Surface Text based Dialogue Models for Virtual Humans

Sudeep Gandhe and **David Traum**
USC Institute for Creative Technologies,
12015 Waterfront Drive, Playa Vista, CA 90094, USA
srgandhe@gmail.com, traum@ict.usc.edu

Abstract

We present virtual human dialogue models which primarily operate on the surface text level and can be extended to incorporate additional information state annotations such as topics or results from simpler models. We compare these models with previously proposed models as well as two human-level upper baselines. The models are evaluated by collecting appropriateness judgments from human judges for responses generated for a set of fixed dialogue contexts. Our results show that the best performing models achieve close to human-level performance and require only surface text dialogue transcripts to train.

1 Introduction

Virtual Humans (VH) are autonomous agents who can play the role of humans in simulations (Rickel and Johnson, 1999; Traum et al., 2005). For these simulations to be convincing these agents must have the ability to communicate with humans and other agents using natural language. Like other dialogue system types, different architectures have been proposed for virtual human dialogue systems. These architectures can afford different features and require different sets of resources. E.g., an information state based architecture such as the one used in SASO-ST (Traum et al., 2005) can model detailed understanding of the task at hand and progression of dialogue, but at the cost of requiring resources such as information state update rules and an annotated corpus or grammar to be able to map surface text to dialogue acts.

For some virtual human dialogue genres such as simple question-answering or some negotiation domains, a simple model of dialogue progression would suffice. In such a case we can build dialogue models that primarily operate on a surface text level. These models only require surface text dialogue transcripts as a resource, and don't require expensive manual update rules, grammars, or even extensive corpus annotation.

In this paper, we describe the construction and evaluation of several models for engaging in dialogue by *selecting* an utterance that has been seen previously in a corpus. We include one model that has been used for this task previously (Gandhe and Traum, 2007b), an adaptation of a model that has been used in a similar manner, though on hand-authored data sets, rather than data sets extracted automatically from a corpus (Leuski and Traum, 2008), as well as a new set of models, using perceptrons on surface text features as well as more abstract information state annotations such as topics. We also tackle the question of evaluating such dialogue models manually as well as automatically, starting with systematically analyzing various decisions involved in the evaluation process. We situate our work with respect to previous evaluation methods.

2 Related Work

The task of a dialogue model is to formulate an utterance given a dialogue context. There are two approaches towards formulating an utterance: *Generation*, where a response is compositionally created from elements of the information state, including the context of previous utterances, and *Selection*, where a response is chosen from previously seen set of responses. In (Gandhe and Traum, 2010), we examined the theoretical potential for the selection approach, looking at a wide variety of domains, and evaluating based on similarity between the actual utterance and the best match in the previously seen corpus. We saw a wide variance in scores across domains, both as to the similarity scores and improvement of scores as more data is considered. For task-oriented planning domains, such as Monroe (Stent, 2000) and

TRAINS (Heeman and Allen, 1994), as well as open conversation in Switchboard (Godfrey et al., 1992), the performance was very low. On the other hand, for more limited domains such as simple question-answering (Leuski et al., 2006) or role-play negotiation in a scenario, the performance was high, with METEOR scores averaging over 0.8.

One possible selection criterion is to assume that the most appropriate response is the most probable response according to a model trained on human-human dialogues. More formally, let there be a dialogue $\langle u_1, u_2, \ldots, u_{t-1}, u_t, \ldots, u_T \rangle$, where utterance u_t appears in $context_t = \langle u_1, u_2, \ldots, u_{t-1} \rangle$. If we have a dialogue model P estimated from the training corpus then the formulated response u_q for some unseen $context_q$ is given by,

$$u_t = \underset{i}{\operatorname{argmax}}\ P(u_i|context_t) \quad \forall u_i \in U_{possible}$$

where $U_{possible}$ is a set of all possible response utterances. Ideally we would like to estimate a probability distribution P, but since it's hard to estimate and we only need *argmax* for this application, we approximate P with a ranking function. We can compare previous work within this framework.

In our previous work (Gandhe and Traum, 2007a), we used context similarity as the ranking function P (see section 3.1 for details). This model is trained from in-domain surface text dialogue transcripts. Leuski et al. (2006) model P as cross-lingual relevance, where the task of selecting an appropriate response is seen as cross-lingual information retrieval where the response utterance u_t is the relevant document and the $context_t$ is treated as a query from different language. This model has been applied to simple question answering where context is the previous utterance and the training data is manually annotated question-answer pairs. DeVault et al. (2011) have proposed to use a multi-class classification model (such as maximum entropy) for estimating P. Their method restricts the set $U_{possible}$ to a set of canonical utterances which represent distinct dialogue acts. This allows for a limited number of classes ($|U_{possible}|$) and also maximizes the number of distinct contexts seen per utterance. This model is also trained from manually annotated utterance-context pairs and can additionally use manually created utterance paraphrases.

