Evaluation of Real-Time Deep Learning Turn-Taking Models for Multiple Dialogue Scenarios

Divesh Lala
Graduate School of Informatics
Kyoto University
Kyoto, Japan
lala@sap.ist.i.kyoto-u.ac.jp

Koji Inoue
Graduate School of Informatics
Kyoto University
Kyoto, Japan
inoue@sap.ist.i.kyoto-u.ac.jp

Tatsuya Kawahara
Graduate School of Informatics
Kyoto University
Kyoto, Japan
kawahara@i.kyoto-u.ac.jp

ABSTRACT

The task of identifying when to take a conversational turn is an important function of spoken dialogue systems. The turn-taking system should also ideally be able to handle many types of dialogue, from structured conversation to spontaneous and unstructured discourse. Our goal is to determine how much a generalized model trained on many types of dialogue scenarios would improve on a model trained only for a specific scenario. To achieve this goal we created a large corpus of Wizard-of-Oz conversation data which consisted of several different types of dialogue sessions, and then compared a generalized model with scenario-specific models. For our evaluation we go further than simply reporting conventional metrics, which we show are not informative enough to evaluate turn-taking in a real-time system. Instead, we process results using a performance curve of latency and false cut-in rate, and further improve our model's real-time performance using a finite-state turn-taking machine. Our results show that the generalized model greatly outperformed the individual model for attentive listening scenarios but was worse in job interview scenarios. This implies that a model based on a large corpus is better suited to conversation which is more user-initiated and unstructured. We also propose that our method of evaluation leads to more informative performance metrics in a real-time system.

CCS CONCEPTS

• **Human-centered computing → Human computer interaction (HCI)**; • **Computing methodologies → Supervised learning by classification**; *Discourse, dialogue and pragmatics*; • **General and reference** → *Evaluation*;

KEYWORDS

dialogue systems; turn-taking; evaluation methods; deep learning; neural networks

ACM Reference Format:
Divesh Lala, Koji Inoue, and Tatsuya Kawahara. 2018. Evaluation of Real-Time Deep Learning Turn-Taking Models for Multiple Dialogue Scenarios. In *ICMI '18: 2018 Int'l Conference on Multimodal Interaction, Oct. 16–20, 2018, Boulder, CO, USA.* ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/3242969.3242994

ICMI '18, October 16–20, 2018, Boulder, CO, USA

ACM ISBN 978-1-4503-5692-3/18/10…$15.00
https://doi.org/10.1145/3242969.3242994

1 INTRODUCTION

Spoken dialogue systems which are able to produce human-like conversation are a major goal for researchers. One major challenge is to be able to generate realistic mechanisms for turn-taking, which regulates who will speak and when they will do so [23]. There is a common goal of turn-taking for all languages to avoid overlaps in speech and to minimize the gaps between speaking turns [20].

Methods such as push-to-talk and magic words have been used in smartphones and smart speakers to know when the system should listen to the user. To determine if the user has ended their turn, a common approach is to wait until they are silent for a period of time. However, the context and task of the dialogue is also influential for turn-taking [21] and the requirements of a suitable turn-taking system will differ between a question-answering system and a conversational chatbot. Turn-taking is of particular importance to humanoid robots, because users will expect them to behave similar to a real human. However, human-like natural turn-taking is still a long way off in spoken dialogue systems.

Research in spoken dialogue systems has mainly addressed one specific problem in turn-taking, which is determining if the user has finished their speaking turn (commonly termed end-of-turn detection). The danger of interrupting the user mid-turn means that systems are often conservative and take a relatively long time to respond to users. This might be acceptable for question-answering systems, but for chat-like systems it means the interaction is not smooth.

Several efforts have been done to improve end-of-turn detection using various modalities such as prosody, linguistics, eye gaze and even respiration [1, 3, 7–10, 13, 14, 19]. Both the types of corpora and the results are varied. This means it is difficult to compare results over different research. Furthermore, the evaluation method in many of these papers is done using conventional metrics such as precision, F1 score and accuracy, but does not fully reflect the real-time capabilities of the system. This is a problem in turn-taking research because we are interested in knowing how fast a system can take to respond and the amount of errors it would generate.

We address this issue by proposing that conventional evaluation metrics should also be supplemented with a performance curve indicating average latency and false cut-in rate. This type of analysis is not new, but is often missing in turn-taking research. Using this metric, it is easier to estimate how well the system would perform in real-time. We will then show that the system may be significantly

improved when we use a finite-state turn-taking machine (FSTTM) approach [16] as opposed to a conventional label-based approach, if applied to a real-time system.

Once we have established this for evaluation, we use it to assess how well a generalized model trained on a wide variety of dialogues can improve on models which are trained only for a specific dialogue. Our hypothesis is that the pattern of turn-taking differs according to the type of scenario. For example, in a formal setting such as a job interview, the interviewee will strive not to take long pauses during their turn, whereas for more casual and spontaneous conversation this might be more acceptable. We propose that training a model which uses a range of conversational scenarios is generally more powerful, but we want to know the type of cases where it has the most benefit. We are not aware of existing research which has compared models in this way.

In this work, we use deep learning methods to detect the end-of-turn of the user using two modalities - acoustics and linguistics. We evaluate both models separately and then construct a multi-modal deep learning model which fuses the hidden states of both unimodal models. As far as we know, combining deep learning models with an FSTTM has not been done in previous work. Although some works recognize latency and false cut-in rate as being valuable metrics, they either only report one point on the performance curve [3], or do not apply an FSTTM to their model [13].

