“The day after the day after tomorrow?” A machine learning approach to adaptive temporal expression generation: training and evaluation with real users

Srinivasan Janarthanam, Helen Hastie, Oliver Lemon, Xingkun Liu
Interaction Lab
School of Mathematical and Computer Sciences (MACS)
Heriot-Watt University
{sc445, h.hastie, o.lemon, x.liu}@hw.ac.uk

Abstract

Generating Temporal Expressions (TE) that are easy to understand, unambiguous, and reasonably short is a challenge for humans and Spoken Dialogue Systems. Rather than developing hand-written decision rules, we adopt a data-driven approach by collecting user feedback on a variety of possible TEs in terms of task success, ambiguity, and user preference. The data collected in this work is freely available to the research community. These data were then used to train a simulated user and a reinforcement learning policy that learns an adaptive Temporal Expression generation strategy for a variety of contexts. We evaluate our learned policy both in simulation and with real users and show that this data-driven adaptive policy is a significant improvement over a rule-based adaptive policy, leading to a 24% increase in perceived task completion, while showing a small increase in actual task completion, and a 16% decrease in call duration. This means that dialogues are more efficient and that users are also more confident about the appointment that they have agreed with the system.

1 Introduction

Temporal Expressions are linguistic expressions that are used to refer to a date and are often a source of confusion in human-human, human-computer and text interactions such as emails and instant messaging. For example, “Let’s meet next Sunday”– “do you mean Sunday this week or a week on Sunday?”. (Mccoy and Strube, 1999) state that changes in temporal structure in text are often indicated by either cue words and phrases (e.g. “next Thursday”, “this week”, “tomorrow”), a change in grammatical time of the verb (e.g. present tense versus future tense), or changes in aspect (e.g. atomic versus extended events versus states as defined by (Moens and Steedman, 1988)). In this study, we will concentrate on the first of these phenomena, generating TEs with the optimal content and lexical choice.

Much work in the field of Natural Language Processing concerns understanding and resolving these temporal expressions in text (Gerber et al., 2002; Pustejovsky et al., 2003; Ahn et al., 2007; Mazur and Dale, 2007; Han et al., 2006), however, little work has looked at how best to plan and realise temporal expressions in order to minimize ambiguity and confusion in a Spoken Dialogue System (SDS). (Reiter et al., 2005) presented a data driven approach to generating TEs to refer to time in weather forecast information where appropriate expressions were identified using contextual features using supervised learning. We adopt an adaptive, data-driven reinforcement learning approach instead. Similar data-driven approaches have been applied to information presentation (Rieser et al., 2010; Walker et al., 2007) where each Natural Language Generation (NLG) action is a sequential decision point, based on the current dialogue context and expected long-term reward of that action. A data-driven approach has also been applied to the problem of referring expression generation in dialogue for expert and novice-users of a SDS (Janarthanam and Lemon, 2010). However, to date, there has been no previous work on adaptive data-driven approaches for *temporal* referring expression generation, where uncertainty in

Proceedings of the SIGDIAL 2011: the 12th Annual Meeting of the Special Interest Group on Discourse and Dialogue, pages 142–151,
Portland, Oregon, June 17-18, 2011. ©2011 Association for Computational Linguistics

the stochastic environment is explicitly modelled.

The data-driven approach to temporal expression generation presented here is in the context of appointment scheduling dialogues. The fact that there are multiple ways that a time slot can be referred to leads to an interesting NLG problem of how best to realise a TE for a particular individual in a particular context for certain domains. For example, the following expressions all vary in terms of length, ambiguity, redundant information and users' preference: "next Friday afternoon" or "Friday next week at the same time", or "in the afternoon, a week on Friday".

Temporal Expressions contain two types of references: absolute references such as "Tuesday" and "12th January", and relative references such as "tomorrow" and "this Tuesday". Generating TEs therefore, involves both in selecting appropriate pieces of information (date, day, time, month, and week) to present and deciding how to present them (absolute or relative reference).

Our objective here is to convey a target appointment slot to users using an expression that is optimal in terms of the trade-off between understandability, length and user preference.

2 Methodology

We address the issue of generating TEs by adopting a data-driven approach that has four stages. Firstly, we define Temporal Expression Units (TEU) as described in Section 2.1. Secondly, we design and implement a web-based data collection, gathering metrics on the TEUs in various contexts for a variety of date types (Section 3). Thirdly, we train a user simulation and use it to learn a policy using reinforcement learning techniques that generates the optimal combination of TEUs for each context (Section 4). Finally, we deploy and evaluate this policy in a Spoken Dialogue System for appointment scheduling and show that our learned policy performs better than a hand-written, adaptive one (results presented in Section 5).