Apart from the models discussed above which have been mainly applied to dialogue domains situated in a story context, there has been some work in surface text based dialogue models for open domains. Ritter et al. (2011) use information retrieval based and statistical machine translation (SMT) based approaches towards predicting the next response in Twitter conversations. Also Chatbots typically use surface text based processing such as string transformations (e.g., AIML rules (Wallace, 2003)). Such rules can also be learned from a dialogue corpus (Abu Shawar and Atwell, 2005). Systems employing SMT or string transformation rules are formulating a response by *Generation* approach and it can be frequently ungrammatical or incoherent, unlike the selection approach which will always pick something that someone has once said (even though it might be inappropriate in the current context).

3 Dialogue Models

3.1 Nearest Context

In previous work (Gandhe and Traum, 2007a), we modeled P as,

$$P(u_i|context_q) \approx Sim(context_i, context_q)$$

where $context_i$ is the context in which utterance u_i was seen in training corpus and Sim is context similarity in a customized vector-space model. The model restricts the set of possible response utterances ($U_{possible}$) to the set of utterances observed in the training data (U_{train}). The context is approximated using the previous two utterances (one from each speaker). This model does not use the contents of the utterance u_i itself.

3.2 Cross-lingual Relevance Model

Leuski et al. (2006) model P as a cross-lingual relevance model. This model takes into account the content of the utterance u_i as well as the content of the context. It does not impose any restriction on $U_{possible}$, but in practice it is restricted to the set of utterances in the training data. The model allows the context to be composed of multiple fields, each with its own weight. This allows us to extend the model where the context is approximated by the previous two utterances. The weights need to be learned using a held-out development set, which presents a challenge in the case of multiple fields (possible if we add more information state annotations), modest amounts of training data and

non-availability of an automatic and reliable estimate of the model's performance. Here, for the first time, we apply this model to automatically extracted pairs of utterance-context and evaluate it. For our model we used the implementation that is available as a part of NPCEditor (Leuski and Traum, 2011) and manually set the field weights corresponding to the two previous utterances to be equal (0.5).

3.3 Perceptron

As discussed earlier, the task of selecting the most appropriate response can be viewed as multi-class classification. But there are a couple of issues. First, since we operate at the surface text level, each unique response utterance will be labeled as a separate class. The number of classes is the number of unique utterances seen in the training set, which is relatively large. As the training data grows, the number of classes will increase. Second, there are very few examples (on average a single example) per class. We need a classifier that can overcome these issues.

The perceptron algorithm and its variants – voted perceptron and averaged perceptron are well known classification models (Freund and Schapire, 1999). They have been extended for use in various natural language processing tasks such as part-of-speech tagging (Collins, 2002), parsing (Collins, 2004) and discriminative language modeling (Roark et al., 2007). Here we use the averaged perceptron model for mapping from dialogue context to an appropriate response utterance.

Collins (2002) outlines the following four components of a perceptron model:

- The training data. In our case it is a set of automatically extracted utterance-context pairs $\{\ldots, \langle u_i, context_i \rangle, \ldots\}$
- A function GEN$(context)$ that enumerates a set of all possible outputs (response utterances) for any possible input (dialogue context)
- A feature extraction function $\Phi : \langle u, context \rangle \rightarrow \mathbb{R}^d$ that is defined over all possible pairings of response utterances and dialogue contexts. d is the total number of possible features.
- A parameter vector $\bar{\alpha} \in \mathbb{R}^d$

Using such a perceptron model, the most appropriate response utterance (u_t) for the given dialogue context ($context_t$) is given by,

$$u_q = \underset{u_i \in \text{GEN}(context)}{\text{argmax}} \Phi(u_i, context_q) \cdot \bar{\alpha}$$

