Installing, configuring, testing and validating three neural dialogue models

Haleema Sheraz
Stefan C. Kremer
Department of Computing and Information Science
University of Guelph

Abstract

Recent advancements in neural dialogue systems struggle to develop a dialogue agent that can communicate naturally with humans. This is due to the system's inability to model the underlying intention. The article titled "Latent Dialogue Model" proposes a model that outperforms all previous models to date by constructing a discrete latent variable to determine the truth behind each dialogue. To explore the properties of the model, the system assists the users to find a restaurant in the Cambridge, UK area. In this paper, I analyze three dialogue models, two of which are widely used, and one being a recent development. The *vanilla neural dialogue model* and the *attention-based neural dialogue model* have been widely applied in previous work, whereas the *latent intention dialogue model* is a new proposition. The conclusion that the *latent intention dialogue model* is a better suited model was determined through training, validating and testing. The probability of success was acquired based upon the metric of each model in which the *latent intention dialogue model* proved most effective. The conclusion of this paper validates the authors' claim that the *latent intention dialogue model* is a practicable approach towards human-computer dialogue.

1 Introduction

Human-computer communication is no longer a question of the past but instead is the reality of the world today. Current developments in dialogue systems have resulted in online technical support services, virtual assistant such as Siri, Alexa, Cortana or even a translation service such as Google Translate. What all these examples have in common is dialogue. Wen et al. (2017a) defines dialogue "as a sequence to sequence mapping problem augmented with a dialogue history and the current data-base search outcome". Hence, a dialogue system can then be summarized as a technique used to process dialogue. Dialogue systems can be divided in to two categories: (1) task-oriented systems and (2) non-task-oriented systems (Chen et al., 2017). The aim of a task-oriented system is to achieve certain tasks (e.g. booking reservations, technical support service) whereas a non-task-oriented system is designed to simulate a conversation with human users with no clear purpose (e.g. chatbots) (Chen et al., 2017). In specific, the pipeline for a task-oriented system is complex in nature which has several limitations (Liu et al., 2018). Firstly, errors made in the upper stream modules can affect the downstream component due to credit assignment (Liu et al., 2018). Not only this, but also noting that as each preceding component is updated in the pipeline as the data is re-trained which causes errors (Liu et al., 2018). To rectify this problem an end-to-end system is suggested which tracks the dialogue state, interfaces with the knowledge base and incorporates structured query results into a system response (Liu et al., 2018). Basically, each system module is trainable from data except for the database operator. For this paper, the objective will be to highlight the intentions of an end-to-end method in a task-oriented system also known as the goal-oriented dialogue to model a latent variable.

Conventional models such as the *vanilla neural dialogue model* and *the attention-based neural dialogue model* are epitomes of an end-to-end model with a task-oriented system. These models tend to provide relevant and appropriate responses at each turn but are unable to generate casual yet diverse responses (Wen et al., 2017a). They also tend to overfit due to the lack of training data present for a goal-oriented dialogue which prevents the model from learning effective and scalable interactions (Wen et al., 2017a). Here the latent intention dialogue model is proposed to eliminate the errors above by composing appropriate responses.

The latent intention dialogue model uses an end-to-end architecture with reinforcement learning as the baseline (Wen et al., 2017a). The end-to-end model employs a discrete latent variable to learn underlying intentions in a dialogue. This helps by guiding the system in generating a natural language response. Using reinforcement learning over the variational inference with a discrete latent variable reduces high variance during sampling (Wen et al., 2017a). This can produce interpretable latent distributions. In the neural variational inference framework, an inference network is constructed that approximates the posterior distribution over the latent intention (Wen et al., 2017a). The intention distribution is then generated by optimising the variational lower bound of the sampled intentions, based on each response. In order to lower the variance further, clustering techniques are used to generate the labels of intentions. Later, the latent intention distribution can be learned in a semi-supervised manner.

In this paper, I evaluate how the *latent intention dialogue model* compares to the *vanilla neural dialogue model* and the *attention-based neural dialogue model*. The first step towards achieving this was to install the necessary applications required to run each model. The next step was research. In order to thoroughly comprehend the model and architecture, prior knowledge of each model was needed. This process was time consuming as it required constant referencing of various articles (see Figure 1A). The results of this paper verify that for corpus-based evaluations, the use of a latent intention variable outperforms all previously published work. The *latent intention dialogue model* attempts to interpret intentions from data by becoming a decision-making basis of a dialogue agent which then improves the conversational strategy maximizing reward.

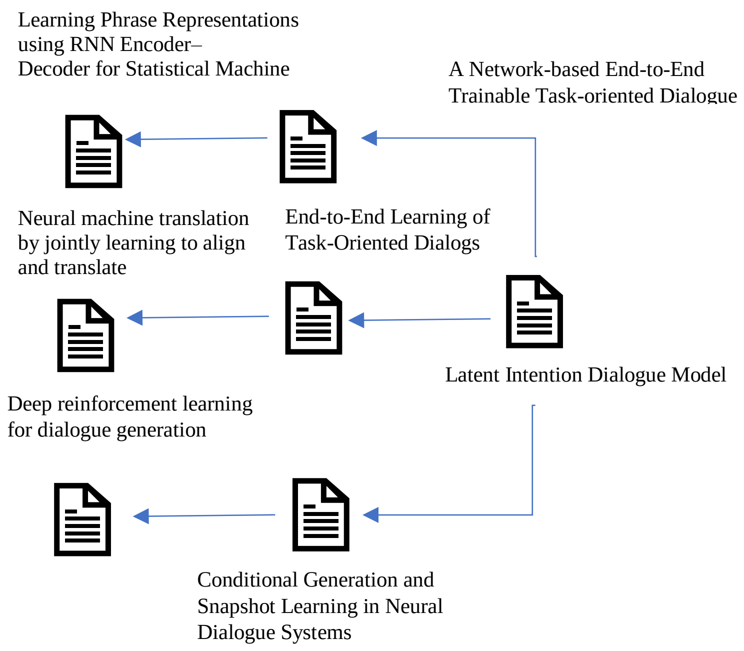

Learning Phrase Representations using RNN Encoder–Decoder for Statistical Machine
A Network-based End-to-End Trainable Task-oriented Dialogue
Neural machine translation by jointly learning to align and translate
End-to-End Learning of Task-Oriented Dialogs
Latent Intention Dialogue Model
Deep reinforcement learning for dialogue generation
Conditional Generation and Snapshot Learning in Neural Dialogue Systems

Figure 1A: The original article was based upon previous work.

2 Task oriented dialogue system

There are two different approaches to a task-oriented dialogue system: (1) pipeline and (2) end-to-end. Since the neural dialogue model is embedded with the end-to-end approach, this paper will summarize the end-to-end method.

2.1 End-to-end method

End-to-end is defined as a method where gradient based learning is applied to a system

(Cho et al., 2014). In a traditional pipeline, dialogue goes through a series of four steps before it can output a response. The four components are as follows: 1) Language understanding 2) dialogue state tracker 3) dialogue policy learning and 4) natural language generation. Instead of using these four components, an end-to-end model uses all modules and treats it as one. Each module is used to interact with the system (Wen et al., 2016). The end-to-end method outperforms the traditional pipeline by bypassing the intermediate steps, such as aligning system optimization targets (Cho et al.,2014). It takes the input sequence and querys the output sequence without having to go through each module. Wen et al., (2017b) introduced a neural dialogue model that is trainable end-to-end. More details regarding the model will be discussed in detail below.

2.2 Encoder-Decoder Model

In the past, an encoder-decoder model was introduced to address the sequence-to-sequence learning for machine translation where the input sequence differs from the output sequence. This model is a recurrent neural network (RNN) that is comprised of two sub-models: an encoder and a decoder (Cho et al., 2014). First, the model *encodes* the input sequence into a fixed length vector called the context vector *c* and then it *decodes* the context vector into the desired output sequence (Cho et al., 2014). The encoder reads an input sentence also known as the sequence vector $x = (x_1, x_2, .., x_T)$ into a vector *c* (Figure 1), which is a summary of the whole input sentence, and as it reads each sequence, the hidden state of the RNN changes with respect to each input (Cho et al., 2014). For example, given an input *Can I have Indian food*? the sentence will be encoded, and each word will represent a value in vector x. Bahdanau et al. (2015) describes that since the most common approach is to use an RNN, the hidden state can be described using Equation 1 and as per c can be described as using Equation 2

$$h_{\langle t \rangle} = \mathrm{F}\left(h_{\langle t-1 \rangle}, x_t\right) \ (1)$$

$$c = q(\{h_1, .., h_T\}) \ (2)$$

where F and q are some nonlinear functions. The decoder generates the output sequence by being trained rigorously to predict the next word y_t , given the context vector *c* and all the previously predicted words $\{y_1, y_2, .., y_{t-1}\}$ (Bahdanau et al., 2015). Since both y_t and $h_{\langle t \rangle}$ are conditioned on y_{t-1} and *c*, Bahdanau et al. (2015) defines the hidden state as

$$h_{\langle t \rangle} = \mathrm{F}\left(h_{\langle t-1 \rangle}, y_{t-1}, c\right) \ (3)$$

whereas the conditional probability is

$$\mathcal{P}(y_t | \{y_1, y_2, .., y_{t-1}\}, c) = g\left(h_{\langle t \rangle}, y_{t-1}, c\right)$$

(4)

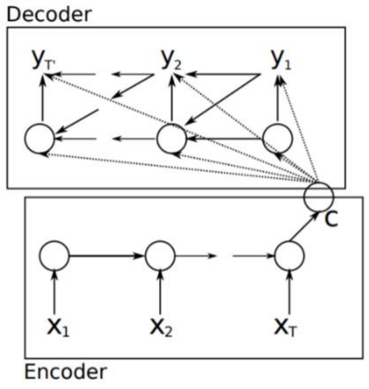


Figure 1: Encoder-Decoder Model from: Cho et al., (2014). “Figure 1: An illustration of the proposed RNN Encoder–Decoder”. *Learning Phrase Representations using RNN Encoder–Decoder for Statistical Machine*

Here g is a nonlinear, potentially multilayered, function that outputs the probability of y_t and $h_{\langle t \rangle}$ in the hidden state (Bahdanau et al., 2015). The difference between g and F is such that F uses an LSTM where as g could use any activation function.

3 Vanilla Neural Dialogue Model (NDM)

An end-to-end trainable neural dialogue model, vanilla neural dialogue model (NDM), is described as a neural network with a task-oriented dialogue system (Wen et al., 2016). Wen et al. (2016) suggests that an end-to- end neural dialogue system can be represented using Figure 2. Here dialogue is treated as a "sequence to sequence mapping problem" where each input sequence goes through the intent network, belief tracker, database operator and policy network, also known as encoder modules, to provide the user with the desired output in the generation network, decoder module.

3.1 Encoder Module

As discussed above, this module *encodes* the text. Each input sequence is converted into a fixed vector (see definition above) which the system can then action. This system action can then be represented by m_t.

3.1.1 Intent Network

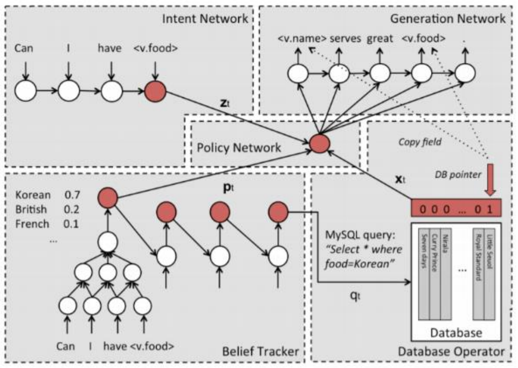


Figure 2: the flow of the framework within a dialogue system from: Wen et al., 2016. "Figure 1: The proposed end-to-end trainable dialogue system framework" *A Network-based End-to-End Trainable Task-oriented Dialogue System*

The intent network takes the input sentence and encodes it with a sequence of input tokens, w_0^t, $w_1^t, .., w_N^t$, which is then distributed into a vector z_t, as seen in Figure 2, at every step t (Wen et al., 2016). Wen et al. (2016) explains that either one can use a long short-term memory (LSTM) network which can be represented by

$$z_t = z_t^N = \text{LSTM}(w_0^t, w_1^t, .., w_N^t) \quad (5)$$

or an alternative method of using a convolutional neural network (CNN) can be used

$$z_t = \text{CNN}(w_0^t, w_1^t, .., w_N^t) \quad (6)$$

3.1.2 Belief Trackers

Considered the core of a task-oriented spoken dialogue system, the neural dialogue system keeps track of the users request by using a set of slot-based belief trackers (Wen et al., 2016). It not only assists to avoid unnecessary long-term dependencies from the input sequence, but it also minimizes the dataset required to train a model (Wen et al., 2016). The focus of the belief tracker is to maintain a multinomial distribution p over values $v \in \mathcal{V}_s$ for each informable slot s, and a binary distribution for each requestable slot (Wen et al., 2016). Here each slot is represented by an ontology $\mathbb{G}$ and each slot has a designated tracker that can be represented by a Jordan-type RNN with a CN-N feature extractor (Wen et al., 2016). $\mathbb{G}$ is a small knowledge graph defining the slot value pairs the system can talk about for a specific task (Wen et al., 2016). This can be seen in Figure 2. The belief state is known as the probability distribution p_s^t of the system. This along with z_t is the system's understanding of the user's request up to turn *t*. Wen et al., (2016) uses the following equation to represent p_s^t

$$p_s^t = \frac{exp(g_v^t)}{exp(g_{0,s}) + \sum_{v' \in V_{,s}} exp(g_{v'}^t)} \quad (7)$$

where $g_v^t = w_s \cdot sigmoid(\text{W}_s f_v^t + b_s) + b_s'$
$$f_v^t = f_{v,cnn}^t \oplus p_v^{t-1} \oplus p_0^{t-1}$$

and

$$f_{v,cnn}^{t} = CNN_{s,v}^{(u)}(u_{\mathrm{t}}) \oplus CNN_{s,v}^{(m)}(m_{\mathrm{t}-1}) \quad (8)$$

Here f_v^t represents the features used when updating each pre-softmax activation g_v^t for a given slot s (Wen et al., 2016). w_s is a vector, $\mathrm{W_s}$ is a matrix, b_s and b_s' are bias terms, and $g_{0,s}$ is a scalar parameter (Wen et al., 2016). p_0^t is the probability that the users have not mentioned that slot up to turn t (Wen et al., 2016). This can be represented by Figure 3 where f_v^t is the concatenation of two CNN derived features of which one is from processing the user's input u_{t} at turn t and $m_{\mathrm{t}-1}$ is the machine response at turn t-1 s (Wen et al., 2016).

3.1.3 Database Operator

Given that the belief state p_s^t is calculated, Wen et al. (2016) suggests that the DB query q_{t} is formed by taking the maximum value of each informable slot

$$q_{\mathrm{t}} = \bigcup_{s' \in S_{\mathrm{I}}} \left\{ \underset{v}{\operatorname{argmax}} \, p_{s'}^t \right\} \quad (9)$$

where S_{I} is defined as the set of informable slots. Next, a vector x_{t} which defines the degree of match within the database is produced by counting the number of matching entries and expressing them as a 6-bin 1-hot encoding (Wen et al., 2016). This can be seen in Figure 2 in the Database Operator section where the 6-bin 1-hot encoding is represented by the term DB pointer.

