AUTOMATED TWO-WAY ENTRAINMENT TO IMPROVE SPOKEN DIALOG SYSTEM PERFORMANCE

José Lopes[1,2], *Maxine Eskenazi*[3], *Isabel Trancoso*[1,2]

[1]INESC-ID Lisboa, Portugal
[2]Instituto Superior Técnico, Lisboa, Portugal
[3]Language Technologies Institute, Carnegie Mellon University, Pittsburgh, PA, USA

jose.david.lopes@l2f.inesc-id.pt

ABSTRACT

This paper proposes an approach to the use of lexical entrainment in Spoken Dialog Systems. This approach aims to increase the dialog success rate by adapting the lexical choices of the system to the user's lexical choices. If the system finds that the users lexical choice degrades the performance, it will try to establish a new conceptual pact, proposing other words that the user may adopt, in order to be more successful in task completion. The approach was implemented and tested in two different systems. Tests showed a relative dialog estimated error rate reduction of 10% and a relative reduction in the average number of turns per session of 6%.

Index Terms— Spoken Dialog Systems, Entrainment

1. INTRODUCTION

Recently spoken dialog systems (SDS) have become more widely available. This has attracted a lot of attention from the research community with an effort to develop SDS that have communication capabilities that are close to humans in terms of speech understanding and decision making. Recent work has shown significant improvement in performance using statistical approaches to dialog management [1, 2]. We believe that if the principle of lexical entrainment [3, 4] is applied to SDSs, performance can improve and interaction will become more natural. The system will entrain to the user's lexical choice. However, if the user's lexical choice produces negative results, it will try to establish a new conceptual pact with the user to successfully achieve task completion. Hence, a very important concept in a dialog context is *priming*, i.e., the process that influences linguistic decision-making, and explains how choices made by one speaker may influence the speech of the other speaker [5].

Our first lexical entrainment approach is described in [6]. The context was the Noctívago system, an experimental agenda-based SDS in European Portuguese that provides schedule information about night buses in Lisbon. The first version of the system was adapted from Let's Go [7], a live system that gives bus schedule information for real users in Pittsburgh since 2005. Both telephone systems were based on the Olympus open-source architecture for SDSs [8], and used Ravenclaw [9], an agenda-based dialog manager.

Our current work involves different versions of these two systems. The new version of Noctívago has a multi-modal web interface. The users interact with it through a virtual agent with a push-to-talk button. This interface makes it easier to recruit new users and test different configurations. Still, the amount of data collected with an experimental system is far from the numbers achieved by a live system. The new version of Let's Go, on the other hand, takes advantage of the large amount of data in order to use a state-of-the-art dialog manager, based on dialog state tracking [1], sometimes also called belief tracking.

In this paper we will combine a dialog manager with a set of rules based on lexical entrainment theory to find the lexical choices that best serve the user and the system. The lexical entrainment rules will be implemented and tested in the two systems which, although targeting the same domain, differ in language, type and number of users, and type of dialog manager.

The paper starts with a review of the related work in section 2. Section 3 describes the first set of entrainment rules. The tests with Noctívago and Let's Go, are reported in sections 4 and 5, respectively. Finally, conclusions and future work are presented in section 6.

2. RELATED WORK

Most of the problems in SDS are caused by speech recognition errors. One possible solution to improve speech recognition would be to select the least confusable words using previously well-known techniques [10, 11, 12]. However, this type of approach ignores the user's lexical choices, and may make the system sound less natural, less engaged and ultimately the overall performance may be negatively affected.

An alternative approach is to try to influence the user's lexical choice towards words that are easier for the ASR to process, by using the principle of lexical entrainment. Entrainment has played an important role in recent SDS research [13]. One of the most important types of entrainment concerns the choice of words. The principle of lexical entrainment has been widely investigated in task-oriented human-human dialogs. In the experiments described in the literature, the subjects establish a conceptual pact in order to achieve success in task-oriented dialogs. If SDSs are able to establish conceptual pacts with the users their performance is likely to increase.

Another interesting finding from [3] and [4] is that there is a high variability of vocabulary between conversations, whereas the variability within the same conversation is relatively low. This suggests that, for SDS, variability should also be controlled during each conversation.

In [14], the problem of which words should be used in a system access was studied. According to the authors, access using the designer's favorite word results in 80-90% failure rates. The problem with SDS is similar, although the fact that most systems are domain-specific helps to reduce the range of possible words.

