Available online at www.sciencedirect.com

ScienceDirect

Computer Speech and Language 31 (2015) 87–112

COMPUTER SPEECH AND LANGUAGE

www.elsevier.com/locate/csl

From rule-based to data-driven lexical entrainment models in spoken dialog systems☆

José Lopes[a,b,*], Maxine Eskenazi[c], Isabel Trancoso[a,b]

[a] *Spoken Language Laboratory, INESC-ID Lisboa, Rua Alves Redol 9, 1000-029 Lisboa, Portugal*
[b] *Instituto Superior Técnico, Avenida Rovisco Pais 1, 1049-001 Lisboa, Portugal*
[c] *Language Technologies Institute, Carnegie Mellon University, 5000 Forbes Avenue, Pittsburgh, PA 15213, USA*

Received 23 August 2013; received in revised form 15 October 2014; accepted 24 November 2014
Available online 4 December 2014

Abstract

This paper presents uses a data-driven approach to improve Spoken Dialog System (SDS) performance by automatically finding the most appropriate terms to be used in system prompts. The literature shows that speakers use one another's terms (entrain) when trying to create common ground during a spoken dialog. Those terms are commonly called "primes", since they influence the interlocutors' linguistic decision-making. This approach emulates human interaction, with a system built to propose primes to the user and accept the primes that the user proposes. These primes are chosen on the fly during the interaction, based on a set of features that indicate good candidate primes. A good candidate is one that we know is easily recognized by the speech recognizer, and is also a normal word choice given the context. The system is trained to follow the user's choice of prime if system performance is not negatively affected. When system performance is affected, the system proposes a new prime. In our previous work we have shown how we can identify the prime candidates and how the system can select primes using rules. In this paper we go further, presenting a data-driven method to perform the same task. Live tests with this method show that use of on-the-fly entrainment reduces out-of-vocabulary and word error rate, and also increases the number of correctly transferred concepts.

Keywords: Lexical entrainment; Spoken dialog systems; Data-driven model; Rule-based model

1. Introduction

The use of Spoken Dialog Systems (SDSs) in everyday life is getting closer to being a reality. However, errors caused by speech recognition and understanding modules may result in incorrect dialog manager decisions concerning the next action to take and may prevent users from trusting and regularly using SDSs. Recent research, especially in the use of statistical frameworks for dialog management (Williams and Young, 2007; Lee and Eskenazi, 2012), has produced significant and promising improvement in performance. In this paper, a new contribution to the improvement of SDSs

☆ This paper has been recommended for acceptance by A. Potamianos.
* Corresponding author at: Spoken Language Laboratory, INESC-ID Lisboa, Rua Alves Redol 9, 1000-029 Lisboa, Portugal. Tel.: +351 919 327 971; fax: +351 213 145 843.
E-mail address: jose.david.lopes@l2f.inesc-id.pt (J. Lopes).

http://dx.doi.org/10.1016/j.csl.2014.11.007
0885-2308/

is presented: the incorporation of the principle of lexical entrainment (Brennan and Clark, 1996; Garrod and Anderson, 1987) to task-oriented SDSs. It uses the same principle of entrainment that has been observed in human–human dialogs (Brennan and Clark, 1996). Often in human communication one speaker will use the terms of the other speaker (entrain) in an effort to create a common ground and to communicate efficiently. Reitter and colleagues (Reitter et al., 2006) introduced priming as the processing of one speaker influencing the linguistic decisions of the other. Hence, the linguistic structures that will be used to influence the linguistic decisions can also be called primes. The goal of our work is to implement this influencing processing in SDSs in both directions, having a system that automatically entrains to the user whenever the resulting word choice does not have a negative impact on system performance. Otherwise, the system should be able to automatically react to an unsuccessful prime by proposing a new prime with the same meaning. We believe that there are several advantages to the use of this principle for SDSs. The interaction will become more natural, the Word Error Rate (WER) is likely to decrease, and consequently system performance should increase.

Our first step towards an SDS performing automated lexical entrainment was described in Lopes et al. (2011). The context was the Noctívago system, an experimental agenda-based SDS in European Portuguese that provides schedule information for night buses in Lisbon. This system was inspired by Lets Go Raux et al. (2005), a live system that gives schedule information for real bus users in the Pittsburgh area since 2005. The choice to cover night buses was made based on the similarity between Lisbons night-bus network and bus frequencies and the Pittsburgh ones. Both systems are telephone-based, use the Olympus open-source architecture for SDSs Bohus et al. (2007), and use Ravenclaw (Bohus and Rudnicky, 2009), an agenda-based dialog manager.

After the above-mentioned trial, the next step was the automation of prime selection for both systems. The automated prime selection was implemented in two different versions of both systems. Noctívagos new version used a multi-modal web interface with an agent and a push-to-talk button. This version enabled us to easily recruit new users to test the different configurations. Nevertheless, the amount of data collected was far less than the amount that can be collected with a live system, like Lets Go. A set of entrainment rules was implemented and tested in the two above-mentioned versions of Noctívago and Lets Go, which target the same domain but differ in language, type and number of users, type of dialog manager (Lets Go used the Cornerstone statistical dialog manager (Lee and Eskenazi, 2012)), speech recognizer and synthesizer, and interface. The results showed that entrainment rules had a positive impact on system performance, especially in Lets Go. In this system, the results showed a 10% relative reduction in the number of unsuccessful dialogs and 6% average relative reduction in the total number of turns per session (Lopes et al., 2013).

This paper describes the next step, going from the use of entrainment rules to the use a statistical-based model for lexical entrainment. First the model was trained using data collected from earlier studies and tested with off-line data from both Noctívago and Lets Go. Then a statistical-based method for prime selection was implemented in the new version of Noctívago for live tests. In this version, same dialog-state tracking module used in Lets Go replaced the agenda-based dialog manager. We have used this system to compare data-driven prime selection to the previously-developed entrainment rules, random and fixed prime selection.

This paper begins with a review of related work in Section 2. Section 3 highlights the differences between the architecture of the systems used. Section 4 describes the lexical entrainment features used in both the rule-based and data driven approaches. Section 5 presents the tests carried out with the rule-based approach for both Noctívago and Let's Go. Section 6 describes the data-driven models for prime selection and results of tests with these models. Section 7 discusses the results and Section 8 presents conclusions and future work directions.

2. Related work

SDS success rate is often affected by speech recognition errors. Many systems use speaker- and noise-adapted models to reduce errors in adverse conditions, but for most SDSs, like Noctívago or Lets Go, which operate in noise-adverse conditions, some of the power of adaptation is lost since caller id is not available, and thus a priori adaptations to the users cannot be implemented. This motivated us to find alternative methods to adapt our system on-the-fly in order to improve speech recognition performance, and consequently the increase the dialog success rate.

A straightforward solution would be to use restricted vocabulary and syntax to provide better recognition from the system perspective (Levow, 2003). This could be done using previously developed techniques (Roe and Riley, 1994; Tan et al., 1999; Anguita et al., 2004). However, this approach ignores the users lexical preference, and may result in the system using terms that users never adopt. As a consequence, the system may sound less natural and the user

may feel less engaged. The user can also feel free to choose any term they wish, thus while the systems vocabulary decreases, the users vocabulary increases. The result is degraded system performance.

Another solution that has been explored is to enrich the confidence score with other sources of information. Basically, this approach uses a set of features based on live knowledge sources to improve the error recovering skills of the system (Bohus, 2007; Schmitt et al., 2011). The features used to train the model could be provided by the ASR (e.g. acoustic confidence or speech rate), the Dialog Manager (e.g. the current dialog state, or if the received answer was more or less expected), or the Language Understanding module (e.g. the number of slots in the parse). They could also be Dialog history features, such as whether the preceding turn was a non-understanding, or Prosody-related features, such as pitch or loudness. Annotation in terms of dialog acts and/or emotions was also used in some studies. These features can be used to train a fully supervised (Litman et al., 1999; Hirschberg et al., 2004; Schmitt et al., 2011) or implicitly supervised (Bohus and Rudnicky, 2007) model for confidence annotation. Based on the given confidence value, the system could modify the strategy to adopt. For instance, if a given slot value has a low confidence score, the system can trigger a confirmation strategy. Confidence annotation is extremely helpful for error recovery in SDSs, however per se it can only influence system actions, without affecting the system's lexical choices.

The current state-of-the-art in SDS dialog management is the use of dialog-state tracking. Dialog managers using this technique have the advantage of keeping a large number of dialog states available. Based on the observation value given by the ASR output and the confidence value given by the confidence annotator, they predict the users next action, which is not observable by the system (Williams and Young, 2007). Training a dialog model using this technique requires large amounts of data. Often user-simulated data is used for this purpose. These modules achieve remarkable improvements when dealing with low-confidence turns. However, at present they have not yet exploited the fact that the system is able to influence the users lexical choices on-the-fly, encouraging the use of words that are easier for the ASR to process, by using the principle of lexical entrainment.

Entrainment is beginning to be recognized as an important direction in SDS research (Hirschberg, 2011). It has been reported that in human–human dialogs entrainment occurs at various levels: lexical, syntactic/semantic and acoustical (Pickering and Garrod, 2004). If entrainment occurs at one of these levels, it should also elicit entrainment at other levels (Pickering and Garrod, 2004). Studies carried out for human-human dialogs have showed that subjects establish implicit conceptual pacts with one another in order to achieve success in task-oriented dialogs (Brennan and Clark, 1996). In these studies, participants collaborated to co-ordinate word choice rather than only using their own preferred words. They followed the output/input coordination principle (Garrod and Anderson, 1987), which states that the next utterance is going to be formulated according to the same principles of interpretation as the previous successful utterances. This coordination is not reached by explicit negotiation of the lexical items to be used, but rather through imitation during the interaction. Frequency is also important. The more common is a particular conceptualization, the stronger is the conceptual pact (Brennan et al., 1996). Evidences of priming are more visible in task-oriented dialogs (Reitter et al., 2006), which is the domain of most SDSs. This constitutes a theoretical background that could be used to implement a similar behavior in an SDS. When combined with dialog-state tracking dialog management and accurate confidence annotation, this is likely to increase the system performance.

Differences between human–human dialogs and human–machine dialogs were studied in Brennan (1991), Brennan et al. (1996), Branigan et al. (2010). Humans use abbreviated and telegraphic strings when communicating with machines. They establish conceptual pacts differently, since they believe that systems are not able to negotiate with them. They tend to adopt system's terms because they expect the system to be inflexible and want to avoid future errors (Brennan et al., 1996). According to Branigan et al. (2010), the motivation for entrainment in human-machine dialogs is to increase successful communication. The use of highly dispreferred linguistic structures, is more likely to occur in human–computer dialogs than in human–human dialogs, if the users believed that this is necessary for successful communication. The computer's ability to understand users is often viewed as limited and domain constrained. The use of lexical entrainment in SDSs should eliminate the use of highly dispreferred lexical items and lessen the implicit belief that SDSs are inflexible.

Lexical entrainment has been successfully tested in a text-based dialog system (Matessa, 2003), both in terms of performance and user preference. Previous work on lexical entrainment for SDSs was also carried out in the context of Let's Go (Stoyanchev and Stent, 2009). Varying syntactic structures and primes was shown to influence the user's choice of words. The different word choices studied here did not correspond directly to system concepts, although they were shown to influence concept acquisition by the system. In Parent and Eskenazi (2010), the authors confirmed that real users also entrained to SDSs. They went further, modifying the primes that the system had been using for a

long time. Unlike (Stoyanchev and Stent, 2009), the modified primes intrinsically affected the system behavior. Parent and Eskenazi (Parent and Eskenazi, 2010) observed that some of new primes were adopted while others were not. This is our evidence that users have prime preferences as to which primes they actually choose to use. This may be explored to automatically identify the most suitable primes. In addition, (Nenkova et al., 2008) found that the success in task-oriented dialogs can be correlated with measure of entrainment.