3.1.4 Policy Network

The policy network combines the vectors z_t, p_s^t and x_{t} into a single action vector $m_t$by a three-way matrix transformation

$$m_t = \tanh\left(\mathrm{W}_{zm} z_t + \mathrm{W}_{xm} x_t + \sum_{s \in \mathbb{G}} \mathrm{W}_{pm}^s p_t^s \right) \quad (10)$$

where $\mathrm{W}_{zm}, \mathrm{W}_{pm}^s$ and W_{xm} are parameters and $\mathbb{G}$ is the domain ontology (Wen et al., 2016).

3.2 Decoder Module

This module *decodes* the output sequence. As seen in section 2 the decoder module uses LS-TM LM to generate the desired sequence output (Wen et al., 2016). The result of this is the actual values of the database entries being substituted with skeletal, slots, sentence structure (Wen et al., 2016).

3.2.1 Generation Network

Long short-term memory models are recurrent neural networks that are capable of learning long-term dependencies (Wen et al., 2016). The generation network is where the output is derived based upon the user's given input. It decodes responses from the action vector m_t. It uses a language probability to determine the template-like sentence token, i.e replace the values of each input with generic tokens for example keywords like *Indian* would be replaced by <v.food> (Wen et al., 2016). Wen et al. (2016) states that this probability can be represented by

$$P\left(w_{j+1}^t \middle| w_j^t, h_{j-1}^t, m_t\right) = LSTM_j\left(w_j^t, h_{j-1}^t, m_t\right) \quad (11)$$

where the h_{j-1}^t is the hidden layer, w_j^t is the last output token and $LSTM_j$ is the conditional LSTM operator for one output step j. As seen in Figure 2 Generation Network, there are three generational architectures that can be used for LSTM:

1 Language Model Type
2 Memory Type
3 Hybrid Type

The aim of a generation network is to replace the generic token with their actual values.

3.2.1.1 Language Model Type

The language model type (LM) can be defined using the following equation:

$$\begin{pmatrix} \mathbf{i}_j \\ \mathbf{f}_j \\ \mathbf{o}_j \\ \hat{\mathbf{c}}_j \end{pmatrix} = \begin{pmatrix} \text{sigmoid} \\ \text{sigmoid} \\ \text{sigmoid} \\ \tanh \end{pmatrix} \mathbf{W}_{4n,3n} \begin{pmatrix} \mathbf{m}_t \\ \mathbf{w}_j \\ \mathbf{h}_{j-1} \end{pmatrix} \quad (12)$$

$$\boldsymbol{c_j = f_j \odot c_{j-1} + i_j \odot \hat{c}_j} \ (13)$$

$$h_j = o_j \odot \tanh(c_j) \ (15)$$

The conditioning vector (m_t) is merged together with the input word (w_j) and hidden layer (h_{j-1}).. j is represented as the generating step, n is the hidden layer size, i_j , f_j, o_j are input, forget and output gates respectively. $\hat{c}_j$ and c_j are proposed cell values and true cell value at step j, and $W_{4n,3n}$ are the model parameters (Wen et al., 2016). Equation 12 represents the compact forms for a forward pass of an LSTM unit with a forget gate. Figure 4a represents the model discussed.

3.2.1.2 Memory Type

The memory type (mem) shown in Figure 4b, in which the conditioning vector m_t is governed by a standalone reading gate $\boldsymbol{r_j}$ (Wen et al., 2016). This reading gate decides how much information should be read from the conditioning vector and directly writes it into the memory cell c_j, where W_c is another weight matrix to learn. $\hat{c}_j$ is the true cell value of the output response in the generation network. Equation 16 represents the compact forms for a forward pass of an LSTM unit with a forget gate. The idea behind this is that the model isolates the conditioning vector from the LM so that the model has more flexibility to learn to trade off between the two (Wen et al., 2016).

$$\begin{pmatrix} \mathbf{i}_j \\ \mathbf{f}_j \\ \mathbf{o}_j \\ \mathbf{r}_j \end{pmatrix} = \begin{pmatrix} \text{sigmoid} \\ \text{sigmoid} \\ \text{sigmoid} \\ \text{sigmoid} \end{pmatrix} \mathbf{W}_{4n,3n} \begin{pmatrix} \mathbf{m}_t \\ \mathbf{w}_j \\ \mathbf{h}_{j-1} \end{pmatrix} \quad (16)$$

$$\hat{c}_j = \tanh\left(W_c(w_j \oplus h_{j-1})\right) \ (17)$$

$$c_j = \boldsymbol{f_j} \odot \boldsymbol{c_{j-1}} + \boldsymbol{i_j} \odot \boldsymbol{\hat{c}_j} + \boldsymbol{r_j} \odot \boldsymbol{m_t} \ (18)$$

$$h_j = o_j \odot \tanh(c_j) \ (19)$$

3.2.1.3 Hybrid Type

The hybrid type network (Figure 4c) states that long term dependency is not needed for the conditioning vector since the information is applied at every step j (Wen et al., 2016). Equation 20 represents the compact forms for a forward pass of an LSTM unit with a forget gate.

$$\begin{pmatrix} \mathbf{i}_j \\ \mathbf{f}_j \\ \mathbf{o}_j \\ \mathbf{r}_j \end{pmatrix} = \begin{pmatrix} \text{sigmoid} \\ \text{sigmoid} \\ \text{sigmoid} \\ \text{sigmoid} \end{pmatrix} \mathbf{W}_{4n,3n} \begin{pmatrix} \mathbf{m}_t \\ \mathbf{w}_j \\ \mathbf{h}_{j-1} \end{pmatrix}$$

$$(20)$$

$$\hat{c}_j = \tanh\left(W_c(w_j \oplus h_{j-1})\right) \ (21)$$

$$c_j = \boldsymbol{f_j} \odot \boldsymbol{c_{j-1}} + \boldsymbol{i_j} \odot \boldsymbol{\hat{c}_j} \ (22)$$

$$h_j = o_j \odot \tanh(c_j) + \boldsymbol{r_j} \odot \boldsymbol{m_t} \ (23)$$

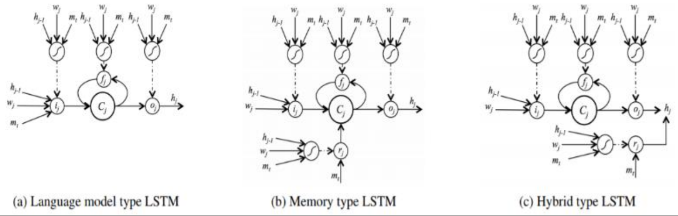


Figure 4: conditional generation architectures: Wen et al., 2016. "Figure 1: Three different conditional generation architectures." *Conditional Generation and Snapshot Learning in Neural Dialogue Systems*

3.3 Snapshot learning

Another method to understand is the snapshot learning. Snapshot learning creates a vector of binary labels as the snapshot of the remaining part of the output sentence from a generation step (Wen et al., 2016). It is used when learning conditional generational models from sequential supervision signals become difficult to understand (Wen et al., 2016). The future event can be predicted by each element in a snap-shot vector which can be obtained from the system response or dialogue context at training time (Wen et al., 2016). Each individual snapshot vector has an indicator function which is depended on a certain event taking place. This can be obtained by the systems response or the dialogue context at training time (Wen et al., 2016). The indicator functions have two forms: (1) whether a slot value (eg. *[v.name]*) is going to occur, and (2) whether the system has offered a venue (Wen et al., 2016). This can be seen in Figure 5. The system then checks for whether the slot value is

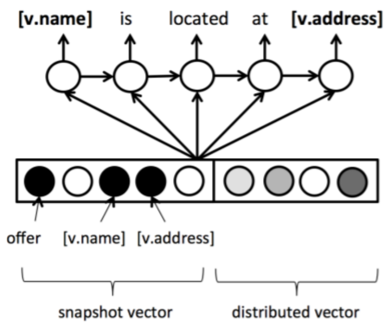


Figure 5: Snapshot vector with slots values present in the response: Wen et al., 2016. "Figure 2: The idea of snapshot learning. The snapshot vector was trained with additional supervisions on a set of indicator functions heuristically labelled using the system response." *Conditional Generation and Snapshot Learning in Neural Dialogue Systems*

in the entire dialogue. If so, every label for each turn is labelled with 1 and if not then it is labelled with a 0.

4 Attention-based model

Attention is defined as a method to both *align* and *translate* (Bahdanau et al., 2015). This model is a mixture of a bidirectional RNN encoder, and a decoder that can search through a long sentence (Bahdanau et al., 2015). The model predicts the output word by soft searching the set of position within the source sentence where the information is found (Bahdanau et al., 2015). The target word is based on the context vector associated with the source position (Bahdanau et al., 2015). The conditional probability can be defined as

$$\mathcal{P}(y_i|\{y_1, y_2, .., y_{i-1}\}, x) = g(h_{\langle i \rangle}, y_{i-1}, c_i) \quad (24)$$

where $h_{\langle i \rangle}$ is the RNN hidden state for time *i*

$$h_{\langle i \rangle} = \mathrm{F}(h_{\langle i-1 \rangle}, y_{i-1}, c_i) (25)$$

The encoder computes the *annotation* $(h_1, .., h_T)$ of each input (Bahdanau et al., 2015). The information regarding each source sentence can be determined by each annotation h_i (Bahdanau et al., 2015). Here, the context vector c_i can be defined by taking the sum of the annotations h_i:

$$c_i = \sum_{j=1}^{T} \propto_{ij} h_j \ (26)$$

The weight $\propto_{ij}$of each annotation can be computed by

$$\propto_{ij} = \frac{exp(e_{ij})}{\sum_{k=1}^{T} exp(e_{ik})} (27)$$

Here, e_{ij} is the *alignment* which determines how well the input at position *j* and the output at position *i* match (Bahdanau et al., 2015)

$$e_{ij} = a(h_{i-1}, y_{i-1}, c_i) \ (28)$$

All of this takes place during encoding, once the *alignment* is complete the process of *translation* can take place. As discussed above, the result of the decoding model is to generate the output sequence given the context vector or, in this case, would be to create a *translation*. In an attention-based model, the notion of an attention mechanism is applied in the decoder. During the process of decoding, the decoder decides on what part of the sentence to pay attention to (Bahdanau et al., 2015). By doing this, the encoder no longer has the pressure to compress the input sentence into a fixed-length vector (Bahdanau et al., 2015). Using this allows the model to spread the information throughout the annotations, which can be selectively retrieved by the decoder (Bahdanau et al., 2015).

4.1 Attentive Generation Network

An action-based mechanism is used to aggregate source embedding at each output step j (Wen et al., 2016). Wen et al. (2016) further points out, that this mechanism can be summarized using the following equation

$$m_t^j = \tanh\left(\mathrm{W}_{zm} z_t + \mathrm{W}_{xm} x_t + \hat{p}_t^{(j)}\right) \quad (27)$$

$$\widehat{p_t^{(j)}} = \sum_{s \in \mathbb{G}} \alpha_s^{(j)} \tanh\left(W_{pm}^s \widehat{p_s^t}\right) \quad (28)$$

and where the attention weights $\alpha_s^{(j)}$ can be calculated and $\mathrm{u}_t$can be defined as

$$\alpha_s^{(j)} = softmax(r^T \tanh(\mathrm{W}_r \cdot \mathrm{u}_t)) \quad (29)$$

$$\mathrm{u}_t = z_t \oplus x_t \oplus \widehat{p_t^s} \oplus\ w_j^t \oplus h_{j-1}^t \quad (30)$$

4.2 Attention-based neural dialogue model

An attention-based neural dialogue model is an end-to-end neural dialogue model that uses attention mechanisms in the generation network. As mentioned above in Section 3, with an attention mechanism we no longer encode the input sentence into a fixed vector but rather *attend* to various parts of the sentences. The model learns what part of a sentence to pay *attention* to and what parts to ignore. This mechanism is applied to a neural dialogue system which produces an effective approach to aggregating various sources for the purpose of prediction (Wen et al., 2017b). Here the attention mechanism is combined with the tracker belief state to produce a policy network with the following equation

$$m_t^j = \tanh\left(\mathrm{W}_{zm} z_t + \mathrm{W}_{xm} x_t + \sum\nolimits_{s \in \mathbb{G}} \alpha_s^j \mathrm{W}_{pm}^s p_t^s\right)$$

$$(31)$$

with the attention weights α_s^j calculated as

$$\alpha_s^j = \mathrm{softmax}\left(\mathrm{r}^{\mathrm{T}} \tanh\left(\mathrm{W}_r \cdot \left(v_t \oplus p_t^s \oplus\ w_j^t \oplus h_{j-1}^t\right)\right)\right)$$

$$(32)$$

Here $v_t = z_t + x_t$ and matrix W_r and vector r are parameters to learn (Wen et al., 2017b). As seen in Figure 2, the dialogue system framework is comprised of five different components: (1) intent network (2) belief tracker (3) database-se operator (4) policy network and (5) generation network (Wen et al., 2016). As summarized above, the outcome of the encoder module is a policy network m_t^j. This policy network is then used to generate the output sequence in the generation network. In Section 3 the definition of an attention-based model was provided where the model was defined as *aligning* and *translating.* This approach is applied to the policy network. To further understand the equation above, the next step will be to break down the equations into their respective parts. The policy network m_t^j is a three-way matrix transformation where W_{zm} and W_{xm} are matrix parameters (Wen et al., 2016). $\mathrm{W}_{zm} z_t$ is the combination of the intent network z_t with the matrix parameter W_{zm}. Here, z_t is characterized by a LSTM network instead of a CNN. Since the $\mathrm{W}_{xm} x_t$ is the join between the vector x_t generated by the data-base operator, the vector representation of x_t is the 6-bin 1-hot encoding where each match is denoted by 1 and W_{xm} is a matrix parameter associated to the vector x_t.Wen et al. (2016) describes $\sum_{s \in \mathbb{G}} \alpha_s^j \mathrm{W}_{pm}^s p_t^s$ as the concatenation of all the summary belief vectors for a given ontology $\mathbb{G}$. The attention weight α_s^j is calculated in Equation 32. The intention network z_t and database operator x_t are summed to vector v_t .This is then concatenated with the belief state p_t^s, the generic name associated to the output w_j^t and the hidden layer h_{j-1}^t. Matrix W_r and vector r are parameters to learn. The analogue of tan, tanh, is then taken which is then applied to the softmax function to be interpreted as a probability. Considering this, the notion of attention is seen here as the mechanism is used to judge the input sequence, what values to pay attention to, instead of treating each value in the input sequence as relevant information.

5 Latent Intention Dialogue Model

A latent intention dialogue model is a dialogue system that utilizes a latent variable to acquire the underlying intention behind each dialogue (Wen et al., 2017a). It uses end-to-end system architectures and consists of three components. Wen et al. (2017a) defines these three components as: (1) Representation Construction; (2) Policy Network; and (3) Generator. This can be seen in Figure 6. The latent variable is considered the hub of decision making for a dialogue agent. In order to interpret the intention behind each dialogue, Wen et al. (2017a) proposed that a neural variational inference framework is used which will learn in a semi-supervised manner. He further suggested to apply reinforcement learning to the observed policy gradient which allows the system to improve itself based upon each interaction.