Previous work on lexical entrainment for spoken dialog systems has been done in [15] where the effect of different syntactic structure and lexical choices were shown to influence the user's choice of words. In [16], the authors also studied the effect of changing words in system prompts that real users have been accustomed to for a long time. They observed that sometimes users kept the older primes. However, there were some cases where users adopted the new primes immediately. This indicates that users showed a preference for some primes. This finding may be explored to determine the most suitable primes automatically.

Apart from lexical entrainment, other forms of entrainment have been studied to improve SDS. In [17], the degree of convergence both in lexical and acoustic/prosodic elements was investigated. In [18], the authors used entrainment to influence the way users speak to a system. They have dealt with loud speech and hyperarticulation, both of which have a negative impact on system performance, by adapting system output to produce the opposite effect of the detected behavior. The system responds more softly when shouting is detected, and it speaks faster to counter hyperarticulation. They found that in most cases the change in system prompts helped users return to more neutral utterances.

3. ENTRAINMENT RULES

Unlike humans, a computer's behavior can easily be controlled. This is an advantage to system developers, since they can program their systems to behave in the way that is most benefiting to the success of the interaction. Ideally, the system should follow the user's behavior, so that the user feels more engaged with it. However, if the user's behavior is not favorable to system outcome (ASR success, for example), the system should then propose a different behavior.

In [6], we found that some primes performed better than others. We also saw that users preferred some primes more than others. The primes with higher error rates were selected as prime candidates and a new set of primes were proposed to replace them. The performance of the set of prime candidates was compared to the new set of primes. The system that used the new set of primes had a better dialog success rate and a lower Word Error Rate (WER). These results proved that the lexical entrainment can play an important role in improving SDS's performance.

In this paper we go a step further to create an algorithm that automatically selects the best primes according to both the system performance and the user's preferences. A set of heuristic rules based on information that can be extracted from events in live interaction will be described in this section. These events are exemplified in Table 1, with 4 excerpts from different Let's Go dialogs. In the first excerpt the user uptakes "new conversation" from the system utterance S1. That is, he uses the prime proposed by the system. This will be considered an uptake event (Up). In the second excerpt the user says "new task" whereas the system proposed "new dialog". This is considered the no uptake condition (NUp). These two types of events had already been analyzed in [6]. Here, we shall also consider the No Previous Use condition (NPU). This is exemplified in the third excerpt, where the user says "now". No other prime for this concept had been used or proposed in this dialog. This means that this prime was not previously used. By knowing if the prime was used and in which conditions it was used, we hope to learn the user prime preference. These prime usage events will help us to do that.

The information about system performance is given both by a confidence score, and by non-understood user turns (NUn). In an NUn turn, the given concepts are not bound to any other concept, like user utterance U7, in the fourth excerpt. These two types of cues

S1: To ask about a different **journey**, you can
say, start a **new conversation**. If you are
finished, you can say goodbye.
U1: START A **NEW CONVERSATION**
...
S2: You can say, tell me about the later bus, when
is the bus prior this one, start a **new dialog**, or say goodbye.
U2: START A **NEW TASK**
...
S3: Welcome to the CMU Let's Go bus information system.
S4: To get support at any time, just say support or press zero.
S5: Where would you like to start from?
U3: SHERADEN
S6: Where do you wanna go?
U4: DOWNTOWN
S7: Going to DOWNTOWN. Did I hear that correctly?
U5: YES
S8: When would you like to travel?
U6: **NOW**
...
S9: Please let me know how I did by pressing, four, if you got
the information you were looking for, or six, if you did not. To get
more information about buses related to this **journey**, you can say,
tell me about the **following** bus, or, when is the bus **prior** to this
one. To ask about a different **journey**, you can say, start a **new conversation**.
If you are finished, you can say goodbye.
U7: 6B LORETTA MONROEVILLE

Table 1. Examples of the events used in the prime choice update. Primes are in bold.

will help the system deciding whether it should update the prime selection. The next section explains how the performance information can be combined with the prime usage events described in the previous paragraph to select the primes to be used.

3.1. Prime Update Algorithm

The SDSs in which the rules were implemented do not have user models. Thus, we decided to break the algorithm into two phases, long-term and short-term entrainment. The first phase focuses on inter-session entrainment, whereas the second phase focuses on entrainment within a session. This way, we believe that we can better deal with the fact that different users can have different prime preferences.