2.1 Temporal Expression Units

For this study, TEs are broken down into 5 categories or units (TEUs) presented in a fixed order: DAY, DATE, MONTH, WEEK and TIME. Each of these units can be expressed relative to the current day and to the current context (i.e. previously mentioned dates). Specifically, there are 3 unit attributes: absolute (e.g. DAY=abs "Tuesday"); relative to current day (e.g. DAY=rel "tomorrow"); and relative to context (e.g. DAY=rc "the following day").

TEU	**Choices**
DAY	abs, rel, rc, nn
DATE	abs, nn
MONTH	abs, nn
WEEK	abs, rel, nn
TIME	abs, rc

Table 1: TEU choices where abs is absolute, rel is relative, rc is relative to context and nn is none

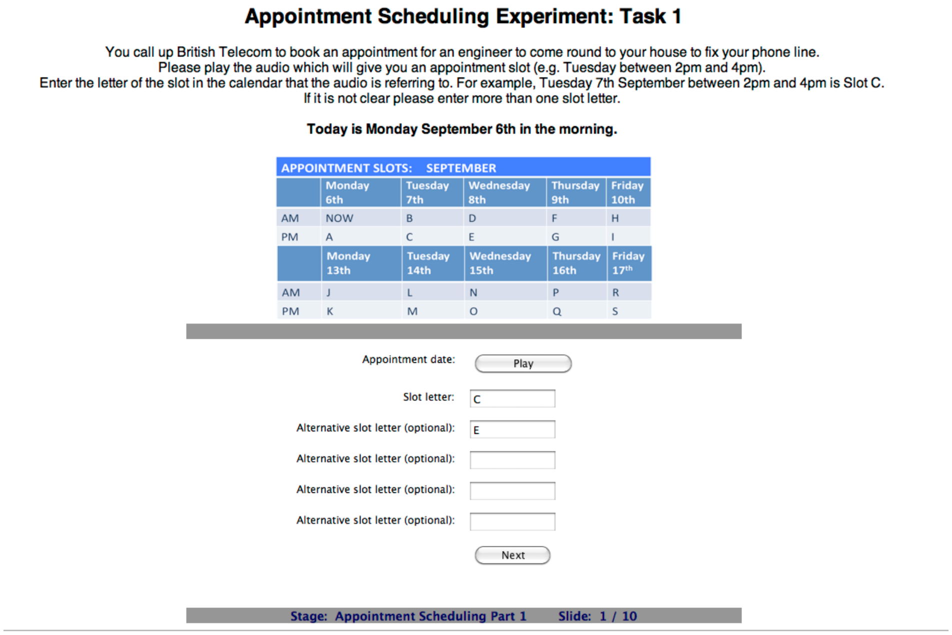
Certain restrictions on possible TEU combinations were imposed, for example, DATE=rc and DAY=rel were combined to be just DAY=rel, and some combinations were omitted on the basis that it is highly unlikely that they would be uttered in natural speech, for example WEEK=rel and MONTH=abs would result in "this week in September". Finally, every TE has to contain a time (am or pm for this application). The possible combinations are summarised in Table 1.

3 Data Collection

The data collection experiment was in two parts (Task 1 and Task 2) and was designed using the Webexp experimental software[1]. Webexp is a client-server set up where a server application hosts the experiment and stores the experimental files, logs and results. The client side runs an applet on the user's web-browser.

In Task 1, participants listened to an audio file containing a TE generated from absolute and relative TEUs (see Figure 1). No relative-context (rc) TEUs were used in Task 1 since the dialogue excerpt presented was in isolation and therefore had no context. Each participant was asked to listen to 10 different audio files in a sequence corresponding to a variety of dates randomly chosen from 8 possible dates. The participant then had to identify the correct appointment slot that the system is referring to. There is scope for the participant to add multiple answers in order to capture potential ambiguity

[1] http://www.webexp.info



Appointment Scheduling Experiment: Task 1

You call up British Telecom to book an appointment for an engineer to come round to your house to fix your phone line.
Please play the audio which will give you an appointment slot (e.g. Tuesday between 2pm and 4pm).
Enter the letter of the slot in the calendar that the audio is referring to. For example, Tuesday 7th September between 2pm and 4pm is Slot C.
If it is not clear please enter more than one slot letter.

Today is Monday September 6th in the morning.