5.1 Representation Construction

In the representation construction the intention of the dialogue is summarized to match with the systems knowledge (Wen et al., 2017a). Wen et al. (2017a) defines the following dialogue state vector as

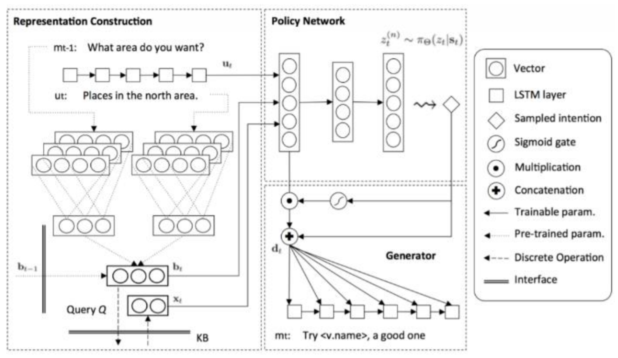


Figure 6 framework of the LIDM: Wen et al., 2017. "*Figure 1.* LIDM for Goal-oriented Dialogue Modeling" *Latent Intention Dialogue Model*

$$s_t = u_t \oplus b_t \oplus x_t \; (33)$$

Here u_t is the user's input and knowledge base KB, b_t is the belief vector and x_t is the database system vector. u_t is trainable from data and so, it encodes the user's utterance with a directional LSTM and concatenates it with the hidden states to form a distributed utterance representation (Wen et al., 2017a). This can then be represented using the following equation

$$u_t = \text{biLSTM}_\ominus(u_t) \; (34)$$

b_t is the belief tracker which can be defined as a set of probability distribution that is determined by using a set of pre-trained RNN-CNN belief trackers in which u_t, m_{t-1} and b_{t-1} are calculated using different CNNs

$$b_t = \text{RNN} - \text{CNN}(u_t, m_{t-1}, b_{t-1}) \; (35)$$

where m_{t-1} is the preceding machine response and b_{t-1} is the preceding belief vector (Wen et al., 2017a). Once the belief vectors are decided, the next step in the system is to create a query Q that takes the union of the maximum values of each slot (Wen et al., 2017a). This query is then used to create a vector x_t by searching the internal knowledge base to find matches (Wen et al., 2017a). The database operator from an end-to-end model can be seen here as the vector

x_t is expressed as a 6-bin 1-hot encoding where 1 is a match and 0 is no match.

5.1.1 Policy Network

z_t is used here to denote the latent intention and s_t is the state. z_t can be rep-resented using a discrete conditional probability distribution and is parameterized by a single layer MLP as seen below

$$\pi_{\ominus}(z_t|s_t) = softmax(W_2^T \cdot \tanh(W_1^T + b_1) + b_2) \quad (36)$$

where W_1, W_2, b_1 and b_2 are parameters. Sampled intention, action $z_t^{(n)}$, from reinforced learning can be calculated by taking the sampled conditional distribution

$$z_t^{(n)} \sim \pi_{\ominus}(z_t|s_t) \quad (37)$$

5.1.2 Generation Network

Once the latent intention is calculated, the next step is move on to the generation network. As seen in Figure 1, the result of a policy network is the conditional probability $z_t^{(n)}$. This distribution is then fed into the generation network which querys out the response. Wen et al. (2017a) suggested that the state vector s_t can be combined with the sampled intention $z_t^{(n)}$ to form vector d_t which can then be used to query out a response m_t. The following equation can be used to calculate d_t

$$d_t = W_4^T \oplus [sigmoid(W_3^T z_t + b_3) \cdot W_5^T s_t] \quad (38)$$

Here b_3 *and* $W_{3\sim5}$ are parameters, z_t is the 1-hot representation of $z_t^{(n)}$ and s_t is the state vector. The parameter W_4 is concatenated with the combination of the sigmoid of z_t and the state vector s_t to form the vector d_t. As described by Wen et al. (2017a), the output response can be represented using a conditional probability

$$p_{\ominus}(m_t|s_t) = \textstyle\sum_{z_t} p_{\ominus}(m_t|z_t, s_t)\, \pi_{\ominus}(z_t|s_t) \quad (39)$$

where $p_{\ominus}(m_t|z_t, s_t)$ is

$$p_{\ominus}(m_t|z_t, s_t) = \textstyle\prod_j p\left(w_{j+1}^t | w_j^t, h_{j-1}^t, d_t\right),$$

w_j^t the last output token, a word, and $h_{j-1}^t$is the last hidden state.

5.2 Inference

$q_{\phi}(z_t|m_t, s_t)$ is used to denote the inference network to approximate the posterior distribution $p(z_t|m_t, s_t)$ in order to optimize the variational lower bound of the joint probability (Wen et al., 2017a). The variational lower bound can be represented as,

$$\mathcal{L} = \mathbb{E}_{q_\phi(z_t)}[log p_{\ominus}(m_t|z_t, s_t)] - \lambda D_{KL}(q_\phi(z_t)||\pi_{\ominus}(z_t|s_t)) \leq log \Sigma p_{\ominus}(m_t|z_t, s_t)\pi_{\ominus}(z_t|s_t)_{z_t} = log p_{\ominus}(m_t|s_t) \quad (40)$$

where $q_\phi(z_t)$ is a short form for $q_\phi(z_t|m_t, s_t)$ and λ is a trade-off factor (Wen et al., 2017a).The inference network can then be calculated by

$$q_{\phi}(z_t|s_t, m_t) = Multi(o_t) = softmax(W_6 o_t) \quad (41)$$

$$o_t = MLP_{\phi}(b_t, x_t, u_t, m_t) \quad (42)$$

$$u_t = biLSTM_{\phi}(u_t), m_t = biLSTM_{\phi}(m_t) \quad (43)$$

where o_t is a joint representation and both u_t and m_t are modeled by a bidirectional LSTM network (Wen et al., 2017a). Keeping in mind here that $q_{\phi}(z_t|s_t, m_t)$ are samples processed during inference to compute the stochastic gradient where as $\pi_{\ominus}(z_t|s_t)$ produces the machine response (Wen et al., 2017a). Since $z_t^{(n)} \sim q_{\phi}(z_t|s_t, m_t)$, two different strategies are used to optimize the parameters ϴ and ϕ against the variational lower bound (Equation 8) (Wen et al., 2017). ϴ is divided into to two sets ϴ = $\{ϴ_1, ϴ_2\}$ where $ϴ_1$, the decoder, can be updated using the back-propagating gradient and $ϴ_2$, in the generative network, are updating by minimizing the KL divergence (Wen et al., 2017a). Equation 43 represents Θ_1 and Equation 13 represents Θ_2.

$$\frac{\partial \mathcal{L}}{\partial \Theta_1} = \mathbb{E}_{q_\phi(z_t|s_t,m_t)}\left[\frac{\partial logp_{\Theta}(m_t|z_t,s_t)}{\partial \Theta_1}\right] \approx \frac{1}{N}\sum_n \frac{\partial logp_{\Theta}\left(m_t|z_t^{(n)},s_t\right)}{\partial \Theta_1}$$

(44)

$$\frac{\partial \mathcal{L}}{\partial \Theta_2} = -\frac{\partial \lambda D_{KL}\left(q_\phi(z_t)||\pi_{\Theta}(z_t|s_t)\right)}{\partial \Theta_2} = \lambda \sum_{z_t} \frac{\partial log\pi_{\Theta}(z_t|s_t)}{\partial \Theta_2}$$

(45)

Moving onto to ϕ, the first step is to define the learning signal $r\left(m_t, z_t^{(n)}, s_t\right)$,

$$r(m_t, z_t^{(n)}, s_t) = \log p_{\Theta_1}(m_t|z_t^{(n)}, s_t) - \lambda\left(logq_\phi(z_t^{(n)}|s_t, m_t) - log\pi_{\Theta}(z_t^{(n)}|s_t)\right)$$

(45)

Then the parameter ϕ is updated by

$$\frac{\partial \mathcal{L}}{\partial \phi} = \mathbb{E}_{q_\phi(a_t|s_t,m_t)}\left[r(m_t,a_t,s_t)\frac{\partial logq_\phi(a_t|m_t,s_t)}{\partial \phi}\right]$$

$$\approx \frac{1}{N}\sum_n r\left(m_t, z_t^{(n)}, s_t\right)\frac{\partial logq_\phi\left(z_t^{(n)}|m_t,s_t\right)}{\partial \phi}$$

(46)

The result of the estimator (Equation 46) is a large variance. This is due to the learning signal $r\left(m_t, z_t^{(n)}, s_t\right)$ (Wen et al., 2017a). A solution to this problem is to apply the reinforcement algorithm. This uses two baselines, the centred learning signal b and the input dependent baseline $b(s_t)$ (Wen et al., 2017a). Here b is a learnable constant and $b(s_t) = MLP(s_t)$ (Wen et al., 2017). The equation below minimizes the distance while using the baselines $\mathcal{L}_b = \left[r\left(m_t, z_t^{(n)}, s_t\right) - b - b(s_t)\right]^2$ (47)

and the gradient estimator ϕ is defined as

$$\frac{\partial \mathcal{L}}{\partial \phi} \approx \frac{1}{N}\sum_n\left[r(m_t, z_t^{(n)}, s_t) - b - b(s_t)\right]\frac{\partial logq_\phi\left(z_t^{(n)}|m_t,s_t\right)}{\partial \phi} \quad (48)$$

5.5 Semi-supervised learning

Semi-supervised learning is a mixture of unsupervised and supervised learning. It uses both labelled and unlabelled data for training. Wen et al. (2017a) suggested that there are two major factors as to why a semi-supervised learning is used rather than unsupervised learning. The first reason being that despite taking the measures stated above, the large variance of the inference network makes it difficult to generate sensible intention samples in the early stages of training. The second reason being the disconnection phenomenon that is observed between the LSTM decoder and the rest of the components. This is due to the LSTM language model's strong discriminative power (Wen et al., 2017a). The equation used to define the semi-supervised learning is represented by the equation below where α controls the supervised and unsupervised examples

$$\mathcal{L}' = \alpha\mathcal{L}_1 + \mathcal{L}_2 \quad (49)$$

During pre-processing, standard clustering algorithms are used on the corpus to generate automatic labels $\widehat{z_t}$ for the training examples $(m_t, s_t, \widehat{z_t}) \in \mathbb{L}$. First, the model is optimized against the modified variational lower bound (Equation 19) by training the model on the unlabeled examples $(m_t, s_t) \in \mathbb{U}$

$$\mathcal{L}_1 = \sum_{(m_t,s_t)\in\mathbb{U}} \mathbb{E}_{q_\phi(z_t|s_t,m_t)}\left[logp_{\Theta}(m_t|z_t,s_t)\right] - \lambda D_{KL}\left(q_\phi((z_t|s_t,m_t))||\pi_{\Theta}(z_t|s_t)\right)$$

(50)

Once the model is trained on the examples from the labeled set $(m_t, s_t) \in \mathbb{U}$, the model is updated based on examples from the labeled set $(m_t, s_t, \widehat{z_t}) \in \mathbb{L}$. It is trained to maximize the joint log-likelihood and the labeled intention $\widehat{z_t}$ is treated as an observed variable

$$\mathcal{L}_2 = \sum_{(m_t,\widehat{z}_t,s_t) \in \mathbb{L}} \left[log p_{\ominus}(m_t|\widehat{z}_t, s_t) \pi_{\ominus}(\widehat{z}_t \,|s_t) q_{\phi}\left((\widehat{z}_t \,|s_t, m_t)\right)\right] \quad (51)$$

Both $\mathcal{L}_1$ and $\mathcal{L}_2$ are combined to form $\mathcal{L}'$ which is the definition of a semi-supervised algorithm.

5.6 Reinforcement learning

Wen et al. (2017a) advises the use of reinforcement learning to refine the model's behaviour towards any new input utterance. The approach here is to maximize the success rate of each output from the policy network by giving the system rewards at each correct response (Li et al., 2016). The rewards are developer-defined (Li et al., 2016). The backbone of this model is an encoder-decoder RNN that predicts all possible responses (Li et al., 2016). Here $\pi_{\ominus}(z_t|s_t)$ is used to denote the policy gradient and so reinforcement learning will be applied to this parameter. Reinforcement learning will be applied to the semi-supervised model where $\mathcal{L}_1$ and $\mathcal{L}_2$ will be treated separately. The reinforcement algorithm is applied at each turn *t* from the learnt policy $z_t^{(n)} \sim \pi_{\ominus}(z_t|s_t)$ of unlabeled examples $\mathbb{U}$, $\mathcal{L}_1$ in a semi supervision, which then receives a reward $r_t^{(n)}$.The policy gradient method is used to then fine tune a subset of the model parameters Θ' and can be defined by

$$\frac{\partial \mathcal{J}}{\partial \Theta\prime} \approx \frac{1}{N} \sum_n r_t^{(n)} \frac{\partial log \pi_{\ominus}\left(z_t^{(n)}|s_t\right)}{\partial \Theta\prime} \quad (52)$$

where $\Theta' = \{W_1, b_1, W_2, b_2\}$ is the parameterized policy network in Equation 36. When the model encounters labeled examples $\mathbb{L}$ it converts the labeled action $z_t^{(n)}$ to $\widehat{z}_t$ and updates the model using Equation 22. Updating the parameter Θ' prevents the model from over-fitting as well as improves decision-making (Wen et al., 2017a).

6 Experiments

6.1 Dataset

The dataset was provided upon request by Tsung-Hsien Wen, one of the authors of the paper. He directed me towards a GitHub link (https://github.com/shawnwun/NNDIAL) that listed instructions on how to assemble each model. The first task was to install all the necessary software's required to run the models.

The dataset used for this model was the CamRest676 corpus that was collected by Wen et al (2017b). In here, the system is assigned the task was to find restaurants in Cambridge, UK. The dataset contains 99 unique restaurants. There are three different informable slots (food, price range and area) that the users can search for and six requestable slots (address, phone, postal code, food, price range and area). In this dataset there are 676 dialogues and 2750 conversational turns.

The dataset was divided into three categories: training, validation and testing. The ratio set for each one of them was 3:1:1. The LSTM hidden layer sizes were set to 50 and the vocabulary size is around 500 after pre-processing (Wen et al., 2017b). The system components were all trained jointly by fixing the pre-trained belief trackers and the database operator with the model's latent intention size *I* which was set to 50, 70 and 100. The trade off constants λ and α which are calculated by the inference and semi-supervision were both set to 0.1.