Other types of entrainment between humans and machines have been found. Users tend to follow the same syntactic pattern of the question they are answering (Branigan et al., 2003). Entrainment at lexical and acoustic/prosodic levels was investigated to find possible correlations with the learning process (Ward and Litman, 2007). Acoustic entrainment was also used to influence the way users speak to the system (Fandrianto and Eskenazi, 2012) in order to influence users to return to a "neutral" speaking style. Higher error rates have been noted in the presence of hyperarticulation and shouting. The authors developed methods to automatically detect these speaking styles. Then, they tested different strategies to deal what they had detected on the fly: explicitly asking to revert to their normal style, changing the dialog slot or changing the system's speaking style. This last strategy tries to make users acoustically entrain to the system, returning them to a "neutral" speaking style (softer getting them to stop shouting, for example) that was more likely to be successfully recognized by the system. Evidence of acoustic entrainment was found. All of the strategies performed better than the baseline system.

3. Systems description

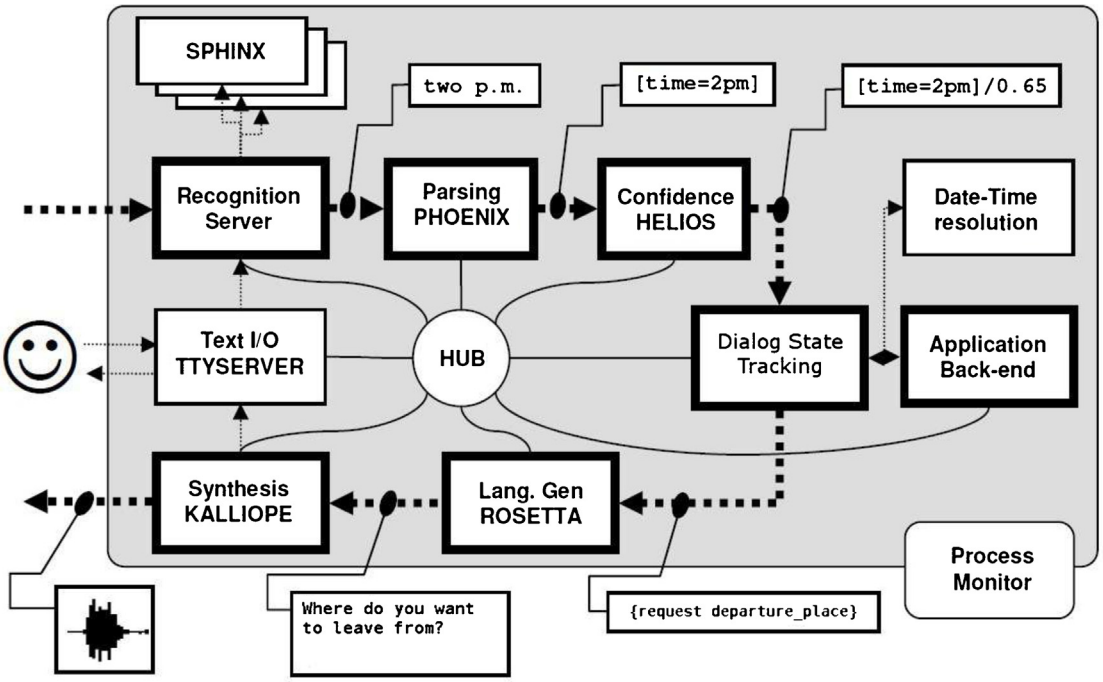
As mentioned in Section 1, two systems were used in the studies described in this paper. Fig. 1 presents both architectures. Some components are shared between the two systems: the Phoenix robust semantic parser (Ward and Issar, 1994), the Helios confidence annotator (Bohus and Rudnicky, 2002), the natural language generator Rosetta (Rudnicky et al., 1999), the Back-end and the date-time resolution. Some of the modules required modifications to European Portuguese, namely new grammars were created for Phoenix and the DataTime parser to convert strings to dates was translated. The schedules from Lisbon's bus network were used as Noctívago's Back-end. New system prompt templates in Portuguese were also created to be used by Rosetta.

The main difference between the architectures of the two systems resides in the speech recognition and synthesis modules, and in the dialog manager. Let's Go uses a recognition server that communicates with different recognizer versions (e.g.: male, female, etc.), running PocketSphinx (Huggins-daines et al., 2006). In Noctívago, the Audimus (Neto et al., 2008) speech recognizer is integrated in a web interface module. Audimus initially uses a language model trained for broadcast news, which was then replaced by domain specific language models trained with a very small 30k artificial corpus generated from the grammar. Let's Go uses the Kalliope synthesis engine (Bohus et al., 2007) using a domain-adapted synthesizer built with the techniques described in Black and Lenzo (2000). Noctívago uses the general-purpose festival-based DIXI synthesizer (Paulo et al., 2008), also integrated in the web interface module. Let's Go uses a recently trained dialog state-tracking dialog manager (DM) (Lee and Eskenazi, 2012), whereas Noctívago uses the agenda-based Ravenclaw DM (Bohus and Rudnicky, 2009). The web interface module of Noctívago also includes a virtual agent presented in Fig. 1b.

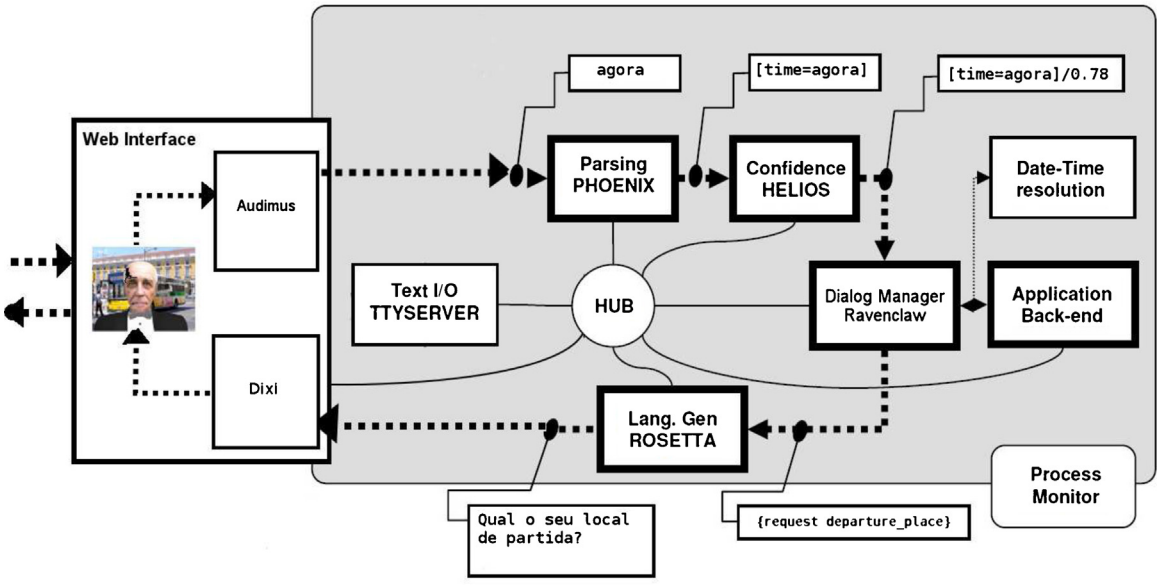
4. Entrainment events

As described in Section 2, there is evidence in the literature of lexical, syntactical and/or acoustic entrainment, human–human and human–computer dialogs. To perform automatic entrainment, this evidence has to be mapped into a format that an SDS can use. The system has to extract information on the fly from live dialogs that indicate whether the user accepted the prime the system proposed, if she proposed another prime or if the prime proposed is hindering the ASR performance. In addition, most SDSs generate a confidence score for each turn that could be used to validate the actions previously enumerated. Based on this information, the system has to decide whether the prime being used should be retained, or replaced by a different prime.

The system logs may be analyzed in chronological order to find the user prime preferences, i.e. if the users took up the system prime choice or used a different term. If the users take up the system prime, then entrainment has taken place. This will be henceforth designated as an *Uptake* (Uptk) event. An example of this behavior can be found in the first excerpt of a Let's Go dialog in Table 1. In this excerpt, the system proposed "new conversation" (S1) and the user followed the system (U1). If a user does not take up the system prime, this can be interpreted as the user deciding



(a) Olympus Architecture used in Let's Go. Adapted from [7].



(b) Olympus Architecture used in Noctívago. Adapted from [7].

Fig. 1. Olympus Architectures used in Let's Go and Noctívago.

not to entrain. This will be called the *No Uptake* (NUptk) event. This behavior can be found in the second excerpt in Table 1, where the system proposed "new dialog" (S2), and the user followed with "new task" (U2). A third event is where the user says a prime for a concept that was not used until that point in the dialog. This will be considered the *No Previous Use* (NPU) event. An example can be found in the third excerpt presented in Table 1, where the user said

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

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: To ask about a different **journey**, you can say, start a **new conversation**. If you are finished, you can say goodbye.
U7: SPRING 94C DUQUESNE
S10: To ask about a different **journey**, you can say, start a **new conversation**. If you are finished, you can say goodbye.
U8: HILL CROSS TURTLE THANK.YOU.
S11: To ask about a different **ride**, you can say, start a **new dialog**. If you are finished, you can say goodbye.
U9: START **NEW DIALOG**

"now" (U6), before the system used this concept. When this event is present, the system should follow up by taking up the prime proposed by the user. These three events are designed to cover all possible user entrainment behaviors, and will be called the *user prime events*.

The fourth event illustrated in Table 1 cannot be considered an entrainment event, but will be later used to represent the system preference. This event is where the dialog-manager is unable to bind the parser output to a concept. It will be called a *Non-Understanding* (NUnd) event. This was the case of the last dialog excerpt (U7 and U8), since the system was not expecting a bus stop or a route at this point of the dialog. If this is a recurring situation for a specific prime, the system should be able to find a different alternative prime (S11) that might work better than the previous one (U9).

5. Heuristic entrainment rules

The selection of the most appropriate prime to use at any given instant should be ideally driven by the four above-mentioned events. Unfortunately, the limited data resources that were initially available lead us to develop a set of heuristics for prime selection that combine the user and system events above mentioned, instead of a data-driven method, to find the best primes.

5.1. Version 1: implementing entrainment heuristics in Noctívago

According to previous findings (Garrod and Anderson, 1987; Brennan and Clark, 1996), different speaker pairs may reveal different strategies in the selection of the primes that are used (and thus use different primes). The ideal solution would be to have a user-dependent model to select the best primes. Since neither Noctívago nor Let's Go have user-dependent dialog models, a two-stage algorithm was adopted to rank all possible primes for each concept. In the first stage, "Long-Term entrainment", the system determines the best prime for any speaker, based on past interactions

it has had with many speakers. The prime selected in this phase will be the first prime that the system uses in a new session. In the second phase, "Short-Term entrainment", the system tries to coordinate the primes with the user's choices on the fly as the dialog progresses, trying to find the best primes for each user.