In the next section we describe the corpus used and the data we collected to train our models. Section 3 describes the deep learning models in detail and we present our evaluation method and results in Section 4 before a discussion and the conclusion of the paper. In this work we use Japanese as the target language.

2 DATA COLLECTION

The corpus we use for our experiments is a collection of one-to-one human-robot interaction sessions with the android robot ERICA. ERICA is a highly realistic robot with the ability to create facial expressions and lip synchronizations similar to a real human, and also move her upper body to execute gestures. Our goal is for ERICA to be able to autonomously engage in conversation using behaviors that are indistinguishable from a real human. Since we want to apply the turn-taking model to ERICA, we use her for our experiments.

The setup was a typical Wizard-of-Oz experiment. The role of ERICA was played by a hidden operator who is a voice actress. We had four different operators play the part of ERICA over the sessions. The operator's room was hidden from the subject, but the operator could see and hear the subject through cameras. For each session we gave some basic instructions to the operators depending on the scenario. The operators controlled ERICA's head movements through a hand-held controller, which triggered non-verbal behaviors such as nods. Figure 1 shows an example of the user interacting with ERICA in a conversational scenario.

There was a total of 105 sessions, with users over a wide range of ages and backgrounds. Each session had a particular scenario to structure the conversation.

DATING (32 sessions) ERICA plays the role of a single woman in a speed-dating simulation. The user talks about their personal interests and preferences with ERICA to try to impress her.

INTERVIEW (30 sessions) ERICA is a job interviewer with the user as a candidate for a job.

LISTENING (20 sessions) ERICA mainly listens to the user while they talk at length about a topic such as a memorable trip. She occasionally responds with questions for the user about their talk.

SECRETARY (19 sessions) ERICA is a university professor's secretary. The user wishes to see the professor in their office but they are currently away, so while they wait ERICA and the user chat casually. ERICA first asks questions about the user and then gives the user the chance to ask about herself.

GUIDE (4 sessions) ERICA explains our laboratory to the user.

Figure 1: A user in a conversational session with ERICA.

We can see that there are a wide range of scenarios and styles of conversation. For example, the job interview is well structured, with ERICA only asking questions and the user responding to them. Meanwhile, attentive listening is somewhat the opposite case, with the user doing a large amount of talking and ERICA only providing short responses and backchannels. The secretary and dating scenarios are in the middle of these, with the conversation being mixed-initiative. Our aim is to determine if a turn-taking model can be generalized to handle all these scenarios, and whether it is better performing than models which are trained on one specific type of dialogue.

The audio data collected from the sessions was captured by two microphones. One was placed in the hidden booth of ERICA's operator. The other was positioned between ERICA and the subject and captured the latter's voice. All sessions were transcribed including notation for backchannels, laughter and fillers.

We extracted IPUs (inter-pausal units) from all sessions, regardless of speaker. An IPU was defined as a segment of speech which did not contain a pause greater than 200ms. We labeled each IPU as being within-turn (WT), where the next IPU was from the same

Table 1: Corpus statistics

	Dialogue scenarios				
	DATING	INTERVIEW	LISTENING	SECRETARY	ALL
Operator IPUs	4569	3195	2334	2067	13298
User IPUs	6595	3832	4563	1874	17011
Operator EOT%	42	26	38	26	32
User EOT %	33	21	21	28	27
Operator mean WT(s) - ms	568	570	539	580	553
User mean WT(s) - ms	620	543	600	497	582
Operator mean turn-grab time - ms	120	365	-34	206	140
User mean turn grab time - ms	55	334	0	170	109

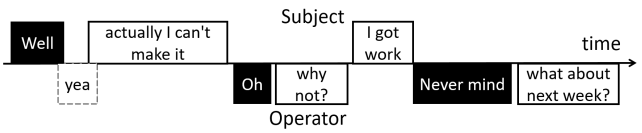

Figure 2: An example of dialogue with IPU samples labeled as within turn (black box) or end-of-turn (white solid box). Backchannel IPUs (dashed box) are ignored.

speaker, or end-of-turn (EOT), where the next IPU was from the other party. For simplification we ignored cases of barge-in. Figure 2 shows an example of the labeling process for turn-taking.

Statistics from the corpus are shown in Table 1, with approximately 30,000 samples being extracted from all sessions. 29.6% of all samples were EOT. We note that this is a smaller percentage than in previous research. In general, users paused within turns more frequently than the operators. We also see some differences in the various scenarios. For example, users paused more frequently per turn in the **LISTENING** and **INTERVIEW** scenarios, but the **DATING** scenario had the largest average pause length while it was less than 500ms in the **SECRETARY** scenario. The time taken for both the operator and subject to take the turn is also very short, with many overlaps, particularly in the **LISTENING** scenario. This result is about the same as real human communication [20].

3 NEURAL NETWORK MODELS

We constructed our prediction models using standard deep learning methods. For our evaluation we implemented models which were trained using only individual data from one of the four scenarios - **DATING**, **INTERVIEW**, **LISTENING** and **SECRETARY** and also trained models which used the entire data set.

Three models were trained for each of the individual data sets and for the whole data set - two unimodal models for acoustic and linguistic features and a fusion model which uses both modalities. The same neural architecture was used for all models. A general diagram of the architecture is shown in Figure 3.