APPOINTMENT SLOTS: SEPTEMBER					
	Monday 6th	Tuesday 7th	Wednesday 8th	Thursday 9th	Friday 10th
AM	NOW	B	D	F	H
PM	A	C	E	G	I
	Monday 13th	Tuesday 14th	Wednesday 15th	Thursday 16th	Friday 17th
AM	J	L	N	P	R
PM	K	M	O	Q	S

Appointment date:

Slot letter: C

Alternative slot letter (optional): E

Alternative slot letter (optional):

Alternative slot letter (optional):

Alternative slot letter (optional):

Stage: Appointment Scheduling Part 1 Slide: 1 / 10

Figure 1: Screen shot of Task 1 in the on-line data collection experiment

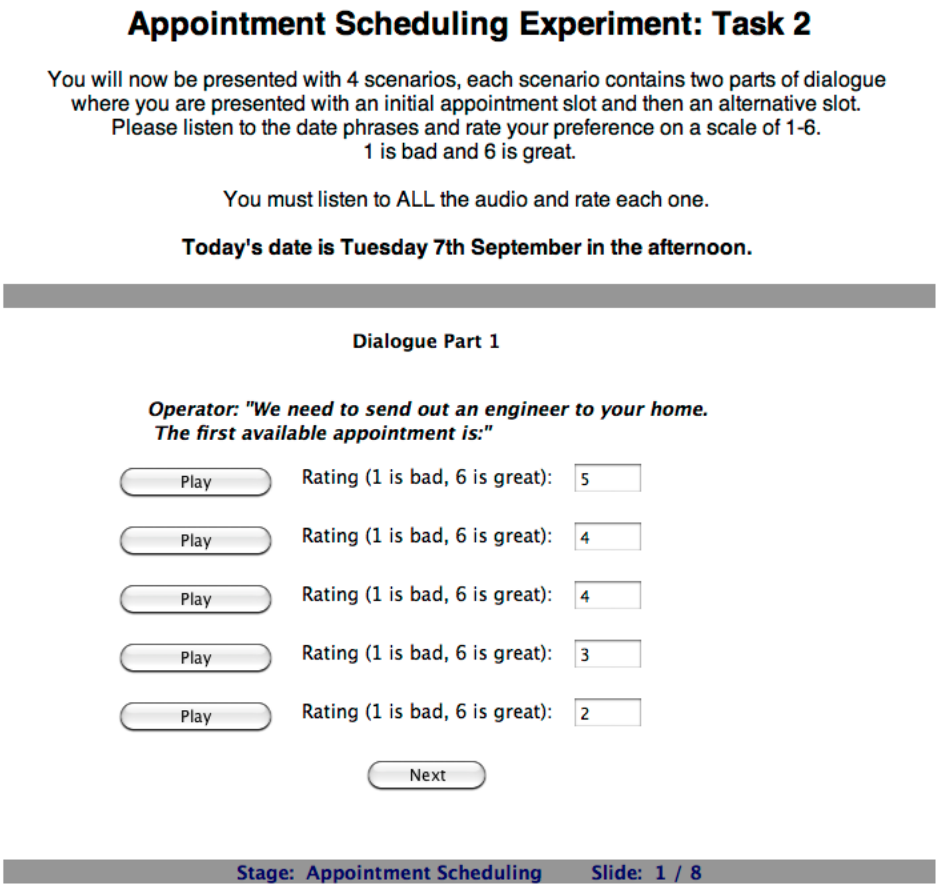
of a TE, and we report on this below. The 8 dates that were used to generate the TEs fell into a two week period in a single month which is in-line with the evaluation set-up of the appointment scheduling SDS discussed in Section 5.3.

For each date, the TE was randomly picked from a set of 30 possible combinations of TEUs. Each TEU was generated by a rule-based realiser and synthesized using the Baratinoo synthesizer (France Telecom, 2011). This realiser generates text from a candidate list for each TEU based on the given date. For example, if the slot currently being discussed is Tuesday 7th, the realiser would generate "tomorrow" for DAY=rel; if the date in discussion was Wednesday 8th then DAY=rel would be realised as "the day after tomorrow". There was potential for overlap of stimuli, as any given TE for any given date may be assessed by more than one participant.

Task 2 of the experiment was in two stages. In the first stage (Task 2A), the participants are given today's date and the following dialogue excerpt; Operator: "We need to send out an engineer to your home. The first available appointment is . . ." (see Figure 2). They are then asked to listen to 5 audio files of the system saying different TEs for the same date and asked to rate preference on a scale of 1-6 (where 1 is bad and 6 is great.) For the second stage (Task 2B), the dialogue is as follows; Operator: "so you can't do Wednesday 8th September in the morning." and then the participants are asked to listen to 5 more audio files that are generated TEs including relative context such as "how about Thursday at the same time?". This two-stage process is then repeated 4 times for each participant.