Algorithm 1 Perceptron Training Algorithm

Initialize: $t \leftarrow 0$; $\bar{\alpha}_0 \leftarrow 0$
for $iter = 1$ to *MAX_ITER* **do**
  for $i = 1$ to N **do**
    $r_i \leftarrow \underset{u \in \text{GEN}(context_i)}{argmax} \Phi(u, context_i) \cdot \bar{\alpha}_t$
    if $r_i \neq u_i$ **then**
      $\bar{\alpha}_{t+1} \leftarrow \bar{\alpha}_t + \Phi(u_i, context_i) - \Phi(r_i, context_i)$
    else
      $\bar{\alpha}_{t+1} \leftarrow \bar{\alpha}_t$
    end if
    $t \leftarrow t + 1$
  end for
end for
return $\bar{\alpha} \leftarrow (\sum_t \bar{\alpha}_t)/(\textit{MAX\_ITER} \times N)$

The parameter vector $\bar{\alpha}$ is trained using the training algorithm described in Algorithm 1. The algorithm goes through the training data one instance at a time. For every training instance, it computes the best response utterance (r_i) for the context based on its current estimate of the parameter vector $\bar{\alpha}_t$. The algorithm changes the parameter vector only if it makes an error ($r_i \neq u_i$). The update drives the parameter vector away from the error (r_i) and towards the correct output (u_i). The final parameter vector $\bar{\alpha}$ is an average of all the intermediate $\bar{\alpha}_t$ values. The averaging of parameter vectors avoids overfitting.

The feature extraction function Φ can list any arbitrary features from the pair $\langle u, context \rangle$. We consider information state annotations (IS_t) along with the surface text corresponding to the previous two turns. The features could also include scores computed from other models, such as those presented in sections 3.1 and 3.2. Figure 1 illustrates an example context and utterance, and several features. We examine several sets of features, Surface text based features (Φ_S), Retrieval model based features (Φ_R), and Topic based features (Φ_T).

Surface text based features (Φ_S) are the features extracted from the surface text of the previous utterances in the dialogue context ($context_j$) and the response utterance (u_i). $\Phi_{S(d)}(u_x, u_y)$ extracts surface text features from two utterances – a response utterance (u_x) and an utterance (u_y) from the context that is (d) utterances away. There are four types of features we extract:

- $common\_term(d, w)$ features indicate the number of times a word w appears in both the utterances. The total number of possible features is $O(|V|)$ and we select a small subset of words ($Selected\_common(d)$) from the vocabulary.
- The $common\_term\_count(d)$ feature indicates the number of words that appear in both utterances.
- The $unique\_common\_term\_count(d)$ feature indicates the number of unique words that appear in both utterances.
- $cross\_term(d, w_x, w_y)$ features indicate the number of times the word w_x appears in the utterance u_x and the word w_y appears in the utterance u_y. The total possible number of such cross features is very large ($O(|V|^2)$), where $|V|$ is the utterance vocabulary size. In order to keep the training tractable and avoid overfitting, we select a small subset of cross_features ($Selected\_cross(d)$) from all possible features.

In this model, we perform feature selection by selecting the subsets $Selected\_cross(d)$ and $Selected\_common(d)$. The training algorithm requires evaluating the feature extraction (Φ_S) function for all possible pairings of response utterances and contexts. One simple feature selection criterion is to allow the features only appearing in *true pairings* of response utterance and context (i.e. features from $\Phi_S(\langle u_i, context_j\rangle)\ \forall i = j$). The subset $Selected\_common(d)$ for $common\_term$ features is selected by extracting features from only such *true pairings*.

For selecting $cross\_term(d, w_x, w_y)$ features we use only *true pairings* but we need to reduce this subset even further. We impose additional constraints based on the collection frequency of lexical events such as, $cf(w_x) > threshold_x$, $cf(w_y) > threshold_y$, $cf(\langle w_x, w_y\rangle) > threshold_{xy}$. Further reduction in size of the selected subset of $cross\_term$ features is achieved by ranking the features using a suitable ranking function and choosing the top n features. In this model, we rank the $cross\_term$ features based on pointwise mutual-information $pmi(\langle w_x, w_y\rangle)$ given by,

$$\log \frac{p(\langle w_x, w_y\rangle)}{p(w_x)p(w_y)} = \log \frac{\left(\frac{\#\langle w_x, w_y\rangle}{\#\langle \cdot,\cdot\rangle}\right)}{\left(\frac{\#\langle w_x, \cdot\rangle}{\#\langle \cdot,\cdot\rangle}\right) \cdot \left(\frac{\#\langle \cdot, w_y\rangle}{\#\langle \cdot,\cdot\rangle}\right)}$$