5.1.1. Long-term entrainment

Our previous study (Lopes et al., 2011) pointed out a possible correlation between the number of *No Uptake* events and the most commonly used primes in daily language. A possible explanation for this is the fact that the terms that are the most frequent in general use are those that the users employ even if the system did not use them. To confirm this, the data collected was analyzed and the frequency of *Uptake* and *No Uptake* events was obtained and correlated with the number of hits each prime had on the *Português Fundamental* corpus (Bacelar et al., 1987), a frequency corpus for spoken European Portuguese. The values found were −0.23 for *Uptake* events and 0.99 for *No Uptake* events. Thus, the primes were ranked based on the number of *No Uptake* events, normalized by the number of times that the prime was used in system prompts. The higher the number, the better the prime. This resulted in the long-term prime ratio for prime i:

$$R(i) = \frac{count_{NUptk}(i)}{count_{system}(i)} \tag{1}$$

5.1.2. Short-term entrainment

In this stage, the goal is to make the users and the system converge in the set primes used during a session. The system is expected to follow the user's choice of terms unless this choice degrades the system performance. To map this behavior, a set of heuristics was designed where three update factors were created for each user prime event: φ_{Uptk}, φ_{NUptk} and φ_{NPU}. These factors will modify the initial long-term prime ratio $R(i)$ according to the following heuristics:

- If an *Uptake* event occurs for prime i, then $R(i)$ is increased by φ_{Uptk}. Example: in the first excerpt from Table 1, R(*new conversation*) will be increased by φ_{Uptk};
- If prime i is used when prime j was proposed, then $R(i)$ is increased by φ_{NUptk} and $R(j)$ is decreased by the same amount. Example: in the second excerpt from Table 1, R(*new task*) will be increased by φ_{NUptk} and R(*new dialog*) will be reduced by φ_{NUptk};
- If prime i is spoken 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 the next user turn is a *Non-Understanding*, then $R(i)$ is reduced by $count_{NUnd}(i)$, where $count_{NUnd}(i)$ is the number of non-understandings for prime i in a session. Example: in the last excerpt from Table 1, the $R(i)$ for "journey" and "new conversation" will be decreased by the number of non-understandings flagged so far in that session (2), for each of them.

5.1.3. Testing entrainment rules on an experimental system

The heuristic rules were implemented in the Noctívago system. Table 2 shows the list of primes used for this study. A prime candidate could, in principle, be any word/expression that has synonyms (Ward and Litman, 2007). For instance, when asking for bus stops or times, users may have different ways to refer to the same thing. This means that they could be considered as primable concepts. However, if they were primable concepts, this would involve several changes to many of the system modules. Thus, we have limited primable concepts selected in our previous study (Lopes et al., 2011) to: *now*, *next bus*, *previous bus* and *new query*. To increase the chances of entrainment by having more primable concepts, the dialog flow was modified. The system performed an explicit confirmation when all of the slots were filled. If the user answered no, meaning that some slot could have been wrongly filled, the system asked which slots were not correct. This change allowed three more primable concepts to be included: *arrival place*, *origin place* and *time*. The last three concepts do not correspond to the slot value (bus stops or time expressions), but rather to the way that they are mentioned in the system prompts. In addition, the system is now able to provide the price of tickets, which, in turn, is another chance to prime the user. Table 2 shows the prime set used in this study. The "type of prime" column indicates whether the content of the slot will have any influence on the course of the dialog. Primes associated with intrinsic slots do influence the course of the dialog. In the dialog presented in Table 1, in U1 "new conversation" is an intrinsic concept. The primes associated with non-intrinsic slots (used in Table 7) are used in system prompts, and the course

Table 2
Primes used by Noctívago in the heuristic method tests.

Type of prime	Concept	Primes
Intrinsic	Next	*próximo/seguinte*
		agora/imediatamente
	Now	*neste momento/o mais rápido possível*
		o mais brevemente possível
	Price	*preço/valor*
		outro percurso/nova procura
	Start Over	*nova pesquisa/procurar novamente*
		outra procura/nova busca
	Arrival Place	*chegada/destino*
	Origin Place	*partida/origem*
	Time	*horas/horário*

of the dialog is not altered if the prime is incorrectly recognized. For instance, if when asking for the departing stop the user answers "start from downtown" and the system recognized "stay from downtown", where "start" is a prime, the slot is filled with the same value, "downtown", in both cases. All of the primes were incorporated in prompts in language generation templates, language models and parsing grammars.

These tests had two main goals. The first was to compare the users' behavior with and without Short-Term entrainment. The second was to find the best way to combine a confidence measure with the prime events detected. For this purpose, four different test sets were created. In Set 1, the dialog confidence score generated in Helios was used to threshold the Short-Term entrainment updates. In Set 2, the ASR confidence score was used for the same purpose. In Set 3 Short-Term entrainment updates were performed regardless of the confidence score value. Finally, Set 4 only performed the Long-Term entrainment updates.

The update factors used in Sets 1–3 were handcrafted. The values for φ_{Uptk}, φ_{NUptk} and φ_{NPU} were set to 1, 2 and 3 respectively. These values were chosen to give more importance to the least frequent events, as the previous findings pointed to their relevance for prime selection. Also, these values ensure that they are superimposed on the initial ratio $R(i)$, as they are at least one order of magnitude higher than the average initial $R(i)$ for the data collected in Lopes et al. (2011), 0.09. This was done in order to reinforce the convergence in prime selection during each session.

Test Set. Users were recruited by e-mail or Facebook event to participate in the experiment. In both cases they were given a short explanation of the requirements to complete the task. Then they were given a web link to access the system via the Flash-based multi-modal web interface of Noctívago (Fig. 1b). They were told to carry out three consecutive requests to the system within one session. This made each dialog longer and consequently gave the system more chances to apply the heuristics. The four Sets ran alternately, one after the other, during the test period, each for the same amount of time, but did not change within an individual session. The users were not aware that the system was running different configurations.

At the end of each dialog, the users were asked to fill in a questionnaire based on the PARADISE framework for SDS evaluation (Walker et al., 1998). The questionnaire also tried to evaluate if the users noticed any difference in the lexical choice, by asking them 'if the system understood them better towards the end of the session'. We hoped that the use of adequate primes would result in better recognition as the session progressed.

Results. Table 3 shows the results of the 160 sessions validated in terms of system performance (Dialog Success and Average Number of Turns). The tests were performed by 83 different people. 33 were female subjects and 46 were male subjects. The system was also tested by five non-native users. 13 users already participated in previous tests

Table 3
Dialog performance results.

	Set 1	Set 2	Set 3	Set 4
Number 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

Table 4
WER and correctly transferred concepts results.

	Set 1	Set 2	Set 3	Set 4
WER (%)	59.7	52.3	53.7	47.9
WER primes (%)	52.6	50.1	54.9	43.4
WER intrinsic (%)	53.6	48.3	58.2	44.8
CTC (%)	47.3	39.6	44.5	51.5
CTC primes (%)	45.6	36.2	37.3	47.2
CTC intrinsic (%)	48.3	39.1	37.1	47.3

Table 5
User satisfaction results.

	Set 1	Set 2	Set 3	Set 4
Average satisfaction	19.4	19.5	19.7	20.3
Adaptation	2.75	2.48	3.16	3.38

with our system (Lopes et al., 2011). System performance includes both the estimated and real success. The estimated success is computed live, since it only takes into account whether the system queried the backend and provided any bus schedule information to the user. Real success is computed *a posteriori*, after listening to each session and verifying if the schedule provided actually corresponded to the user's request.

Set 4 achieved the best performance in terms of real dialog success and the second lowest average number of turns per session, although the Chi-square test for dialog success and the one-way ANOVA tests for Number of Turns revealed no statistical significance differences between versions. Since one of our goals was to compare the performance with different confidence measures, Table 3 shows that among those Sets that performed Short-Term entrainment Set 1, which used the dialog confidence measure, was the one with the best performance.

Table 4 shows Word Error Rate (WER) and percentage of correctly transferred concepts (% CTC) for the same tests. A concept is considered correctly transferred when the parsed ASR result is equal to the parsed transcription. Despite the fact that one-way ANOVA tests did not reveal statistical significance, results show Set 4 achieved the lowest WER and the highest percentage of CTC. The relation between the WER and the system performance reveals that Set 1, although it has the highest WER, has the second best real success rate. We observe that the WER for primes is lower than the global WER for all the Sets, except Set 3. Against our expectation, the % CTC is lower for Primes than for the other concepts. However, Sets 1 and 2 achieved the lowest loss in % CTC. The % CTC for intrinsic primes even increased in Set 1, compared to other concepts' % CTC, where Short-Term entrainment is performed and the threshold for the entrainment rules is set.

If we analyze the results from Tables 3 and 4 together, we see that despite the highest WER, Set 1 was the second best in terms of system performance.

The results of the satisfaction quesationnaire are given in Table 5. The analysis of the Adaptation question result, suggests that the users evaluation is more correlated with the session success (0.41, $p-value<0.001$) than with the adaptation of lexical choices (0.13, $p-value=0.14$). Set 4 received the highest rating. However, Set 3 was second highest rated, despite being the worst performing. These results could mean that the users did not notice that the system was adapting to them. They also could mean that the question regarding adaptation possibly induced the users to evaluate other than the lexical choices.

Table 6 presents the results of the entrainment events detected during live interactions. The event percentage was computed as:

$$Event(\%) = \frac{\sum_{i=1}^{P} count_{event}(i)}{\sum_{i=1}^{P} count_{system}(i)} \tag{2}$$

where the *events* are *Uptakes*, *No Uptakes*, *No Previous Use* or *Non-Understandings*.

Table 6
Entrainment events relative frequency.

	Set-up 1	Set-up 2	Set-up 3	Set-up 4
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

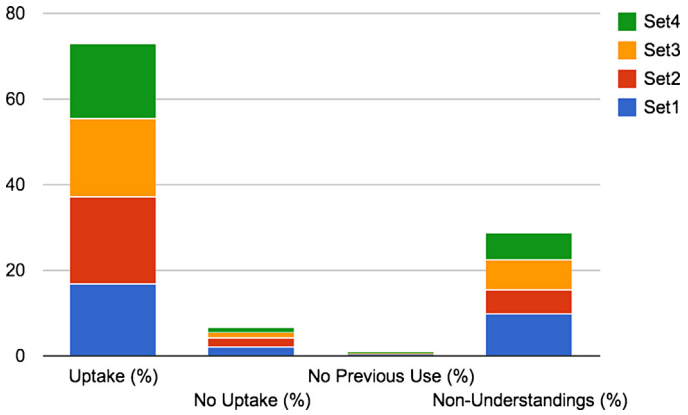


Fig. 2. Accumulated of events percentage.

We see that *Uptake* events are much more frequent than the other events. This signifies that users followed the system proposed prime much more than they used terms of their own choice. This confirms previous findings about the behavior of novice users (Levow, 2003).

A detailed analysis of the systems logs showed that the initial prime rank given by Eq. (1), rarely changed from session to session, unless a *No Uptake* event occurred. There were very few of these events when compared to *Uptake* events, as we also see in Fig. 2. We also examined reactions to non-understandings. With the strategy proposed in Section 5.1.2, a prime could be changed before there was the necessary evidence that it was degrading system performance (Branigan et al., 2010).

The results of these tests do not reveal an improvement in the system performance when Short-Term entrainment is used. This brings us to envisage future experiments using the entrainment rules. The dialog confidence score could be used as a threshold for Short-Term entrainment. Since *Uptake* events are much more frequent than any other prime event, they could also be used to compute the initial prime rank in Long-Term entrainment. An increase in the update factor φ_{Uptk} should also enhance the convergence between user and system during the session. Finally, the approach to the update at a non-understanding event could also be modified to allow the system to gather more evidence that the prime is hindering ASR performance.

5.2. Version 2: implementing entrainment heuristics in let's go

The tests described in Section 5.1.3 revealed some interesting trends. We do note that many of the numbers are not statistically significant and thus do not permit us to draw strong conclusions about the effect of the entrainment rules on system performance. These trends motivated modifications in the entrainment rules. Firstly, the dialog confidence score was chosen to be the only one used as threshold. Second, the initial prime update ratio was modified to include a weighted sum of the normalized number of *No Uptake* and *Uptake* events:

$$R(i) = \frac{count_{NUptk}(i)}{count_{system}(i)} + w_{Uptk} \times \frac{count_{Uptk}(i)}{count_{system}(i)} \tag{3}$$

Table 7
Primes used by let's go before and after the entrainment rules were implemented.