Table 2 summarizes the metrics collected in the different parts of the experiment. The metric *Distance* is calculated in terms of the number of slots from the current date to the target date (TD). Instances were grouped into four distance groups: G1: TD is 1-2 slots away; G2: TD is 3-6 slots away; G3: TD is 7-11 slots away and G4: TD more than 11 slots away. P_replay is calcuated by the total number of replays divided by the total number of plays for that temporal expression, i.e. the probability that the temporal expression played is requested to be replayed. P_ambiguous is calculated by the number of times a given temporal expression is given more than 1 interpretation divided by the total number of times that the same given referring expression is answered.

In total there were 73 participants for Task 1 and



Appointment Scheduling Experiment: Task 2

You will now be presented with 4 scenarios, each scenario contains two parts of dialogue
where you are presented with an initial appointment slot and then an alternative slot.
Please listen to the date phrases and rate your preference on a scale of 1-6.
1 is bad and 6 is great.

You must listen to ALL the audio and rate each one.

Today's date is Tuesday 7th September in the afternoon.

Dialogue Part 1

Operator: "We need to send out an engineer to your home.
The first available appointment is:"

Rating (1 is bad, 6 is great): 5

Rating (1 is bad, 6 is great): 4

Rating (1 is bad, 6 is great): 4

Rating (1 is bad, 6 is great): 3

Rating (1 is bad, 6 is great): 2

Stage: Appointment Scheduling Slide: 1 / 8

Figure 2: Screen shot of Task 2 in the on-line data collection experiment

730 TE samples collected. Although Task 2 directly followed on from Task 1, there was a significant drop out rate as only 48 participants completed the second task resulting in 1,920 TE samples. Participants who completed both tasks were rewarded by a chance to win an Amazon voucher.

3.1 Data Analysis

Figure 3 shows various metrics with respect to TE absoluteness and relativeness is the number of absolute and relative TEUs respectively. These two graphs represent the state space that the generation policy described in Section 4 is exploring, trading off between various features such as *Length, taskSuccess* and *userPref.*

As we can see, there is a tendency for average *taskSuccess* to increase as absoluteness increases whereas, for relativeness the distribution is more even. The TE with the greatest *taskSuccess* has an absoluteness of 4 and zero relativeness: DATE=abs, MONTH=abs, WEEK=abs, TIME=abs (e.g. “11th September, the week starting the 10th, between 8am and 10am”) and the TE with the least *taskSuccess* has an absoluteness of only 2, again with no relativeness: DATE=abs, TIME=abs, (e.g. “8th between 8am and 10am”).

Average *userPref* stays level and then decreases if absoluteness is 5. We infer from this that although long utterances that are completely explicit are more clear in terms of *taskSuccess*, they are not necessarily preferred by users. This is likely due to TE length increasing. On average, the inclusion of one relative expression is preferred over none at all or two. The most preferred TE has an absoluteness of 3 with a relativeness of 2: DAY=rel, DATE=abs, MONTH=abs, WEEK=rel, TIME=abs (e.g. “Tomorrow the 7th of September, this week, between 8am and 10am”).

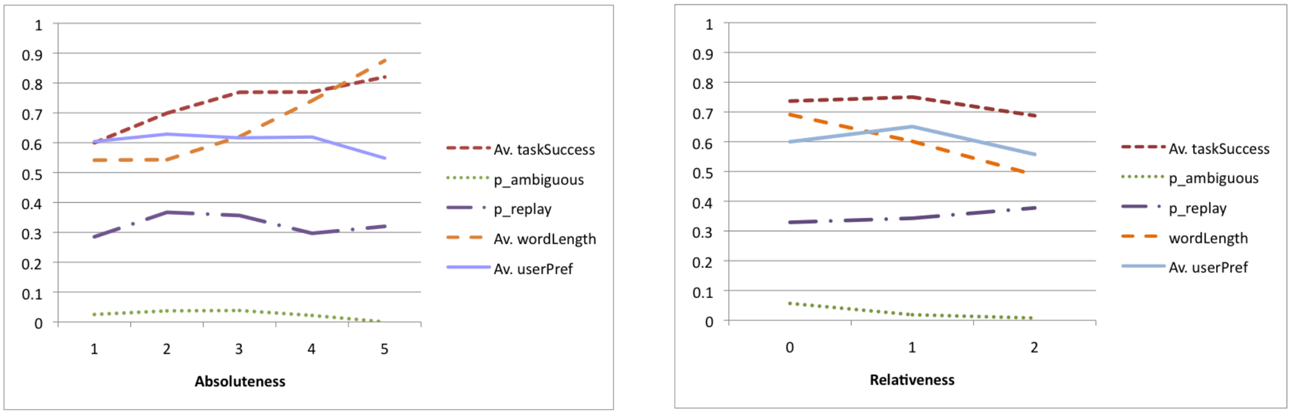


Figure 3: Graph showing the trade-offs between various metrics with respect to absoluteness and relativeness (number of absolute/relative TEUs) in terms of probabilities or normalised values.

