$ $y_6^{(2)}$
x_7 $c_l(w_7)$ Bank $c_r(w_7)$ $y_7^{(2)}$
$y^{(3)}$

Figure 3.5: Fig Source: Bengio et al. [2] The structure of the recurrent convolutional neural network

3.3 Proposed Architectures

The proposed architectures thus have three key components a word embedding initialization layer, a recurrent neural network layer and a convolutional neural network layer. The embedding layer can have task specific embeddings trained through word2vec or glove. The recurrent layer, which can be a simple recurrent neural network, a long short term memory network or a gated recurrent unit. This is followed by multi-filter convolutional neural networks. The entire architecture has been shown in Figure 3.6

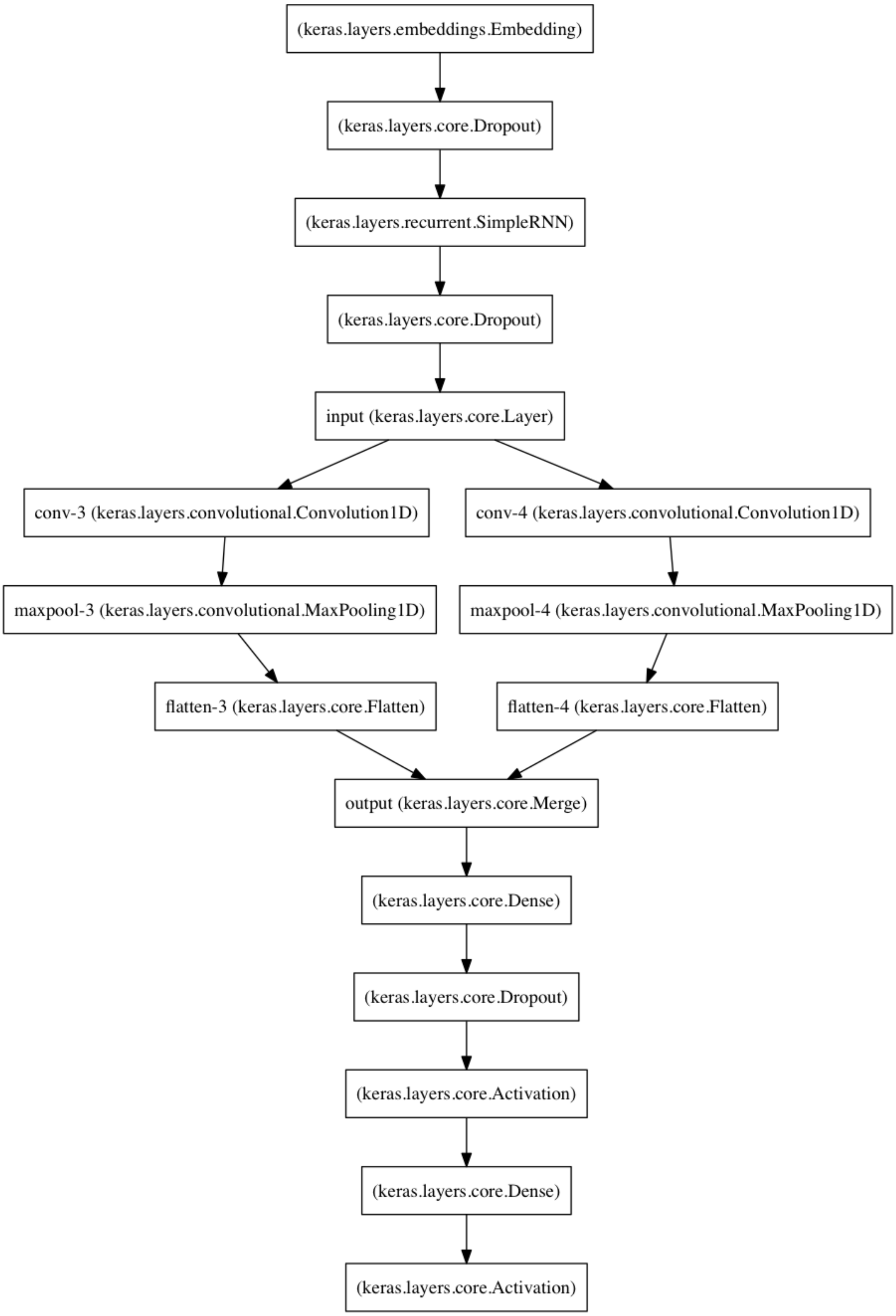

(keras.layers.embeddings.Embedding)
(keras.layers.core.Dropout)
(keras.layers.recurrent.SimpleRNN)
(keras.layers.core.Dropout)
input (keras.layers.core.Layer)
conv-3 (keras.layers.convolutional.Convolution1D)
conv-4 (keras.layers.convolutional.Convolution1D)
maxpool-3 (keras.layers.convolutional.MaxPooling1D)
maxpool-4 (keras.layers.convolutional.MaxPooling1D)
flatten-3 (keras.layers.core.Flatten)
flatten-4 (keras.layers.core.Flatten)
output (keras.layers.core.Merge)
(keras.layers.core.Dense)
(keras.layers.core.Dropout)
(keras.layers.core.Activation)
(keras.layers.core.Dense)
(keras.layers.core.Activation)

Figure 3.6: Final proposed architecture

Chapter 4

System Evaluation and Validation

4.1 Evaluation on TREC Dataset

The TREC dataset has 5500 questions in 6 categories. The categories are shown Table 4.1. The TREC test set has 500 questions. A multi-channel version of a CNN along with an RNN (Simple RNN, GRU or LSTM) has been implemented for evaluation.

Coarse	Fine
ABBREVIATION	Abbreviation, expansion
DESCRIPTION	Definition, description, manner, reason
ENTITY	Animal, body, color, creative, currency, medical disease, event, food, instrument, language, letter, other, plant, product, religion, sport, substance, symbol, technique, term, vehicle, word
HUMAN	Description, group, individual, title
LOCATION	City, country, mountain, other, state
NUMERIC	Code, count, date, distance, money, order, other, percent, period, speed, temperature, size, weight

Table 4.1: TREC QA data variable descriptions

WEI Method	CNN	CNN + SimpleRNN	CNN+ GRU	CNN+ LSTM
Random	0.8927	0.2416	0.9138	**0.9134**
Word2vec Static	0.5361	0.2395	0.4138	0.3909
Word2vec Non-Static	0.7097	0.2295	**0.9128**	0.9156

Table 4.2: TREC Dataset results using the proposed architectures

RNNs capture temporal dependencies, and CNNs capture the most important part of the sentence through max pooling. Even though RNNs can capture local dependencies, in practice, they don't generalize well for earlier time steps because of the vanishing gradient problem. The TREC dataset [17] involves classifying a question into 6 question types, whether the question is about a person, location, numeric information, etc. as shown in 4.1. In our example cases we found that static word embeddings performed the worst irrespective of the network architecture which essentially shows how the word embeddings are critical to the performance of the overall architecture. Simple RNNs did not perform well primarily because of the vanishing gradient problem. Long

Short Term Memory Architectures and Gated Recurrent Units performed comparable to existing state-of-the-art approaches. This architecture also did not use any hand coded feature as in previous SVM approaches [9].