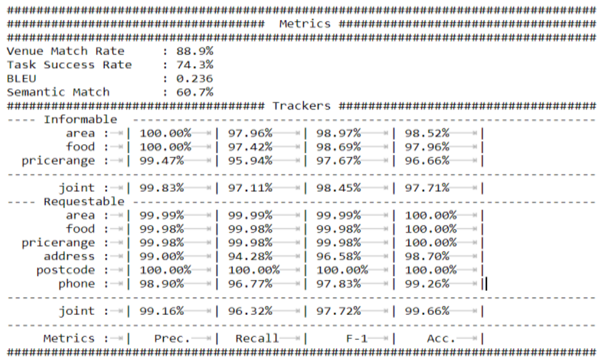
The reinforcement fine-tuning process consisted of generating a sentence m_t from the model that replaces the ground truth $\widehat{m_t}$ at each turn (Wen et al., 2016). This then receives a reward as to whether m_t can improve the dialogue success relative to $\widehat{m_t}$ and the sentence BLEU score where the constant η was set to 0.5 (Wen et al., 2016). Papineni et al., (2002) developed the BLEU score of evaluation. The BLEU score is also known as the Bilingual Evaluation Understudy Score which is an algorithm that evaluates the quality of text which has been machine translated

$$r_t = \eta \cdot sBLEU(m_t, \widehat{m_t}) + \begin{cases} 1 \; m_t \; improves \\ 0 \; m_t \; degrades \\ -1 \quad otherwise \end{cases}$$

The model parameters were fine-tuned using RL for only 3 epochs (Wen et al., 2016). The most probable intention and beam search was selected with the beam width set to 10 for when the response was decoded (Wen et al., 2017a). The average log-probability of tokens in the response was set to the decoding criterion (Wen et al., 2016).

6.2 Metric

The *latent intention dialogue model* was compared to the *vanilla neural dialogue model* and the *attention-based neural dialogue model*. The evaluation of these models was based upon the task success rate of whether the system was able to understand the intention behind each input. There are three evaluation categories: (1) BLEU score, (2) venue matching rate and (3) success rate. The dialogue is considered a success if the system answered all the information requested (e.g. *what is the phone number?)* and whether the entity matches the specified task by the user (Wen et al., 2016). The venue matching rate is seen as percentage of generic values (e.g. *[s.food]*) appear in the candidate as well as the reference (Wen et al., 2016). The BLEU score is calculated over skeletal form (sentences consists of generic values for example *[s.food]*, *[s.address]*) before it is substituted with actual values. The informable tracker is a probability distribution that represent the system's understanding of each slot requested. An example of the display is seen in Figure 7.



```
##############################################################################
###################################  Metrics ##################################
##############################################################################
Venue Match Rate      : 88.9%
Task Success Rate     : 74.3%
BLEU                  : 0.236
Semantic Match        : 60.7%
################################### Trackers ##################################
---- Informable  -------------------------------------------------------------
       area :   | 100.00%   | 97.96%    | 98.97%    | 98.52%    |
       food :   | 100.00%   | 97.42%    | 98.69%    | 97.96%    |
 pricerange :   | 99.47%    | 95.94%    | 97.67%    | 96.66%    |
------------------------------------------------------------------------------
      joint :   | 99.83%    | 97.11%    | 98.45%    | 97.71%    |
---- Requestable -------------------------------------------------------------
       area :   | 99.99%    | 99.99%    | 99.99%    | 100.00%   |
       food :   | 99.98%    | 99.98%    | 99.98%    | 100.00%   |
 pricerange :   | 99.98%    | 99.98%    | 99.98%    | 100.00%   |
    address :   | 99.00%    | 94.28%    | 96.58%    | 98.70%    |
   postcode :   | 100.00%   | 100.00%   | 100.00%   | 100.00%   |
      phone :   | 98.90%    | 96.77%    | 97.83%    | 99.26%    ||
------------------------------------------------------------------------------
      joint :   | 99.16%    | 96.32%    | 97.72%    | 99.66%    |
------------------------------------------------------------------------------
    Metrics :   |   Prec.   |  Recall   |    F-1    |    Acc.   |
##############################################################################
```

Figure 7: Example of the metric of the vanilla neural dialogue model.

6.3 Setup and results

Each script was organized accordingly based on the layout on GitHub. The scripts were altered based upon each unique error. Initially this task was time-consuming. But as the understanding of each script developed, finding solutions to these errors became an easy task. Most of the errors that occurred were due to incorrect file path. Table 1 presents the evaluation results. It compares how the results I achieved by testing vary from that of the author's (Wen et al., 2017). Here three baseline models are tested: (1) the vanilla neural dialogue mode (NDM), (2) the attention-based neural dialogue model, and (3) the LIDM. As part of this research, the LIDM model with and without reinforced learning fine-tuning was tested to determine the effectiveness of the architecture. As seen in the table below, the results of the LIDM+RL model outperformed the three baseline models with the baseline being low. This is proven true for both the author's case and the testing that I conducted. In order to achieve these results, there were specific measures taken. Each model was individually trained, validated and tested. Notice how in Table 1 the success rate and BLEU evaluation rate vary from that of the author's. This is because I trained, validated and tested the data multiple times. As mentioned above, the ratio of split for training, validating and testing was 3:1:1 which is exactly how the author divided up the dataset. The data was the trained, validated and tested rigorously multiple times hence the difference in values.

MODEL	SUCCESS	BLEU	SUCCESS	BLEU
NDM	76.1%	0.212	74.3%	0.206
NDM+ATT	79.0%	0.224	77.1%	0.214
LIDM	63.2%	**0.242**	65.8%	**0.248**
LIDM+RL	**84.6%**	0.240	**83.8%**	0.233

Table 1: The results from evaluating the model. The blue font represents the results of when I tested the model. The black represents the author's results. The bolded values represent the highest output.

6.3.1 Training

The process of training is used to fit the model. Here training was divided into two sections. First, the belief tracker model called the *CamRest.tracker.model* was trained using the cross entropy errors between tracker labels y_s^t and p_s^t . Once that was complete, the next step was to train the remaining parts of the model with the cross-entropy errors from the generation network language model. The commands below were used to train the belief tracker and the other models:

```
TRACKER="CamRest.tracker.model"

THEANO_FLAGS="optimizer=fast_compile"

python./notebooks/nndial.py-config
./notebooks/config/tracker.cfg -mode train
```

Code 1: command used to set up the belief tracker

```
cp./notebooks/model/demo/$TRACKER
./notebooks/model/demo/CamRest.NDM.model

THEANO_FLAGS="optimizer=fast_compile"
```

Code 2: command used to train the *vanilla neural dialogue model*

```
python./notebooks/nndial.py-config
./notebooks/config/NDM.cfg -mode adjust

cp./notebooks/model/demo/$TRACKER
./notebooks/model/demo/CamRest.Att-NDM.model

THEANO_FLAGS="optimizer=fast_compile"

python./notebooks/nndial.py-config
./notebooks/config/Att-NDM.cfg -mode adjust
```

Code 3: command used to train the *attention-based neural dialogue model*

```
cp./notebooks/model/demo/$TRACKER
./notebooks/model/demo/CamRest.LIDM.model

THEANO_FLAGS="optimizer=fast_compile"
python./notebooks/nndial.py-config
./notebooks/config/LIDM.cfg -mode adjust
```

Code 4: command used to train the *latent intention dialogue model*

6.3.2 Validation

Validation is used to tune the hyperparameters. The dataset in validation is periodically being evaluated. Based upon the frequent evaluation results, the model will tune its parameters. The hyperparameters for these model's can be seen in Table 2. They can be found in the configuration folder within the GitHub page. All the hyperparameters were set to values that the author had used. The hyperparameter varies depending upon the structure of each model. Keeping this in mind, the hyperparameters listed below varied model to model

Short form	**Description**
lr	Initial learning rate of Adam
lr_decay	Learning rate decay
stop_count	The maximum number of times when validation gets worse
cur_stop_count	current early stopping step
l2	l2 regularisation weight
random_seed	Random seed
min_impr	The relative minimal improvement allowed

debug	Debug flag
llogp	log prob in the last epoch
grad_clip	gradient clipping parameter

Table 2: the hyperparameters used

The commands used to validate each model is as follows:

```
cp./notebooks/model/demo/$TRACKER
./notebooks/model/demo/CamRest.NDM.model

THEANO_FLAGS="optimizer=fast_compile"
python ./notebooks/nndial.py -config
./notebooks/config/NDM.cfg -mode valid
```

Code 5: command to validate the *vanilla neural dialogue model*

```
cp ./notebooks/model/demo/$TRACKER
./notebooks/model/demo/CamRest.Att-NDM.model

THEANO_FLAGS="optimizer=fast_compile"
python ./notebooks/nndial.py -config
./notebooks/config/Att-NDM.cfg -mode valid
```

Code 6: command to validate the dataset using the *attention-based neural dialogue model*

```
cp ./notebooks/model/demo/$TRACKER
./notebooks/model/demo/CamRest.LIDM.model
THEANO_FLAGS="optimizer=fast_compile"

python ./notebooks/python nndial.py -config
./notebooks/config/LIDM.cfg -mode valid
```

Code 7: command to validate the dataset with the *latent intention dialogue model*

6.3.3 Testing

The testing case is the final evaluation of the model. This is crucial in providing an unbiased evaluation of the final model fit on the training dataset. The following commands were used to test the data:

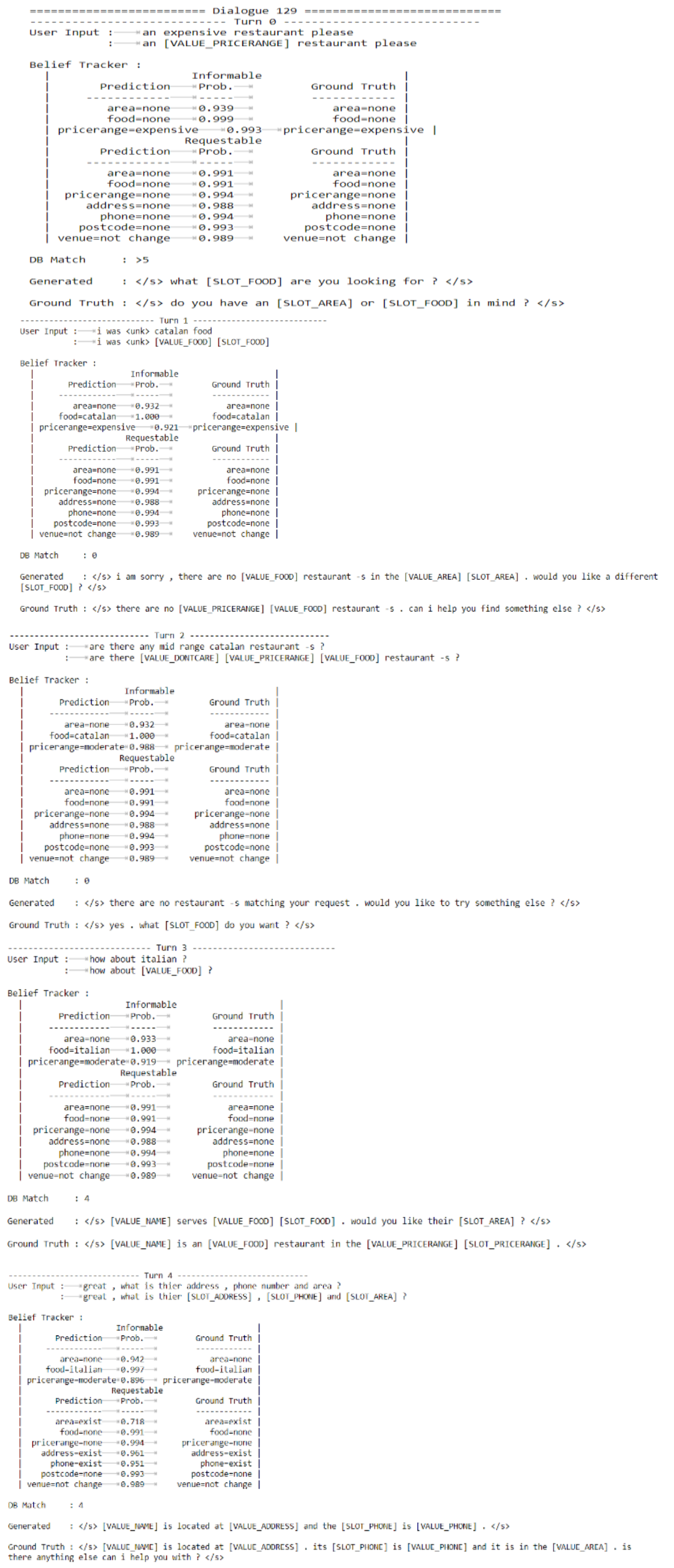
```
cp./notebooks/model/demo/$TRACKER
./notebooks/model/demo/CamRest.NDM.model

THEANO_FLAGS="optimizer=fast_compile"
 python ./notebooks/nndial.py -config
./notebooks/config/NDM.cfg -mode test
```

Code 8: command to test the *vanilla neural dialogue model*

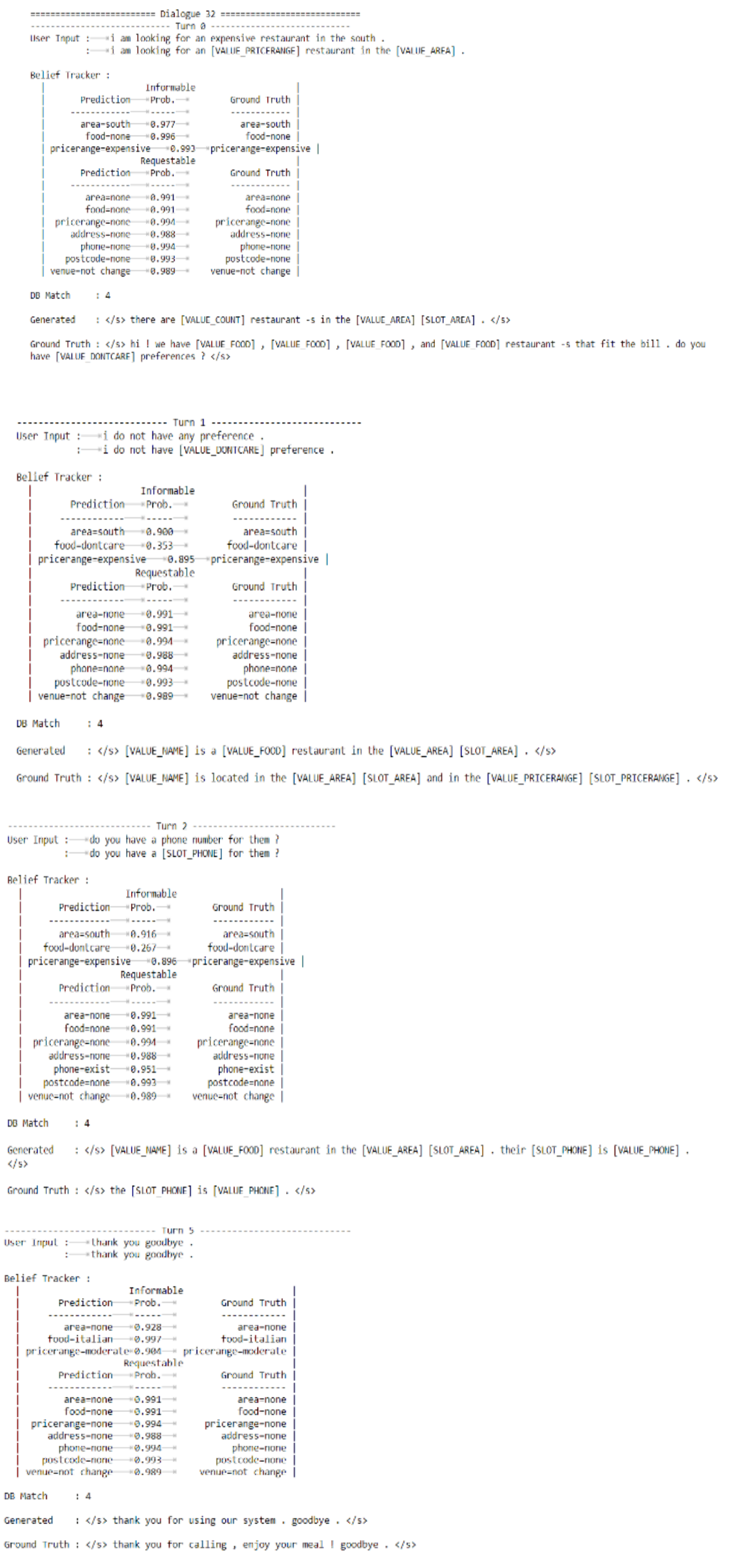
The result of the query above was a series of 135 dialogues generated for the *vanilla neural dialogue model*. These dialogues had no fixed set of turns. With each turn the user asks a question until the goal is reached. The belief tracker is included in every turn. As explained in 3.1.2, the aim of the belief tracker is to maintain a multinomial distribution *p* over values for each informable slot (Wen et al., 2016). And so, the probability distribution that is seen in each tur represent the probability of each value occurring based upon the input. The longest turn recorded was five whereas the shortest was two. The hidden layer size was set to 50, and the weights were randomly initialised between -0.3 and 0.3 including word embeddings (Wen et al., 2016). Below are a few examples of how the testing process took place.