Metric	Description	Task
P_ambiguous	Probability that the expression is ambiguous to the user	1
taskSuccess	Correct slot identified	1
P_replay	Probability of replay (measure of understandability)	1 & 2
Length	Expression length in terms of number of TEUs that are non null divided by the total number of possible TEUs (5)	1 & 2
wordLength	Expression length in words normalised over max num of words (15)	1 & 2
userPref	Preference rating of audio from 1-6	2
Distance	Distance from target date (TD) to current date in terms of number of slots	1 & 2

Table 2: Metrics collected in various parts of the experiment

The probability of ambiguity and replay does not seem to be affected by absoluteness. The most ambiguous TE has an absoluteness of 3 and zero relativeness: DAY=abs MONTH=abs TIME=abs, (e.g. “Tuesday September between 8am and 10am”) indicating that a date is needed for precision. The TEs that the participants were most likely to replay tended to be short e.g. “Tomorrow at the same time”. This may be due to the clarity of the speech synthesiser.

4 Learning a TE generation policy

Reinforcement learning is a machine learning approach based on trial and error learning, in which a learning agent learns to map sequences of “optimal” actions to environment or task states (Sutton and Barto, 1998). In this framework the problem of generating temporal expressions is presented as a Markov Decision Process. The goal of the learning agent is to learn to choose those actions that obtain maximum expected reward in the long run. In this section, we present the reinforcement learning setup for learning temporal expression generation policies.

4.1 Actions and States

In this learning setup, we focus only on generating the formal specification and treat the set of TEU choices as the sequential actions of the learning agent. Table 1 presents the choices that are available for each TEU.

The actions are taken based on two factors: the distance (in terms of time slots: morning or afternoon appointments) between (1) the current date and the target slot and (2) the current date and the slot in context. Based on the distance, the target slot was classified to belong to one of the four distance groups (G1-G4). The slot in context represents whether there was any other slot already mentioned in the conversation so far, so that the system has an option to use "relative_context" expressions to present day and time information. Information concerning the target slot's group and the slot in context make up the state space of the Markov Decision Process (MDP).

4.2 User Simulation

We built a user simulation to simulate the dialogue behaviour of a user in appointment scheduling conversations based on the data from real users described in Section 3. It responds to the TE used by the system to refer to an appointment slot. It responds by either accepting, rejecting, or clarifying the offered slot based on the user's own calendar of available slots. For instance, the simulated user rejects an offered slot if the user is not available at that time. If they accept or reject an offered slot, the user is assumed to understand the TE unambiguously. However, if the user is unable to resolve the appointment slot from the TE, it responds with a clarification request. The simulation responded with a dialogue action ($A_{u,t}$) to TEs based on the system's dialogue act ($A_{s,t}$), system's TE ($TE_{s,t}$). The following probabilistic model was used to generate user dialogue actions:

$$P(A_{u,t}|A_{s,t}, TE_{s,t}, G, C, Cal)$$

In addition to $TE_{s,t}$ and $A_{s,t}$, other factors such as distance between the target slot and the current slot (G), the previous slot in context (C), and the user's calendar (Cal) were also taken into account. G is either G1, G2, G3 or G4 as explained in Section 3. The User's dialogue action ($A_{u,t}$) is one of the three: Accept_slot, Reject_slot or Request_Clarification. The probability of clarification request was calculated as the average of the ambiguity and replay probabilities seen in real user data.

4.3 Reward function

The learning agent was rewarded for each TE that it generated. The reward given to the agent was based on trade-offs between three variables: User preference (UP), Length of the temporal expression (L), and Clarification request probability (CR). UP for each TE is obtained from Task 2 of the data collection. In the following reward function, UP is normalised to be between 0 and 1. L is based on number of TEUs used. The maximum number of TEUs that can be used is 5 (i.e. DAY, DATE, WEEK, MONTH, TIME). L is calculated as follows:

$$\text{Length of TE (L)} = \frac{No.\ of\ used\ TEUs}{Max.\ no.\ of\ TEUs}$$

The clarification request (CR) is set to be 1 if the user responds to the TE with a Request_Clarification and 0 otherwise. Reward is therefore calculated on a turn-by-turn basis using the following formula:

$$Reward = UP * 10.0 - L * 10.0 - CR * 10.0$$

In short, we chose a reward function that penalises TEs that are long and ambiguous, and which rewards TEs that users prefer. It also indirectly rewards task success by penalising ambiguous TEs resulting in clarification requests. This trade-off structure is evident from the data collection where TEs that are too long are dispreferred by the users (see Figure 3). The maximum possible reward is 6 (i.e. UP=1, CR=0, L=2/5) and the minimum is -20 (i.e. UP=0, CR=1, L=1). Note that other reward functions could be explored in future work, for example maximising only for user preference or length.