4.2 Use Case: Virtual Human Medical Training System for Interprofessional Education

4.2.1 Background and Summary of Problem

In the realm of health care practices, inter-professional education (IPE) has been identified as a key mechanism to prepare students to function effectively in health care teams. IPE has been shown to improve knowledge and attitudes about collaboration and team functionality; however, studies using simulated IPE experiences have revealed only short-term associations between simulation and inter-professional collaborative behaviors [45]. According to the Joint Commission [46] , communication failure within health-care teams is the leading cause of medical errors. Inter-professional education (IPE) aimed at improving communication among members of the health care team plays an essential role in preparing students and clinicians to deliver safe high-quality team-based collaborative patient care.

Growing evidence supports the position that IPE and collaborative care are essential elements of health care education and practice [47, 48]. Nonetheless, there remains significant scheduling, resource, and faculty development barriers to integrating IPE experiences in meaningful and measurable ways [49–51]. Furthermore, there is not a standard mechanism for IPE training and assessment in health professions education making IPE challenging to evaluate.

A few institutions are developing and testing Inter-professional Teamwork Objective Structured Clinical Examinations (ITOSCEs) for the assessment of teamwork competencies using standardized patients [10–14]. Students who participate in one of these IPE training programs are assessed using a Collaborative Behaviors Observational Assessment Tool (CBOAT), which measures the specific inter-professional teamwork behaviors associated with the IPE competency goals for that IPE experience [11, 13].

We have developed a reusable architecture that addresses the issue of Virtual OSCEs (Objective Structured Clinical Examinations) in IPE scenarios. The primary purpose of the design of the architecture was not to have a Virtual Human System that is general purpose and used in a wide range of scenarios like VHToolkit [52], but instead focused entirely on the specific needs of Inter-professional Teamwork Objective Structured Clinical Examinations education for health profession students and clinicians. This system is also designed to be lightweight enough to be deployed over the web to have the widest potential use to participants. Additionally, the architecture of the system itself is designed to allow for rapid development of new training scenarios.

We have developed a virtual human training tool called the Virtual Patient Provider Platform for Inter-professional Teamwork Objective Structured Clinical Examinations (VPP ITOSCE) that integrates virtual human technology with evaluated, structured methods of IPE education. In the VPP ITOSCE, a nursing or medical student performs an assessment of a virtual patient and must engage in teamwork communication with a virtual provider while continuing to provide care to the virtual patient. The student is scored on interview and assessment skills, sequenced steps in patient management, and collaboration with the virtual provider using the validated checklist of behaviors described in an associated Collaborative Behaviors Observational Assessment Tool (CBOAT). As an initial prototype, we have integrated a scenario that allows a nursing student to interact with a virtual patient in a Rapid Response ITOSCE. To demonstrate the entire architecture this VPP ITOSCE focuses on a patient suffering from an acute case of Chronic Obstructive Pulmonary Disorder (COPD) that is refusing treatment. The user's goal is to interact with the virtual patient and provider and the virtual environment in general in order to successfully address the patient's concerns and resolve the situation. We have also prototyped the intelligent response system utilizing retrospective data obtained from videotaped interactions of students with standardized patients and providers in the Rapid Response ITOSCE and associated CBOAT implemented for all nursing and medical students at the University of Virginia in 2013 and 2014.

4.2.1.1 Clinical Training with Simulations

There is evidence that experiential learning, which is defined as learning that takes place as a result of an encounter with an experience that is planned by teachers within a course or curriculum, [53] is an effective approach to learn IPE concepts [54]. Students participating in high-fidelity and standardized patient IPE simulated scenarios are provided experiential, reflective and contextual educational experiences in which to learn the skills required to practice collaboratively. Through simulated IPE scenarios, students are fully engaged in the educational experience, and must integrate the required IPE knowledge, behaviors, and competencies in order to respond effectively in a collaborative team practice setting. Following the simulation, they also have the opportunity to observe the outcomes of their actions and clinical decisions through debriefing, videotapes, and observer feedback. Thus, simulation provides a realistic practice setting in which to learn the concepts, behaviors, and competencies of collaborative practice without the possibility of placing a live patient at risk. In one study, it was demonstrated that simulated operating room (OR) team training increased self-efficacy related to interdisciplinary team work [55]. Furthermore, there is accumulating evidence that competencies learned through simulation are transferred to the practice setting [56].

4.2.2 System Architecture

The VPP ITOSCE tool is designed to address the needs of interprofessional education among clinicians. We first demonstrate the feasibility of the VPP ITOSCE to a Rapid Response scenario which requires medical or nursing students to interact with a virtual patient as well as collaborate with a virtual provider in treating the patient. Based on these specific needs of IPE the system uses state-of-the-art techniques in the development of virtual humans and is portable and lightweight for web-deployment. The Rapid Response Scenario was chosen because students must engage in patient communication and complete an assessment of the patient's condition while under stress and then utilize interprofessional team collaboration competencies with other health-care providers to resolve the patients concerns promptly.

The system architecture as in Figure 4.1 is broadly divided into the following categories:

1. Scenario

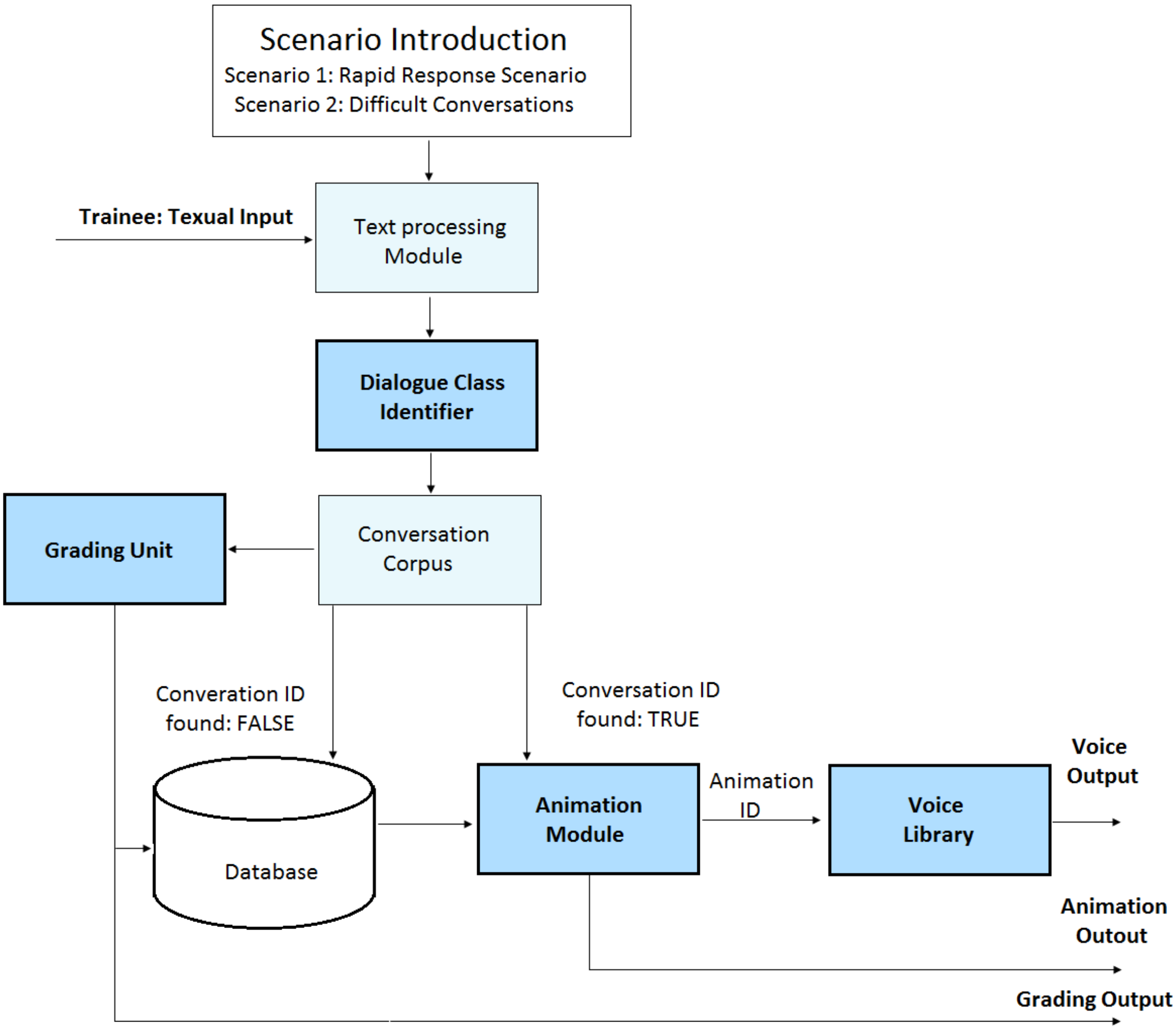


Figure 4.1: Architecture of the COPD IPE system

2. Dialogue Classification
3. Animation Module and Behavior Generation
4. Evaluation Tool or Grading Unit

The scenario introduction is the first stage which is animation and video combined to introduce the present condition of the patient. This involves a brief medical history of the patient and what is expected of the nurse in the

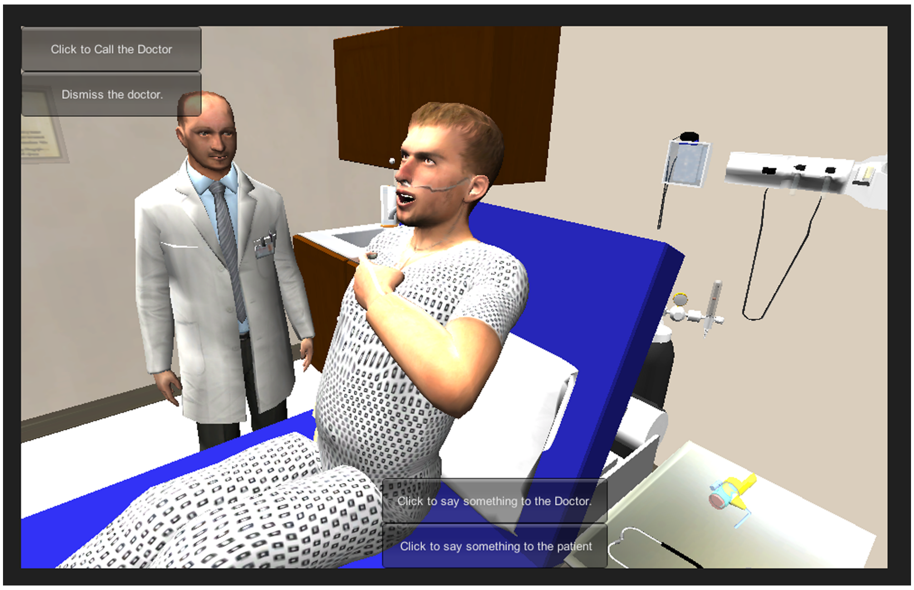

Click to Call the Doctor
Dismiss the doctor.
Click to say something to the Doctor.
Click to say something to the patient

Figure 4.2: Virtual Patient Provider ITOSCE Platform

training session.

The conversation manager and the knowledge base concentrates on not only two way conversation but extends to multiparty conversations. The team building aspect of IPEs is repeatedly highlighted in the various training modules.

The animation module controls activating the correct animation given information regarding user actions in the environment and statements addressed towards the virtual patient or the virtual provider.

The grading unit controls the automated evaluation of the nurse. The behavior of this unit is closely integrated with the functions of the conversation manager and also communicates with the state of the virtual environment regarding patient condition and actions taken to examine the patient.

4.2.3 Scenario

The scenario developed into the VPP ITOSCE portrays a virtual patient with COPD exhibiting shortness of breath. The motivation behind developing this rapid response scenario is that COPD affects more than 24 million Americans and claims over 120,000 lives each year [57]. The student and a virtual clinician must work together to convince the virtual patient to take his/her breathing treatment as shown in the Figure 4.2. Another Critical Care/Rapid Response scenario under development involves sepsis care which is the cause of approximately 570,000 emergency department visits annually and results in approximately 200,000 deaths [58]. In this scenario, the student must assess a virtual patient for sepsis, and work with a virtual clinician to provide effective sepsis care treatment. Using the Collaborative Care Best Practice Models approach to design IPE experiences, other scenarios beyond the Critical Care/Rapid Response scenarios can be developed which pertain to different practice settings and the specific variables unique to the interaction and structure of the collaborative team in those settings. Two examples are Chronic Progressive Illness scenarios such as a pediatric patient with Duchenne muscular dystrophy and Transitions for the Cognitively Impaired scenarios such as an elderly patient with Alzheimer's Disease, who is making the transition from the hospital to home.

4.2.4 IPE Dialogue System and Data

Currently, the majority of virtual human interactions involve only a single user and one or two virtual human conversation partners. The user is predominantly the lead of the conversation with scenario scripts using a natural language processing approach or a speech-trigger matching approach that links leader utterances (typed, spoken, or selected from a multiple choice list) to a particular virtual human and the appropriate response [59]. These approaches work well for information-gathering interactions (e.g. doctors speaking with patients [60]). However, team-training and other multiparty interactions are usually goal-oriented interactions centered on the patient. In such interactions, each individual has goals for the information and actions they are striving to achieve. Prior work into multiparty dialogue systems has generated systems capable of multiparty conversations focused on open-ended conversations. However, since the rapid response CBOAT is structure and primarily driven by the user interactions, we will extend conversational modeling infrastructures described in [59, 61, 62].