Summing up, $\Phi_{S(d)}(u_x, u_y) =$

$$\begin{aligned}
&\{cross_term(d, w_x, w_y) : w_x \in u_x \wedge \\
&\quad w_y \in u_y \wedge \langle w_x, w_y\rangle \in Selected_cross(d)\} \\
\cup\ &\{common_term(d, w) : w \in u_x \wedge w \in u_y \wedge \\
&\quad w \in Selected_common(d)\} \\
\cup\ &\{common_term_count(d)\} \\
\cup\ &\{unique_common_term_count(d)\}
\end{aligned}$$

Retrieval model based features (Φ_R) are the scores computed in a fashion similar to the *Nearest Context* model. $Sim(u_x, u_y)$ is a cosine similarity function for tf-idf weighted vector space representations of utterances and $Sim(context_a, context_b)$ is the same function from *Nearest Context* model. We define three features,

- $retrieval\_score = \max_{k=1}^{|\mathcal{L}|} Sim(context_j, context_k) \cdot Sim(u_i, u_k)$
- $context\_sim@best\_utt\_match = Sim(context_j, context_b)$ where, $b = \operatorname{argmax}_{k=1}^{|\mathcal{L}|} Sim(u_i, u_k)$
- $utt\_sim@best\_context\_match = Sim(u_i, u_b)$ where, $b = \operatorname{argmax}_{k=1}^{|\mathcal{L}|} Sim(context_j, context_k)$

$$\begin{aligned}
\Phi_R(\langle u_i, context_j\rangle) = \{&retrieval_score, \\
&context_sim@best_utt_match, \\
&utt_sim@best_context_match\}
\end{aligned}$$

Topic based feature (Φ_T) tracks the topic similarity between the topic of the dialogue context and the response utterance. A topic is marked as mentioned if a set of keywords triggering that topic have been previously mentioned in the dialogue. Each information state (IS) consists of a topic signature which can be viewed as a boolean vector representing mentions of topics.

$$\begin{aligned}
\Phi_T(\langle u_i, context_j\rangle) &= \{topic_similarity\} \\
topic_similarity &= cosine(IS_i, IS_j)
\end{aligned}$$

where, IS_i is the topic and is part of $context_i$ which is the context associated with the utterance u_i.

The perceptron model presented here allows novel combinations of resources such as combining surface text transcripts with information state annotations for tracking topics in the conversation. As compared to the generative cross-lingual relevance model approach, the perceptron model is a discriminative model. It is also a parametric model and the inference requires linear time with respect to the size of candidate utterances ($|\textsc{gen}(context)|$) and the number of features ($|\bar{\alpha}|$). Although, computing some of the features themselves (e.g., Φ_R features) requires linear time with

		⋮	
$context_j$	$[u_{j(-2)}]$	**Doctor**	you are the threat i need protection from you
	$[u_{j(-1)}]$	**Captain**	no no you do you do not need protection from me i am here to help you uh what i would like to do is move your your clinic to a safer location and uh give you money and medicine to help build it
$utterance$	$[u_i]$	**Doctor**	i have no way of moving

$$\Phi_S(\langle u_i, context_j \rangle) = \{\ cross\_term(-2, \text{“moving”}, \text{“need”}) = 1,$$
$$common\_term(-2, \text{“i”}) = 1,$$
$$common\_term\_count(-2) = 1, unique\_common\_term\_count(-2) = 1,$$
$$cross\_term(-1, \text{“moving”}, \text{“give”}) = 1,$$
$$common\_term(-1, \text{“i”}) = 1, common\_term(-1, \text{“no”}) = 1,$$
$$common\_term\_count(-1) = 2, unique\_common\_term\_count(-1) = 2,$$
$$retrieval\_score = 0.198, context\_sim@best\_utt\_match = 0.198,$$
$$utt\_sim@best\_context\_match = 0,$$
$$topic\_similarity = 0.667\ \}$$

Figure 1: Features extracted from a context ($context_j$) and a response utterance (u_i)

respect to the size of the training data. The perceptron model can rank an arbitrary set of utterances given a dialogue context. But some of the features (e.g., $topic\_similarity$) require that the utterance $u_i (u_i \in |\text{GEN}(context)|)$ be associated with a known context ($context_i$). For all our models we use $\text{GEN}(context) = U_{train}$.

We have implemented three different variants of the perceptron model based on the choice of features used. ***Perceptron(surface)*** model uses only surface text features ($\Phi = \Phi_S$). The other two models are ***Perceptron(surface+retrieval)*** where $\Phi = \Phi_S \cup \Phi_R$ and ***Perceptron(surface+retrieval+topic)*** where $\Phi = \Phi_S \cup \Phi_R \cup \Phi_T$.