4.4 Training

We trained a TE generation policy using the above user simulation model for 10,000 runs using the SARSA reinforcement learning algorithm (Sutton and Barto, 1998). During the training phase, the learning agent generated and presented TEs to the user simulation. When a dialogue begins, there is no appointment slot in context (i.e. C = 0). However, if the user rejects the first slot, the dialogue system sets C to 1 and presents the next slot. This is again reset at the beginning of the next dialogue. The agent was rewarded at the end of every turn based on the user's response, length of the TE, and user preference scores as shown above. It gradually explored all possible combinations of TEUs and identified those TEUs in different contexts that maximize

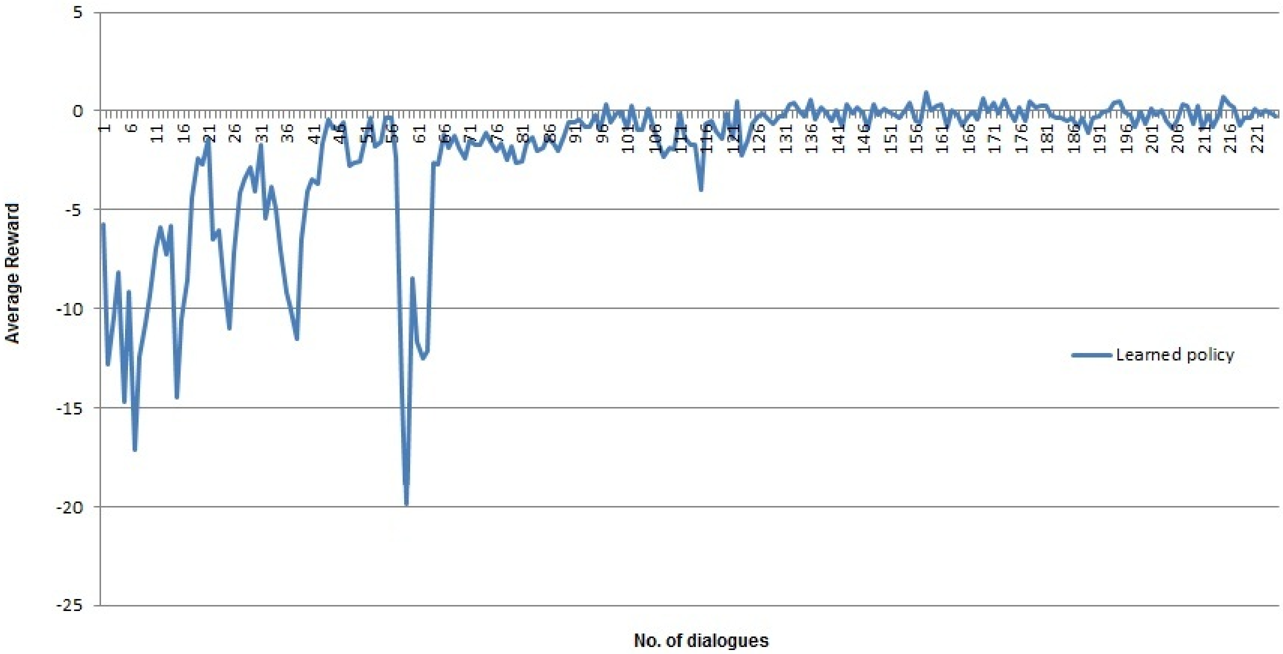


Figure 4: Learning curve

the long-term reward. Figure 4 shows the learning curve of the agent.

Table 3 presents the TE generation policy learned by the agent. As one can observe, it used a minimum number of TEUs to avoid length penalties in the reward. In all cases, MONTH and WEEK information have not been presented at all. For target slots that were closest (in group G1) and the farthest (in group G4), it used relative forms of day (e.g. “tomorrow”, “next Tuesday”, etc.). This is probably because users dispreferred day information for in-between slots (e.g. “the day after the day after tomorrow”). Also, MONTH information may have been considered to be irrelevant due to the fact that the two week window over which the data has been collected do not span over two different months.

5 Evaluation

In this section, we present the baseline policies that were evaluated along with the learned policy. We then present the results of evaluation.