The acoustic model used features extracted from each IPU. Extraction was done using the HTK Speech Recognition Toolkit[1] and

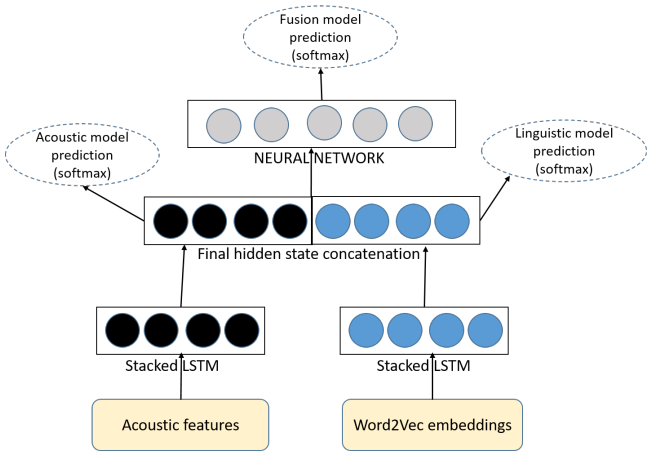

Figure 3: General architecture of the neural network model used.

we used 40 log mel-filter bank features in 10 ms intervals. We selected these features as they are the same as we use for ERICA's speech recognition, and so can be easily integrated into the live system. Interestingly, unlike other works, adding prosodic features of power and pitch did not improve the model. This could be due to the fact that the microphone used in the data collection is not located as close to the speaker as in other settings. We trained the data using a 3-layer stacked LSTM model for training. The size of each hidden layer of the LSTM was 128 and all implemented a 20% dropout rate. We found that using the final 500ms of the IPU was sufficient, rather than using the acoustic features of the entire IPU.

The linguistic model was based on the transcripts used over the sessions. We note that this is not the ideal approach, because the ASR system could provide different outputs than the transcripts. We tried two different Japanese word tokenizers, MeCab[2] and JUMAN[3] and found that the former had the better results. Word embedding was conducted using Word2Vec [15] on the tokenizations with a dimension of 100. Tokens which did not have a word embedding were given a random vector with values uniformly distributed between -0.25 to 0.25, which has been used in previous work [11].

[1] http://htk.eng.cam.ac.uk/

[2] http://taku910.github.io/mecab/

[3] http://nlp.ist.i.kyoto-u.ac.jp/index.php?JUMAN

The word embeddings were then used as input for a 3-layer stacked LSTM model. Like the acoustic model, the linguistic model had hidden layers of size 128 with dropout of 20%.

The fusion model concatenated the final hidden states of the acoustic and linguistic models. This type of pooling approach for similar features has been implemented in previous works [2, 12]. This concatenated layer was then used as input for a final neural network with just one hidden layer with a size of 256 and a dropout rate of 20%. For this model we used ReLU activation functions.

One notable observation in our fusion model is that weight initialization was done using He's algorithm [6]. We found that using random truncated weights caused our model to become extremely confident in prediction, with most probabilities very close to one or zero. This is not suitable for our FSTTM approach, which requires probabilities to be somewhat close to a representation of confidence in prediction. This type of phenomena of overconfident models has been previously observed, with possible countermeasures including temperature scaling [4].

We used two different approaches to train the fusion model. In the first approach, we trained all models separately, and stored the final hidden states of the training set. We then trained the fusion model using these states as inputs. In the second approach, we trained all models at the same time. This meant that the fusion model was trained using the hidden states for every training epoch. The difference is that the fusion model sees a larger variety of samples, but sees many samples only once, because the hidden states for the unimodal models will change over each epoch. We found that the latter approach gave the best results.

Data was split by session, with a 60:20:20 split used for training, validation and testing. For the generalized model, the sessions of the test data matched the distribution of the entire sample. We found no improvement when balancing the samples, since the absolute number of samples in each class was sufficiently large. We trained over 50 epochs and used the model with highest F1 macro score on the validation set as the test model. Since we ran the models simultaneously we can train all three at once on the same data to make a proper comparison.

4 EVALUATION

We first demonstrate the effectiveness of FSTTM using the latency vs. false cut-in metric and train a generalized fusion model on all the corpus data. Next, we compare scenario-specific models trained with all corpus data to those trained on only the individual scenario. Although we trained using both the operator and speaker's IPUs, in order to increase the amount of training data we only test on the user's IPUs since this is the task in the actual system.

We also note that our claims about evaluation in this section are restricted to the class of models which classify IPUs in an event-wise manner. Continuous turn-taking models which predict the next speaker also exist and have a different decision function [5, 19], but comparisons with these models are outside the scope of this work.

4.1 Measuring turn-taking performance

The performance of turn-taking can be measured using conventional metrics such as precision, recall, F1 score and accuracy, and this has been done in many previous works. These metrics can be used to compare models on the same dataset. They also imply a binary prediction - take the turn or wait for the user to finish.

However, if only these metrics are used to evaluate turn-taking models, they are not fully informative in terms of system performance. For turn-taking, we can also evaluate a model by its performance across two dimensions - average latency (time to take the turn after the user has finished) and false-cut in rate (the rate at which the user is interrupted during their turn). Optimizing these dimensions will reduce the time between speaking turns and minimize overlap, which is a fundamental goal of turn-taking and is done rapidly in mixed-initiative human conversation [18, 20].

Our corpus also shows turn-grabbing times similar to real human communication. An important point here is that achieving these times are practically impossible since we require a silence of at least 200ms to detect an IPU in addition to any signal processing, while the time between speakers is often less than that. This is the major weakness of event-based prediction as opposed to continuous prediction models. We simplify our analysis so that processing times are ignored, but our conclusions remain applicable.