In the first phase, long-term entrainment, the algorithm tries to select the best primes from a history of interactions. The goal is to capture the prime preference from a population of users. The conclusions from [6] pointed to a possible correlation between the number of no uptake events and the most common primes in daily language. This was further investigated. The first step is to compute the number of uptake and no uptake events for each prime. Next, the prime usage events were correlated with the numbers of hits of each prime in a web search engine. A correlation of 0.14 and 0.61 was found for uptake and no uptake events, respectively. This lead to the conclusion that the no uptake events could be helpful in finding the best primes from a population of users. Thus, the prime are ranked according to the number of NUp events for prime i , $\#_{NUp}(i)$, normalized by the number of system usages for the same prime, $\#_{SU}(i)$, to compute the initial prime update ratio, $R(i)$ for each prime i:

$$R(i) = \frac{\#_{NUp}(i)}{\#_{SU}(i)} \quad (1)$$

In short-term entrainment, the goal is to make the system adapt as much as possible to user preferences, making it more robust to prime change. This will produce less variability within a dialog as suggested by [3] and [4]. At the same time, the system has to rapidly react if a prime degrades system performance. This kind of behavior can be achieved by a set of heuristics that are based on the use of three update factors, φ_{Up}, φ_{NUp} and φ_{NPU} that will update the initial ratio given by Equation 1, each time a prime usage event is detected after a user turn. These factors represent the importance of each event in the prime choice, and ideally should be trained from data. After each user turn, the ratio $R(i)$ is updated according to the following heuristics:

- If an uptake occurs for prime i, then $R(i)$ is increased by φ_{Up}. Example: in the first excerpt from Table 1 the $R(new\ conversation)$ will be increased by φ_{Up};
- If prime i is used when prime j was proposed, then $R(i)$ is increased by φ_{NUp} and $R(j)$ is subtracted by the same quantity. Example: in the second excerpt from Table 1 the $R(new\ task)$ will be increased by φ_{NUp} and $R(new\ dialog)$ will be subtracted φ_{NUp};
- If prime i was used without being previously used in that session either by the user or the system, then $R(i)$ is increased by φ_{NPU}. Example: in the third excerpt from Table 1, $R(now)$ will be increased by φ_{NPU};
- If prime i was proposed and a non-understanding was generated in the next user turn, then $R(i)$ is subtracted by $\#_{NUn}(i)$, where $\#_{NUn}(i)$ is the number of non understandings for prime i in a session. Example: in the last excerpt from Table 1, the R for "journey", "following", "prior" and "new conversation" will be subtracted by the number of previous non-understandings they had in that session.

4. TESTING THE ENTRAINMENT RULES WITH NOCTÍVAGO

The first tests of the dialog system implemented with the entrainment rules were conducted with the Noctívago web-based system. At this point, the collected data was not sufficient to train the entrainment factors φ_{Up}, φ_{NUp} and φ_{NPU}. The factors were set to 1, 2 and 3, respectively. These values assign a higher weight to the least frequent events, since they seem very important to find the user prime preference. The values also ensure that they superimpose to the initial $R(i)$, since they are one order of magnitude higher than the average value for $R(i)$, 0.09, for the data collected in [6].

These tests aimed at investigating how to combine the prime usage information with the system performance information, namely the ASR confidence score or the dialog confidence score. Noctívago ran alternately using different confidence scores. Set-up 1 used the dialog confidence score. Set-up 2 used the ASR confidence score. Set-up 3 performed short-term entrainment updates regardless of the given confidence score. Set-up 4 performed only long-term entrainment.

The callers were asked to interact with the multi-modal version of Noctívago, making three consecutive calls, without knowing that the system had different configurations running alternately.

Table 2 summarizes the details of 160 sessions collected, in terms of dialog success rate, average number of turns, WER, percentage of prime usage events and percentage of non-understandings.

Despite using different system versions that may have influenced the result, the new version of the system has an estimated success rate at least 15% higher when compared to the best results obtained in [6]. The real success rate has also an absolute 10% increase. The WER has decreased in all the set-ups compared to the best results obtained with the previous version, the telephone-based Noctívago, 52.3%.