Slots	Specification learned

1-2
> 11
	DAY=rel;DATE=abs;MONTH=nn;
	WEEK=nn;TIME=abs

3-11	DAY=nn;DATE=abs;MONTH=nn;
	WEEK=nn;TIME=abs

Table 3: Learned policy

5.1 Baseline policies

The following are the baseline TEG policies:

1. **Absolute policy**: always use absolute formats for all TEUs (i.e. DAY=abs; DATE=abs; MONTH=abs; WEEK=abs; TIME=abs)

2. **Minimal policy**: always use a minimal format with only date, month and time information in their absolute forms (i.e. DAY=nn; DATE=abs; MONTH=abs; WEEK=nn; TIME=abs)

3. **Random policy**: select possible formats randomly for each TEU.

TEG Policy	Average reward
Learned	-0.071* (±3.75)
Absolute	-4.084 (±4.36)
Minimal	-1.340 (±4.2)
Random	-8.21 (±7.72)

Table 4: Evaluation with simulated users (* $p < 0.05$, two-tailed independent samples t-test)

5.2 Results

We evaluated the learned policy and the three other hand-coded baseline TE generation policies with our user simulation model. Each policy generated 1,000 TEs in different states. Table 4 present the results of evaluation with simulated users. On average, the learned policy scores higher than all the baseline policies and the differences between the average reward of the learned policy and the other baselines are statistically significant. This shows that target slots can be presented using different TEs depending on how far they are from the current date and such adaptation can produce less ambiguous, shorter and user preferred expressions.

5.3 Evaluation with real users

The policy was also integrated into an NLG component of a deployed Appointment Scheduling spoken dialogue system. Please note that this is different from the web environment in which the training data was collected. Our data-driven policy was activated when the system informs the user of an available time slot. This system was compared to the exact same system but with a *rule-based* adaptive baseline system. In the rule-based policy MONTH, DATE and TIME were always absolute, DAY was relative if the target date was less than three days away (i.e. “today, tomorrow, day after tomorrow”), and WEEK was always relative (i.e. “this week, next week”). All 5 information units were included in the realisation (e.g. “Thursday the 15th July in the afternoon, next week”) although the order was slightly different (DAY-DATE-MONTH-TIME-WEEK).

In this domain, the user tries to make an appointment for an engineer to visit their home. Each user is given a set of 2-week calendars which shows their availability and the goal is to arrange an appointment when both they and the engineer are available. There were 12 possible scenarios that were evenly rotated across participants and systems. Each scenario is categorised in terms of scheduling difficulty (Hard/Medium/Easy). Scheduling difficulty is calculated for User Difficulty (UD) and System Difficulty (SD) separately to assess the system’s mixed initiative ability. Scheduling difficulty is calculated as the ordinal of the first session that is free for both the User and the System. Hard scenarios are with an ordinal of 3 or 4; Medium with an ordinal of 2, and Easy with an ordinal of 1. There are 4 scenarios in each of these difficulty categories for both the user and system. To give an example, in Scenario 10, the user can schedule an appointment on Wednesday afternoon but he/she also has one free session on the previous Tuesday afternoon when the engineer is busy therefore UD = 2. For the system, in this scenario, the first free session it has is on the Wednesday afternoon therefore SD=1. In this case, the scenario is easier for the system than the user because the system could just offer the first session that it has free.

605 dialogues were collected and analysed. The system was evaluated by employees at France Telecom and students of partner universities who have never used the appointment scheduling system before. After each scenario, participants were then asked to fill out a questionnaire on perceived task success and 5 user satisfaction questions on a 6-point Likert Scale (Walker et al., 2000). Results from the real user study are summarised in Table 5. The data-driven policy showed significant improvement in Perceived Task Success (+23.7%) although no significant difference was observed between the two systems in terms of Actual Task Success (Chi-square test, df=1). Perceived Task Success is users’ perception of whether they completed the task successfully or not. Overall user satisfaction (the average score of all the questions) was also significantly higher (+5%)[2]. Dialogues with the learned policy were significantly shorter with lower Call Duration in terms of time (-15.7%)[2] and fewer average words per system turn (-23.93%)[2]. Figure 5 shows the length results in time for systems of varying UD and SD. We can see that the data-driven adaptive policy consistently results in a shorter dialogue across all levels of difficulty. In summary, these results show that using a policy trained on the data collected here

Parameters	**Learned TEG**	**Baseline TEG**
Actual Task Success	80.05%	78.57%
Perceived Task Success	74.86%*	60.50%
User satisfaction	4.51*	4.30
No. system turns	22.8	23.2
Words per system turn	13.16*	17.3
Call duration	88.60 sec *	105.11 sec

Table 5: Results with real users (* statistically significant difference at p<0.05)

results in shorter dialogues and greater confidence in the user that they have had a successful dialogue. Although the learned policy was trained to generate optimal TEs within a two week window and therefore is not general policy for all TE generation problems, we believe that the data-driven approach that we have followed can generalise to other TE generation tasks.