We make the following assumption that for *a fixed cut-in rate, the better turn-taking model is the one which has the lowest average latency.* We propose that this evaluation is better because it is more relevant to live systems where there is generally a trade-off between the two dimensions. Furthermore, the model should have multiple points at which the dimensions can be assessed, and so evaluation should be done through by analyzing a performance curve constructed using these dimensions.

A naive model will simply define a time x as the amount of silence to wait for from the user before taking a turn. Although this model is trivial to implement, we can imagine that if x is sufficiently large, then it scores well under conventional metrics since it only takes the turn when the user is very likely to have finished their turn. However, for a real-time system this is impractical since the user has to wait a long time for any system responses.

Given a prediction model, we can estimate its latency and false cut-in performance through a label-based approach where we use the model's label prediction to decide whether to take the turn or wait. Since we have no other information, we assume the decision is taken immediately. However, we still have the problem of deciding x because if our model predicts a false negative (the system wrongly predicts the end-of-turn), the system will stay silent until the user speaks again. By setting x we cannot then guarantee that a prediction labeled negative is correct, because the system can still potentially cut in. This means that conventional metrics are not fully representative of actual performance in a real-time system.

Another approach is to use a finite-state turn-taking machine (FSTTM), introduced by Raux and Eskenazi [16, 17]. Details of the model can be referred to in the papers, but we will provide a brief summary below. In this approach, there are costs associated with grabbing the turn or waiting ($C(Grab|O_t)$ and $C(Wait|O_t)$, respectively). If we define τ as the current length of a pause in ms (the amount of time with no voice activity from the user), then the following formulas are applied:

$$C(Grab|O_t) = (1 - P(F|O_t)) * C_g$$

Table 2: Conventional metrics for end-of-turn detection in combined scenario models.

Model	Precision	Recall	F1	F1 macro	Accuracy
Baseline	0.295	0.295	0.295	0.500	0.584
Acoustic	0.652	0.409	0.503	0.675	0.766
Linguistic	0.684	0.508	0.583	0.721	0.790
Fusion	**0.686**	**0.521**	**0.592**	**0.726**	**0.792**

$$C(Wait|O_t) = P(F|O_t) * C_w(\tau)$$

where C_g and $C_w(\tau)$ are grab and wait coefficients and $P(F|O_t)$ is the probability that the floor is open given the observations at time t. Using Bayes rule, $P(F|O_t)$ is calculated using the formula:

$$\frac{P(F|O)}{P(d \geq \tau|O,U) * (1 - P(F|O)) + P(F|O)}$$

where $P(F|O)$ is the probability of the end of the user's turn given by the model and $P(d \geq \tau|O,U)$ is a function of mean pause duration which follows an exponential distribution. In this case, the parameter for the distribution can be calculated from the corpus.

As τ increases, the wait cost increases while the grab cost decreases. Eventually, when the wait cost exceeds the grab cost, the system makes the decision to take the turn. Unlike the label-based approach, FSTTM uses the probabilistic output of the model to decide *when* the turn should be taken, not *if* the turn should be taken. A higher probability of end-of-turn given by the model results in less silence needed before the grab action is taken. This approach removes the need to try to estimate a good value of x. Instead, the parameter to be adjusted is a coefficient to weight the cost of grabbing the turn, C_g. Although this method of decision-making is useful, it seems that few researchers apply FSTTM to their models, even though it greatly improves real-time system performance.

The performance curve has another key advantage, in that we know the performance of different levels of x or C_g. This becomes necessary if we have some benchmark we wish to compare (e.g. a false cut-in rate of less than 10%) or if we want to adjust the length of the response according to the user. We can easily visualize the performance for various parameterizations.

In this work, we will show that latency vs. false cut-in performance curves should be used when analyzing turn-taking because it is more indicative of the "true" performance of a model intended to be used in real-time. We will also show that a model which uses a traditional label-based approach to making the turn-taking decision may perform worse than one which uses an FSTTM to decide when to take the turn.

4.2 Evaluation of generalized model

We first begin by conventionally evaluating three models in the combined scenarios for end-of-turn detection - acoustic only, linguistic only, and the fusion model. Results are shown in Table 2. The fusion model is the best performing model but only by a slight amount over the linguistic model. The baseline model randomly selects EOT or WOT with the probability according to the corpus distribution. To demonstrate the effectiveness of FSTTM, we draw performance curves of average latency (l) and false cut-in rate (fc) through the following equations:

$$l = \sum_{i=1}^{m} t(EOT_i)/m$$

$$fc = (\sum_{i=1}^{n} WT_i = fc/n) * 100$$

where $t(EOT_i)$ is the time taken for a ground truth end-of-turn sample i to be responded to by the system, and $WT_i = fc$ is a binary value which denotes if a ground truth WT sample i is incorrectly predicted. Variables m and n are the number of EOT and WOT samples respectively.