The proposed dialogue management framework thus needed to do fine-grained dialogue classification and intent recognition for the specific domain of Chronic Obstructive Pulmonary Disorder or COPD. The dialogue system should be capable of fine-grained analysis and distinction of the various components of the evaluation of the nursing student in training as shown in Figure 4.3.

The COPD IPE dataset was transcribed from 54 videos of nursing students with standardized patients. Each of these videos had roughly 20 interactions with a total of 2300 sentences.

The COPD IPE data has conversations that can be broadly divided into categories like introduction, patient condition inquiry, reassurance and so on as shown in Figure 4.3. Some sample questions and the corresponding answers to each of the categories is shown in Figure 4.3.

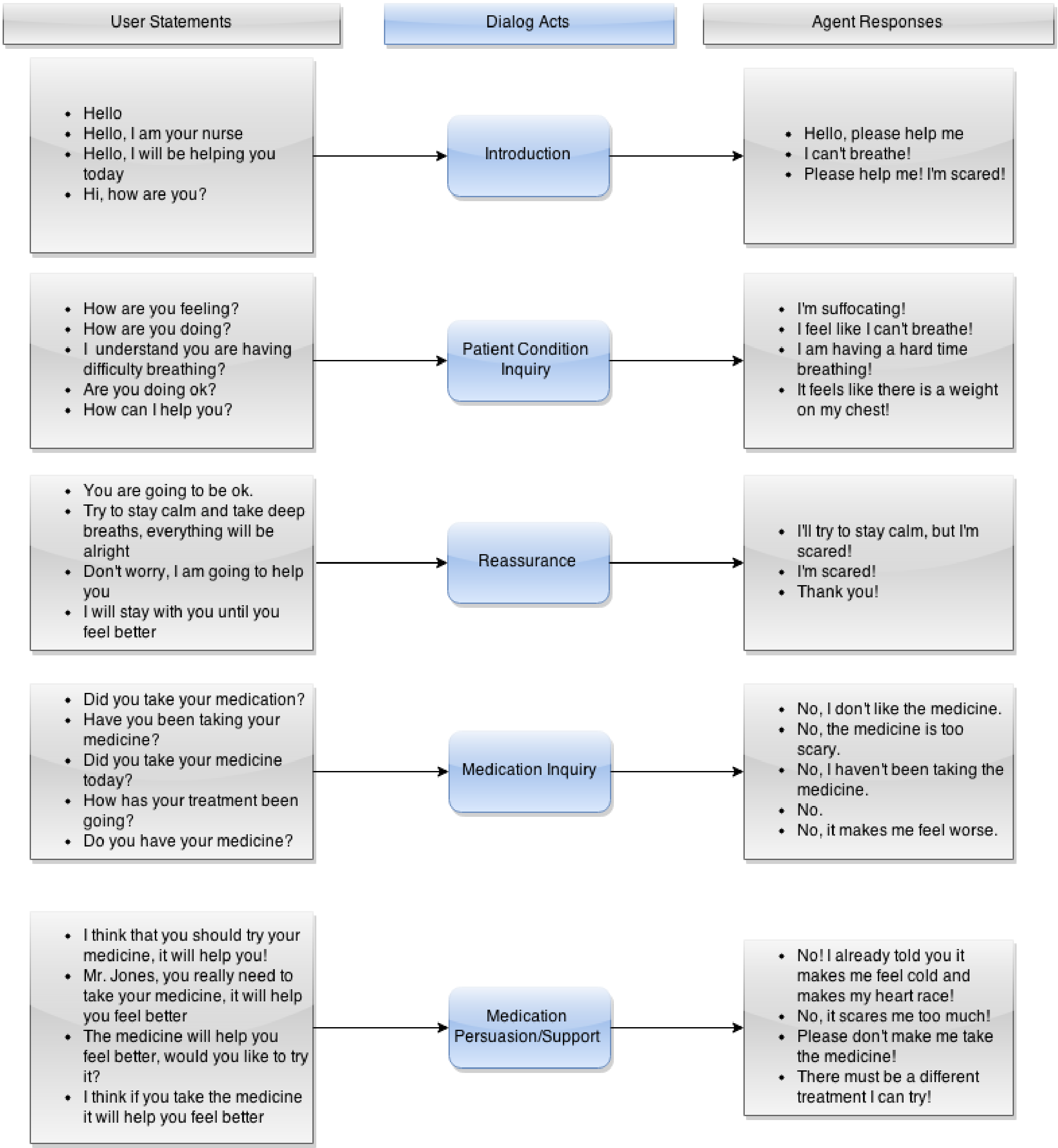
4.2.5 Evaluation on the COPD IPE Data

The COPD IPE data was split into a training set with 2000 sentences and a test set with 300 sentences with an approximately equal number of each available category in the test set, i.e from each possible category for SUC in the IPE dataset- Figure 4.3.

For the IPE COPD Scenario, since the number of sentences in the training example was much lower than the TREC dataset, CNN non-static performed equally well compared to CNN-LSTM non-static and CNN-GRU non-static. This was because the number of unique words in the vocabulary was very low.

Since there is no existing benchmarks to compare this result, we performed a test set evaluation on the COPD dataset and achieved an accuracy of 0.9833.

These results demonstrate the effectiveness in the proposed approach for semantic utterance classification in context-specific dialogue systems. This



User Statements	Dialog Acts	Agent Responses
• Hello • Hello, I am your nurse • Hello, I will be helping you today • Hi, how are you?	Introduction	• Hello, please help me • I can't breathe! • Please help me! I'm scared!
• How are you feeling? • How are you doing? • I understand you are having difficulty breathing? • Are you doing ok? • How can I help you?	Patient Condition Inquiry	• I'm suffocating! • I feel like I can't breathe! • I am having a hard time breathing! • It feels like there is a weight on my chest!
• You are going to be ok. • Try to stay calm and take deep breaths, everything will be alright • Don't worry, I am going to help you • I will stay with you until you feel better	Reassurance	• I'll try to stay calm, but I'm scared! • I'm scared! • Thank you!
• Did you take your medication? • Have you been taking your medicine? • Did you take your medicine today? • How has your treatment been going? • Do you have your medicine?	Medication Inquiry	• No, I don't like the medicine. • No, the medicine is too scary. • No, I haven't been taking the medicine. • No. • No, it makes me feel worse.
• I think that you should try your medicine, it will help you! • Mr. Jones, you really need to take your medicine, it will help you feel better • The medicine will help you feel better, would you like to try it? • I think if you take the medicine it will help you feel better	Medication Persuasion/Support	• No! I already told you it makes me feel cold and makes my heart race! • No, it scares me too much! • Please don't make me take the medicine! • There must be a different treatment I can try!