As seen in Figure 8, notice how the slot values are still in skeletal form. The system tries to generate results based on a price range. Here the output of "expensive resturants" is greater than 5 which can then be narrowed down depending on the user's input in turn 1. In Figure 9, it can be see that the system fails to generate a valid response. At turn 3, the user's input is *have a good evening*. The system in unable to understand what the user means and so it generates a random response*; </s> [VALUE_NAME] is a nice [VALUE_FOOD] restaurant in the [VALUE_AREA] [SLOT_AREA]* . *</s>*. This is how the system fails to understand the underlying meaning behind the dialogue. Notice how the response to each input is direct and closed ended. This seems less like a human-human interaction and like a computer-human interaction. This model surpasses the *latent intention dialogue model*

(LIDM) without RL. According to Table 1 the success rate of this model is seen to be 74.3% with a BLEU score of 0.206.

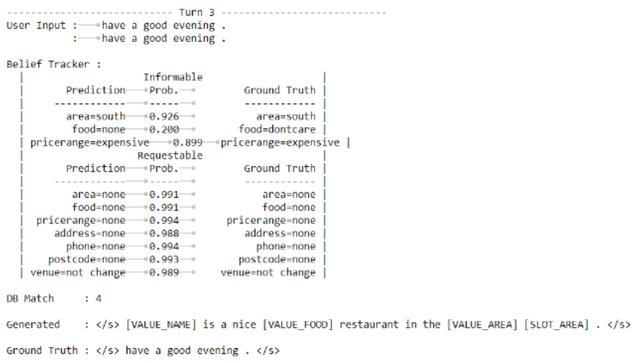


```
========================= Dialogue 129 ============================
--------------------------- Turn 0 ---------------------------
User Input :→an expensive restaurant please
           :→an [VALUE_PRICERANGE] restaurant please

Belief Tracker :
  |                   Informable                          |
  |         Prediction→Prob.→              Ground Truth   |
  |       ------------→-----→              ------------   |
  |          area=none→0.939→                 area=none   |
  |          food=none→0.999→                 food=none   |
  | pricerange=expensive→0.993→pricerange=expensive |
  |                  Requestable                          |
  |         Prediction→Prob.→              Ground Truth   |
  |       ------------→-----→              ------------   |
  |          area=none→0.991→                 area=none   |
  |          food=none→0.991→                 food=none   |
  |    pricerange=none→0.994→           pricerange=none   |
  |       address=none→0.988→              address=none   |
  |         phone=none→0.994→                phone=none   |
  |      postcode=none→0.993→             postcode=none   |
  |  venue=not change→0.989→           venue=not change   |

DB Match      : >5

Generated     : </s> what [SLOT_FOOD] are you looking for ? </s>

Ground Truth : </s> do you have an [SLOT_AREA] or [SLOT_FOOD] in mind ? </s>
--------------------------- Turn 1 ---------------------------
User Input :→i was <unk> catalan food
           :→i was <unk> [VALUE_FOOD] [SLOT_FOOD]

Belief Tracker :
  |                   Informable                          |
  |         Prediction→Prob.→              Ground Truth   |
  |       ------------→-----→              ------------   |
  |          area=none→0.932→                 area=none   |
  |       food=catalan→1.000→              food=catalan   |
  | pricerange=expensive→0.921→pricerange=expensive |
  |                  Requestable                          |
  |         Prediction→Prob.→              Ground Truth   |
  |       ------------→-----→              ------------   |
  |          area=none→0.991→                 area=none   |
  |          food=none→0.991→                 food=none   |
  |    pricerange=none→0.994→           pricerange=none   |
  |       address=none→0.988→              address=none   |
  |         phone=none→0.994→                phone=none   |
  |      postcode=none→0.993→             postcode=none   |
  |  venue=not change→0.989→           venue=not change   |

DB Match      : 0

Generated     : </s> i am sorry , there are no [VALUE_FOOD] restaurant -s in the [VALUE_AREA] [SLOT_AREA] . would you like a different
[SLOT_FOOD] ? </s>

Ground Truth : </s> there are no [VALUE_PRICERANGE] [VALUE_FOOD] restaurant -s . can i help you find something else ? </s>

--------------------------- Turn 2 ---------------------------
User Input :→are there any mid range catalan restaurant -s ?
           :→are there [VALUE_DONTCARE] [VALUE_PRICERANGE] [VALUE_FOOD] restaurant -s ?

Belief Tracker :
  |                   Informable                          |
  |         Prediction→Prob.→              Ground Truth   |
  |       ------------→-----→              ------------   |
  |          area=none→0.932→                 area=none   |
  |       food=catalan→1.000→              food=catalan   |
  | pricerange=moderate→0.988→  pricerange=moderate   |
  |                  Requestable                          |
  |         Prediction→Prob.→              Ground Truth   |
  |       ------------→-----→              ------------   |
  |          area=none→0.991→                 area=none   |
  |          food=none→0.991→                 food=none   |
  |    pricerange=none→0.994→           pricerange=none   |
  |       address=none→0.988→              address=none   |
  |         phone=none→0.994→                phone=none   |
  |      postcode=none→0.993→             postcode=none   |
  |  venue=not change→0.989→           venue=not change   |

DB Match      : 0

Generated     : </s> there are no restaurant -s matching your request . would you like to try something else ? </s>

Ground Truth : </s> yes . what [SLOT_FOOD] do you want ? </s>

--------------------------- Turn 3 ---------------------------
User Input :→how about italian ?
           :→how about [VALUE_FOOD] ?

Belief Tracker :
  |                   Informable                          |
  |         Prediction→Prob.→              Ground Truth   |
  |       ------------→-----→              ------------   |
  |          area=none→0.933→                 area=none   |
  |       food=italian→1.000→              food=italian   |
  | pricerange=moderate→0.919→  pricerange=moderate   |
  |                  Requestable                          |
  |         Prediction→Prob.→              Ground Truth   |
  |       ------------→-----→              ------------   |
  |          area=none→0.991→                 area=none   |
  |          food=none→0.991→                 food=none   |
  |    pricerange=none→0.994→           pricerange=none   |
  |       address=none→0.988→              address=none   |
  |         phone=none→0.994→                phone=none   |
  |      postcode=none→0.993→             postcode=none   |
  |  venue=not change→0.989→           venue=not change   |

DB Match      : 4

Generated     : </s> [VALUE_NAME] serves [VALUE_FOOD] [SLOT_FOOD] . would you like their [SLOT_AREA] ? </s>

Ground Truth : </s> [VALUE_NAME] is an [VALUE_FOOD] restaurant in the [VALUE_PRICERANGE] [SLOT_PRICERANGE] . </s>


--------------------------- Turn 4 ---------------------------
User Input :→great , what is thier address , phone number and area ?
           :→great , what is thier [SLOT_ADDRESS] , [SLOT_PHONE] and [SLOT_AREA] ?

Belief Tracker :
  |                   Informable                          |
  |         Prediction→Prob.→              Ground Truth   |
  |       ------------→-----→              ------------   |
  |          area=none→0.942→                 area=none   |
  |       food=italian→0.997→              food=italian   |
  | pricerange=moderate→0.896→  pricerange=moderate   |
  |                  Requestable                          |
  |         Prediction→Prob.→              Ground Truth   |
  |       ------------→-----→              ------------   |
  |         area=exist→0.718→                area=exist   |
  |          food=none→0.991→                 food=none   |
  |    pricerange=none→0.994→           pricerange=none   |
  |      address=exist→0.961→             address=exist   |
  |        phone=exist→0.951→               phone=exist   |
  |      postcode=none→0.993→             postcode=none   |
  |  venue=not change→0.989→           venue=not change   |

DB Match      : 4

Generated     : </s> [VALUE_NAME] is located at [VALUE_ADDRESS] and the [SLOT_PHONE] is [VALUE_PHONE] . </s>

Ground Truth : </s> [VALUE_NAME] is located at [VALUE_ADDRESS] . its [SLOT_PHONE] is [VALUE_PHONE] and it is in the [VALUE_AREA] . is
there anything else can i help you with ? </s>
```



```
========================= Dialogue 32 ============================
--------------------------- Turn 0 ---------------------------
User Input :→i am looking for an expensive restaurant in the south .
           :→i am looking for an [VALUE_PRICERANGE] restaurant in the [VALUE_AREA] .

Belief Tracker :
  |                   Informable                          |
  |         Prediction→Prob.→              Ground Truth   |
  |       ------------→-----→              ------------   |
  |         area=south→0.977→                area=south   |
  |          food=none→0.996→                 food=none   |
  | pricerange=expensive→0.993→pricerange=expensive |
  |                  Requestable                          |
  |         Prediction→Prob.→              Ground Truth   |
  |       ------------→-----→              ------------   |
  |          area=none→0.991→                 area=none   |
  |          food=none→0.991→                 food=none   |
  |    pricerange=none→0.994→           pricerange=none   |
  |       address=none→0.988→              address=none   |
  |         phone=none→0.994→                phone=none   |
  |      postcode=none→0.993→             postcode=none   |
  |  venue=not change→0.989→           venue=not change   |

DB Match      : 4

Generated     : </s> there are [VALUE_COUNT] restaurant -s in the [VALUE_AREA] [SLOT_AREA] . </s>

Ground Truth : </s> hi ! we have [VALUE_FOOD] , [VALUE_FOOD] , [VALUE_FOOD] , and [VALUE_FOOD] restaurant -s that fit the bill . do you
have [VALUE_DONTCARE] preferences ? </s>


--------------------------- Turn 1 ---------------------------
User Input :→i do not have any preference .
           :→i do not have [VALUE_DONTCARE] preference .

Belief Tracker :
  |                   Informable                          |
  |         Prediction→Prob.→              Ground Truth   |
  |       ------------→-----→              ------------   |
  |         area=south→0.900→                area=south   |
  |      food=dontcare→0.353→             food=dontcare   |
  | pricerange=expensive→0.895→pricerange=expensive |
  |                  Requestable                          |
  |         Prediction→Prob.→              Ground Truth   |
  |       ------------→-----→              ------------   |
  |          area=none→0.991→                 area=none   |
  |          food=none→0.991→                 food=none   |
  |    pricerange=none→0.994→           pricerange=none   |
  |       address=none→0.988→              address=none   |
  |         phone=none→0.994→                phone=none   |
  |      postcode=none→0.993→             postcode=none   |
  |  venue=not change→0.989→           venue=not change   |

DB Match      : 4

Generated     : </s> [VALUE_NAME] is a [VALUE_FOOD] restaurant in the [VALUE_AREA] [SLOT_AREA] . </s>

Ground Truth : </s> [VALUE_NAME] is located in the [VALUE_AREA] [SLOT_AREA] and in the [VALUE_PRICERANGE] [SLOT_PRICERANGE] . </s>


--------------------------- Turn 2 ---------------------------
User Input :→do you have a phone number for them ?
           :→do you have a [SLOT_PHONE] for them ?

Belief Tracker :
  |                   Informable                          |
  |         Prediction→Prob.→              Ground Truth   |
  |       ------------→-----→              ------------   |
  |         area=south→0.916→                area=south   |
  |      food=dontcare→0.267→             food=dontcare   |
  | pricerange=expensive→0.896→pricerange=expensive |
  |                  Requestable                          |
  |         Prediction→Prob.→              Ground Truth   |
  |       ------------→-----→              ------------   |
  |          area=none→0.991→                 area=none   |
  |          food=none→0.991→                 food=none   |
  |    pricerange=none→0.994→           pricerange=none   |
  |       address=none→0.988→              address=none   |
  |        phone=exist→0.951→               phone=exist   |
  |      postcode=none→0.993→             postcode=none   |
  |  venue=not change→0.989→           venue=not change   |

DB Match      : 4

Generated     : </s> [VALUE_NAME] is a [VALUE_FOOD] restaurant in the [VALUE_AREA] [SLOT_AREA] . their [SLOT_PHONE] is [VALUE_PHONE] .
</s>

Ground Truth : </s> the [SLOT_PHONE] is [VALUE_PHONE] . </s>


--------------------------- Turn 5 ---------------------------
User Input :→thank you goodbye .
           :→thank you goodbye .

Belief Tracker :
  |                   Informable                          |
  |         Prediction→Prob.→              Ground Truth   |
  |       ------------→-----→              ------------   |
  |          area=none→0.928→                 area=none   |
  |       food=italian→0.997→              food=italian   |
  | pricerange=moderate→0.904→  pricerange=moderate   |
  |                  Requestable                          |
  |         Prediction→Prob.→              Ground Truth   |
  |       ------------→-----→              ------------   |
  |          area=none→0.991→                 area=none   |
  |          food=none→0.991→                 food=none   |
  |    pricerange=none→0.994→           pricerange=none   |
  |       address=none→0.988→              address=none   |
  |         phone=none→0.994→                phone=none   |
  |      postcode=none→0.993→             postcode=none   |
  |  venue=not change→0.989→           venue=not change   |

DB Match      : 4

Generated     : </s> thank you for using our system . goodbye . </s>

Ground Truth : </s> thank you for calling , enjoy your meal ! goodbye . </s>
```

Figure 8: Dialogue 129 with 5 turns. The system was successful in generating a relevant response.



```
--------------------------- Turn 3 ---------------------------
User Input :    have a good evening .
           :    have a good evening .

Belief Tracker :
  |                 Informable                     |
  |       Prediction    Prob.         Ground Truth |
  |     ------------    -----         ------------ |
  |       area=south    0.926           area=south |
  |        food=none    0.200        food=dontcare |
  | pricerange=expensive    0.899   pricerange=expensive |
  |                 Requestable                    |
  |       Prediction    Prob.         Ground Truth |
  |     ------------    -----         ------------ |
  |        area=none    0.991            area=none |
  |        food=none    0.991            food=none |
  |  pricerange=none    0.994      pricerange=none |
  |     address=none    0.988         address=none |
  |       phone=none    0.994           phone=none |
  |    postcode=none    0.993        postcode=none |
  | venue=not change    0.989     venue=not change |

DB Match     : 4

Generated    : </s> [VALUE_NAME] is a nice [VALUE_FOOD] restaurant in the [VALUE_AREA] [SLOT_AREA] . </s>

Ground Truth : </s> have a good evening . </s>
```