The baseline is the silence threshold model described in Section 4.1. To draw the performance curve we calculate average latency and false cut-in rate for various values of user silence threshold x ms using all our test samples. The value $t(WT_i)$ indicates the length of time after a WT sample after which the user began to speak again, which we calculated from our corpus. For each EOT and WT, we calculate latency and false cut-in respectively as follows:

$$t(EOT_i) = x$$
$$WT_i = fc \quad \text{if } t(WT_i) > x$$

Next we evaluate our trained fusion prediction model. The first type of evaluation is the label-based approach. Again, we calculate the metrics using various values of x. Recall that in the label-based approach, the system takes the turn straight away if the model predicts to do so, else it waits until x ms have passed. The value y indicates the minimum amount of time for IPU detection after which a decision will be made. We fix y as 200ms for our evaluation. In practice, we use a fast end-to-end speech recognition system so this value is reasonable [22], but for other speech recognition systems this value will be higher. In any case, comparisons can still be made since a larger value of y will only reduce the rate of false cut-ins for all models. The equations of the label-based approach are:

$$t(EOT_i) = \begin{cases} \text{y}, & \text{if correctly predicted} \\ \text{x}, & \text{otherwise} \end{cases}$$

$$\begin{cases} WT_i = fc & \text{if incorrectly predicted} \\ WT_i = fc & \text{if correctly predicted and } t(WT_i) > x \end{cases}$$

Finally, we calculate the FSTTM for the fusion model. In this case, we are not adjusting x but the cost coefficient of grabbing the turn (C_g). For simplicity we fix the coefficient of waiting (C_w) to the amount of user silence time. We denote z as the time when $C(Grab|O_t) > C(Wait|O_t)$ for a sample i. Therefore, to determine a false cut-in we can simply check if the system would have taken the turn before $t(WT_i)$ ms had passed.

$$t(EOT_i) = z$$
$$WT_i = fc \quad \text{if } t(WT_i) < z$$

We can see the performance curves in Figure 4. Models with a performance curve closer to the lower left-corner of the graph are better. The FSTTM outperforms the baseline and label-based approach at smaller false cut-in rates. Interestingly, the label-based

Table 3: Average latencies at fixed false cut-in rates for combined scenario fusion model. Values are in ms.

	False cut-in rate			
Model	20%	15%	10%	5%
Baseline	816	940	1133	1511
Label-based	**581**	**724**	N/A	N/A
FSTTM	611	729	**917**	**1208**

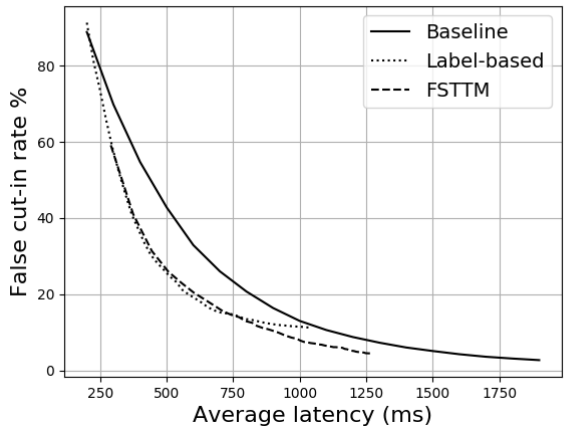

Figure 4: Comparison of label-based and FSTTM approaches for generalized (combined scenario) fusion model.

approach becomes worse than the baseline after a certain latency. We note that as x becomes large, the label-based approach will eventually level off because it can only increase latency but not improve the rate of false cut-ins, since these are due to the system having to make a hard decision even for incorrect predictions. Our results confirm that FSTTM is the best approach for making the real-time turn-taking decision, given that we have used the exact same model for the label-based and FSTTM approaches. We can also show results in terms of the latencies at various levels of false cut-in rate, as in Table 3. From our previous discussion, we note that smaller latencies are better given a fixed false cut-in threshold. Using an FSTTM approach cuts the latency of turn-taking by approximately 20% from the baseline while maintaining the same false cut-in rate.

Based on this result, all models will be evaluated using FSTTM. Furthermore, we report average latencies and false cut-in rate for comparison purposes as in Tables 4 and 5.

4.3 Evaluation of scenario models

We now evaluate the models trained on only a particular scenario. The acoustic, linguistic, and fusion model results for each scenario are shown in Table 4. Notice that the baseline values are markedly different for each scenario.

Firstly we note that in every scenario, the fusion model was better than the unimodal models, except **SECRETARY**, where the individual acoustic model was slightly better. Acoustic-only models

Table 4: Average latencies vs. false cut-ins for models trained on individual scenarios. Values are in ms.

	False cut-in rate			
DATING	20%	15%	10%	5%
Baseline	888	1051	1284	1731
Acoustic-only	820	1036	1302	N/A
Linguistic-only	878	1023	1226	N/A
Fusion	**776**	**958**	**1185**	N/A
INTERVIEW	20%	15%	10%	5%
Baseline	706	815	966	1313
Acoustic-only	290	348	451	824
Linguistic-only	613	709	930	1262
Fusion	**270**	**330**	**401**	**692**
LISTENING	20%	15%	10%	5%
Baseline	**854**	**964**	**1133**	**1462**
Acoustic-only	962	1106	1246	N/A
Linguistic-only	1026	1116	1232	1585
Fusion	929	1048	1188	1484
SECRETARY	20%	15%	10%	5%
Baseline	723	830	979	1286
Acoustic-only	573	**647**	**744**	**987**
Linguistic-only	716	782	846	1206
Fusion	**552**	648	769	994

Table 5: Comparison of best individual model and best combined scenario model. Values are in ms.