Type of primes	Concept	Old primes	New primes
Intrinsic	*next bus*	next	following/subsequent later/after
	now	now	immediately/right now right away/as soon as possible
	previous bus	previous	preceding/prior/before
	start over	route schedule	itinerary/trip ride/journey
	confirm	right	alright/correct/okay
	help	help	assistance/support more information
	new query	query	request/task dialog/route/conversation
Non-Intrinsic	*origin place*	leaving leave	departing/depart starting/start

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

$$w_{Uptk} = \frac{\sum_{i=1}^{P} count_{Uptk}(i)}{\sum_{i=1}^{P} count_{NUptk}(i)} \tag{4}$$

where P is the total number of primes. Thirdly, the update factor for uptake events, φ_{Uptk}, was increased to 2, in order to enhance the convergence during each session. Finally, the update after a non-understanding were only performed after the second non-understanding, instead of the first non-understanding. In addition, instead of subtracting $count_{NUnd}(i)$ from $R(i)$, $w_{NUnd} \times count_{NUnd}(i)$ is now subtracted from $R(i)$, where w_{NUnd} is computed similarly to w_{Uptk}:

$$w_{NUnd} = \frac{\sum_{i=1}^{P} count_{NUnd}(i)}{\sum_{i=1}^{P} count_{NUptk}(i)} \tag{5}$$

5.2.1. Testing entrainment rules in a real system

The modified set of rules was tested in a live system with real users. Let's Go has been running live since 2005, receiving an average of 40 calls during weekdays and 90 during weekends. The set of primes for this study was extended from the prime set used in Parent and Eskenazi (2010) and is shown in Table 7. This has also increased the number of different prompts that the system could produce, consequently making the system sound more natural.

At the time of the study, Let's Go had a set of regular users who are believed to be familiar with the system's lexical choices. In order to force the use of the new primes, a threshold was set so that the system only used the primes in "Old Primes" column if that was reached.

The Let's Go with the entrainment rules ran live during 31 consecutive days. Its performance was compared to performance of the live Let's Go that ran during the a few weeks just before the rules were tested, over approximately the same period of time.

In order to better understand the type of behavior the system has after the implementation of the entrainment rules, Table 8 shows two dialog excerpts where prime choice was affected. In the first dialog, the user said "new conversation", but the recognition output was not bound to any concept, and the system generated a non-understanding. When the system proposed "new dialog", the user accepted the proposed prime and it was correctly recognized. In the second

Table 8
Excerpts of dialogs where entrainment rules changed the system's normal behavior. Primes affected in bold.

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.

excerpt, the system proposed "following", however the user preferred to use "after". The next time the system had to use that concept, "after" was adopted. These examples show that the system was able to adapt in both directions on-the-fly.

Table 9 presents the dialog performance measures for each of the two versions of Let's Go. The estimated dialog success has increased by more than 2% (10% relative error reduction) and the number of turns has also decreased by almost one turn (6% relative reduction). The prime usage events reflect the fact that the users were familiar with the "Old primes", and they had used more of those terms in the Baseline version than the ones in the entrainment rules version, where the variety of primes was much larger. This also resulted in the increase of the *No Uptake* events in this version. This is another confirmation of the studies that contrast the behavior of experimented and novice users. *Non-understanding* events also increased relative to the baseline version. One possible reason for this is that some of the new primes were not available in the data that was used to train the language and acoustic models, since some of them are new to the system. For this reason they were later manually added to language models and lexicon. The last column in Table 9 shows that many of the results were statistically significant.

5.2.2. Acoustic distance

The previous section showed that entrainment rules used for prime selection can improve system performance. However, the performance numbers do not show whether the system always used the same prime or if the prime selected for each concept varied considerably, if the prime selected corresponded to the most acoustically distinct, if the prime corresponded to the most common word or if the primes used belong to the old or new prime set. In this section, we will try analyze the criteria that may be the reasons for prime selection.

Table 9
Results for let's go tests. Statistically significant differences in bold.

	Baseline	Entrainment rules	Statistical significance test result
Number of sessions	1542	1792	–
Estimated Dialog Success (%)	75.11	77.64	Fisher's. No statistically significant difference.
Avg. number of turns	**12.24**	**11.47**	Two-way ANOVA. $F(1) = 8.131$; $p = 0.004$ ..
Total Uptake (%)	**5.35**	**2.39**	One-way ANOVA. $F(1) = 120.579$; $p = 0.000$.
Total No Uptake (%)	**0.56**	**0.78**	One-way ANOVA. $F(1) = 4.421$; $p = 0.036$.
Total No Previous Usage (%)	**1.92**	**1.75**	One-way ANOVA. $F(1) = 4.496$; $p = 0.034$.
Total Non Understanding (%)	6.33	9.07	One-way ANOVA, $F(1) = 0.240$; $p = 0.624$

Table 10
Primes selected according to the minimal and average acoustic distance for each language model.

Dialog state	Concept	Prime min. distance	Max min. dist. (dB)	Prime avg. distance	Max avg. dist. (dB)
Request place	*Origin place*	Departing	4.35	Leave	15.58
	Now	As soon as possible	5.57	Immediately	13.42
	Help	Assistance	5.01	Support	13.58
Request time	*Now*	As soon as possible	5.07	Now	12.93
	Origin place	Departing	4.18	Leave	14.56
	Help	Assistance	4.99	More information	12.00
Explicit confirmation	*Confirmation*	Alright	4.46	Correct	12.51
	Help	Assistance	4.11	More information	12.00
Request next query	*Next query*	Request	4.35	Query	12.54
	Start over	Itinerary	3.97	Ride	10.76
	Next bus	Next	3.74	Next	11.40
	Previous bus	Preceding	4.48	Before	10.79
	Help	Assistance	4.11	More information	12.00

To start with, the most acoustically distinct prime for each concept at different dialog states was found. Each prime and all the entries in the state-specific language models that included any prime from Table 7 were synthesized with 3 different voices using Flite (Black and Lenzo, 2001), in order to introduce more variability when computing the acoustic distance. The acoustic distances between the synthesized samples of the primes and the remaining entries in each language model were computed with Dynamic Time Warping using the method described in Toda et al. (2007). Finally, the average and minimum distances were computed. Table 10 shows the best primes to use in each dialog state.

5.2.3. Prime usage evolution

In order to capture how prime usage evolved over the 31 days of the study, the percentage of usage of each prime was computed for each concept. Figs. 3–5 show the resulting prime usage.

For the *confirm* concept the system started by proposing "okay", however after "correct" was proposed it remained the most-used prime for the rest of the test period. In this case of *confirm*, the former prime, "right", was rarely used. Compared to the acoustic prime choice, the most used prime coincides with the prime selected with maximum average distance.

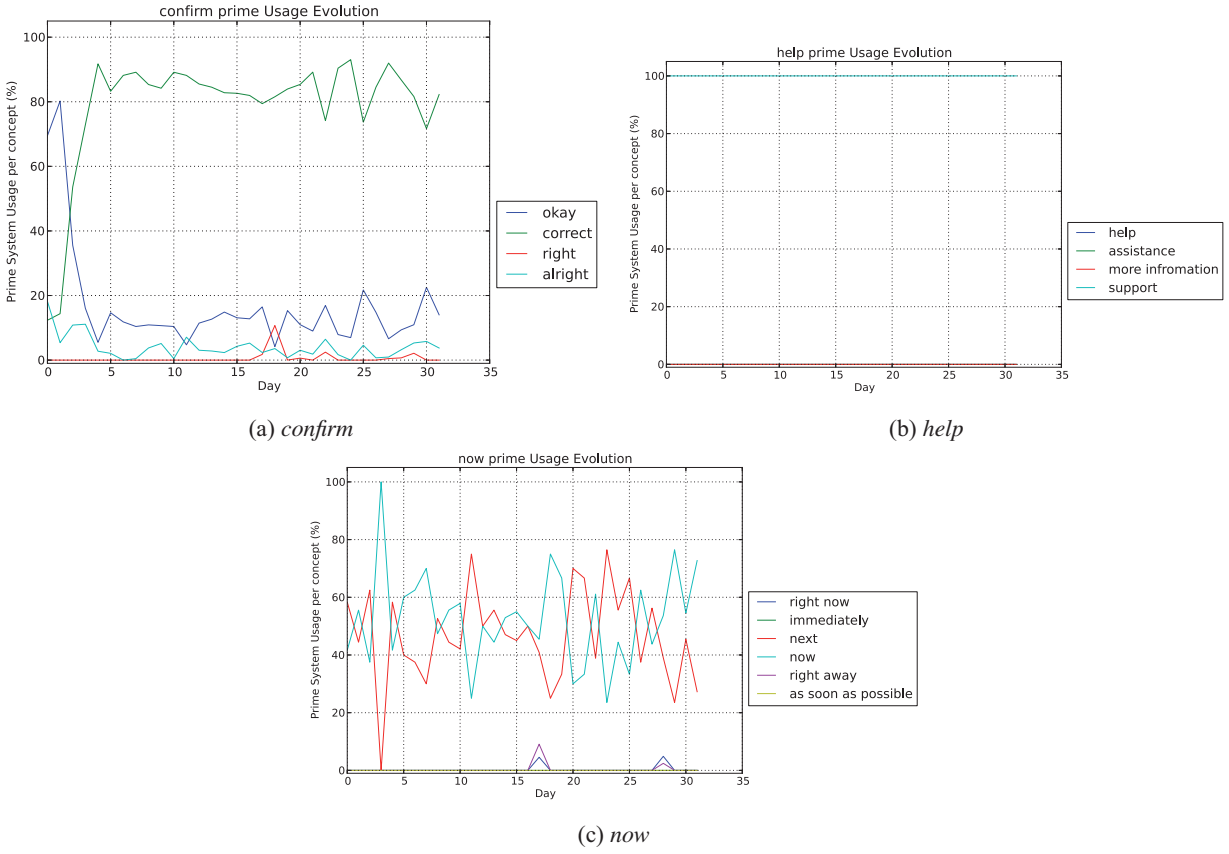
"Support" was always used as the *help* prime. There are two reasons for this. The first one is that the concept only appeared twice in user turns during the test period and the users entrained to the system's choice of prime. The second is that the *help* prime was always followed by the system prompt asking for the place of departure. This means that the initial prime ratio $R(i)$ will never be subtracted by an update factor, since *Non-Understandings* and *No-Uptakes* will never occur. Unless the user picked a different prime, the initial prime will always remain as the chosen prime. The acoustic prime choice would be "support", "more information" or "assistance" depending on the language model used and the average metric chosen.

Despite the "Old prime" restriction, "now" was still the most used prime for the *now* concept together with *next*. This can be explained by the fact that it was primarily detected in *No Previous Use* events, which have a higher update factor. According to the average acoustic distance, "now" is also a possibility for the prime used by the system to ask for travel time. If the minimal distance is taken into account, then "as soon as possible" should be used. Fig. 3c shows that this prime was never used.

The *new query* rule-based prime choice alternated between "dialog", "conversation" and "route", whereas the acoustic prime choice would be "request" or "query". Fig. 4a shows that the system tried "request", however the attempt did not seem successful. "Query" was never used, because of the "old prime" restriction mentioned in Section 5.2.1.

The system started by proposing "start" as *origin place* prime, however, once it changed to "depart", it used any prime except that one. In this case, most of the primes listed were never used. Since it is a non-intrinsic concept, most of the time the concept is not used in user turns, and the prime rank is not updated. None of these primes match the acoustic distance prime choice.

According to Fig. 4 there is no clear prime choice for the *start over* prime. The use of *start over* has the same effect in the dialog of the *next query* concept, i.e., after the user received the schedule information the system will ask for the



(a) *confirm* (b) *help*

(c) *now*

Fig. 3. Prime Usage over time for the concepts in *confirmation*, *help* and *now*.

next action. If the answer is *next query* or *start over*, the system will restart the dialog from the beginning. At this point of the dialog, there is a large number of *Non-Understandings*, which often results in the update of the prime ratio. In addition, since the prompt explicitly directs the user to the *new query* prime (see S6 in Table 8), this concept is less used and consequently the prime ratio is only updated after *Non-Understandings*, which results in significant prime variance.