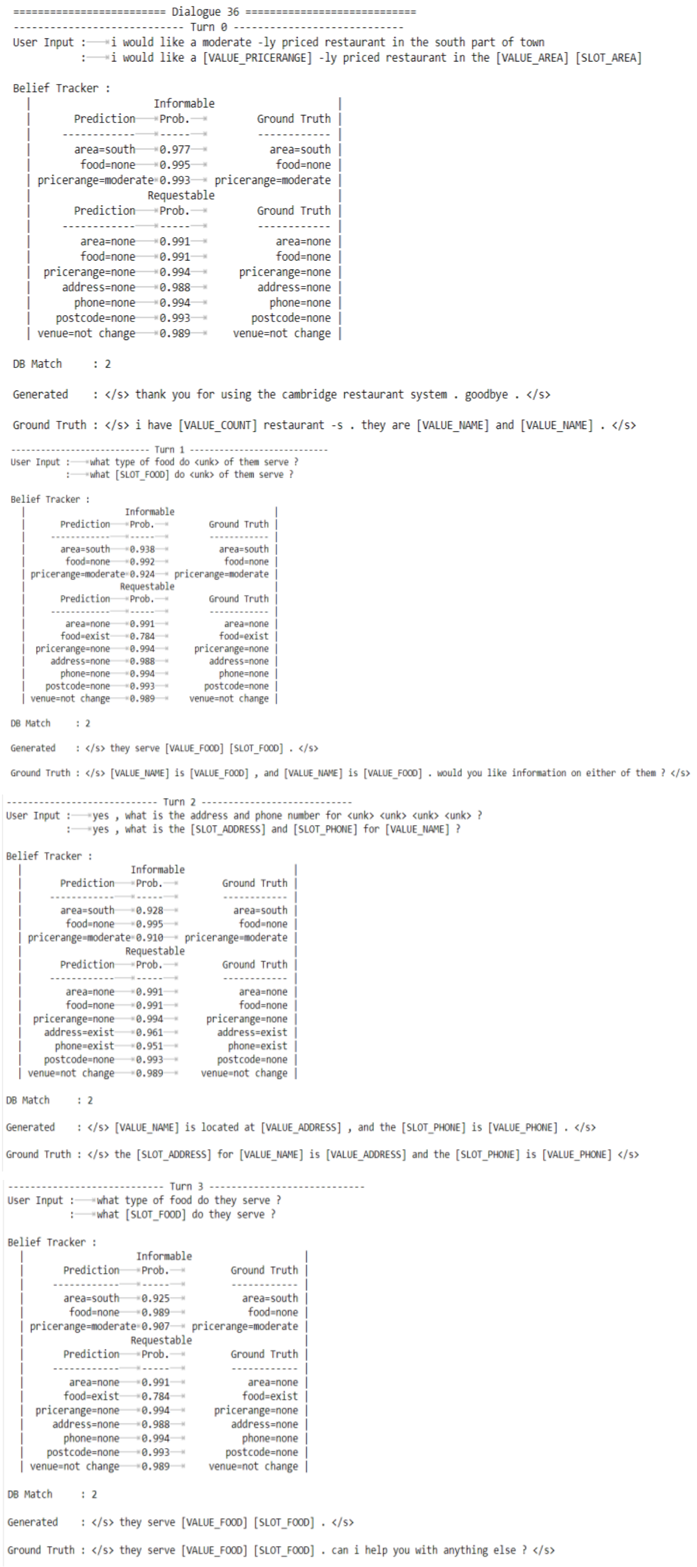
Figure 9: Dialogue 32 out of 132 with 3 turns. As seen the system fails to generate an appropriate response on turn 3.

The command used to test the dataset for the *attention-based neural dialogue model* is:

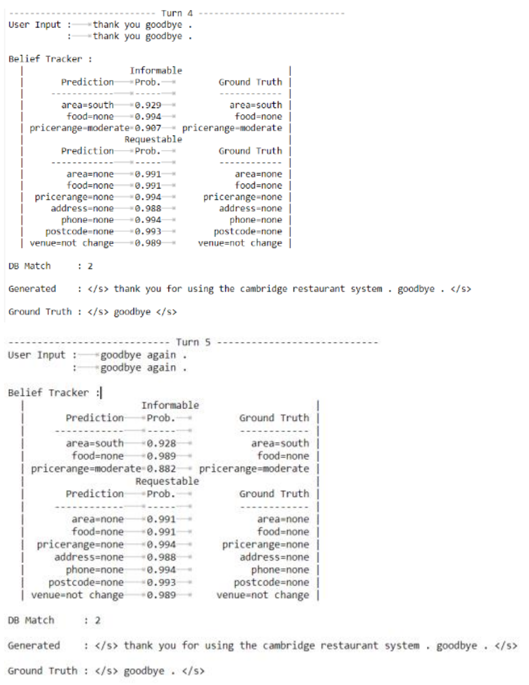
```
cp ./notebooks/model/demo/$TRACKER
./notebooks/model/demo/CamRest.Att-NDM.model

THEANO_FLAGS="optimizer=fast_compile"
  python ./notebooks/nndial.py -config
./notebooks/config/Att-NDM.cfg -mode test
```

Code 9: command to test data using the *attention-based neural dialogue model*

Two examples are shown below of which one is a success and the other is where the system fails. Figure 10 is an example where the *attention-based neural dialogue model* successfully performs the task assigned. This example is interesting as during turn 5 the system can understand the meaning behind the user's response. As mentioned above, at times the *attention-based dialogue model* fails to generate an appropriate response. Even though in turn 5 the input was *goodbye again*, the system was able to ignore the word *again* and generate a valid response.



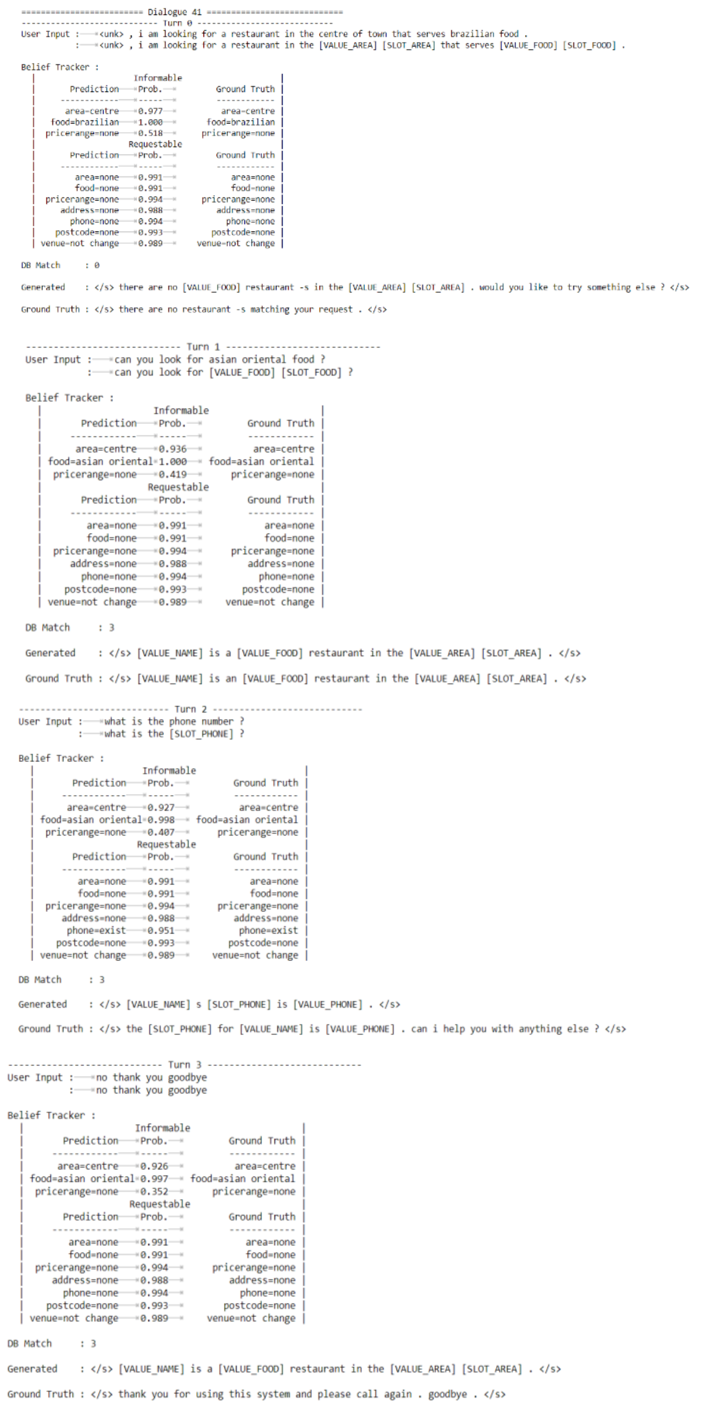
```
========================= Dialogue 36 ===========================
---------------------------- Turn 0 ----------------------------
User Input :    i would like a moderate -ly priced restaurant in the south part of town
           :    i would like a [VALUE_PRICERANGE] -ly priced restaurant in the [VALUE_AREA] [SLOT_AREA]

Belief Tracker :
  |                 Informable                     |
  |       Prediction    Prob.         Ground Truth |
  |     ------------    -----         ------------ |
  |       area=south    0.977           area=south |
  |        food=none    0.995            food=none |
  | pricerange=moderate 0.993   pricerange=moderate |
  |                 Requestable                    |
  |       Prediction    Prob.         Ground Truth |
  |     ------------    -----         ------------ |
  |        area=none    0.991            area=none |
  |        food=none    0.991            food=none |
  |  pricerange=none    0.994      pricerange=none |
  |     address=none    0.988         address=none |
  |       phone=none    0.994           phone=none |
  |    postcode=none    0.993        postcode=none |
  | venue=not change    0.989     venue=not change |

DB Match     : 2

Generated    : </s> thank you for using the cambridge restaurant system . goodbye . </s>

Ground Truth : </s> i have [VALUE_COUNT] restaurant -s . they are [VALUE_NAME] and [VALUE_NAME] . </s>
---------------------------- Turn 1 ----------------------------
User Input :    what type of food do <unk> of them serve ?
           :    what [SLOT_FOOD] do <unk> of them serve ?

Belief Tracker :
  |                 Informable                     |
  |       Prediction    Prob.         Ground Truth |
  |     ------------    -----         ------------ |
  |       area=south    0.938           area=south |
  |        food=none    0.992            food=none |
  | pricerange=moderate 0.924   pricerange=moderate |
  |                 Requestable                    |
  |       Prediction    Prob.         Ground Truth |
  |     ------------    -----         ------------ |
  |        area=none    0.991            area=none |
  |       food=exist    0.784           food=exist |
  |  pricerange=none    0.994      pricerange=none |
  |     address=none    0.988         address=none |
  |       phone=none    0.994           phone=none |
  |    postcode=none    0.993        postcode=none |
  | venue=not change    0.989     venue=not change |

DB Match     : 2

Generated    : </s> they serve [VALUE_FOOD] [SLOT_FOOD] . </s>

Ground Truth : </s> [VALUE_NAME] is [VALUE_FOOD] , and [VALUE_NAME] is [VALUE_FOOD] . would you like information on either of them ? </s>

---------------------------- Turn 2 ----------------------------
User Input :    yes , what is the address and phone number for <unk> <unk> <unk> <unk> ?
           :    yes , what is the [SLOT_ADDRESS] and [SLOT_PHONE] for [VALUE_NAME] ?

Belief Tracker :
  |                 Informable                     |
  |       Prediction    Prob.         Ground Truth |
  |     ------------    -----         ------------ |
  |       area=south    0.928           area=south |
  |        food=none    0.995            food=none |
  | pricerange=moderate 0.910   pricerange=moderate |
  |                 Requestable                    |
  |       Prediction    Prob.         Ground Truth |
  |     ------------    -----         ------------ |
  |        area=none    0.991            area=none |
  |        food=none    0.991            food=none |
  |  pricerange=none    0.994      pricerange=none |
  |    address=exist    0.961        address=exist |
  |      phone=exist    0.951          phone=exist |
  |    postcode=none    0.993        postcode=none |
  | venue=not change    0.989     venue=not change |

DB Match     : 2

Generated    : </s> [VALUE_NAME] is located at [VALUE_ADDRESS] , and the [SLOT_PHONE] is [VALUE_PHONE] . </s>

Ground Truth : </s> the [SLOT_ADDRESS] for [VALUE_NAME] is [VALUE_ADDRESS] and the [SLOT_PHONE] is [VALUE_PHONE] </s>

---------------------------- Turn 3 ----------------------------
User Input :    what type of food do they serve ?
           :    what [SLOT_FOOD] do they serve ?

Belief Tracker :
  |                 Informable                     |
  |       Prediction    Prob.         Ground Truth |
  |     ------------    -----         ------------ |
  |       area=south    0.925           area=south |
  |        food=none    0.989            food=none |
  | pricerange=moderate 0.907   pricerange=moderate |
  |                 Requestable                    |
  |       Prediction    Prob.         Ground Truth |
  |     ------------    -----         ------------ |
  |        area=none    0.991            area=none |
  |       food=exist    0.784           food=exist |
  |  pricerange=none    0.994      pricerange=none |
  |     address=none    0.988         address=none |
  |       phone=none    0.994           phone=none |
  |    postcode=none    0.993        postcode=none |
  | venue=not change    0.989     venue=not change |

DB Match     : 2

Generated    : </s> they serve [VALUE_FOOD] [SLOT_FOOD] . </s>

Ground Truth : </s> they serve [VALUE_FOOD] [SLOT_FOOD] . can i help you with anything else ? </s>
```



```
--------------------------- Turn 4 ---------------------------
User Input :    thank you goodbye .
           :    thank you goodbye .

Belief Tracker :
   |                  Informable                        |
   |       Prediction    Prob.            Ground Truth |
   |     ------------    -----            ------------ |
   |       area=south    0.929              area=south |
   |        food=none    0.994               food=none |
   | pricerange=moderate 0.907     pricerange=moderate |
   |                  Requestable                       |
   |       Prediction    Prob.            Ground Truth |
   |     ------------    -----            ------------ |
   |        area=none    0.991               area=none |
   |        food=none    0.991               food=none |
   |  pricerange=none    0.994         pricerange=none |
   |     address=none    0.988            address=none |
   |       phone=none    0.994              phone=none |
   |    postcode=none    0.993           postcode=none |
   | venue=not change    0.989        venue=not change |

DB Match     : 2

Generated    : </s> thank you for using the cambridge restaurant system . goodbye . </s>

Ground Truth : </s> goodbye </s>

--------------------------- Turn 5 ---------------------------
User Input :    goodbye again .
           :    goodbye again .

Belief Tracker :|
   |                  Informable                        |
   |       Prediction    Prob.            Ground Truth |
   |     ------------    -----            ------------ |
   |       area=south    0.928              area=south |
   |        food=none    0.989               food=none |
   | pricerange=moderate 0.882     pricerange=moderate |
   |                  Requestable                       |
   |       Prediction    Prob.            Ground Truth |
   |     ------------    -----            ------------ |
   |        area=none    0.991               area=none |
   |        food=none    0.991               food=none |
   |  pricerange=none    0.994         pricerange=none |
   |     address=none    0.988            address=none |
   |       phone=none    0.994              phone=none |
   |    postcode=none    0.993           postcode=none |
   | venue=not change    0.989        venue=not change |

DB Match     : 2

Generated    : </s> thank you for using the cambridge restaurant system . goodbye . </s>

Ground Truth : </s> goodbye . </s>
```