Although most of the results obtained are statistically insignificant, against our initial expectations, set-up 4 achieved the best real dialog success, whereas set-up 3 achieved the lowest average number of turns. The reason for this may be related to the fact that long-term entrainment is only based on no uptake events. The initial prime update ratio rarely changed from session to session, since no uptake events were less frequent than uptake events. This resulted in less prime variation across sessions, even if a prime was degrading the system performance. This problem could be solved by including the uptake events in the prime update ratio for long-term entrainment.

	Set-up 1	Set-up 2	Set-up 3	Set-up 4
# of sessions	40	42	44	34
Estimated Dialog Success (%)	92.5	95.2	95.5	91.2
Real Dialog Success (%)	70.3	63.2	67.2	74.5
Average Number of Turns	**9.24**	**9.13**	8.12	8.92
WER (%)	50.9	42.2	49.4	45.8
Total Uptake (%)	16.8	20.3	18.4	17.6
Total No Uptake (%)	2.03	2.31	1.21	1.20
Total No Previous Usage (%)	0.38	0.12	0.13	0.33
Total Non Understanding (%)	9.77	5.78	6.85	6.50

Table 2. Results from prime updating tests. Statistically significant ($p - value < 0.05$) values are bold marked.

Another factor that may have contributed to the better performance of set-up 4, is the update after a non-understanding. Other configurations updated the score after every non-understanding, which may influence the prime rank. According to [19], it might be too early to gather enough evidence that the prime needs to be changed.

5. ENTRAINMENT RULES WITH REAL USERS IN A DIALOG STATE TRACKING SYSTEM

Although some interesting findings came out of Section 4, most of the results in Table 2 are not statistically significant. Using Let's Go as our platform, the impact in system performance will hopefully become more clear. Since it is a real system, the number of calls is much higher, typically averaging 40 calls during weekdays and 100 calls during weekends.

The entrainment rules were implemented in the dialog state tracking version of Let's Go [20] with slight differences from what was described in Section 3.1, following the conclusions of the first tests held with Noctívago. Long-term entrainment was changed to also accommodate uptake events. This initial score is now given by:

$$R(i) = \frac{\#_{NUp}(i)}{\#_{SU}(i)} + w_{Up} \times \frac{\#_{Up}(i)}{\#_{SU}(i)} \tag{2}$$

where $\#_{Up}(i)$ is the past number of uptakes for prime i and w_{Up} is given by the ratio between the total uptake events and the total no uptake condition events:

$$w_{Up} = \frac{\sum_{i=1}^{P} \#_{Up}(i)}{\sum_{i=1}^{P} \#_{NUp}(i)} \tag{3}$$

where P is the total number of primes. The w_{Up} allows the system to enhance to the most frequent event, either uptake or no uptake.

In short-term entrainment, the factor φ_{Up} was set to 2 to strengthen the importance of the uptake events and, consequently, increase the degree of convergence between the system and the user. The factors φ_{NUp} and φ_{NPU} kept the same value. The update after a non-understanding was also modified. A threshold was set to avoid updating the prime ratio until a minimum number of non-understandings occurred in one session. Since we do not dispose of enough data to calibrate this threshold, it was initially set to

2. This follows the intuition that at the second non-understanding, the system is having difficulties dealing with that prime. Once the number of non-understandings reaches the threshold, the $R(i)$ is subtracted by $w_{NUn} \times \#_{NUn}(i)$. w_{NUn} is a weight factor for non-understandings, given by:

$$w_{NUn} = \frac{\sum_{i=1}^{P} \#_{NUn}(i)}{\sum_{i=1}^{P} \#_{NUp}(i)} \quad (4)$$

We decided to use the dialog confidence score as the threshold to perform the updates in short-term entrainment. This score has more context information which can make it more reliable than the ASR confidence score. In addition, according to Table 2 the real dialog success rate was 7% higher for Set-up 1, which uses the dialog confidence score, when compared with Set-up 2, which uses the ASR confidence score.

The set of prime choices available was extended from those used in [16]. We suspect that the increased amount of possible utterances at each point in the dialog will produce more natural system behavior. Table 3 shows the primes available in this dialog. The remaining concepts were bus stops which could not be considered prime candidates since they have much fewer synonyms [21].