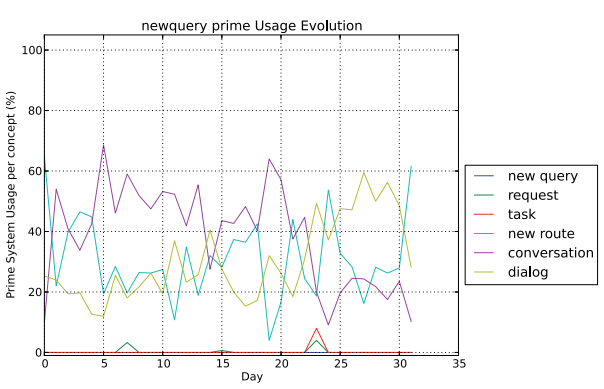
For the *next bus* primes the system first proposed "later" and "after", but when "following" was proposed, the system kept it as the most used prime for the rest of the test period. The old prime "next", was only used for very limited periods due to the "Old prime" restriction. The acoustic distance would have recommended the use of "next".

The system alternated between "prior" and "preceding" for *previous bus* concepts. Occasionally, it used the prime "before", and rarely the old prime "previous". "Before" was the prime chosen according to the average distance. Apparently the system could not choose one prime as being better than the other. "Preceding" corresponds to the acoustic choice according to the minimum distance.

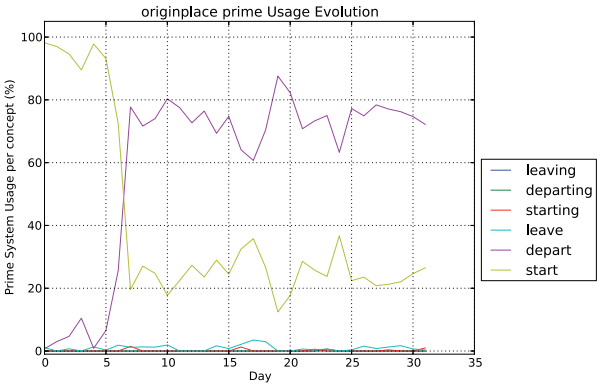
The comparison between the acoustic distance prime selection in Table 10 and the entrainment rules prime selection in Figs. 3–5 shows that the two methods can lead to different prime choices, as is expected given that prime choice was a result of the system and user preferences. In addition, since prime selection with entrainment rules constantly adapts to each user, the prompts have more variety and the system sounds more natural. However, the acoustic and entrainment driven prime selection lead sometimes to the same primes. This means that the acoustic distance could be used as an alternative to rank primes, if no prior entrainment study had been carried out.

6. Data-driven prime selection

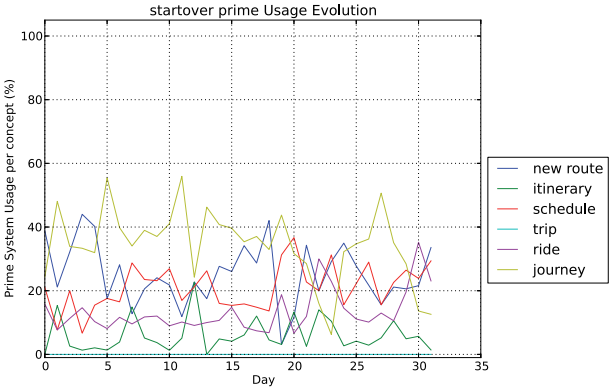
The use of entrainment rules improved the system performance. However, a data-driven method could be a more robust approach to the problem. Also, if data was available, it would be easier to generalize a method to perform live



(a) *new query*

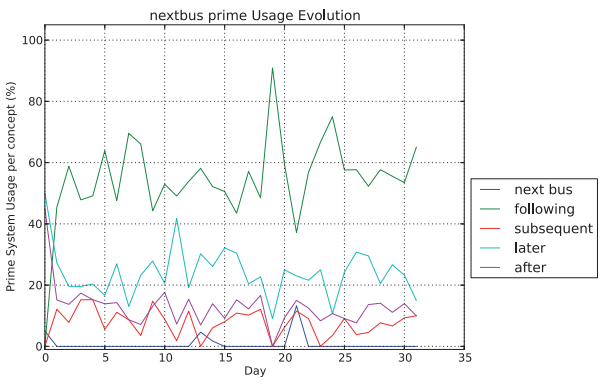


(b) *origin place*

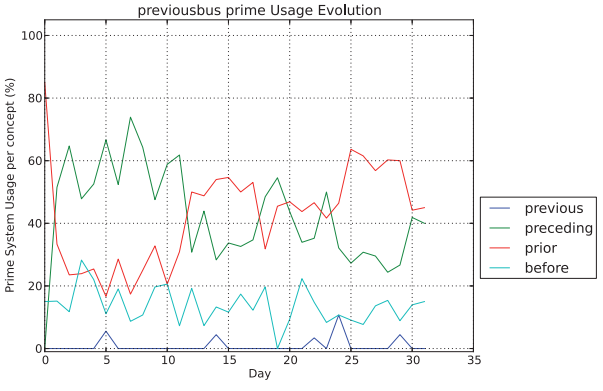


(c) *start over*

Fig. 4. Prime Usage over time for the concepts *next query*, *origin place* and *start over*.



(a) *next bus*



(b) *previous bus*

Fig. 5. Prime Usage over time for the concepts *next bus* and *previous bus*.

Table 11
Example of how the prime distance was computed.

[$d_{new\ conversation} = 0$] S1: To ask about a different **journey**, you can say, start a **new conversation**. If you are finished, you can say goodbye.
[$d_{new\ conversation} = 1$] U1: START A **NEW CONVERSATION**
...
[$d_{now} = 0$] S3: Welcome to the CMU Let's Go bus information system.
[$d_{now} = 0$] S4: To get support at any time, just say support or press zero.
[$d_{now} = 0$] S5: Where would you like to start from?
[$d_{now} = 1$] U3: SHERADEN
[$d_{now} = 1$] S6: Where do you wanna go?
[$d_{now} = 2$] U4: DOWNTOWN
[$d_{now} = 2$] S7: Going to DOWNTOWN. Did I hear that correctly?
[$d_{now} = 3$] U5: YES
[$d_{now} = 3$] S8: When would you like to travel?
[$d_{now} = 4$] U6: **NOW**

entrainment for any system, rather than each system developer creating its own heuristics. In this section a statistical model will be described along with the first off-line results of its use in Let's Go and Noctívago. Finally, a live test using Noctívago is also described.

6.1. Prime selection model

A statistical model for prime selection shares the goals of the heuristic entrainment rules: adapt the system's choice of prime in view of improving system performance. Performance mprovement may be achieved by combining the system and user prime preferences to reduce the WER of the primable concepts. A model that uses entrainment-related features could be trained to predict the WER of each prime at a given point of the dialog.

Empoying transcribed data, a supervised learning method could be used to train a regression for a feature set derived from the entrainment events and other sources of information that could help to predict the WER. The prime with the lowest WER prediction should be the one used by the system.

The events presented in Section 4, *Uptakes*, *No Uptakes* and *No Previous Use* as user events, and system's *Non Understandings*, will be treated as binary features. The dialog confidence score that was previously used to validate the detected user events will be now used as a feature. Two more features were added to the feature set: the distance to the previous use of the prime and an entrainment measure created for human-human dialogs (Nenkova et al., 2008).

The distance can be important in validating the user events. There is a high probability that the user will entrain during the turn immediately after the prime was presented. If the distance to the prime is short, the prime is more likely to be correctly recognized. This is confirmed by the correlation found between the distance and the confidence score, −0.35, and between the average distance and the session estimated success, −0.54, for the data collected with Let's Go with the entrainment rules (both values are statistically significant). The distance is given by the number of user turns between the last time the system used the prime and the current turn when the prime was used. Table 11 shows how to compute the distance for "new conversation" and "now". In the case of "now", since the prime had never been used in the session before that, the distance is simply the number of user turns since the beginning of the session. The minimum value is 1.

The entrainment measure for prime p, adapted from Nenkova et al. (2008), in an SDS is given by:

$$Entr(p) = -\left|\frac{count_{user}(p)}{ALL_{user}} - \frac{count_{system}(p)}{ALL_{system}}\right| \tag{6}$$

where $count_{user}(p)$ is the number of times that user used prime p and ALL_{user} is the total number of words said by the user. This measure gives the similarity of use of prime p between the user and the system during a session and it was correlated with task success in human-human task-oriented dialogs.

Table 12
Number of turns used to train the prime selection regression.

(a) Noctívago

Dialog State	# Turns
Request next query	331
Generic	369

(b) Let's go

Dialog State	# Turns
Request next	1428
Request time	1040
Request stop	143
Inform	434
Generic	1733

6.2. Training and testing the model

A prime selection model was trained for Noctívago and Let's Go. The Noctívago model was trained using the data transcribed from the first studies described in Section 5.1.3. The Let's Go model was trained with two months of data transcribed using crowdsourcing, that was released for the Spoken Dialog Challenge (SDC) (Black et al., 2010). It is important to point out that the data used for the Noctívago model came from a version of the system that already performed automated prime selection, as described in Section 5.1.3, whereas the system used to collect the SDC data did not.

System logs were analyzed to extract the features described in Section 6.1. One feature vector, F, was generated for each user turn where the presence of a prime was detected. The feature vectors were grouped by dialog states, to train different models for each dialog state, since primes are used differently from state to state. However, to have significant amounts of data for each state, the states with similar prime behavior were merged into a single category. For instance, the *Request Origin Place* and *Request Destination* place were merged into the *Request Stop* category. In the Noctívago dataset, however, since many states are under-resourced, the states with less than 300 samples were grouped to create a *Generic* category. The distribution of data per state in both corpora is given in Table 12.

The datasets were split in 75% for train and 25% for test. Several regression methods were tested. Once the model was trained, the correlation between the predicted WER and the actual WER was computed for the test set, together with the coefficient of determination (R^2) which measures the quality of the model (the closer to 1.0, the better the model). The results achieved for both datasets using Linear Regression (LR) and SVM regression (SVM-R) with linear kernel are presented in Table 13. These regression methods outperformed the other methods we tried.

The results for the Noctívago regressions show remarkable correlation values, especially in the *Generic* model, and in both cases the correlation is statistically significant. Nevertheless, the coefficient of determination is not very high in the *Request Next Query* model.

The regressions trained for Let's Go, apart from the *Request Next* model, have lower correlation values when compared with the models trained for Noctívago. There are several factors that may have contributed to this result. The first one is the data set used to train the regression which was generated with a version of the system that did not have an entrainment policy, and consequently a shorter set of primes, which means that the data collected might have fewer examples of the user prime events described in Section 4. Second, the context of the two data collections is substantially different. While the Noctívago data was collected from paid users in an experimental set-up, Let's Go was collected with real users, some of them believed to be regular users of the system. Generally, low experienced users tend to entrain more to the system primes (Levow, 2003). Finally, despite the fact that both systems target the same domain, the Noctívago dialog-flow has more chances of entrainment than Let's Go. In the Noctívago dataset there was an average of 2.1 turns with primes per query, while in the Let's Go dataset the average number of turns with primes per query is 1.24. Further research is needed to find a model that better fits Let's Go user's entrainment behavior.

Table 13
Prediction results for prime selection models. Statistically significant correlation values ($p - value < 0.05$) are in **bold**.

(a) Noctívago models.

Model	Measure	LR	SVM-R
Request next query	Correlation	**0.32**	**0.36**
	Coeff. det. (R^2)	0.08	0.11
Generic	Correlation	**0.63**	**0.62**
	Coeff. det. (R^2)	0.39	0.35

(b) let's go models.