Figure 4.3: COPD IPE sample conversation and SUC categories

Architecture Type	3,4 filters	2,3,4 filters	2,3,4,5 filters
CNN Static	0.4315	0.4254	0.3824
CNN Rand	0.7035	**0.8139**	0.7751
CNN Non-Static	0.6585	0.6912	0.7239

Table 4.3: Final Results on the optimal number of filters in the COPD IPE dataset

	CNN	CNN + SimpleRNN	CNN + GRU	CNN + LSTM
Random	0.9506	0.9684	0.9560	0.9431
Word2vec Static	0.4872	0.5762	0.5814	0.5612
Word2vec Non-Static	0.9666	**0.9833**	0.9640	0.9586

Table 4.4: Results on the COPD IPE dataset

dataset is more representative than the TREC data for intent determination since the questions are all from the same domain.

Chapter 5

Conclusions and Future Work

5.1 Summary

This thesis defines the multi-label classification problem for intent determination or semantic utterance classification. We proposed a deep learning methodology specifically targeted at SUC. In the proposed method, the random or pre-trained word embedding is fed into a type of Recurrent Neural Network (LSTM, GRU or a Simple RNN) to capture dependencies among words in the sentences. The output from the RNN is then fed into multi-channel convolutional layers to capture local semantics. The max over time pooling layers capture global semantic features followed by a fully connected layer with dropout to summarize the features. Our experiments show that this approach outperformed traditional feature engineered approaches for intent determination tasks. Shallow CNNs captured semantic similarities better than Deep CNNs because in intent determination sentences are generally short with 6-8 words per sentence. For longer sentences, deep CNN's performed better as in sentence classification tasks [1]. These methods lay the foundation for implementing high performance, context-specific dialogue systems.

5.2 Limitations

One of the key limitation of the work proposed in this thesis is that the TREC dataset and COPD IPE dataset are much smaller than those traditionally used in deep learning architectures. Even though we used shallow networks in this thesis, and the methods performed comparable to existing benchmarks, the methodology needs to be validated with larger, more heterogeneous datasets. Since the improvements in performance were minimal from CNN to CNN with RNN, it would be interesting to see if stacked CNNs with pretraining of each layer can perform comparable to the results that have been obtained in this thesis.

Another limitation of this work is that SUC is domain specific. Open-domain systems do not have a robust approach to question answering, and it is extremely difficult to model such large vocabulary. The work presented here focuses on datasets with a small number of categories, and it is unclear how robust this approach will be with increasing numbers of categories. Hierarchical CNNs along with the network architectures that have been mentioned in this thesis are one potential approach.

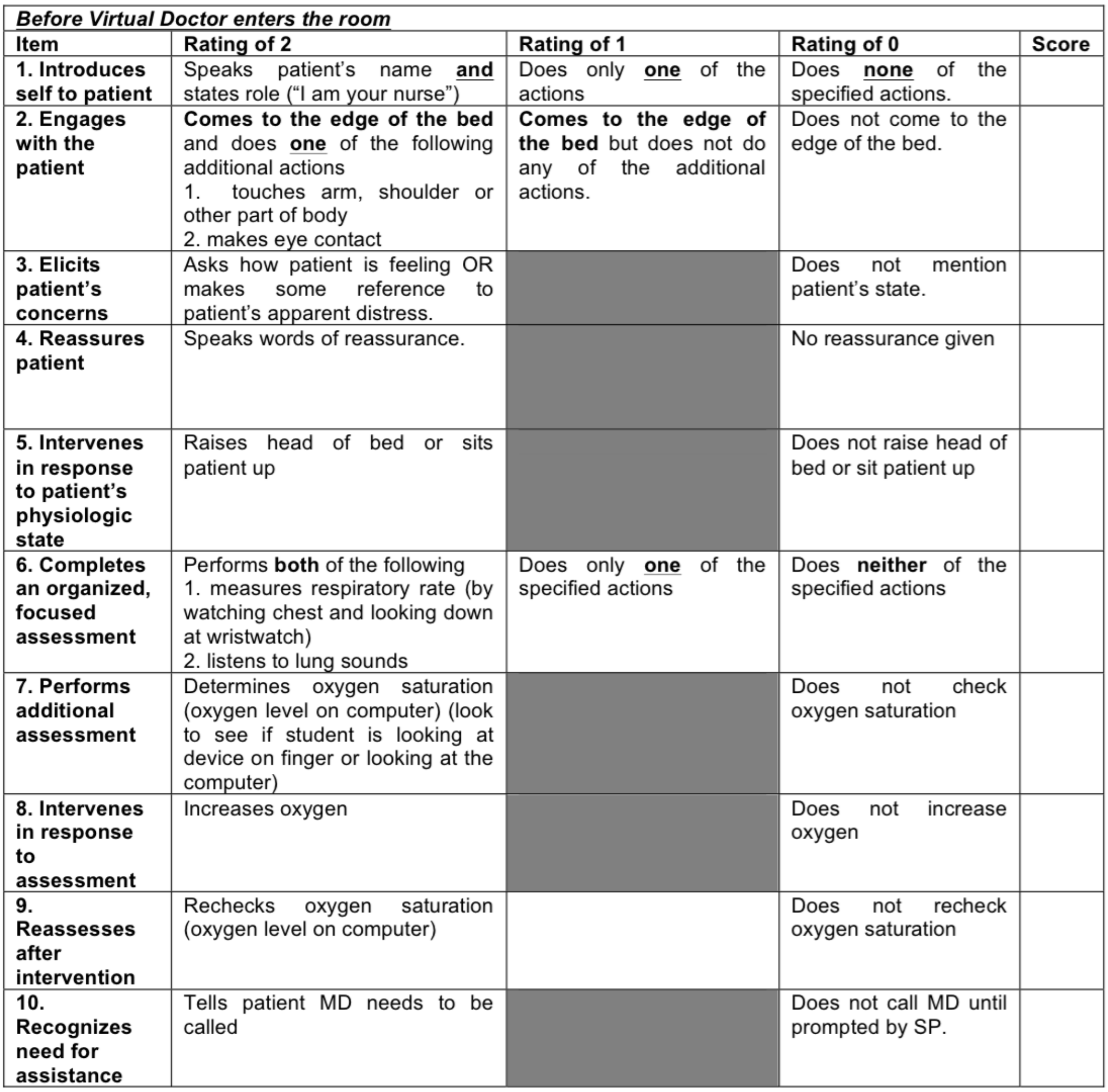
5.3 Future Work

The proposed deep learning library implements many architectures for intent determination. Even though these perform optimally for the TREC dataset [17] and the COPD IPE dataset, these methods need to be validated with more data to determine optimal network architectures for intent determination. In future work, methods for domain detection and slot filling will also be integrated into the deep learning library.