Figure 10: Dialogue 36 out of 132 with 5 turns. The system can successfully generate a response

Figure 11 is a representation of the *attention-based neural dialogue model* failing to generate an appropriate response. As seen in Figure 10, the system was able to generate an appropriate response for turn 5 even though there was an additional word after *goodbye*. But what is seen in Figure 11 is that the system is unable to conclude with a valid response. This could be due to the model's inability to generate a response that is valid which was highlighted in the beginning of this paper. The 132 dialogues generated for this model seem to fail due to the reason mentioned above. The success of the dialogues seems to outweigh the failures. As seen in Table 1, the metric for this model shows a success rate of 77.1% with a BLEU rate of 0.214. This model outperforms the *vanilla neural dialogue model* (NDM) and the *latent intention dialogue model (*LIDM) without RL.



```
========================= Dialogue 41 ============================
--------------------------- Turn 0 ---------------------------
User Input :    <unk> , i am looking for a restaurant in the centre of town that serves brazilian food .
           :    <unk> , i am looking for a restaurant in the [VALUE_AREA] [SLOT_AREA] that serves [VALUE_FOOD] [SLOT_FOOD] .

Belief Tracker :
   |                  Informable                        |
   |       Prediction    Prob.            Ground Truth |
   |     ------------    -----            ------------ |
   |      area=centre    0.977             area=centre |
   |   food=brazilian    1.000          food=brazilian |
   |  pricerange=none    0.518         pricerange=none |
   |                  Requestable                       |
   |       Prediction    Prob.            Ground Truth |
   |     ------------    -----            ------------ |
   |        area=none    0.991               area=none |
   |        food=none    0.991               food=none |
   |  pricerange=none    0.994         pricerange=none |
   |     address=none    0.988            address=none |
   |       phone=none    0.994              phone=none |
   |    postcode=none    0.993           postcode=none |
   | venue=not change    0.989        venue=not change |

DB Match     : 0

Generated    : </s> there are no [VALUE_FOOD] restaurant -s in the [VALUE_AREA] [SLOT_AREA] . would you like to try something else ? </s>

Ground Truth : </s> there are no restaurant -s matching your request . </s>


--------------------------- Turn 1 ---------------------------
User Input :    can you look for asian oriental food ?
           :    can you look for [VALUE_FOOD] [SLOT_FOOD] ?

Belief Tracker :
   |                  Informable                        |
   |       Prediction    Prob.            Ground Truth |
   |     ------------    -----            ------------ |
   |      area=centre    0.936             area=centre |
   | food=asian oriental 1.000     food=asian oriental |
   |  pricerange=none    0.419         pricerange=none |
   |                  Requestable                       |
   |       Prediction    Prob.            Ground Truth |
   |     ------------    -----            ------------ |
   |        area=none    0.991               area=none |
   |        food=none    0.991               food=none |
   |  pricerange=none    0.994         pricerange=none |
   |     address=none    0.988            address=none |
   |       phone=none    0.994              phone=none |
   |    postcode=none    0.993           postcode=none |
   | venue=not change    0.989        venue=not change |

DB Match     : 3

Generated    : </s> [VALUE_NAME] is a [VALUE_FOOD] restaurant in the [VALUE_AREA] [SLOT_AREA] . </s>

Ground Truth : </s> [VALUE_NAME] is an [VALUE_FOOD] restaurant in the [VALUE_AREA] [SLOT_AREA] . </s>

--------------------------- Turn 2 ---------------------------
User Input :    what is the phone number ?
           :    what is the [SLOT_PHONE] ?

Belief Tracker :
   |                  Informable                        |
   |       Prediction    Prob.            Ground Truth |
   |     ------------    -----            ------------ |
   |      area=centre    0.927             area=centre |
   | food=asian oriental 0.998     food=asian oriental |
   |  pricerange=none    0.407         pricerange=none |
   |                  Requestable                       |
   |       Prediction    Prob.            Ground Truth |
   |     ------------    -----            ------------ |
   |        area=none    0.991               area=none |
   |        food=none    0.991               food=none |
   |  pricerange=none    0.994         pricerange=none |
   |     address=none    0.988            address=none |
   |      phone=exist    0.951             phone=exist |
   |    postcode=none    0.993           postcode=none |
   | venue=not change    0.989        venue=not change |

DB Match     : 3

Generated    : </s> [VALUE_NAME] s [SLOT_PHONE] is [VALUE_PHONE] . </s>

Ground Truth : </s> the [SLOT_PHONE] for [VALUE_NAME] is [VALUE_PHONE] . can i help you with anything else ? </s>

--------------------------- Turn 3 ---------------------------
User Input :    no thank you goodbye
           :    no thank you goodbye

Belief Tracker :
   |                  Informable                        |
   |       Prediction    Prob.            Ground Truth |
   |     ------------    -----            ------------ |
   |      area=centre    0.926             area=centre |
   | food=asian oriental 0.997     food=asian oriental |
   |  pricerange=none    0.352         pricerange=none |
   |                  Requestable                       |
   |       Prediction    Prob.            Ground Truth |
   |     ------------    -----            ------------ |
   |        area=none    0.991               area=none |
   |        food=none    0.991               food=none |
   |  pricerange=none    0.994         pricerange=none |
   |     address=none    0.988            address=none |
   |       phone=none    0.994              phone=none |
   |    postcode=none    0.993           postcode=none |
   | venue=not change    0.989        venue=not change |

DB Match     : 3

Generated    : </s> [VALUE_NAME] is a [VALUE_FOOD] restaurant in the [VALUE_AREA] [SLOT_AREA] . </s>

Ground Truth : </s> thank you for using this system and please call again . goodbye . </s>
```

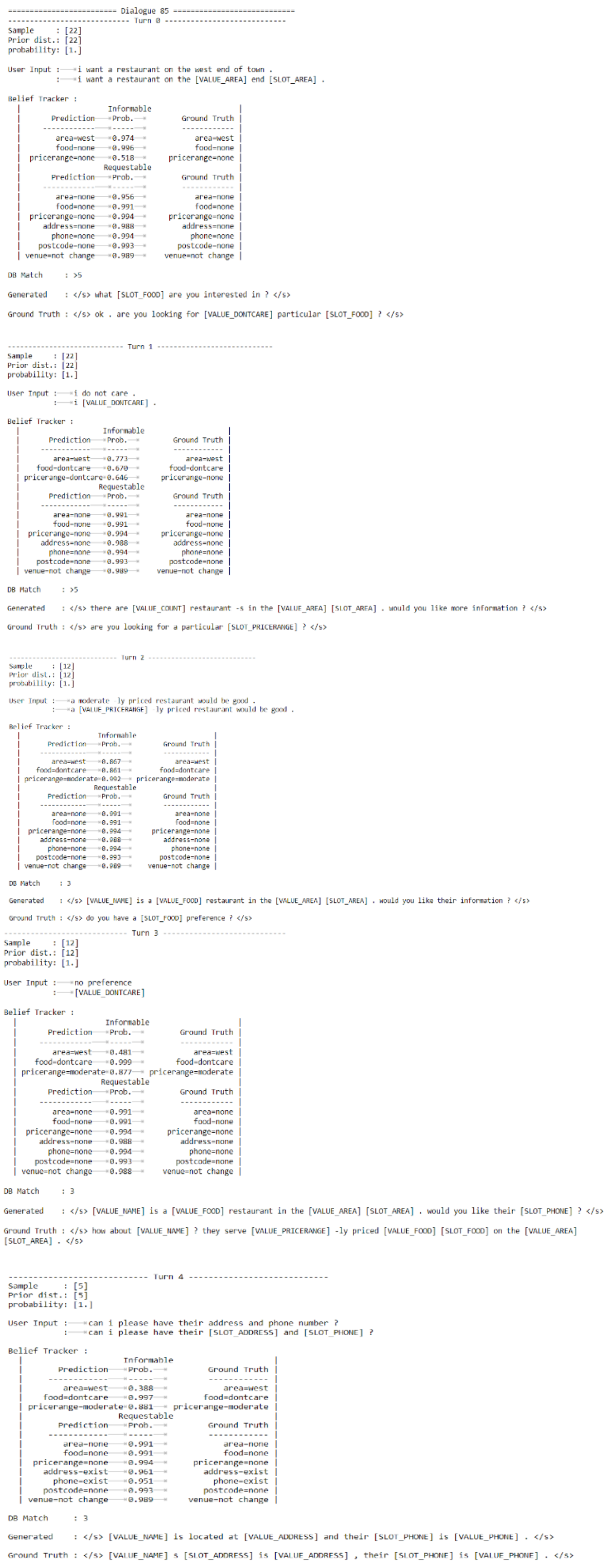
Figure 11: Dialogue 41 out of 132 with 3 turns. The system fails to generate a valid response.

The command to test the *latent intention dialogue model* is described as follows:

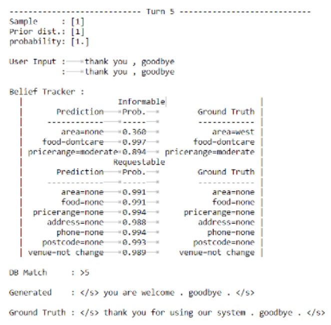
```
cp ./notebooks/model/$TRACKER
model/CamRest.LIDM.model

THEANO_FLAGS="optimizer=fast_compile" python
nndial.py -config ./notebooks/config/LIDM.cfg
-mode adjust
```

Code 10: command to test the *latent intention dialogue model* without reinforced learning

There will be two examples that will show the success and failure case of this model. Figure 12 shows the success of the model. Here, one can tell that the responses the system is giving are more human-like. Notice how in turn 5 the system responds with *you are welcome* instead of the generic *goodbye* most commonly used in the NDM and ATT+NDM. During turn 1 the system provides the user with a count of restaurants available based upon the type of food requested. In the models above, the system would randomly choose a restaurant if the options were more than one. The system can understand sentences such as *I don't care* in turn 1 and *no preferences* in turn 3. This is important as the ATT+NDM was unable to pick up words that were not in the database. As seen in Table 1, the LIDM has a success rate as low as 65.8%. this is the lowest amongst the three. This is proven true as most of the dialogues suggested are similar to Figure 13 where it is unable to understand the truth behind the dialogue. During turn 2, the ground truth is that there is no expensive European food in the area but instead the model gave an incorrect response by giving the name of the restaurant with the location. This should be noted for future references as this is where the model is lacking. Surprisingly, the quality of text compared to the other two models is better. The system seems to respond in a humanistic aspect. Due to this, the BLEU of the LIDM is 0.248. the highest amongst the three.