Category of primes	Old Primes	New Primes
Bus immediacy	next	following / subsequent later / after
	now	immediately / right now right away / as soon as possible
	previous	preceding / prior / before
Action	leaving leave	departing / depart starting / start
Domain Specific	route schedule	itinerary / trip ride / journey
Agreement	right	alright / correct / okay
System Interaction	help	assistance / support more information
	query	request / task dialog / route / conversation

Table 3. Primes used by Let's Go before and after the entrainment rules were implemented.

Let's Go has been running live since 2005, with a set of regular users who are believed to be familiar with the system's lexical choices. To avoid the use of the former primes, a threshold was set so that the system only used of any of the former primes if their prime update ratio was above that threshold defined.

5.1. Results

The performance of the system version where automated choices were made using the entrainment rules was compared to the one without entrainment, over approximately the same period of time. Table 4 shows two examples where the implemented rules affected the prime choice.

In the first excerpt, there was a non-understanding "new conversation". The system changed to "new dialog" and the user successfully used the prime. In the second excerpt, the system proposed "following" in S3. The user answered with "after", a different prime for the same category. The next time the system had to use a prime for that category, in S6, the system adapted to the user's choice using "after" instead of "following".

Table 5 presents the number of sessions, estimated dialog success rate, average number of turns for both conditions, the prime update events and the non-understandings. The estimated dialog success shows an absolute gain of more than 2% which represents

S1: To ask about a different **journey**, you can say, start a **new conversation**.
If you are finished, you can say goodbye.
U1: 13 TREE STATION
S2: To ask about a different **ride**, you can say, start a **new dialog**.
If you are finished, you can say goodbye.
U2: START A **NEW DIALOG**
...
S3: You can say, tell me about the **following** bus, when is the bus prior this one,
start a new conversation, or goodbye.
U3: **AFTER** THAT
S4: Okay.
S5: There is a 71D that departs from HAY STREET AT PENN AVENUE at 8:13 a.m..
It will arrive at FIFTH AVENUE AT MCMASTERS at 8:48 a.m..
S6: Please let me know how I did by pressing, four, if you got the information
you were looking for, or six, if you did not. To get more information about
buses related to this journey, you can say, what time is the bus **after** this one, or,
when is the bus prior this one. To ask about a different journey, you can say,
start a new conversation. If you are finished, you can say goodbye.

Table 4. Excerpts of dialogs where entrainment rules changed the system's normal behavior. Primes affected in bold.

a 10% relative reduction in the estimated error rate. There was also a 6% relative reduction in the number of turns per session.

Looking at the prime usage events, the results reflect the fact that Let's Go users were familiar to the old primes. In the baseline system with fewer primes, the number of uptake events increased and the number of no uptake events decreased. For the system with the entrainment rules the number of no-uptakes increased and the uptakes decreased, due the the larger variety of primes.

Some of the new primes introduced were manually added to lexicon and language models, trained with data from previous dialogs. This can help to explain the increase of the non understandings in the entrainment rules system. Nevertheless, the system showed a better performance.

	Baseline	Entrainment Rules
# of sessions	1542	1792
Estimated Dialog Success (%)	75.11	77.64
Avg. number of turns	12.24	11.47
Total Uptake (%)	6.02	2.48
Total No Uptake (%)	0.55	0.64
Total No Previous Usage (%)	1.82	1.66
Total Non Understanding (%)	4.71	4.84

Table 5. Results for Let's Go tests.

6. CONCLUSIONS AND FUTURE WORK

This paper presented a novel approach to the incorporation of lexical entrainment principles in SDSs. The approach is based on a set of heuristics, the entrainment rules, which were implemented and tested in two different SDSs, Noctívago and Let's Go. The results showed a positive impact on the system performance, especially with Let's Go. The estimated dialog success increased, together with the reduction of the average number of turns per session. Besides the quantitative results, some expert users also reported that the system sounded more natural. This is something that we would like properly assess in the near future. We hope that the data collected can help finding a statistically based approach to the use of lexical entrainment in SDSs.

Acknowledgments

José Lopes and Isabel Trancoso were supported by FCT grants SFRH/BD/47039/2008 and PEst-OE/EEI/LA0021/2011.

7. REFERENCES

[1] Sungjin Lee and Maxine Eskenazi, "Exploiting machine-transcribed dialog corpus to improve multiple dialog states tracking method," in *Proc. SIGDIAL 2012*, 2012.