	False cut-in rate			
DATING	20%	15%	10%	5%
Individual-fusion	776	958	1185	N/A
Combined-fusion	691	845	1011	N/A
Improvement	85	113	174	N/A
INTERVIEW	20%	15%	10%	5%
Individual-fusion	270	330	401	692
Combined-fusion	501	613	755	1029
Improvement	-231	-283	-354	-337
LISTENING	20%	15%	10%	5%
Individual-fusion	929	1048	1188	1484
Combined-fusion	650	773	960	1235
Improvement	279	275	228	249
SECRETARY	20%	15%	10%	5%
Individual-acoustic	573	647	744	987
Combined-fusion	459	534	694	979
Improvement	114	113	50	8

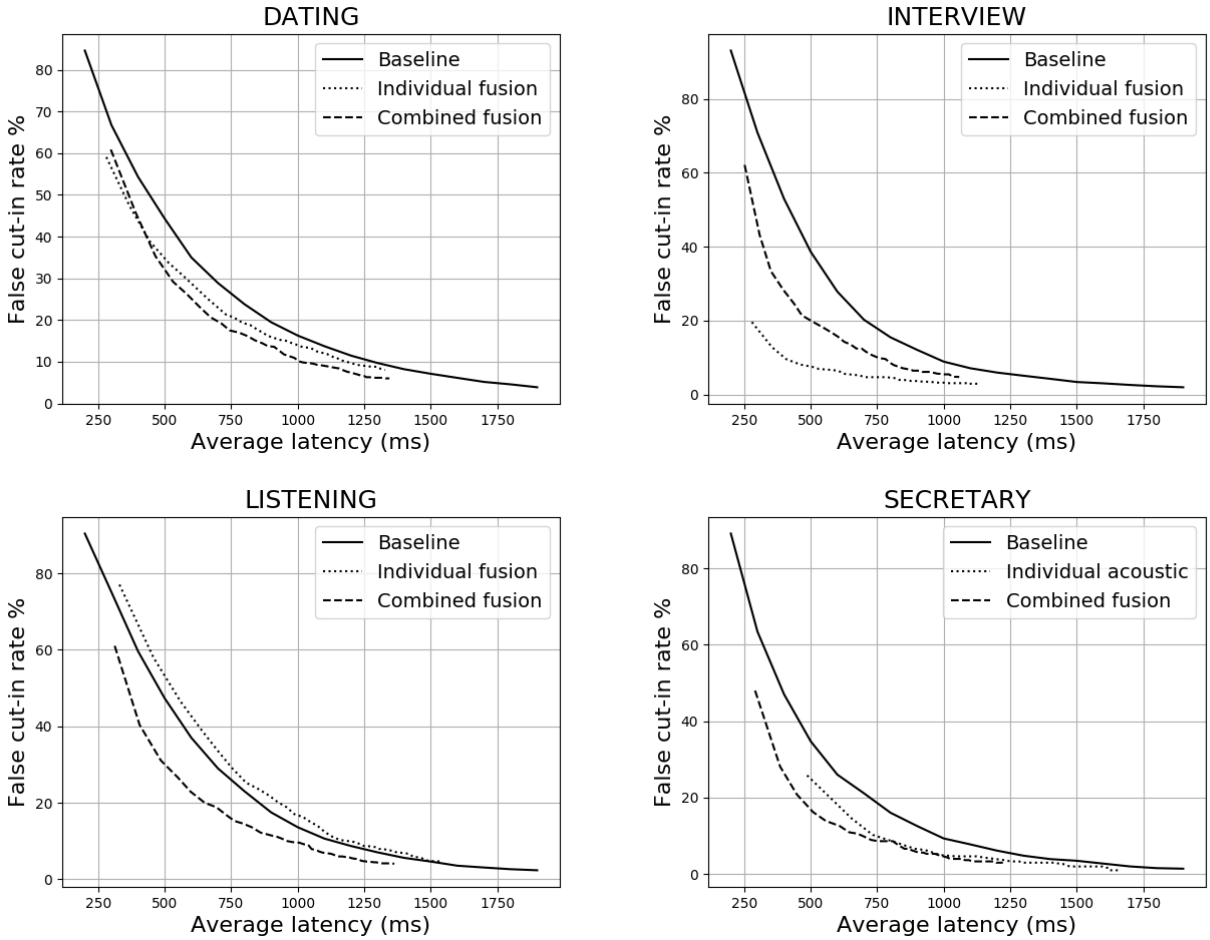

Figure 5: Comparison of best individual model with combined fusion for each scenario.

generally seem to outperform linguistic-only models. This could be because the training of the word embeddings uses a smaller subset of data than the combined scenario models.

We also observe that all models outperformed the baseline, except for the **LISTENING** scenario, where they were all worse. Our hypothesis is that this is due to the unstructured nature of attentive listening dialogue. We require much more variation in samples, both acoustically and linguistically, and this variation is not sufficient when only attentive listening scenarios are used for training.

Next, we compared the combined-fusion model with the best individual model in each scenario, to determine if training on the whole dataset produces a better model than training on a specific scenario. The results are shown in Table 5 and performance curves for each scenario are shown in Figure 5.

We see that there exists some variation in the improvement of the combined model over each scenario. For example, in the **INTERVIEW** scenario, the individual model is clearly better. On the other hand, for the **DATING** scenario, the combined model improves performance and in the **LISTENING** scenario the combined model is able to better the baseline, which couldn't be achieved with the individual fusion model.

The extent of improvement of the combined model appears to be somewhat related to the style of dialogue. The improvement is greatest in the **LISTENING** scenario, where most of the dialogue initiative is provided by the user. On the hand, in the **INTERVIEW** scenario, the dialogue is more structured and language more formal, with the user mostly answering standard questions. The **DATING** and **SECRETARY** scenarios are somewhat in between.