Figure 2 shows a schematic representation of these models along with the set of resources being used by each model. The figure also shows the relationships between these models. The arrows point from a less informative model to a more informative model and the annotations on these arrows indicate the additional information used.

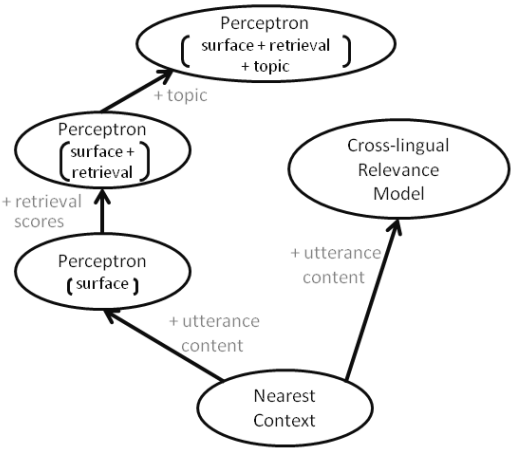

Figure 2: A schematic representation of implemented unsupervised dialogue models and the relationships between the information used by their ranking functions.

4 Evaluation

For the experiments reported in this paper, we used the human-human spoken dialogue corpus collected for the project SASO-ST (Traum et al., 2005). In this scenario, the trainee acts as an Army Captain negotiating with a simulated doctor to convince him to move his clinic to another location. The corpus is a collection of 23 roleplay dialogues and 13 WoZ dialogues lasting an average of 40 turns (a total of ≈ 1400 turns and ≈ 30k words).

We perform a *Static Context* evaluation (Gandhe, 2013). In *Static Context* evaluation, all the dialogue models being evaluated receive the same set of contexts as input. These dialogue contexts are extracted from actual in-domain

human-human dialogues and are not affected by the dialogue model being evaluated. For every turn whose role is to be played by the system, we predict the most appropriate response in place of that turn given the dialogue context.

Since the goal for virtual humans is to be as human-like as possible, a suitable evaluation metric is how appropriate or human-like the responses are for a given dialogue context. The evaluation reported here employs human judges. We set up a simple subjective 5-point likert scale for rating appropriateness – 1 being a very inappropriate nonsensical response and 5 being a perfectly appropriate response.

We built five dialogue models to play the role of the doctor in SASO-ST domain, viz.: *Nearest Context* (section 3.1), *Cross-lingual Relevance Model* (section 3.2) and three *perceptron* models (section 3.3) with different feature sets. These dialogue models are evaluated using 5 in-domain human-human dialogues from the training data (2 roleplay and 3 WoZ dialogues, referred to as test dialogues). A dialogue model is trained in a leave-one-out fashion where the training data consists of all dialogues except the one test dialogue that is being evaluated. A dialogue model trained in this fashion is then used to predict the most appropriate response for every context that appears in the test dialogue. This process is repeated for each test dialogue and for each dialogue model being evaluated. In this evaluation setting, the actual response utterance found in the original human-human dialogue may not belong to the set of utterances being ranked by the dialogue model. We also compare these five dialogue models with two human-level upper baselines. Figure 4 in the appendix shows some examples of utterances returned by a couple of the models.

4.1 Human-level Upper Baselines

In order to establish an upper baseline for human-level performance for the evaluation task, we conducted a wizard data collection. We asked human volunteers (wizards) to perform a similar task to that performed by the dialogue models being evaluated. The wizard is presented with a set of utterances (U_{train}) and is asked to select a subset from these that will be appropriate as a response for the presented dialogue context. Compared to this, the task of the dialogue model is to select a single most appropriate response for the given context.

DeVault et al. (2011) carried out a similar wizard data collection but at the dialogue act level, where wizards were asked to select only one response dialogue act for each dialogue context. Their findings suggest that there are several valid response dialogue acts for a dialogue context. A specific dialogue act can be realized in several ways at the surface text level. For these reasons we believe that for a given dialogue context there are often several appropriate response utterances at the surface text level. In our setting the dialogue models work at the surface text level and hence the wizards were asked to select a subset of surface text utterances that would be appropriate responses. Each wizard was asked to select several (ideally between five and ten, but always at least one) appropriate responses for each dialogue context. Four wizards participated in this data collection with each wizard selecting responses for the contexts from the same five human-human test dialogues. The set of utterances to chose from (U_{train}) for every test dialogue was built in the same leave-one-out fashion as used for evaluating the implemented dialogue models.

There are a total of 89 dialogue contexts where the next turn belongs to *doctor*. As expected, wizards frequently chose multiple utterances as appropriate responses (mean = 7.80, min = 1, max = 25).