Model	Measure	LR	SVM-R
Request next	Correlation	**0.35**	**0.34**
	Coeff. det. (R^2)	0.11	0.05
Request time	Correlation	−0.01	0.06
	Coeff. det. (R^2)	−0.13	−0.02
Request stop	Correlation	0.16	0.04
	Coeff. det. (R^2)	−0.84	−0.11
Inform	Correlation	**0.22**	**0.28**
	Coeff. det. (R^2)	−0.03	0.03
Generic	Correlation	**0.13**	**0.13**
	Coeff. det. (R^2)	0.01	0.09

6.3. Testing the prime selection regression in a live system

The Noctívago SVM regression model was then tested in a live system. The architecture used in these tests has two major differences from the one used in the previous tests: a new web interface with a new agent developed with Unity 3D and new dialog manager. The new architecture is shown in Fig. 6. The agent replacement was mainly due to problems verified in the audio capture. Moreover, the new interface has other advantages regarding its predecessor. First, since the video in generated on the client side, the server does not need an advanced Graphics Processing Unit to generate the video. Second, for the same reason the clients with lower speed connections would be able to watch the

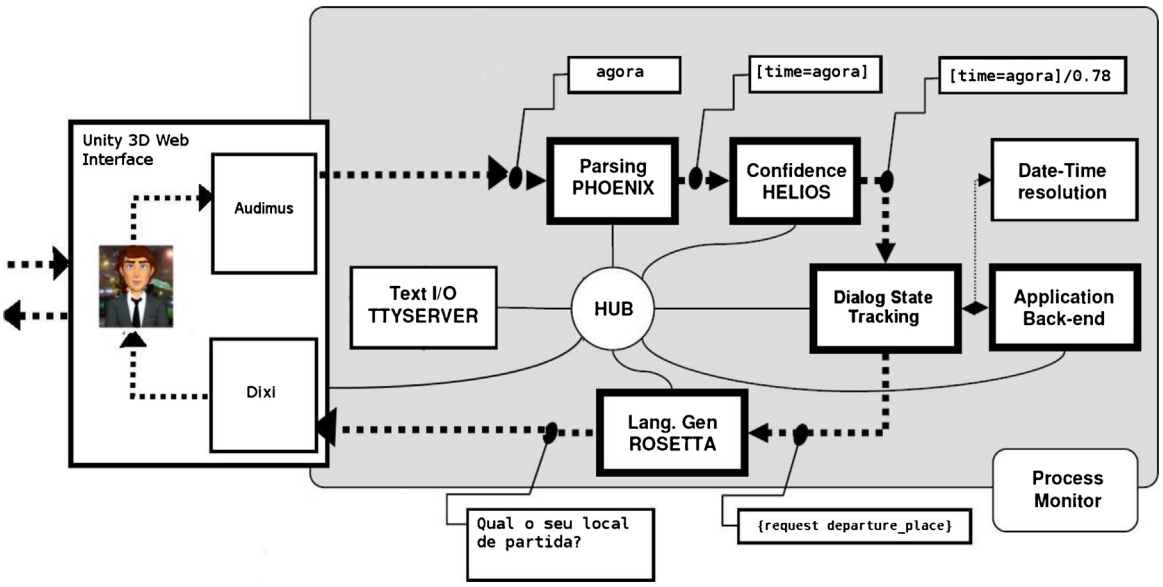


Fig. 6. Noctívago architecture with the Unity 3D interface and state-tracking dialog manager.

video without delay interferences. And finally, the new agent looked more appealing and friendly than the previous one.

The Ravenclaw Dialog Manager was replaced by the Dialog State Tracking dialog manager used in Let's Go (Lee and Eskenazi, 2012). The user action model is the same as the one used in Let's Go, since both systems target the same domain and the Noctívago dataset was not sufficient to train a dedicated user model. This had some implications in the dialog flow, but minor modifications to the original Let's Go state-tracking dialog manager enabled us to include the majority of the prompts previously created in order to increase the chances of entrainment. The confidence calibration model, however, was trained with the Noctívago dataset.

The language models used in these tests were also improved. The previously-used language models were trained with artificially created corpora generated from the parser grammar specification, for *Place* and *Time* models. The corpora used had 30k sentences. The corpora used here was 10 times larger, and the grammar was generated according to the frequencies observed in previous tests, in order to have a similar frequency mapped in the artificial corpora. This was done, since the data collected was still very scarce to train a language model. The *Confirmation* and *Next Query* models were built from the Speech Recognition Grammars Specification (SRGS) standard, since there are fewer options available. This specification was already used in previous tests, but it was now enriched with answers collected from previous dialogs.

In the tests performed with off-line data, each feature vector was considered as an isolated observation. However, since the prime selection depends on the evidence built in previous turns (Garrod and Anderson, 1987), the predicted WER for turn t, $\widehat{WER}_t$, should also incorporate a scaling factor, representing those turns. The WER prediction from the previous turn, $\widehat{WER}_{t-1}$, was chosen to be the scaling factor. Thus, the predicted WER for prime p at turn t is given by:

$$\widehat{WER}_t(p) = \widehat{WER}_{t-1}(p) \times r(F) \tag{7}$$

where, $r(F)$ is the prediction value given by the trained regression for $F = <f_1, \cdots, f_n>$, the feature vector generated during live interaction for every turn where a prime was recognized. Since at the beginning of the dialog, there is no feature vector that can be used to compute $\widehat{WER}$, the system assumes that the most frequently used primes are more likely to have lower WER. Hence, the initial prediction value is given by:

$$\widehat{WER}_0(p) = 1 - \frac{count_{user}(p)}{count_{user}(C)}, \; for \; p \in C \tag{8}$$

where $count_{user}(C)$ corresponds to the sum of all the primes used for concept C. The relative frequency is subtracted to 1, so that the more frequent words have the lowest $\widehat{WER}_0$. In the beginning of each session the primes are ranked according to this value.

6.3.1. Test set

To compare the performance of several algorithms for prime selection, the system using the data-driven method was compared to three other system versions, all of them running the same components, but with different methods for prime selection. The three other versions were the *entrainment rules*, the *random primes* and the *fixed primes* versions. The *entrainment rules* version performed prime selection with the rules described in Section 5.2. The *random primes* version selected the primes randomly. The *fixed primes* version used only the most acoustically distinct primes, computed based on a pre-defined acoustic distance between pairs of phones. The systems ran alternately on different days. The users tested the system using the multimodal web interface without knowing which version they were testing. Once they completed the test, they were asked to fill in a questionnaire based on the PARADISE questionnaire (Walker et al., 1998). Two more questions were added to the questionnaire mentioned in Section 5.1.3: 'if the system was able to proposed alternatives when it encountered problems' and 'if you felt that the system was proposing words in a smart way'. These new questions were used to help to confirm whether users detected the system's adaptation of lexical choices.

The list of primes used is presented in Table 14. Due to the modifications in the course of the dialog caused by the replacement of RavenClaw by the Dialog-State tracking dialog manager, Cornerstone, the explicit confirmation strategy described in Section 5.1.3 where the users were supposed to say which of the given concepts were wrong, was not used. For that reason the concepts *Arrival Place*, *Origin Place* and *Time* that were Intrinsic in the tests described

Table 14
Primes used by Noctívago in the data-driven model tests.

Type of prime	Concept	Primes
Intrinsic	Next	*próximo/seguinte*
		agora/imediatamente
	Now	*neste momento/o mais rápido possível*
		o mais brevemente possível
	Price	*preço/valor*
		outro percurso/nova procura
	Start Over	*nova pesquisa/procurar novamente*
		outra procura/nova busca
Non-Intrinsic	*Arrival Place*	*chegada/destino*
	Origin Place	*partida/origem*
	Time	*horas/horário*

earlier (Section 5.1.3) were Non-Intrinsic here since they were only used in system prompts, and the course of the dialog was not modified when they were not correctly recognized.

6.3.2. Results

A total of 214 dialogs were collected during these tests. 88 sessions were completed by female subjects. 53 of these sessions were performed people from the laboratory. 96 different users have participated in these tests and 4 of them were non-native speakers. 8 people that participated in these tests had already participated in tests described in Lopes et al. (2011) and 23 participated in the tests described in Section 5.1.3.

The dialogs were orthographically transcribed to compute prime and word level performance metrics, since these are expected to be greatly influenced by prime selection methods. The transcriptions were also parsed using an off-line version of the Phoenix semantic parser, and the same grammar used in the live system. Table 15 shows the performance in terms of Out-of-vocabulary words (OOV), WER and CTC. For WER and CTC, the results were further analyzed at the prime level, and at the intrinsic prime level.

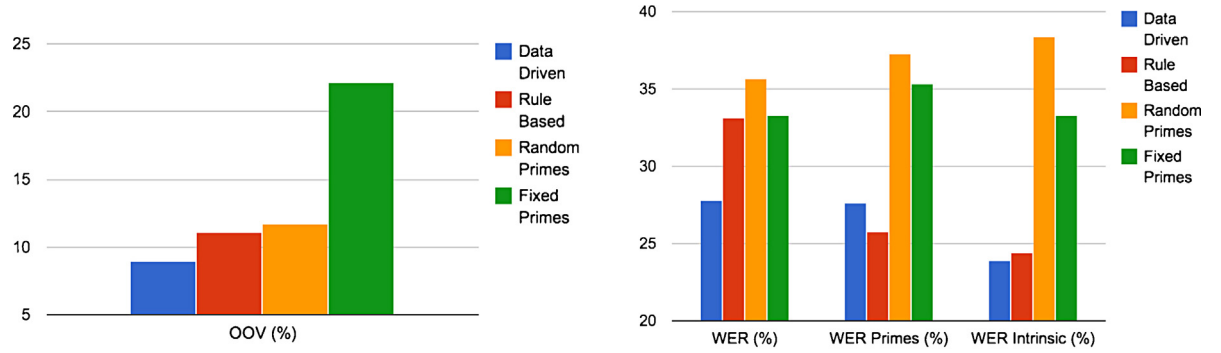
These results show that both versions with prime entrainment policy clearly outperformed the random and fixed primes both in terms of reducing OOV (Fig. 7a), WER (Fig. 7b) and increasing the percentage of CTC in the dialogs (Fig. 7c). The differences are particularly remarkable when the analysis is restricted to primes, and even more for intrinsic Primes, as can be seen in Fig. 7c. Both Data Driven and Rule Based methods were able to improve system performance when dealing with turns involving primes. The Rule Based prime selection method has slightly better performance than the Data Driven one.

Table 16 shows the results for the percentage of entrainment events found in this dataset. The percentage is computed as in Eq. (2).

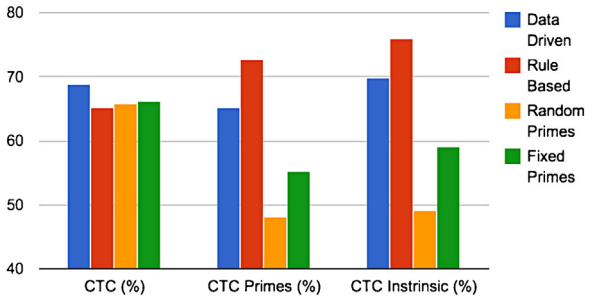
The Entrainment Event results show a higher *Uptake* percentage for the versions with prime selection algorithms. The results for *No Uptakes* show that the highest percentage occurs with the Random Primes version. Algorithms that used different primes were less prone to *Non-Understandings* than Fixed Primes.

Table 15
OOV, WER and CTC Results for the different versions. Statistically significant results in bold (one-way ANOVA with $F(3) = 2.881$ and $p - value = 0.037$).