Appendix A

CBoat Grading Criteria

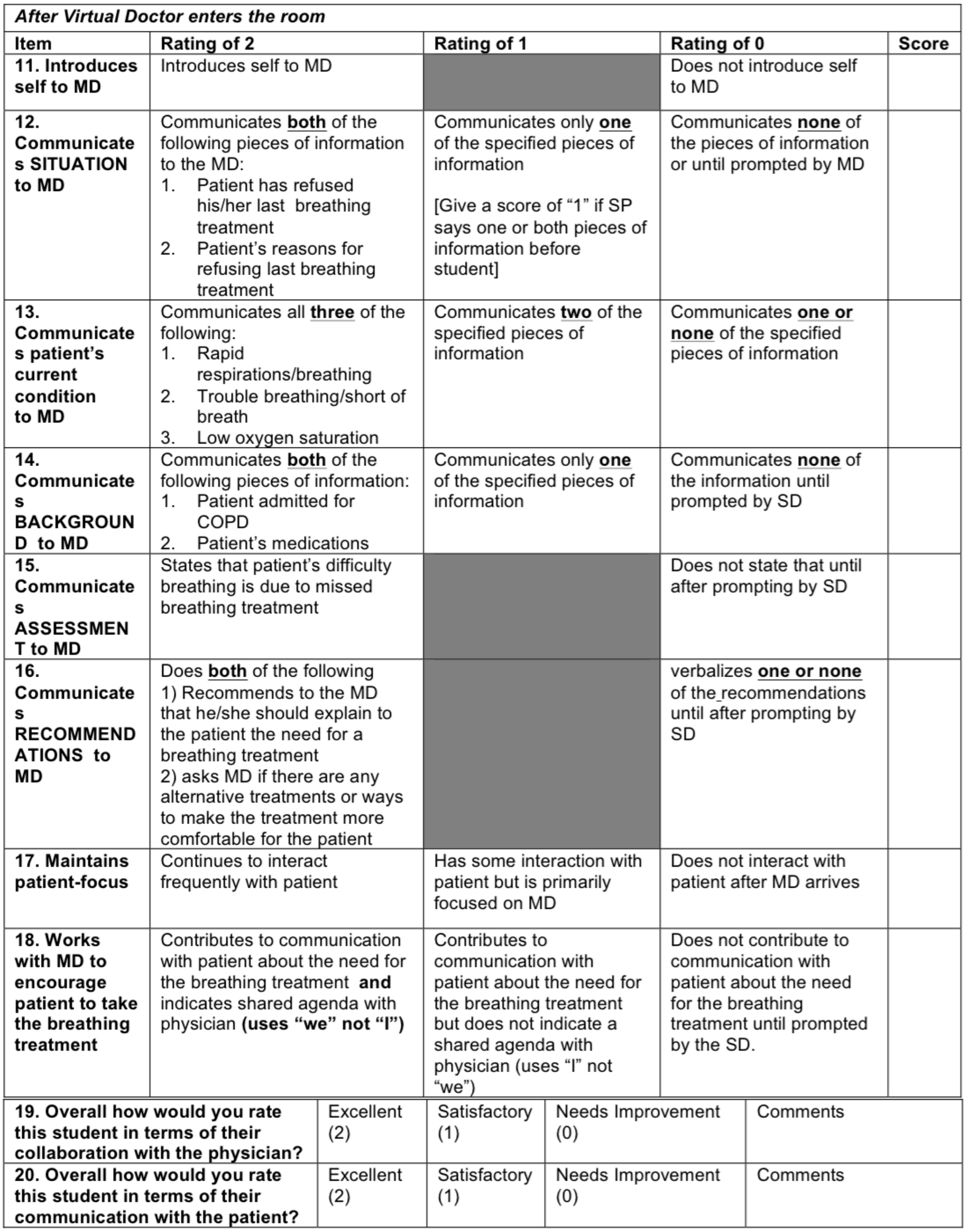
APPENDIX A. CBOAT GRADING CRITERIA



Before Virtual Doctor enters the room				
Item	**Rating of 2**	**Rating of 1**	**Rating of 0**	**Score**
1. Introduces self to patient	Speaks patient's name **and** states role ("I am your nurse")	Does only **one** of the actions	Does **none** of the specified actions.	
2. Engages with the patient	**Comes to the edge of the bed** and does **one** of the following additional actions 1. touches arm, shoulder or other part of body 2. makes eye contact	**Comes to the edge of the bed** but does not do any of the additional actions.	Does not come to the edge of the bed.	
3. Elicits patient's concerns	Asks how patient is feeling OR makes some reference to patient's apparent distress.		Does not mention patient's state.	
4. Reassures patient	Speaks words of reassurance.		No reassurance given	
5. Intervenes in response to patient's physiologic state	Raises head of bed or sits patient up		Does not raise head of bed or sit patient up	
6. Completes an organized, focused assessment	Performs **both** of the following 1. measures respiratory rate (by watching chest and looking down at wristwatch) 2. listens to lung sounds	Does only **one** of the specified actions	Does **neither** of the specified actions	
7. Performs additional assessment	Determines oxygen saturation (oxygen level on computer) (look to see if student is looking at device on finger or looking at the computer)		Does not check oxygen saturation	
8. Intervenes in response to assessment	Increases oxygen		Does not increase oxygen	
9. Reassesses after intervention	Rechecks oxygen saturation (oxygen level on computer)		Does not recheck oxygen saturation	
10. Recognizes need for assistance	Tells patient MD needs to be called		Does not call MD until prompted by SP.	

Figure A.1: CBOAT grading scale: Part 1

APPENDIX A. CBOAT GRADING CRITERIA



After Virtual Doctor enters the room				
Item	**Rating of 2**	**Rating of 1**	**Rating of 0**	**Score**
11. Introduces self to MD	Introduces self to MD	▨	Does not introduce self to MD	
12. Communicates SITUATION to MD	Communicates **both** of the following pieces of information to the MD: 1. Patient has refused his/her last breathing treatment 2. Patient's reasons for refusing last breathing treatment	Communicates only **one** of the specified pieces of information [Give a score of "1" if SP says one or both pieces of information before student]	Communicates **none** of the pieces of information or until prompted by MD	
13. Communicates patient's current condition to MD	Communicates all **three** of the following: 1. Rapid respirations/breathing 2. Trouble breathing/short of breath 3. Low oxygen saturation	Communicates **two** of the specified pieces of information	Communicates **one or none** of the specified pieces of information	
14. Communicates BACKGROUND to MD	Communicates **both** of the following pieces of information: 1. Patient admitted for COPD 2. Patient's medications	Communicates only **one** of the specified pieces of information	Communicates **none** of the information until prompted by SD	
15. Communicates ASSESSMENT to MD	States that patient's difficulty breathing is due to missed breathing treatment	▨	Does not state that until after prompting by SD	
16. Communicates RECOMMENDATIONS to MD	Does **both** of the following 1) Recommends to the MD that he/she should explain to the patient the need for a breathing treatment 2) asks MD if there are any alternative treatments or ways to make the treatment more comfortable for the patient	▨	verbalizes **one or none** of the recommendations until after prompting by SD	
17. Maintains patient-focus	Continues to interact frequently with patient	Has some interaction with patient but is primarily focused on MD	Does not interact with patient after MD arrives	
18. Works with MD to encourage patient to take the breathing treatment	Contributes to communication with patient about the need for the breathing treatment **and** indicates shared agenda with physician **(uses "we" not "I")**	Contributes to communication with patient about the need for the breathing treatment but does not indicate a shared agenda with physician (uses "I" not "we")	Does not contribute to communication with patient about the need for the breathing treatment until prompted by the SD.	