This data collected from wizards is used to build two human-level upper-baseline models for the task of selecting a response utterance given a dialogue context:

Wizard Max Voted model returns the response which gets the maximum number of votes from the four wizards. Ties are broken randomly.

Wizard Random model returns a random utterance from the list of all utterances marked as appropriate by one of the wizards.

4.2 Comparative Evaluation of Models

We performed a static context evaluation using four judges for the above-mentioned two human-level baselines (*Wizard Random* and *Wizard Max Voted*) and five dialogue models (*Nearest Context*, *Cross-lingual Relevance Model* and three *perceptron* models), as described in section 3.3. We tune the parameters used for the perceptron

models based on the automatic evaluation metric, *Weak Agreement* (DeVault et al., 2011). According to this evaluation metric a response utterance is judged as perfectly appropriate (a score of 5) if any of the wizards chose this response utterance for given context and inappropriate (a score of 0) otherwise. The *Perceptron(surface)* model was trained using 30 iterations, the *Perceptron(surface+retrieval)* using 20 iterations, and the *Perceptron(surface+retrieval+topic)* was trained using 25 iterations. For all perceptron models we used $threshold_x = threshold_y = threshold_{xy} = 3$.

For a comparative evaluation of dialogue models, we need an evaluation setup where judges could see the complete dialogue context along with the response utterances generated by the dialogue models to be evaluated. In this setup, we show all the response utterances next to each other for easy comparison and we do not show the actual response utterance that was encountered in the original human-human dialogue. We built a web interface for collecting appropriateness ratings that addresses the above requirements. Figure 3 shows the web interface used by the four judges to evaluate the appropriateness of response utterances for given dialogue context. The appropriateness was rated on the same scale of 1 to 5. The original human-human dialogue (roleplay or WoZ) is shown on the left hand side and the response utterances from different dialogue models are shown on the right hand side. In cases where different dialogue models produce the same surface text response only one candidate surface text is shown to judge. Once the judge has rated all the candidate responses they can proceed to the next dialogue context. This setting allows for comparative evaluation of different dialogue models. The presentation order of responses from different dialogue models is randomized. Two of the judges also performed the role of the wizards in our wizard data collection as outlined in section 4.1, but the wizard data collection and the evaluation tasks were separated by a period of over 3 months.

Table 1 shows the results of our comparative evaluation for each judge and averaged over all judges. We also computed inter-rater agreement for individual ratings for all response utterances using Krippendorff's α (Krippendorff, 2004). There were a total of $n = 397$ distinct response utterances that were judged by the evaluators. The Krippendorff's α for all four judges was 0.425 and it ranges from 0.359 to 0.495 for different subsets of judges. The value of α indicates that the inter-rater agreement is substantially above chance ($\alpha > 0$), but indicates a fair amount of disagreement, indicating that judging appropriateness is a hard task even for human judges. Although there is low inter-rater agreement at the individual response utterance level there is high agreement at the dialogue model level. Pearson's correlation between the average appropriateness for different dialogue models ranges from 0.928 to 0.995 for different pairs of judges.

We performed a paired Wilcoxon test to check for statistically significant differences in different dialogue models. *Wizard Max Voted* is significantly more appropriate than all other models ($p < 0.001$). *Wizard Random* is significantly more appropriate than *Cross-lingual Relevance Model* ($p < 0.05$) and significantly more appropriate than the three perceptron models as well as *Nearest Context* model ($p < 0.001$). *Cross-lingual Relevance Model* is significantly more appropriate than *Nearest Context* ($p < 0.01$). All other differences are not statistically significant at the 5 percent level.

We found that adding topic annotations did not help. This is in contrast with previous observation (Gandhe and Traum, 2007b), where topic information helped when evaluation was performed in *Dynamic Context* setting. In *Dynamic Context* setting, the dialogue model is used in an online fashion where the response utterances it generates become part of the dialogue contexts with respect to which the subsequent responses are predicted and evaluated. The topic information ensures systematic progression of dialogue. But for static context evaluation such help is not required as the dialogue contexts are extracted from human human dialogues and are fixed.