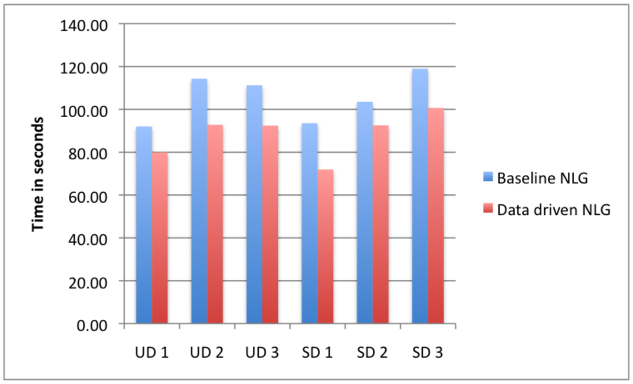


Figure 5: Graph comparing length of dialogues for user (UD) and system difficulty (SD)

6 Conclusion

We have presented a principled statistical learning method for generating Temporal Expressions (TEs) that refer to appointment slots in natural language utterances. We presented a method for gathering data on TEs with an on-line experiment and showed how we can use these data to generate TEs using a Markov Decision Process which can be optimised using reinforcement learning techniques. We showed that a TEG policy learned using our framework performs signifcantly better than hand-coded adaptive policies with real users as well as with simulated users.

The data collected in this work has been freely released to the research community in 2011[3].

Acknowledgements

The research leading to these results has received funding from the EC's 7th Framework Programme (FP7/2007-2013) under grant agreement no. 216594 (CLASSiC project `www.classic-project.org`), (FP7/2011-2014) under grant agreement no. 248765 (Help4Mood project), (FP7/2011-2014) under grant agreement no. 270435 (JAMES project), (FP7/2011-2014) under grant agreement no. 270019 (SpaceBook project), and from the EPSRC, project no. EP/G069840/1. We would also like to thank our CLASSiC project colleagues at Cambridge University and France Telecom / Orange Labs.

References

D. Ahn, J. van Rantwijk, and M. de Rijke. 2007. A Cascaded Machine Learning Approach to Interpreting Temporal Expressions. In *Proceedings of NAACL-HLT 2007*.

France Telecom. 2011. Baratinoo expressive speech synthesiser. *http://tts.elibel.tm.fr*.

L. Gerber, L. Ferro, I. Mani, B. Sundheim, G. Wilson, and R. Kozierok. 2002. Annotating Temporal Information: From Theory to Practice. In *Proceedings of HLT*.

B. Han, D. Gates, and L. Levin. 2006. Understanding temporal expressions in emails. In *HLT-NAACL 2006*.

Srinivasan Janarthanam and Oliver Lemon. 2010. Learning to adapt to unknown users: referring expression generation in spoken dialogue systems. In *ACL '10*.

P. Mazur and R. Dale. 2007. The DANTE Temporal Expression Tagger. In *Proceedings of the 3rd Language and Technology Conference, Poznan, Poland*.

Kathleen F. Mccoy and Michael Strube. 1999. Taking time to structure discourse: Pronoun generation beyond accessibility. In *Proc. of the 21th Annual Conference of the Cognitive Science Society*.

M. Moens and M. Steedman. 1988. Temporal ontology and temporal reference. In *Computational Linguistics*, volume 14(2), pages 15–28.

[2] independent two-tailed t-test p < 0.05

[3] Sec 2.6 at http://www.macs.hw.ac.uk/ilabarchive/classicproject/data/

J. Pustejovsky, J. Castano, R. Ingria, R. Sauri, R. Gaizauskas, A. Setzer, G. Katz, and D. Radev. 2003. TimeML: Robust specification of event and temporal expressions in text. In *AAAI Spring Symposium on New Directions in Question-Answering, Stanford, CA*.

E. Reiter, S. Sripada, J. Hunter, and J. Yu. 2005. Choosing words in computer-generated weather forecasts. *Artificial Intelligence*, 167:137169.

Verena Rieser, Oliver Lemon, and Xingkun Liu. 2010. Optimising information presentation for spoken dialogue systems. In *Proc. ACL 2010*.

R. Sutton and A. Barto. 1998. *Reinforcement Learning*. MIT Press.

Marilyn A. Walker, Candace A. Kamm, and Diane J. Litman. 2000. Towards Developing General Models of Usability with PARADISE. *Natural Language Engineering*, 6(3).

Marilyn Walker, Amanda Stent, François Mairesse, and Rashmi Prasad. 2007. Individual and domain adaptation in sentence planning for dialogue. *Journal of Artificial Intelligence Research (JAIR)*, 30:413–456.