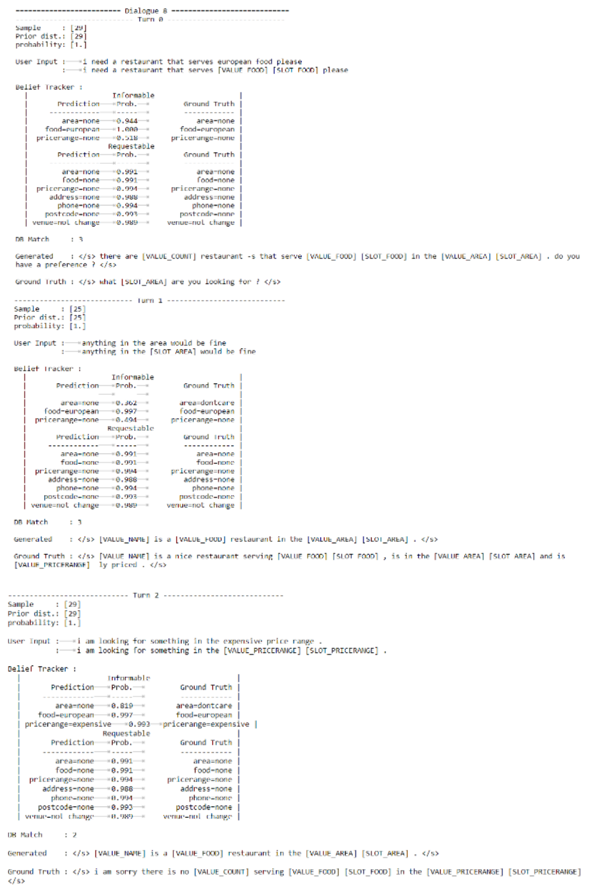


```
======================== Dialogue 85 ============================
---------------------------- Turn 0 ----------------------------
Sample     : [22]
Prior dist.: [22]
probability: [1.]

User Input :    i want a restaurant on the west end of town .
           :    i want a restaurant on the [VALUE_AREA] end [SLOT_AREA] .

Belief Tracker :
   |                Informable                        |
   |       Prediction   Prob.          Ground Truth   |
   |     ------------   -----          ------------   |
   |        area=west   0.974             area=west   |
   |        food=none   0.996             food=none   |
   |  pricerange=none   0.518       pricerange=none   |
   |               Requestable                        |
   |       Prediction   Prob.          Ground Truth   |
   |     ------------   -----          ------------   |
   |        area=none   0.956             area=none   |
   |        food=none   0.991             food=none   |
   |  pricerange=none   0.994       pricerange=none   |
   |     address=none   0.988          address=none   |
   |       phone=none   0.994            phone=none   |
   |    postcode=none   0.993         postcode=none   |
   | venue=not change   0.989      venue=not change   |

DB Match     : >5

Generated    : </s> what [SLOT_FOOD] are you interested in ? </s>

Ground Truth : </s> ok . are you looking for [VALUE_DONTCARE] particular [SLOT_FOOD] ? </s>


---------------------------- Turn 1 ----------------------------
Sample     : [22]
Prior dist.: [22]
probability: [1.]

User Input :    i do not care .
           :    i [VALUE_DONTCARE] .

Belief Tracker :
   |                 Informable                       |
   |         Prediction   Prob.        Ground Truth   |
   |       ------------   -----        ------------   |
   |          area=west   0.773           area=west   |
   |      food=dontcare   0.670       food=dontcare   |
   | pricerange=dontcare  0.646     pricerange=none   |
   |                Requestable                       |
   |         Prediction   Prob.        Ground Truth   |
   |       ------------   -----        ------------   |
   |          area=none   0.991           area=none   |
   |          food=none   0.991           food=none   |
   |    pricerange=none   0.994     pricerange=none   |
   |       address=none   0.988        address=none   |
   |         phone=none   0.994          phone=none   |
   |      postcode=none   0.993       postcode=none   |
   |   venue=not change   0.989    venue=not change   |

DB Match     : >5

Generated    : </s> there are [VALUE_COUNT] restaurant -s in the [VALUE_AREA] [SLOT_AREA] . would you like more information ? </s>

Ground Truth : </s> are you looking for a particular [SLOT_PRICERANGE] ? </s>


---------------------------- Turn 2 ----------------------------
Sample     : [12]
Prior dist.: [12]
probability: [1.]

User Input :    a moderate -ly priced restaurant would be good .
           :    a [VALUE_PRICERANGE] -ly priced restaurant would be good .

Belief Tracker :
   |                 Informable                          |
   |         Prediction   Prob.           Ground Truth   |
   |       ------------   -----           ------------   |
   |          area=west   0.867              area=west   |
   |      food=dontcare   0.861          food=dontcare   |
   | pricerange=moderate  0.992    pricerange=moderate   |
   |                Requestable                          |
   |         Prediction   Prob.           Ground Truth   |
   |       ------------   -----           ------------   |
   |          area=none   0.991              area=none   |
   |          food=none   0.991              food=none   |
   |    pricerange=none   0.994        pricerange=none   |
   |       address=none   0.988           address=none   |
   |         phone=none   0.994             phone=none   |
   |      postcode=none   0.993          postcode=none   |
   |   venue=not change   0.989       venue=not change   |

DB Match     : 3

Generated    : </s> [VALUE_NAME] is a [VALUE_FOOD] restaurant in the [VALUE_AREA] [SLOT_AREA] . would you like their information ? </s>

Ground Truth : </s> do you have a [SLOT_FOOD] preference ? </s>
---------------------------- Turn 3 ----------------------------
Sample     : [12]
Prior dist.: [12]
probability: [1.]

User Input :    no preference
           :    [VALUE_DONTCARE]

Belief Tracker :
   |                 Informable                          |
   |         Prediction   Prob.           Ground Truth   |
   |       ------------   -----           ------------   |
   |          area=west   0.481              area=west   |
   |      food=dontcare   0.999          food=dontcare   |
   | pricerange=moderate  0.877    pricerange=moderate   |
   |                Requestable                          |
   |         Prediction   Prob.           Ground Truth   |
   |       ------------   -----           ------------   |
   |          area=none   0.991              area=none   |
   |          food=none   0.991              food=none   |
   |    pricerange=none   0.994        pricerange=none   |
   |       address=none   0.988           address=none   |
   |         phone=none   0.994             phone=none   |
   |      postcode=none   0.993          postcode=none   |
   |   venue=not change   0.988       venue=not change   |

DB Match     : 3

Generated    : </s> [VALUE_NAME] is a [VALUE_FOOD] restaurant in the [VALUE_AREA] [SLOT_AREA] . would you like their [SLOT_PHONE] ? </s>

Ground Truth : </s> how about [VALUE_NAME] ? they serve [VALUE_PRICERANGE] -ly priced [VALUE_FOOD] [SLOT_FOOD] on the [VALUE_AREA]
[SLOT_AREA] . </s>


---------------------------- Turn 4 ----------------------------
Sample     : [5]
Prior dist.: [5]
probability: [1.]

User Input :    can i please have their address and phone number ?
           :    can i please have their [SLOT_ADDRESS] and [SLOT_PHONE] ?

Belief Tracker :
   |                 Informable                          |
   |         Prediction   Prob.           Ground Truth   |
   |       ------------   -----           ------------   |
   |          area=west   0.388              area=west   |
   |      food=dontcare   0.997          food=dontcare   |
   | pricerange=moderate  0.881    pricerange=moderate   |
   |                Requestable                          |
   |         Prediction   Prob.           Ground Truth   |
   |       ------------   -----           ------------   |
   |          area=none   0.991              area=none   |
   |          food=none   0.991              food=none   |
   |    pricerange=none   0.994        pricerange=none   |
   |      address=exist   0.961          address=exist   |
   |        phone=exist   0.951            phone=exist   |
   |      postcode=none   0.993          postcode=none   |
   |   venue=not change   0.989       venue=not change   |

DB Match     : 3

Generated    : </s> [VALUE_NAME] is located at [VALUE_ADDRESS] and their [SLOT_PHONE] is [VALUE_PHONE] . </s>

Ground Truth : </s> [VALUE_NAME] s [SLOT_ADDRESS] is [VALUE_ADDRESS] , their [SLOT_PHONE] is [VALUE_PHONE] . </s>
```



```
--------------------------- Turn 5 ---------------------------
Sample      : [1]
Prior dist.: [1]
probability: [1.]

User Input :    thank you , goodbye
           :    thank you , goodbye

Belief Tracker :
  |                  Informable|                              |
  |       Prediction    Prob.            Ground Truth |
  |     ------------    -----            ------------ |
  |        area=none    0.360               area=west |
  |    food-dontcare    0.997           food-dontcare |
  | pricerange=moderate 0.894     pricerange=moderate |
  |                  Requestable                       |
  |       Prediction    Prob.            Ground Truth |
  |     ------------    -----            ------------ |
  |        area=none    0.991               area=none |
  |        food-none    0.991               food-none |
  |   pricerange-none   0.994         pricerange-none |
  |     address=none    0.988            address=none |
  |       phone-none    0.994              phone-none |
  |    postcode=none    0.993           postcode=none |
  | venue-not change    0.989        venue-not change |

DB Match      : >5

Generated     : </s> you are welcome . goodbye . </s>

Ground Truth : </s> thank you for using our system . goodbye . </s>
```

Figure 12: Dialogue 85 out of 132 with 5 turns. The system is successful in generating a valid response. Notice the change in response.



```
------------------------ Dialogue 8 ---------------------------
--------------------------- Turn 0 ---------------------------
Sample      : [29]
Prior dist.: [29]
probability: [1.]

User Input :    i need a restaurant that serves european food please
           :    i need a restaurant that serves [VALUE FOOD] [SLOT FOOD] please

Belief Tracker :
  |                  Informable                  |
  |       Prediction    Prob.       Ground Truth |
  |     ------------    -----       ------------ |
  |        area-none    0.944          area-none |
  |    food=european    1.000      food-european |
  |  pricerange-none    0.518    pricerange-none |
  |                  Requestable                 |
  |       Prediction    Prob.       Ground Truth |
  |                                              |
  |        area-none    0.991          area-none |
  |        food-none    0.991          food-none |
  |  pricerange-none    0.994    pricerange-none |
  |     address=none    0.988       address=none |
  |       phone-none    0.994         phone-none |
  |    postcode-none    0.993      postcode-none |
  | venue=not change    0.989   venue=not change |

DB Match      : 3

Generated     : </s> there are [VALUE_COUNT] restaurant -s that serve [VALUE_FOOD] [SLOT_FOOD] in the [VALUE_AREA] [SLOT_AREA] . do you
have a preference ? </s>

Ground Truth : </s> what [SLOT_AREA] are you looking for ? </s>

--------------------------- Turn 1 ---------------------------
Sample      : [25]
Prior dist.: [25]
probability: [1.]

User Input :    anything in the area would be fine
           :    anything in the [SLOT AREA] would be fine

Belief Tracker :
  |                  Informable                  |
  |       Prediction    Prob.       Ground Truth |
  |                                              |
  |        area=none    0.362      area=dontcare |
  |    food-european    0.997      food-european |
  |  pricerange-none    0.494    pricerange-none |
  |                  Requestable                 |
  |       Prediction    Prob.       Ground Truth |
  |     ------------    -----       ------------ |
  |        area-none    0.991          area-none |
  |        food-none    0.991          food-none |
  |  pricerange=none    0.994    pricerange=none |
  |     address-none    0.988       address-none |
  |       phone=none    0.994         phone=none |
  |    postcode-none    0.993      postcode-none |
  | venue=not change    0.989   venue=not change |

DB Match      : 3

Generated     : </s> [VALUE_NAME] is a [VALUE_FOOD] restaurant in the [VALUE_AREA] [SLOT_AREA] . </s>

Ground Truth : </s> [VALUE NAME] is a nice restaurant serving [VALUE FOOD] [SLOT FOOD] , is in the [VALUE AREA] [SLOT AREA] and is
[VALUE_PRICERANGE]  ly priced . </s>


--------------------------- Turn 2 ---------------------------
Sample      : [29]
Prior dist.: [29]
probability: [1.]

User Input :    i am looking for something in the expensive price range .
           :    i am looking for something in the [VALUE_PRICERANGE] [SLOT_PRICERANGE] .

Belief Tracker :
  |                  Informable                  |
  |       Prediction    Prob.       Ground Truth |
  |     ------------    -----       ------------ |
  |        area=none    0.819      area=dontcare |
  |    food-european    0.997      food-european |
  | pricerange=expensive    0.993   pricerange=expensive |
  |                  Requestable                 |
  |       Prediction    Prob.       Ground Truth |
  |     ------------    -----       ------------ |
  |        area=none    0.991          area=none |
  |        food-none    0.991          food-none |
  |  pricerange=none    0.994    pricerange=none |
  |     address-none    0.988       address-none |
  |       phone-none    0.994         phone-none |
  |    postcode-none    0.993      postcode-none |
  | venue-not change    0.989   venue-not change |

DB Match      : 2

Generated     : </s> [VALUE_NAME] is a [VALUE_FOOD] restaurant in the [VALUE_AREA] [SLOT_AREA] . </s>

Ground Truth : </s> i am sorry there is no [VALUE_COUNT] serving [VALUE_FOOD] [SLOT_FOOD] in the [VALUE_PRICERANGE] [SLOT_PRICERANGE]
</s>
```

Figure 13: Dialogue 8 out of 132 with 2 turns. The system fails to in generating a correct response. Notice in turn 2 how the ground truth is different from what is generated

As seen above, the LIDM model was tested and it was determined that the success rate was lower than the other two models. In order to increase the success rate reinforcement learning fine-tuning was applied. The command below, takes the tested LIDM and adds an extra component to it. Applying this step to the LIDM resulted in a higher success rate and allowed the model to understand the intention behind each input. As seen in Table 1, the success rate increased to 83.8% and the BLEU score was 0.233. Which shows that even though the LIDM without RL was not able to query out correct responses, the LIDM with RL is able to predict better responses with a high degree of quality in each response.

```
cp ./notebooks/model/CamRest.LIDM.model
model/CamRest.LIDM-RL.model


THEANO_FLAGS="optimizer=fast_compile"
python ./notebooks/nndial.py -config
./notebooksconfig/LIDM-RL.cfg -mode rl
```

Code 11: command used to test the *latent intention dialogue model* with reinforced learning

7 Conclusion

This paper investigated the claim that the *latent intention dialogue model (LIDM)* outperforms the *vanilla neural dialogue model* and the *attention-based neural dialogue model*. The analysis of each model displayed that the results of the *latent intention dialogue model* were the highest amongst the three. It resulted in a higher success rate for predicting the underlying truth behind each input. Not only was the rate of success high but the model was also able to carry out quality responses with the user. This was only true for when reinforcement learning fine tuning was applied to the model. This meant that the model received a reward each time a correct response was generated. Looking at the results presented above it is safe to say that the *latent intention dialogue model* is a step forward towards the future of dialogue agents.

To conclude that the *latent intention dialogue model* is indeed a better approach to model dialogue, training, validation and testing was conducted. During the process of setting up the model, the biggest challenge was to grasp a deeper understanding of the models suggested, the architectures being used and how they are connected. In order to overcome this barrier, rigorous research was conducted on dialogue systems and the struggles of developing one. There were no concise instructions given on how to set up each model which resulted in a tedious and time-consuming task. Since the author directed me towards his GitHub page, it did not consist of a list of instructions that should be followed. As a result, the method of trial and error was applied to get the models up and running. The training and validation process were difficult as the python scripts gave an output as an error. The most common error was associated to the path for each model and architecture. Therefore, the model was trained, validated and tested multiple times as the scripts were not properly configured.

The next step towards further analyzing the capability of the *latent intention dialogue model* would be to test different datasets. In CIS 4900 the *latent intention dialogue model* was set up to mimic the results to that of the author's. In CIS 4910 the aim should be to prove the validity of the *latent intention dialogue model* by applying it to different types of datasets. A key factor to keep in mind would be to model data that is large. This will not only reduce bias but will allow us to make better predictions regarding the limitations of the *latent intention dialogue model*. Another task I would like to conduct is to compare the discrete latent variable to the continuous latent variable and note the difference. The article Latent Intention Dialogue Model makes use of a discrete latent variable with reinforcement learning due to high variability. In order to reduce variability reinforcement learning was applied. What if reinforcement learning was applied to a continuous latent variable how would the results vary and determine whether a continuous latent variable could be applied to this model. The ideal achievement would be to add a component to the model that categorizes each data value and querys out a list of possible options for the user to choose from. For example, if there are two restaurants in a specific area the system would query out a list from which the user can choose from. These are some ideas that could be further analyzed. Depending on the level of difficulty and the time available, these ideas can change.

At the end, conducting this research contributed to my knowledge by allowing me to explore a new concept within machine learning. I was introduced to the idea of dialogue systems and the possibilities within. The underlying truth behind the *vanilla neural dialogue model*, the *attention-based neural dialogue* model and the *latent intention dialogue model* was investigated by carrying out training, validation and testing. The conclusion that the *latent intention dialogue model* is a better suited model is presented in this paper.

References

Bahdanau, D., Cho, K., and Bengio, Y. (2015) Neural machine translation by jointly learning to align and translate. *Proceedings of the 2015 International Conference on Learning Representations (ICLR).* doi: 1409.0473/v7

Chen, H., Liu, X., Yin, D., & Tang, J. (2017). A Survey on Dialogue Systems: Recent Advances and New Frontiers. *ACM SIGKDD Explorations Newsletter,* 19(2), 25-35. doi:10.1145/3166054.3166058

Cho, K., Merrienboer, B. V., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H., & Bengio, Y. (2014). Learning Phrase Representations using RNN Encoder–Decoder for Statistical Machine Translation. *Proceedings of the 2014 Conference on Empirical*

Li, Jiwei, Monroe, Will, Ritter, Alan, Jurafsky, Dan, Galley, Michel, and Gao, Jianfeng. (2016). Deep reinforcement learning for dialogue generation. *Proceedings of the 2016 Conference on Empirical Method in Natural Language Processing*., pp. 1192– 1202

Liu, B., & Lane, I. (2018). End-to-End Learning of Task-Oriented Dialogs. *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Student Research Workshop*. doi: 10.18653/v1/n18-4010

Papineni, K., Roukos, S., Ward, T., & Zhu, W.-J. (2002). Bleu. Proceedings of the 40th Annual Meeting on Association for Computational Linguistics - ACL 02. doi: 10.3115/1073083.1073135

Wen, T., Gasic, M., Mrkšić, N., Barahona, L. M., Su, P., Ultes, S., . . . Young, S. (2016). Conditional Generation and Snapshot Learning in Neural Dialogue Systems. *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*. doi:10.18653/v1/d16-1233

Wen, T., Miao, Y., Blunsom, P., Young, S. (2017a). Latent Intention Dialogue Models. *Proceedings of the 34th International Conference on Machine Learning,* 1-10. doi: 1705.10229/v1

Wen, T., Gasic, M., Mrkšić, N., Barahona, L. M., Su, P., Ultes, S., . . . Young, S. (2017b). A Network-based End-to-End Trainable Task-oriented Dialogue System. *Proceedings of 2017 Conference on European Chapter of the Association of Computational Linguistics.* doi:1604.0456/v3