4.4 Effects of weighted samples

Up until now, we have assumed that false cut-in rate and average latency are equally important. In a live system we may want to modify this assumption. For example, it is likely that the user will tolerate an extra 100ms of latency if the false cut-in rate can be reduced by 10%. One approach could be to simply adjust the grab coefficient C_g for our existing models. Another approach is to train the model to heavily penalize false positives (WT samples which are labeled as EOT), to reduce the false cut-in rate.

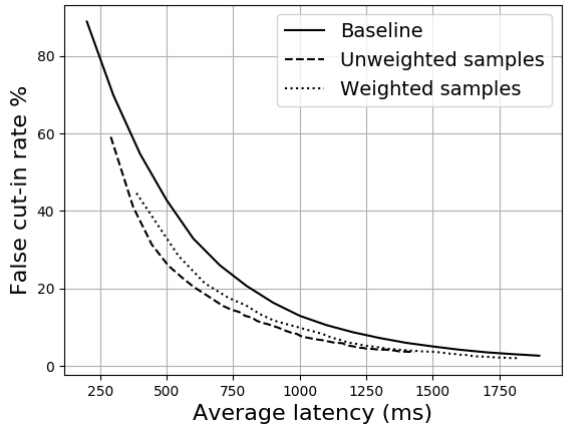

Figure 6: Comparison of generalized models using weighted and unweighted samples. The unweighted model is the same as in Figure 4

We trained another generalized fusion model using weighted cross-entropy to increase the loss on false positives, with a weight coefficient of 10. We again trained over 50 epochs, but only tested with the model with the highest precision for end-of-turn prediction. Naturally, our model had very high precision (0.92) but the F1 macro score was barely above the baseline. We compared its performance curve to the unweighted sample model as shown in Figure 6.

The unweighted model is in general still better performing. However, it reaches a limit at around 1400ms where neither latency nor false cut-in rate will change even by increasing C_u. The weighted model is still able to decrease false cut-in rate past this point in time, and at around 1800ms average latency, the false cut-in rate is approximately 2%. Intuitively, this makes sense, because the model is more conservative, with the ability to wait longer to reduce the number of false cut-ins. We can also observe this phenomenon in the **SECRETARY** scenario. It would appear that increasing precision has the effect of being able to increase the limit of the performance curve, allowing for longer average latency times.

Although we have demonstrated model evaluation from a quantitative perspective, we do not know how significant these results are according to a subjective evaluation. We expect that users will notice false cut-ins much more than a small decrease in latency, so being able to tune the model to a desired rate of interruptions is a useful property of FSTTM.

5 DISCUSSION

In this work, we emphasize two important points in terms of model evaluation. Firstly, the latency vs. false cut-off performance curve is the more informative way to evaluate turn-taking models. Secondly, using an FSTTM can greatly improve a model in terms of this metric. We showed that using a label-based approach as implied by only reporting conventional metrics was not suitable in the context of a real-time system. We encourage future research on turn-taking to use performance curves.

We also found that the type of dialogue task used for turn-taking influences model performance. An interview-type task appears to be the easiest, while a listening-style scenario was more difficult. We found that for the listening scenario, using a wide variety of data greatly improved the model. This suggests that a generalized model would be better suited for unstructured, informal conversation. On the other hand, a job interview has less variation in terms of dialogue and structure, and so it seems including other types of dialogue as training data makes the model too generalized, leading to lower performance. Unstructured tasks such as attentive listening should be explored further, as question-answering type systems have been the focus of a large amount of work on turn-taking.

There are several issues with the construction of our models. We used simple architectures and pooling strategies, but there are many hyperparameter combinations that may have improved the models. Our method of pooling through concatenating hidden states was quite basic, and there may be better ways to fully harness both models. We also did not test our models with techniques such as attention or pre-training using auto-encoders. These are all issues which we are able to address in the future.

One aspect touched upon previously was the overconfidence of deep learning models. If the FSTTM approach is to be used, then for evaluation purposes an overconfident model reduces to a label-based approach. We therefore recommend that when using deep learning models, this overconfidence be addressed in some way through calibration. In our case, using different weight initializations was helpful, but the optimal approach is dependent on the type of network used. This result also means that models which score higher under conventional metrics are not necessarily better than ones which score lower but are calibrated correctly.

6 CONCLUSION

In this work we evaluated turn-taking models for a conversational robot by training with acoustic and linguistic features over several types of scenarios. Our goal was to assess how well a generalized model could improve upon a model trained for a specific scenario. The method of evaluation was also critical in our work. We proposed that conventional metrics such as precision and F1 score cannot satisfactorily measure the true real-time performance of turn-taking models and a performance curve assessing average latency and false cut-in should be constructed. Due to this, a finite-state turn-taking machine is able to improve the model by using probabilistic values rather than labels for decision making.

We used an LSTM-based approach together with concatenation pooling to train the models. Our results showed that a generalized turn-taking model could improve all scenarios except for the job interview. It was most effective at improving the attentive listening scenario. We propose that a generalized turn-taking model is more suited for unstructured, informal conversation. The next step in our work is to integrate these models into a humanoid robot and perform evaluations in a live scenario.

ACKNOWLEDGMENTS

This work was supported by JST ERATO Ishiguro Symbiotic Human-Robot Interaction program (Grant Number JPMJER1401), Japan.