5 Conclusion

In this paper we introduced dialogue models that can be trained simply from in-domain surface text dialogue transcripts. Some of these models also allow for incorporating additional information state features such as topics or results of simpler models. We have evaluated the appropriateness of responses and have compared these models with two human-level baselines. Evaluating response appropriateness is highly subjective as

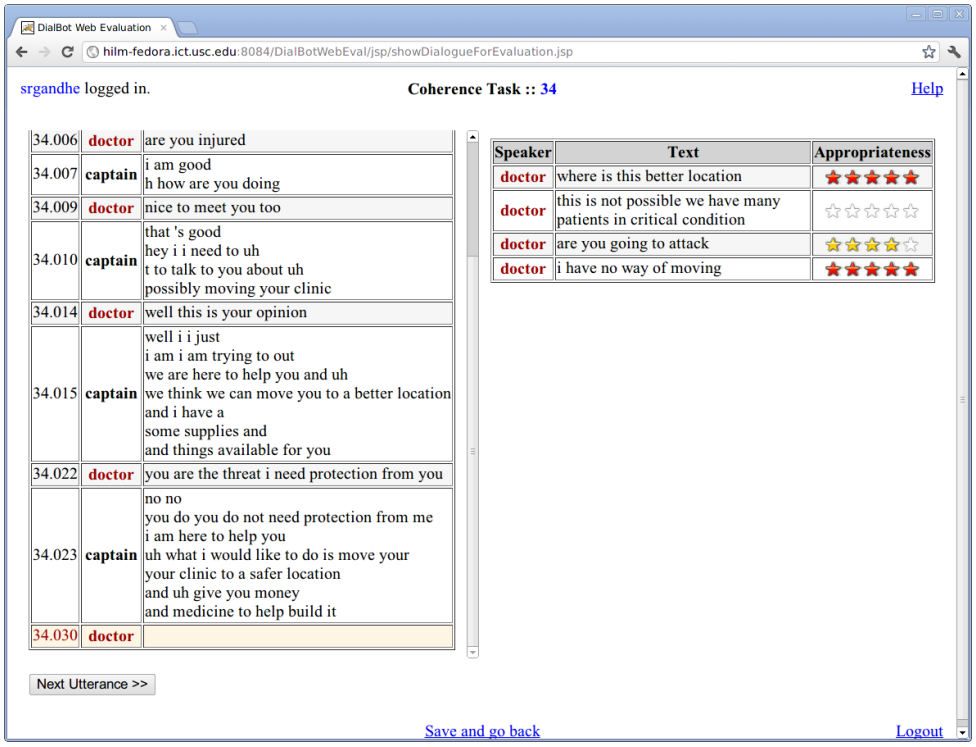

Figure 3: Screenshot of the user interface for static context comparative evaluation of dialogue models

Model	#Utts	Avg. appropriateness				Appropriateness (All judges)	
		Judge 1	Judge 2	Judge 3	Judge 4	Avg	stddev
Nearest Context	89	4.12	3.98	3.40	3.53	3.76	1.491
Perceptron(surface)	89	3.97	4.11	3.51	3.62	3.80	1.445
Perceptron (surface+retrieval)	89	4.26	4.12	3.51	3.72	3.90	1.414
Perceptron (surface+retrieval+topic)	89	4.21	4.09	3.51	3.57	3.85	1.433
Cross-lingual Relevance Model	89	4.28	4.31	3.70	3.91	4.05	1.314
Wizard Random	89	4.55	4.55	4.03	4.16	4.32	1.153
Wizard Max Voted	89	4.76	4.84	4.40	4.52	4.63	0.806

Table 1: Offline comparative evaluation of dialogue models.

can be seen from the fact that utterances which receive more wizard votes (*Wizad Max Voted*) receive significantly higher appropriateness ratings than those which receive fewer votes (*Wizard Random*). The performance of best performing dialogue models are close to human-level baselines. In future we plan to use larger datasets which should be easy, since no additional annotations are required for training these dialogue models.

Acknowledgments

The effort described here has been sponsored by the U.S. Army. Any opinions, content or information presented does not necessarily reflect the position or the policy of the United States Government, and no official endorsement should be inferred.

References

Bayan Abu Shawar and Eric Atwell. 2005. Using corpora in machine-learning chatbot systems. *International Journal of Corpus Linguistics*, 10:489–516.

Michael Collins. 2002. Discriminative training methods for hidden markov models: theory and experiments with perceptron algorithms. In *Proceedings of the ACL-02 conference on Empirical methods in natural language processing - Volume 10*, EMNLP '02, pages 1–8, Stroudsburg, PA, USA. Association for Computational Linguistics.

Michael Collins, 2004. *Parameter estimation for statistical parsing models: theory and practice of distribution-free methods*, pages 19–55. Kluwer Academic Publishers, Norwell, MA, USA.