	Data driven	Rule based	Random primes	Fixed primes
OOV (%)	8.92	11.10	11.68	22.18
WER (%)	27.84	33.05	35.68	33.29
WER primes (%)	27.53	25.77	37.38	35.25
WER intrinsic (%)	23.90	24.38	38.42	33.33
CTC (%)	68.88	65.24	65.78	66.18
CTC primes (%)	65.22	72.58	48.20	55.28
CTC intrinsic (%)	**69.78**	**75.93**	**49.15**	**58.95**



(a) *OOV* (b) *WER*



(c) *CTC*

Fig. 7. OOV, WER and CTC results for the different configurations tested.

Table 16
Entrainment events and non-understandings relative frequencies. One-way ANOVA tests revealed no statistical significance.

	Data driven	Rule based	Random primes	Fixed primes
Uptakes (%)	9.54	9.97	9.08	8.60
No uptakes (%)	2.26	1.64	3.94	1.76
Non-understandings (%)	0.57	1.03	0.72	1.14

Table 17 shows the high-level dialog results for these tests: distribution of the dialogs per version, the estimated dialog success, the real dialog success, the average number of turns per dialog and the average number of turns with primes per dialog.

These high level measures show that Fixed Primes was the version with best success rate and lowest average number of turns per dialog. One-way ANOVA tests performed with Real Success Rate, Number of Turns and Number of turns with primes revealed no statistically significant differences between versions in any of these measures. However, since the session success is computed based on the information returned to the user, it might not be the best measure to

Table 17
Dialog performance results.

	Data driven	Rule based	Random primes	Fixed primes
Number of dialogs	57	53	52	52
Estimated success rate (%)	96.5	98.1	94.2	98.1
Real success rate (%)	82.4	82.6	84.9	86.9
Number of turns (avg. per dialog)	14.75	13.21	13.73	13.15
Number of turns w/primes (avg. per dialog)	2.82	2.34	2.67	2.37

Table 18
Questionnaire results for the different versions.

	Data driven	Rule based	Random primes	Fixed primes
Number of questionnaires	50	46	47	41
Average satisfaction score	22.1	20.9	22.7	22.8
Average entrainment question score	9.1	8.3	9.7	9.7

evaluate how effective was the prime choice. In fact, a session could be successful even without using any prime. For instance, this could happen if a user makes a single request asking for a bus for a specific hour instead of asking for the next bus and ends the dialog as soon as she/he receives the correct information. This dialog is considered successful while no prime was used in it.

Finally, Table 18 shows the results of the questionnaire that many of the testers answered (184 questionnaires, for 214 tests). The distribution of questionnaires per version, the average overall satisfaction and the average score in the entrainment related questions are detailed. The maximum value admitted for overall satisfaction was 33 and for entrainment satisfaction was 12.

ANOVA tests revealed no statistical significance in the questionnaires results. However results confirm the informal comments from some users that they hardly noticed any difference between the different versions, which is supported by the findings that state that entrainment is an unconscious phenomenon. It should be a difficult task to find some sort of adaptation when a dialogs are less than 14 turns long, and only 2 or 3 turns involve primes. Even regarding the entrainment question set, the versions that had no algorithm implemented achieved higher scores, even if in the case of Fixed Primes the primes were not changing at all. Since there was a lot of variation in the Random Primes version, the users might have thought that the system was adapting someway, although it was not. This could explain the highest entrainment question score was achieved in this version.

6.3.3. Prime usage evolution: rule based versus data driven

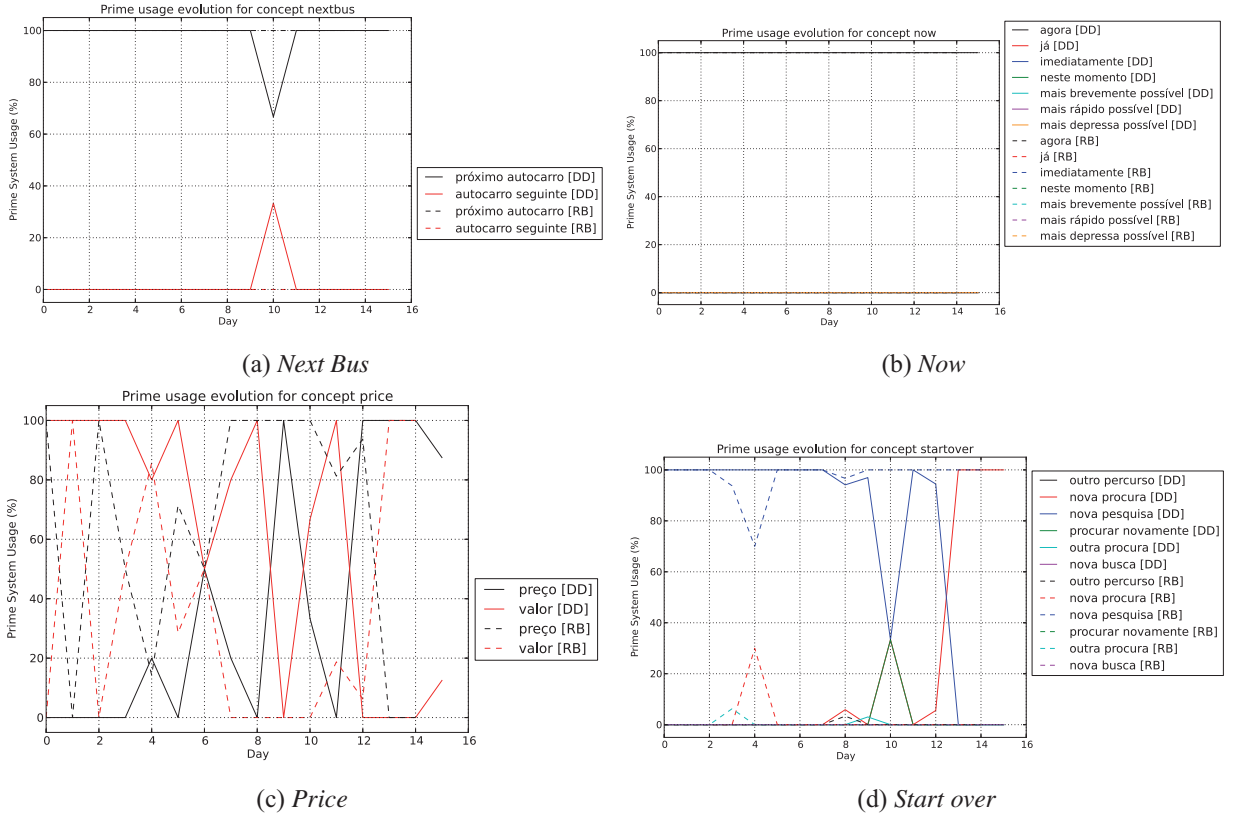
In order to have an overview of the prime selection, this sections aims at further analyzing and comparing the two entrainment policies that were tested. Figs. 8 and 9 show the prime usage evolution during the testing days for the Data Driven (DD) and Rule Based (RB) methods, similarly to what was done in Section 5.2.3 for the prime evolution in Let's Go primes. Fig. 8 shows the evolution for intrinsic primes, whereas Fig. 9 shows it for non-intrinsic primes.

The choice of primes for *Next Bus* and *Now* almost never changed during the test period with both methods. *Próximo autocarro* and *agora* are by far the primes most chosen by both methods (Figs. 8a and b). This could mean that a large majority of the users prefer to use those primes instead of the other options available. The *Price* prime shows much variation in both methods, although the Data Driven method only used *valor* for the first few days (Fig. 8c). This could be a prime where each user has her/his own preference, and the system is continuously changing to adapt to each user preference. This is confirmed by the high number of *No Uptake* events for this concept.

The *Start Over* prime selection frequency presents some differences between the two methods (Fig. 8d). The Rule Based method used *Nova Pesquisa* during the majority of the test period. On the other hand, the Data Driven method modified the prime used towards the end of the test period. Both methods used only a small subset of the prime candidates available.

Concerning the non-intrinsic primes the choice for *Arrival Place* and *Origin Place* primes is similar (Figs. 9a and b). There is a clear preference for one of the primes, *chegada* for *Arrival Place* and *partida* for *Origin Place*. For the *Time* prime (Fig. 9c), both methods kept changing the prime used during the test period. This was due the fact that at the point of the dialog where the system uses this prime, a statistical language model is used, whereas in the other points where the system tries to entrain, an SRGS grammar is used. Unlike statistical language models, SRGS grammars do not allow any type of Non-Understandings, since the grammar specification only includes in-grammar utterances.

According to these results (especially Figs. 8c and 9c), the Rule Based method prime choice had more variability than the Data Driven method. This is probably due to the Short-Term entrainment phase within dialogs where there is a stronger adaptation to the user.



(a) *Next Bus* (b) *Now*

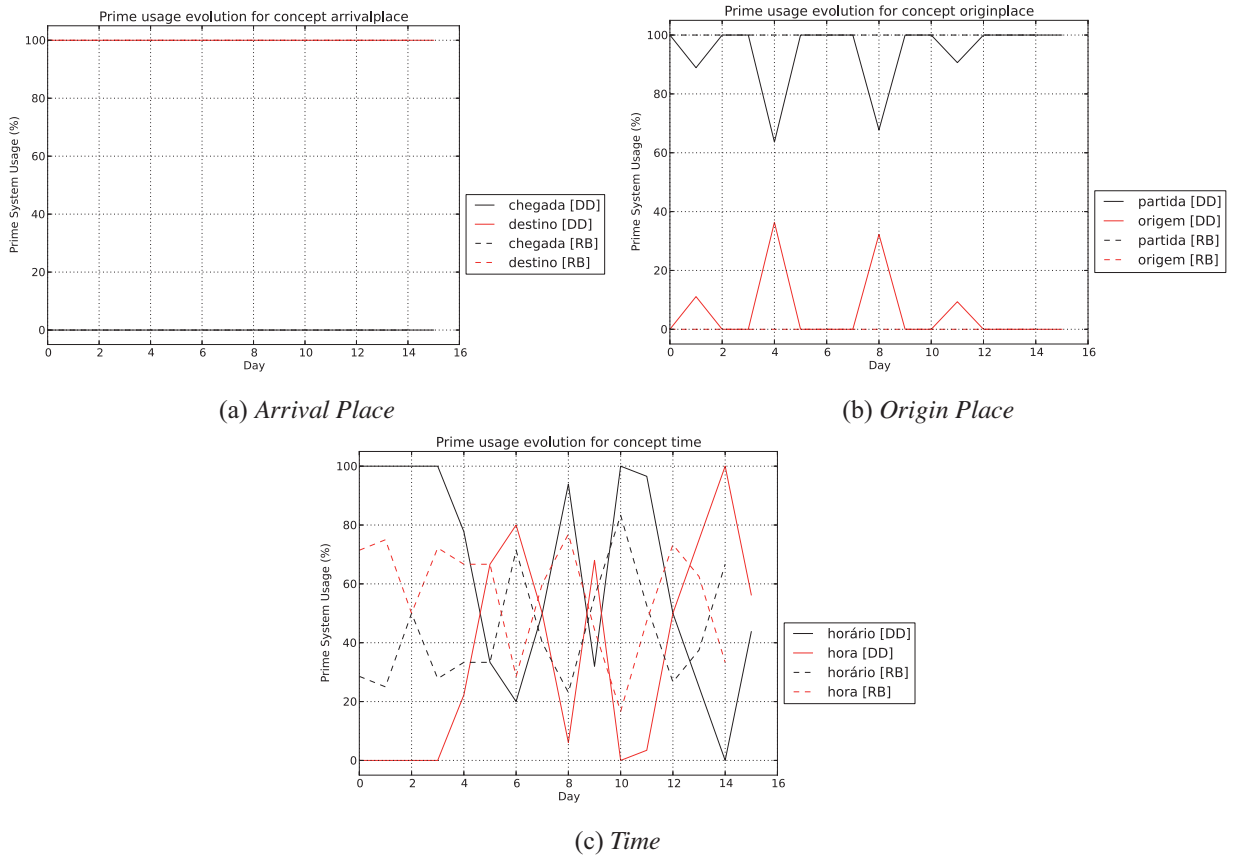
(c) *Price* (d) *Start over*

Fig. 8. Comparison between intrinsic prime usage in prompts between data driven (DD) and rule based (RB) prime selection.