REFERENCES

[1] Zakaria Aldeneh, Dimitrios Dimitriadis, and Emily Mower Provost. 2018. Improving End-of-turn Detection In Spoken Dialogues By Detecting Speaker Intentions As A Secondary Task. In *ICAASP*.

[2] Zakaria Aldeneh, Soheil Khorram, Dimitrios Dimitriadis, and Emily Mower Provost. 2017. Pooling acoustic and lexical features for the prediction of valence. In *Proceedings of the 19th ACM International Conference on Multimodal Interaction*. ACM, ACM, New York, NY, USA, 68–72.

[3] Harish Arsikere, Elizabeth Shriberg, and Umut Ozertem. 2014. Computationally-efficient endpointing features for natural spoken interaction with personal-assistant systems. In *Acoustics, Speech and Signal Processing (ICASSP), 2014 IEEE International Conference on*. IEEE, 3241–3245.

[4] Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q Weinberger. 2017. On calibration of modern neural networks. *arXiv preprint arXiv:1706.04599* (2017).

[5] Kohei Hara, Koji Inoue, Katsuya Takanashi, and Tatsuya Kawahara. 2018. Prediction of Turn-taking Using Multitask Learning with Prediction of Backchannels and Fillers. In *INTERSPEECH*. To appear.

[6] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2015. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *Proceedings of the IEEE international conference on computer vision*. IEEE Computer Society, Washington, DC, USA, 1026–1034.

[7] Ryo Ishii, Kazuhiro Otsuka, Shiro Kumano, and Junji Yamato. 2014. Analysis of Respiration for Prediction of "Who Will Be Next Speaker and When?" in Multi-Party Meetings. In *Proceedings of the 16th International Conference on Multimodal Interaction (ICMI '14)*. ACM, New York, NY, USA, 18–25.

[8] Yuichi Ishimoto, Takehiro Teraoka, and Mika Enomoto. 2017. End-of-Utterance Prediction by Prosodic Features and Phrase-Dependency Structure in Spontaneous Japanese Speech. In *Proceedings of Interspeech 2017*. 1681–1685.

[9] Kristiina Jokinen, Kazuaki Harada, Masafumi Nishida, and Seiichi Yamamoto. 2010. Turn-alignment using eye-gaze and speech in conversational interaction. In *Eleventh Annual Conference of the International Speech Communication Association*.

[10] Tatsuya Kawahara, Takuma Iwatate, and Katsuya Takanashi. 2012. Prediction of turn-taking by combining prosodic and eye-gaze information in poster conversations. In *Thirteenth Annual Conference of the International Speech Communication Association*.

[11] Yoon Kim. 2014. Convolutional neural networks for sentence classification. *arXiv preprint arXiv:1408.5882* (2014).

[12] Chaoran Liu, Carlos Ishi, and Hiroshi Ishiguro. 2017. Turn-Taking Estimation Model Based on Joint Embedding of Lexical and Prosodic Contents. In *Proc. Interspeech 2017*. 1686–1690.

[13] Angelika Maier, Julian Hough, and David Schlangen. 2017. Towards Deep End-of-Turn Prediction for Situated Spoken Dialogue Systems. In *Proceedings of INTERSPEECH 2017*.

[14] Ryo Masumura, Taichi Asami, Hirokazu Masataki, Ryo Ishii, and Ryuichiro Higashinaka. 2017. Online End-of-Turn Detection from Speech based on Stacked Time-Asynchronous Sequential Networks. In *Proc. Interspeech 2017*. 1661–1665.

[15] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781* (2013).

[16] Antoine Raux and Maxine Eskenazi. 2009. A finite-state turn-taking model for spoken dialog systems. In *Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics*. Association for Computational Linguistics, 629–637.

[17] Antoine Raux and Maxine Eskenazi. 2012. Optimizing the turn-taking behavior of task-oriented spoken dialog systems. *ACM Transactions on Speech and Language Processing (TSLP)* 9, 1 (2012), 1.

[18] Emanuel A. Schegloff. 2006. Interaction: The infrastructure for social institutions, the natural ecological niche for language, and the arena in which culture is enacted. In *Roots of Human Sociality*, Nick J. Enfield and Stephen C. Levinson (Eds.). Berg, London, 70–96.

[19] Gabriel Skantze. 2017. Towards a General, Continuous Model of Turn-taking in Spoken Dialogue using LSTM Recurrent Neural Networks. In *Proceedings of the 18th Annual SIGdial Meeting on Discourse and Dialogue*. 220–230.

[20] Tanya Stivers, Nicholas J Enfield, Penelope Brown, Christina Englert, Makoto Hayashi, Trine Heinemann, Gertie Hoymann, Federico Rossano, Jan Peter De Ruiter, Kyung-Eun Yoon, et al. 2009. Universals and cultural variation in turn-taking in conversation. *Proceedings of the National Academy of Sciences* 106, 26 (2009), 10587–10592.

[21] L. ten Bosch, N. Oostdijk, and Jan de Ruiter. 2004. Turn-taking in social talk dialogues: temporal, formal and functional aspects. In *SPECOM 2004*.

[22] Sho Ueno, Hirofumi Inaguma, Masato Mimura, and Tatsuya Kawahara. 2018. Acoustic-to-word attention-based model complemented with character-level CTC-based model. In *Proceedings of IEEE-ICASSP*. IEEE, 5804–5808.

[23] Nigel G. Ward and David DeVault. 2017. Challenges in Building Highly-Interactive Dialog Systems. *AI Magazine* 37, 4 (2017), 7–18.