David DeVault, Anton Leuski, and Kenji Sagae. 2011. Toward learning and evaluation of dialogue policies with text examples. In *Proceedings of the SIGDIAL 2011 Conference*, pages 39–48, Portland, Oregon, June. Association for Computational Linguistics.

Yoav Freund and Robert E. Schapire. 1999. Large margin classification using the perceptron algorithm. *Mach. Learn.*, 37:277–296, December.

Sudeep Gandhe and David Traum. 2007a. Creating spoken dialogue characters from corpora without annotations. In *Proceedings of Interspeech-07*.

Sudeep Gandhe and David Traum. 2007b. First steps towards dialogue modeling from an un-annotated human-human corpus. In *5th Workshop on knowledge and reasoning in practical dialogue systems*, Hyderabad, India.

Sudeep Gandhe and David Traum. 2010. I've said it before, and i'll say it again: an empirical investigation of the upper bound of the selection approach to dialogue. In *Proceedings of the 11th Annual Meeting of the Special Interest Group on Discourse and Dialogue*, SIGDIAL '10, pages 245–248, Stroudsburg, PA, USA. Association for Computational Linguistics.

Sudeep Gandhe. 2013. *Rapid prototyping and evaluation of dialogue systems for virtual humans*. Ph.D. thesis, University of Southern California.

J. J. Godfrey, E. C. Holliman, and J. McDaniel. 1992. Switchboard: Telephone speech corpus for research and development. In *Proc. of ICASSP-92*, pages 517–520.

P. A. Heeman and J. Allen. 1994. The TRAINS 93 dialogues. TRAINS Technical Note 94-2, Department of Computer Science, University of Rochester.

Klaus Krippendorff. 2004. *Content Analysis, An Introduction to Its Methodology 2nd Edition*. Sage Publications.

Anton Leuski and David Traum. 2008. A statistical approach for text processing in virtual humans. In *Proccedings of 26th Army Science Conference*.

Anton Leuski and David Traum. 2011. Npceditor: Creating virtual human dialogue using information retrieval techniques. *AI Magazine*, 32(2):42–56.

Anton Leuski, Ronakkumar Patel, David Traum, and Brandon Kennedy. 2006. Building effective question answering characters. In *Proceedings of the 7th SIGdial Workshop on Discourse and Dialogue*, pages 18–27, Sydney, Australia, July. Association for Computational Linguistics.

Jeff Rickel and W. Lewis Johnson. 1999. Virtual humans for team training in virtual reality. In *Proceedings of the Ninth International Conference on Artificial Intelligence in Education*, pages 578–585. IOS Press.

Alan Ritter, Colin Cherry, and William B. Dolan. 2011. Data-driven response generation in social media. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, EMNLP '11, pages 583–593, Stroudsburg, PA, USA. Association for Computational Linguistics.

Brian Roark, Murat Saraclar, and Michael Collins. 2007. Discriminative n-gram language modeling. *Comput. Speech Lang.*, 21:373–392, April.

A. J. Stent. 2000. The monroe corpus. Technical Report 728, Computer Science Dept. University of Rochester.

David Traum, William Swartout, Jonathan Gratch, and Stacy Marsella. 2005. Virtual humans for non-team interaction training. In *AAMAS-05 Workshop on Creating Bonds with Humanoids*, July.

Richard Wallace. 2003. *AIML Overview*. ALICE A. I. Foundation.

	Original Human-Human Dialogue	Model response
Captain	hello how are you doing sir	
Doctor	what do you want i have patients waiting for me	hello what can i do for you mr
Captain	sir i come by to offer you some assistance ah on behalf of my commander ah we want to try to support you in this hospital and ah improving the health and welfare of the civilians that you have under your care	
Doctor	well this is your opinion	i see captain is this a is this a suggestion from your commander
Captain	i am sorry sir could you say again	
Doctor	well this is your opinion	you are the threat i need protection from you

(a) *Perceptron(surface)*

	Original Human-Human Dialogue	Model response
Captain	hello doctor perez	
Doctor	hello	hello what was your name captain dezois very nice to meet you i am sorry but i am very busy today so i only have a limited amount of time what can i help you with
Captain	i am captain xx	
Doctor	so do you need help	ok what do you want i have patients waiting for me
Captain	yes i have a very urgent matter to discuss with you	
Doctor	are you injured	pl please captain i i am sorry to cut you off but i really must uh go i was in the middle of examining a patient

(b) *Cross-lingual Relevance Model*

Figure 4: Example interaction for the dialogue models in static context setting. The second column shows the original human-human dialogue and the third column shows the dialogue model's response for the corresponding system turn.