19. Overall how would you rate this student in terms of their collaboration with the physician?	Excellent (2)	Satisfactory (1)	Needs Improvement (0)	Comments
20. Overall how would you rate this student in terms of their communication with the patient?	Excellent (2)	Satisfactory (1)	Needs Improvement (0)	Comments

Figure A.2: CBOAT grading scale: Part 2

Bibliography

[1] Y. Kim, "Convolutional neural networks for sentence classification," 2014.

[2] S. Lai, L. Xu, K. Liu, and J. Zhao, "Recurrent convolutional neural networks for text classification." 2015.

[3] B. Rossen and B. Lok, "A crowdsourcing method to develop virtual human conversational agents," *International Journal of HCS*, pp. 301–319, 2012.

[4] T. Bickmore, L. Bukhari, L. Pfeiffer, M. Paasche-Orlow, and C. Shanahan, "Hospital buddy: A persistent emotional support companion agent for hospital patients," *Springer Berlin / Heidelberg*, pp. 492–495, 2012.

[5] S. K. D'Mello and A. C. Graesser, "Autotutor and affective autotutor: Learning by talking with cognitively and emotionally intelligent computers that talk back." *ACM Transactions on Interactive Intelligent Systems*, vol. 2, 2012.

[6] H. C. Lane, D. Noren, D. Auerbach, M. Birch, and W. Swartout, "Tutoring

goes to the museum in the big city: A pedagogical agent for ise," *Artificial Intelligence in Education*, pp. 155–162, 2011.

[7] W. L. Johnson and A. Valente, "Tactical language and culture training systems: Using ai to teach foreign languages and cultures," pp. 72–83, 2009.

[8] J. Campbell, M. Core, R. Artstein, L. Armstrong, A. Hartholt, C. Wilson, K. Georgila, F. Morbini, E. Haynes, D. Gomboc, M. Birch, J. Bobrow, H. Lane, J. Gerten, A. Leuski, D. Traum, M. Trimmer, R. DiNinni, M. Bosack, T. Jones, R. Clark, and K. Yates, "Developing inots to support interpersonal skills practice." *In Proceedings of the Thirty-second Annual IEEE Aerospace Conference*, pp. 1–14, 2011.

[9] J. Silva, L. Coheur, A. C. Mendes, and A. Wichert, "From symbolic to subsymbolic information in question classification," *Artificial Intelligence Review*, vol. 35, no. 2, pp. 137–154, 2011.

[10] L. Blackhall, J. Owen, A. Chapin, S. Thomas *et al.*, "Development of collaborative behaviors objective assessment tools (poster presentation)," *Collaborating Across Borders IV*, 2013.

[11] V. Brashers, J. Owen, L. Blackhall, J. Erickson, and C. Peterson, "A program design for full integration and assessment of clinically relevant interprofessional education into the clinical/clerkship year for nursing and

medical students," *Journal of Interprofessional Care*, vol. 26, no. 3, pp. 242–244, 2012.

[12] D. Marshall, P. Hall, and A. Taniguchi, "Team osces: evaluation methodology or educational encounter?" *Medical education*, vol. 42, no. 11, pp. 1129–1130, 2008.

[13] J. Owen, T. Brashers, C. Peterson, L. Blackhall, and J. Erickson, "Collaborative care best practice models: A new educational paradigm for developing interprofessional educational (ipe) experiences," *Journal of interprofessional care*, vol. 26, no. 2, pp. 153–155, 2012.

[14] I. Symonds, L. Cullen, and D. Fraser, "Evaluation of a formative interprofessional team objective structured clinical examination (itosce): A method of shared learning in maternity education," *Medical Teacher*, vol. 25, no. 1, pp. 38–41, 2003.

[15] J. D. Williams, "A case study of applying decision theory in the real world: Pomdps and spoken dialog systems," *Decision Theory Models for Applications in Artificial Intelligence: Concepts and Solutions*, pp. 315–342, 2010.

[16] R. E. Schapire and Y. Singer, "Boostexter: A boosting-based system for text categorization," *Machine learning*, vol. 39, no. 2, pp. 135–168, 2000.

[17] D. R. Xin Li, "Learning question classifiers," *COLING'02*, 2002.

[18] R. Socher, A. Perelygin, J. Y. Wu, J. Chuang, C. D. Manning, A. Y. Ng, and C. Potts, “Recursive deep models for semantic compositionality over a sentiment treebank,” vol. 1631, p. 1642, 2013.

[19] J. Chung, Ç. Gülçehre, K. Cho, and Y. Bengio, “Empirical evaluation of gated recurrent neural networks on sequence modeling,” *CoRR*, vol. abs/1412.3555, 2014. [Online]. Available: http://arxiv.org/abs/1412.3555

[20] N. Kalchbrenner, E. Grefenstette, and P. Blunsom, “Convolutional neural network for modelling sentences.” *Modelling Sentences. In Proceedings of ACL 2014*, 2014.

[21] A. L. Gorin, G. Riccardi, and J. H. Wright, “How may i help you?” *Speech communication*, vol. 23, no. 1, pp. 113–127, 1997.

[22] P. Price, “Evaluation of spoken language systems: The atis domain,” in *Proceedings of the Third DARPA Speech and Natural Language Workshop*. Morgan Kaufmann, 1990, pp. 91–95.

[23] G. Tur and R. De Mori, *Spoken language understanding: Systems for extracting semantic information from speech*. John Wiley & Sons, 2011.

[24] G. Tur, L. Deng, D. Hakkani-Tür, and X. He, “Towards deeper understanding: Deep convex networks for semantic utterance classification,”

in *Acoustics, Speech and Signal Processing (ICASSP), 2012 IEEE International Conference on*. IEEE, 2012, pp. 5045–5048.

[25] T. Mikolov, M. Karafiát, L. Burget, J. Cernockỳ, and S. Khudanpur, "Recurrent neural network based language model." in *INTERSPEECH*, vol. 2, 2010, p. 3.

[26] R. Collobert, J. Weston, L. Bottou, M. Karlen, K. Kavukcuglu, and P. Kuksa, "Natural language processing (almost) from scratch." *Journal of Machine Learning Research*, pp. 2493–2537, 2011.