7. Discussion

The tests presented in this paper have shown that applying lexical entrainment in both directions can have an impact on SDS performance. Let's Go success rate was increased, and, most importantly, the number of turns per session was reduced. In Noctívago, the prime selection methods have clearly outperformed Fixed and Random prime selection in terms of prime concept acquisition (which can be evaluated by the WER and CTC results for primes) and reducing the OOV.

Unlike in the Let's Go tests, none of the tests performed with Noctívago with different entrainment policies resulted in the expected impact in terms of dialog success rate. The first reason for this was already mentioned and it is the fact that a dialog can be successful without the use of any primes. But other reasons can explain why the dialog success rate did not improve in the studies ran with Noctívago. One thing that might have made the algorithms for prime selection more effective in Let's Go than in Noctívago was the fact that Let's Go uses context-dependent statistical language models in every dialog stage. This increases the number of outputs that the ASR can produce and gives more freedom to the user. On the other hand, this also generates more *Non-Understandings*, which means more opportunities to adapt the prime choice. In Noctívago, in order to have a more robust system, we have opted for SRGS grammars in some points of the dialog. The price paid for boosting the ASR performance was the reduction of the *Non-understandings*, since the result of the speech recognition was always parsable. The results for *Non-Understandings* in Table 16 confirm this observation. However, in future experiments we should give more freedom to the user. This will give the chance to the system to adjust the prime selection as the *Non-Understandings* take place. This strategy, may lead to longer dialogs in the beginning, but as soon as the prime selection is refined by the interactions, the system performance will certainly improve.



(a) *Arrival Place* (b) *Origin Place*

(c) *Time*

Fig. 9. Comparison between non-intrinsic prime usage in prompts between data driven (DD) and rule based (RB) prime selection.

8. Conclusions and future work

This paper described new methods for the integration of lexical entrainment in SDS. The first part of the paper, reports two studies using a Rule Based method for prime selection. This method was implemented and tested in two different systems, Noctívago and Let's Go. The Noctívago tests revealing interesting trends that were later taken into account for the algorithm implemented in Let's Go. The tests with Let's Go showed an improvement in the system performance when using the rules for prime selection.

The second part of the paper presents a first data-driven approach to perform lexical entrainment in both directions in an SDS. The experimental tests held with Noctívago revealed a similar performance to the Rule Based method, and both clearly outperformed Random and Fixed prime choice in terms of understanding turns with primes.

The application of the data-driven algorithm to other systems working in different domains, the extension of lexical entrainment to more concepts in the dialog and experiments with real users would help us to confirm the results found in this paper.

In order to improve the algorithm for prime selection new features and statistical methods should be investigated. Among the new features that could enhance a future Data-Driven model, prosody would be an interesting one to study. A possible solution to improve the statistical model for prime selection would be the integration of lexical entrainment in dialog-state tracking framework. This way, the system could learn the best primes in an unsupervised way.

Acknowledgements

The authors would like to thank Sungjin Lee and Alan W. Black for their contributions to this work. This research work was funded by Fundação para a Ciência e Tecnologia (FCT, Portugal) through the Ph.D. grant with reference SFRH/BD/47039/2008 and project PEst-OE/EEI/LA0021/2013.

References

Anguita, J., Peillon, S., Hernando, J., Bramoulle, A.,2004. Word confusability prediction in automatic speech recognition. In: INTERSPEECH 2004 – ICSLP, 8th International Conference on Spoken Language Processing. ISCA.

Bacelar do Nascimento, M.F., Marques, M.L.G., da Cruz, M.L.S., 1987. Português Fundamental, Métodos e Documentos, tomo 1, Inquérito de Frequência. INIC, CLUL, Lisboa.

Black, A.W., Lenzo, K.A., 2000. Limited domain synthesis. In: ICSLP, pp. 411–414.

Black, A.W., Lenzo, K.A., 2001. Flite: A small fast run-time synthesis engine. In: 4th isca tutorial and research workshop on speech synthesis, Perthshire, Scotland, pp. 20–24.

Black, A.W., Burger, S., Langner, B., Parent, G., Eskenazi, M., 2010. Spoken dialog challenge 2010. In: SLT, pp. 448–453.

Bohus, D., Rudnicky, A.,2002. Integrating multiple knowledge sources for utterance-level confidence annotation in the cmu communicator spoken dialog system, Tech. rep., Roots in the Town. In: 2nd International Workshop on Community Networking. 1995. IEEE Comm. Soc, Princeton (NJ).

Bohus, D., Rudnicky, A.I., 2007. Implicitly-supervised learning in spoken language interfaces: an application to the confidence annotation problem. In: SIGDIAL.

Bohus, D., Rudnicky, A.I., 2009. The ravenclaw dialog management framework: Architecture and systems. Comput. Speech Lang. 23 (3), 332–361.

Bohus, D., Raux, A., Harris, T.K., Eskenazi, M., Rudnicky, A.I.,2007. 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, NAACL-HLT-Dialog '07. Association for Computational Linguistics, Stroudsburg, PA, USA, pp. 32–39.

Bohus, D., 2007. Error Awareness and Recovery in Conversational Spoken Language Interfaces. Carnegie Mellon University, Pittsburgh, PA, USA, aAI3277260 (Ph.D. thesis).

Branigan, H.P., Pickering, M.J., Pearson, J., McLean, J.F., Nass, C., 2003. Syntactic alignment between computers and people: The role of belief about mental states. In: Proceedings of the 25th Annual Conference of the Cognitive Science Society (CogSci).

Branigan, H., Pickering, J., Pearson, J., McLean, J., 2010. Linguistic alignment between people and computers. J. Pragmat. 42 (9), 2355–2368.

Brennan, S.E., Clark, H.H., 1996. Conceptual pacts and lexical choice in conversation. J. Exp. Psychol.: Learn. Mem. Cognit. 22, 1482–1493.

Brennan, S., Ries, P., Rubman, C., Lee, G., 1996. The vocabulary problem in spoken language systems. In: Luperfoy, S. (Ed.), Automated Spoken Dialog Systems. MIT Press, p. 1998.

Brennan, S.E., 1991. Conversation with and through computers. User Model. User-Adapted Interact. 1 (1), 67–86.

Fandrianto, A., Eskenazi, M., 2012. Prosodic entrainment in an information-driven dialog system. In: Proceedings of Interspeech 2012, Portland, Oregon, USA.

Garrod, S., Anderson, A., 1987. Saying what you mean in dialogue: a study in conceptual and semantic coordination. Cognition 27, 181–218.

Hirschberg, J., Litman, D.J., Swerts, M., 2004. Prosodic and other cues to speech recognition failures. Speech Commun. 43 (1–2), 155–175.

Hirschberg, J., 2011. Speaking more like you: entrainment in conversational speech. In: Proc. INTERSPEECH.

Huggins-daines, D., Kumar, M., Chan, A., Black, A.W., Ravishankar, M., Rudnicky, A.I., Pocketsphinx, 2006. A free, real-time continuous speech recognition system for hand-held devices. In: Proceedings of ICASSP 2006.

Lee, S., Eskenazi, M., 2012. Pomdp-based let's go system for spoken dialog challenge. In: Proc. IEEE SLT Workshop.

Levow, G.-A., 2003. Learning to speak to a spoken language system: Vocabulary convergence in novice users. In: Proc. SIGDIAl.

Litman, D.J., Walker, M.A., Kearns, M.S.,1999. Automatic detection of poor speech recognition at the dialogue level. In: Proceedings of the 37th annual meeting of the Association for Computational Linguistics on Computational Linguistics, ACL '99. Association for Computational Linguistics, Stroudsburg, PA, USA, pp. 309–316.

Lopes, J., Eskenazi, M., Trancoso, I.,2011. Towards choosing better primes for spoken dialog systems. In: Proc. ASRU. Waikiloa, Hawaii, USA, pp. 306–311.

Lopes, J., Eskenazi, M., Trancoso, I., 2013. Automated two-way entrainment to improve spoken dialog system performance. In: Proceedings of ICASSP.

Matessa, M., 2003. Measures of Adaptive Communication.

Nenkova, A., Gravano, A., Hirschberg, J.,2008. High frequency word entrainment in spoken dialogue. In: Proceedings of ACL-08: HLT. Association for Computational Linguistics.

Neto, J., Meinedo, H., Viveiros, M., Cassaca, R., Martins, C., Caseiro, D., 2008. Broadcast news subtitling system in portuguese. In: ICASSP'08, pp. 1561–1564.

Parent, G., Eskenazi, M., 2010. Lexical entrainment of real users in the let's go spoken dialog system. In: INTERSPEECH, pp. 3018–3021.

Paulo, S., Oliveira, L.C., Mendes, C., Figueira, L., Cassaca, R., Viana, C., Moniz, H., 2008. Dixi – a generic text-to-speech system for european portuguese. In: Proceedings of the 8th international conference on Computational Processing of the Portuguese Language, PROPOR '08, pp. 91–100.

Pickering, M.J., Garrod, S., 2004. Towards a mechanistic psychology of dialogue. Behav. Brain Sci. 27 (2), 169–190.

Raux, A., Langner, B., Bohus, D., Black, A.W., Eskenazi, M., 2005. Let's go public! taking a spoken dialog system to the real world. In: Proceedings of Interspeech 2005.

Reitter, D., Keller, F., Moore, J.D., 2006. Computational modelling of structural priming in dialogue. In: Proceedings of HLT-NAACL, pp. 121–124.

Roe, D.B., Riley, M.D., 1994. Prediction of word confusabilities for speech recognition. In: Proceedings of ICSLP.

Rudnicky, A.I., Thayer, E.H., Constantinides, P.C., Tchou, C., Shern, R., Lenzo, K.A., Xu, W., Oh, A., 1999. Creating natural dialogs in the carnegie mellon communicator system. In: EUROSPEECH, ISCA.

Schmitt, A., Schatz, B., Minker, W.,2011. Modeling and predicting quality in spoken human-computer interaction. In: Proceedings of the SIGDIAL 2011 Conference, SIGDIAL '11. Association for Computational Linguistics, Stroudsburg, PA, USA, pp. 173–184.

Stoyanchev, S., Stent, A., 2009. Lexical and syntactic priming and their impact in deployed spoken dialog systems. In: Proceedings of HLT-NAACL 2009, NAACL-Short'09, Stroudsburg, PA, USA, pp. 189–192.

Tan, B.T., Gu, Y., Thomas, T., 1999. Word confusability measures for vocabulary selection in speech recognition. In: Proceedings of ASRU.

Toda, T., Black, A., Tokuda, K., 2007. Voice conversion based on maximum-likelihood estimation of spectral parameter trajectory. IEEE Trans. Audio, Speech, Lang. Process. 15 (8), 2222–2235.

Walker, M.A., Litman, D.J., Kamm, C.A., Kamm, A.A., Abella, A., 1998. Evaluating Spoken Dialogue Agents with Paradise: Two Case Studies.

Ward, W., Issar, S., 1994. Recent improvements in the cmu spoken language understanding system. In: Proceedings of the workshop on Human Language Technology, HLT '94, pp. 213–216.

Ward, A., Litman, D.,2007. Automatically measuring lexical and acoustic/prosodic convergence in tutorial. In: Proceedings of SLaTE 2007. Framington, Pennsylvania, USA.

Ward, A., Litman, D., 2007. Measuring Convergence and Priming in Tutorial Dialog, Tech. Rep. University of Pittsburgh.

Williams, J.D., Young, S., 2007. Partially observable markov decision processes for spoken dialog systems. Comput. Speech Lang. 21 (2), 393–422.