[2] Jason D. Williams and Steve Young, "Partially observable markov decision processes for spoken dialog systems," *Comput. Speech Lang.*, vol. 21, no. 2, pp. 393–422, 2007.

[3] Simon Garrod and Anthony Anderson, "Saying what you mean in dialogue: A study in conceptual and semantic coordination," *Cognition*, vol. 27, pp. 181–218, 1987.

[4] Susan E. Brennan and Herbert H. Clark, "Conceptual pacts and lexical choice in conversation," *Journal of Experimental Psychology: Learning, Memory, and Cognition*, vol. 22, pp. 1482–1493, 1996.

[5] David Reitter, Frank Keller, and Johanna D. Moore, "Computational modelling of structural priming in dialogue," in *In Proc. HLT-NAACL*, 2006, pp. 121–124.

[6] José Lopes, Maxine Eskenazi, and Isabel Trancoso, "Towards choosing better primes for spoken dialog systems," in *Proc. ASRU*, Waikiloa, Hawaii, USA, 2011, pp. 306–311.

[7] Antoine Raux, Brian Langner, Dan Bohus, Alan W Black, and Maxine Eskenazi, "Lets go public! taking a spoken dialog system to the real world," in *in Proc. of Interspeech 2005*, 2005.

[8] Dan Bohus, Antoine Raux, Thomas K. Harris, Maxine Eskenazi, and Alexander I. Rudnicky, "Olympus: an open-source framework for conversational spoken language interface research," in *Proceedings of the Workshop on Bridging the Gap: Academic and Industrial Research in Dialog Technologies*, Stroudsburg, PA, USA, 2007, NAACL-HLT-Dialog '07, pp. 32–39, Association for Computational Linguistics.

[9] Dan Bohus and Alexander I. Rudnicky, "The ravenclaw dialog management framework: Architecture and systems," *Comput. Speech Lang.*, vol. 23, no. 3, pp. 332–361, July 2009.

[10] David B. Roe and Michael D. Riley, "Prediction of word confusabilities for speech recognition," in *Proc. ICSLP*, 1994.

[11] Beng T. Tan, Yong Gu, and Trevor Thomas, "Word confusability measures for vocabulary selection in speech recognition," in *Proc. ASRU*, 1999.

[12] Jan Anguita, Stephane Peillon, Javier Hernando, and Alexandre Bramoulle, "Word confusability prediction in automatic speech recognition," in *INTERSPEECH 2004 - ICSLP, 8th International Conference on Spoken Language Processing*, 2004.

[13] Julia Hirschberg, "Speaking more like you: Entrainment in conversational speech," in *Proc. INTERSPEECH*, 2011.

[14] G. W. Furnas, T. K. Landauer, L. M. Gomez, and S. T. Dumais, "The vocabulary problem in human-system communication," *Commun. ACM*, vol. 30, no. 11, pp. 964–971, Nov. 1987.

[15] Svetlana Stoyanchev and Amanda Stent, "Lexical and syntactic priming and their impact in deployed spoken dialog systems," in *Proceedings of HLT-NAACL 2009*, Stroudsburg, PA, USA, 2009, NAACL-Short '09, pp. 189–192.

[16] Gabriel Parent and Maxine Eskenazi, "Lexical entrainment of real users in the let's go spoken dialog system," in *INTERSPEECH*, 2010, pp. 3018–3021.

[17] Arthur Ward and Diane Litman, "Automatically measuring lexical and acoustic/prosodic convergence in tutorial," in *Proceedings of SLaTE 2007*, Framington, Pennsylvania, USA, 2007.

[18] Andrew Fandrianto and Maxine Eskenazi, "Prosodic entrainment in an information-driven dialog system," in *Proceedings of Interspeech 2012*, Portland, Oregon, USA, 2012.

[19] H. Branigan, J. Pickering, J. Pearson, and J. McLean, "Linguistic alignment between people and computers," *Journal of Pragmatics*, vol. 42, no. 9, pp. 2355–2368, Sept. 2010.

[20] Sungjin Lee and Maxine Eskenazi, "Pomdp-based let's go system for spoken dialog challenge," in *Proc. IEEE SLT Workshop*, 2012.

[21] Arthur Ward and Diane Litman, "Measuring convergence and priming in tutorial dialog," Tech. Rep., University of Pittsburgh, 2007.