[27] R. Collobert and J. Weston, "A unified architecture for natural language processing: Deep neural networks with multitask learning," in *Proceedings of the 25th international conference on Machine learning*. ACM, 2008, pp. 160–167.

[28] R. Johnson and T. Zhang, "Effective use of word order for text categorization with convolutional neural networks," *arXiv preprint arXiv:1412.1058*, 2014.

[29] Y. Goldberg, "A primer on neural network models for natural language processing," *arXiv preprint arXiv:1510.00726*, 2015.

[30] Y. LeCun, S. Chopra, and R. Hadsell, "A tutorial on energy-based learning," 2006.

[31] Y. LeCun and F. J. Huang, “Loss functions for discriminative training of energy-based models.” in *AISTATS*, 2005.

[32] L. Bottou, “Stochastic gradient descent tricks,” in *Neural Networks: Tricks of the Trade*. Springer, 2012, pp. 421–436.

[33] J. Pennington, R. Socher, and C. D. Manning, “Glove: Global vectors for word representation.”

[34] T. Mikolov, I. Sutskever, K. Chen, G. Corrado, and J. Dean, “Distributed representations of words and phrases and their compositionality,” *CoRR*, vol. abs/1310.4546, 2013. [Online]. Available: http://arxiv.org/abs/1310.4546

[35] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, “Gradient-based learning applied to document recognition,” *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.

[36] W. Yih, X. He, and C. Meek, “Semantic parsing for single-relation question answering.” *In Proceedings of ACL 2014*, 2014.

[37] Y. Shen, X. He, J. Gao, L. Deng, and G. Mesnil, “Learning semantic representations using convolutional neural networks for web search,” *In Proceedings of WWW 2014*, 2014.

[38] A. Razavian, H. Azizpour, J. Sullivan, and S. Carlsson, “Cnn features off-the-shelf: an astounding baseline for recognition,” 2014.

[39] W. Ling, T. Luís, L. Marujo, R. F. Astudillo, S. Amir, C. Dyer, A. W. Black, and I. Trancoso, “Finding function in form: Compositional character models for open vocabulary word representation,” *CoRR*, vol. abs/1508.02096, 2015. [Online]. Available: http://arxiv.org/abs/1508.02096

[40] P. J. Werbos, “Backpropagation through time: what it does and how to do it,” *Proceedings of the IEEE*, vol. 78, no. 10, pp. 1550–1560, 1990.

[41] S. Hochreiter and J. Schmidhuber, “Long short-term memory,” *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[42] J. Chung, C. Gulcehre, K. Cho, and Y. Bengio, “Empirical evaluation of gated recurrent neural networks on sequence modeling,” *arXiv preprint arXiv:1412.3555*, 2014.

[43] M. Schuster and K. K. Paliwal, “Bidirectional recurrent neural networks,” *Signal Processing, IEEE Transactions on*, vol. 45, no. 11, pp. 2673–2681, 1997.

[44] R. Jozefowicz, W. Zaremba, and I. Sutskever, “An empirical exploration of recurrent network architectures,” in *Proceedings of the 32nd International Conference on Machine Learning (ICML-15)*, 2015, pp. 2342–2350.

[45] C. Kenaszchuk, K. MacMillan, M. van Soeren, and S. Reeves, "Interprofessional simulated learning: short-term associations between simulation and interprofessional collaboration," *BMC medicine*, vol. 9, no. 1, p. 1, 2011.

[46] "Joint commission and sentinel events." [Online]. Available: http://www.jointcommission.org/sentinel_event.aspx

[47] E. Knebel, A. C. Greiner *et al.*, *Health Professions Education:: A Bridge to Quality*. National Academies Press, 2003.

[48] P. G. Baker *et al.*, "Framework for action on interprofessional education and collaborative practice," 2010.

[49] D. S. Freeth, M. Hammick, S. Reeves, I. Koppel, and H. Barr, *Effective interprofessional education: development, delivery, and evaluation*. John Wiley & Sons, 2008.

[50] J. H. V. Gilbert, "Interprofessional learning and higher education structural barriers," *Journal of Interprofessional Care*, 2005.

[51] S. Reeves, J. Goldman, A. Burton, and B. Sawatzky-Girling, "Synthesis of systematic review evidence of interprofessional education," *Journal of Allied Health*, vol. 39, no. Supplement 1, pp. 198–203, 2010.

[52] A. Hartholt, D. Traum, S. C. Marsella, A. Shapiro, G. Stratou,

A. Leuski, L.-P. Morency, and J. Gratch, “All Together Now: Introducing the Virtual Human Toolkit,” in *13th International Conference on Intelligent Virtual Agents*, Edinburgh, UK, Aug. 2013. [Online]. Available: http://ict.usc.edu/pubs/All%20Together%20Now.pdf

[53] D. Kolb, “Experiential learning. englewood cliffs,” *NJ: Prentice Hall*, 1984.

[54] C. Baker, C. Pulling, R. McGraw, J. D. Dagnone, D. Hopkins-Rosseel, and J. Medves, “Simulation in interprofessional education for patient-centred collaborative care,” *Journal of Advanced Nursing*, vol. 64, no. 4, pp. 372–379, 2008.

[55] J. T. Paige, V. Kozmenko, T. Yang, R. P. Gururaja, C. W. Hilton, I. Cohn, and S. W. Chauvin, “High-fidelity, simulation-based, interdisciplinary operating room team training at the point of care,” *Surgery*, vol. 145, no. 2, pp. 138–146, 2009.

[56] K. K. P. Chang, J. W.-Y. Chung, and T. K. S. Wong, “Learning intravenous cannulation: a comparison of the conventional method and the cathsim intravenous training system,” *Journal of Clinical Nursing*, vol. 11, no. 1, pp. 73–78, 2002.

[57] “Medicines in development copd,” *America’s Biopharmaceutical Reseacrch Companies*, 2012.

[58] "Southeastern and mid-atlantic states have highest rate of sepsis-related deaths," *Rockville, MD2010*, 2010.

[59] D. Traum, "Practical language processing for virtual humans," *IAAI Conference*, 2010.

[60] K. Johnsen, A. Raij, A. Stevens, D. S. Lind, and B. Lok, "The validity of a virtual human experience for interpersonal skills education," in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, ser. CHI '07. New York, NY, USA: ACM, 2007, pp. 1049–1058. [Online]. Available: http://doi.acm.org/10.1145/1240624.1240784

[61] B. Morshedi, A. V. Chaet, C. Brown, G. Arroyo, S. Proctor, R. Valdez *et al.*, "User preferences influencing the design of a tailored virtual patient educator in a latina farm worker community," *American Medical Informatics Association Annual Symposium*, 2014.

[62] D. Sonntag, "Towards combining finite state, ontologies, and data driven approaches to dialogue management for multimodal question answering," *Information Society Language Technologies Conference*, 2006.