Crowdsourcing Language Generation Templates for Dialogue Systems

Margaret Mitchell
Microsoft Research
Redmond, WA USA
memitc@microsoft.com

Dan Bohus
Microsoft Research
Redmond, WA USA
dbohus@microsoft.com

Ece Kamar
Microsoft Research
Redmond, WA USA
eckamar@microsoft.com

Abstract

We explore the use of crowdsourcing to generate natural language in spoken dialogue systems. We introduce a methodology to elicit novel templates from the crowd based on a dialogue seed corpus, and investigate the effect that the amount of surrounding dialogue context has on the generation task. Evaluation is performed both with a crowd and with a system developer to assess the naturalness and suitability of the elicited phrases. Results indicate that the crowd is able to provide reasonable and diverse templates within this methodology. More work is necessary before elicited templates can be automatically plugged into the system.

1 Introduction

A common approach for natural language generation in task-oriented spoken dialogue systems is template-based generation: a set of templates is manually constructed by system developers, and instantiated with slot values at runtime. When the set of templates is limited, frequent interactions with the system can quickly become repetitive, and the naturalness of the interaction is lost.

In this work, we propose and investigate a methodology for developing a corpus of natural language generation templates for a spoken dialogue system via crowdsourcing. We use an existing dialogue system that generates utterances from templates, and explore how well a crowd can generate reliable paraphrases given snippets from the system's original dialogues. By utilizing dialogue data collected from interactions with an existing system, we can begin to learn different ways to converse while controlling the crowd to stay within the scope of the original system. The proposed approach aims to leverage the system's existing capabilities together with the power of the crowd to expand the system's natural language repertoire and create richer interactions.

Our methodology begins with an existing corpus of dialogues, extracted from a spoken dialogue system that gives directions in a building. Further details on this system are given in §4.1. The extracted dialogue corpus contains phrases the system has generated, and crowd-workers construct alternates for these phrases, which can be plugged back into the system as *crowd templates*. We investigate via crowdsourcing the effect of the amount of surrounding context provided to workers on the perceived meaning, naturalness, and diversity of the alternates they produce, and study the acceptability of these alternates from a system developer viewpoint. Our results indicate that the crowd provides reasonable and diverse templates with this methodology. The developer evaluation suggests that additional work is necessary before we can automatically plug crowdsourced templates directly into the system.

We begin by discussing related work in §2. In §3, we detail the proposed methodology. In §4, we describe the experimental setup and results. Directions for future work are discussed in §5.

2 Related Work

Online crowdsourcing has gained popularity in recent years because it provides easy and cheap programmatic access to human intelligence. Researchers have proposed using crowdsourcing for a diverse set of natural language processing tasks, including paired data collection for training machine translation systems (Zaidan and Callison-Burch, 2011), evaluation of NLP systems (Callison-Burch and Dredze, 2010) and speech transcriptions (Parent and Eskenazi, 2010). A popular task targeting language diversity is paraphrase generation, which aims at collecting diverse phrases while preserving the original meaning. Crowdsourcing paraphrase generation has

Proceedings of the INLG and SIGDIAL 2014 Joint Session, pages 16–24,
Philadelphia, Pennsylvania, 19 June 2014. ©2014 Association for Computational Linguistics

been studied for the purposes of plagiarism detection (Burrows and Stein, 2013), machine translation (Buzek et al., 2010), and expanding language models used in mobile applications (Han and Ju, 2013). Automated and crowd-based methods have been proposed for evaluating paraphrases generated by the crowd (Denkowski and Lavie, 2010; Tschirsich and Hintz, 2013). Researchers have proposed workflows to increase the diversity of language collected with crowd-based paraphrase generation (Negri et al., 2012) and for reducing the language bias in generation by initiating generation with visual input (Chen and Dolan, 2011). While paraphrase generation typically aims to preserve the meaning of a phrase without considering its use beyond the sentence level, we focus on collecting diverse language to be used directly in a dialogue system in a way that agrees with the full dialogue context.

Manually authoring dialogue systems has been identified as a challenging and time-consuming task (Ward and Pellom, 1999), motivating researchers to explore opportunities to use the crowd to improve and evaluate dialogue systems. Wang et al. (2012) proposed methods to acquire corpora for NLP systems using semantic forms as seeds, and for analyzing the quality of the collected corpora. Liu et al. (2010) used crowdsourcing for free-form language generation and for semantic labeling, with the goal of generating language corpora for new domains. Crowd-workers contribute to dialogue generation in real-time in the Chorus system by providing input about what the system should say next (Lasecki et al., 2013). Crowdsourcing has also been used with some success for dialogue system evaluation (Jurčíček et al., 2011).

Previous work on increasing language diversity in dialogue systems with crowdsourcing has focused on learning about diversity in user input to improve components such as speech recognition and language understanding (e.g., Wang et al. (2012)). Instead, our work focuses on adding diversity to system outputs. Mairesse et al. (2010) followed a similar approach to the work reported here, using crowdsourcing to collect paraphrases for a dialogue system in the restaurant domain. However, the focus of the Mairesse et al. work was on training an NLG module using this data. Our work focuses on crowdsourcing techniques to extract relevant paraphrases, examining the effect of context on their suitability and generalizability.

3 Methodology

Our methodology for developing natural language generation templates is illustrated by the pipeline in Figure 1. This pipeline is designed for dialogue systems that use a template-based natural language generation component. It assumes that the given system has an initial set of language generation templates that have been manually authored, and expands from there. The initial system is used to collect a corpus of dialogues, which we will refer to as the **dialogue seed corpus**, through interactions with users. Based on the dialogue seed corpus, we automatically construct a set of **generation HITs**, web-based crowdsourcing tasks that are used to elicit paraphrases from crowd-workers for instantiated system templates. A generation HIT displays one of the system turns extracted from a system dialogue, with a phrase highlighted, and different amounts of surrounding context in different conditions. The worker is asked to replace the phrase with another one that keeps the same meaning and the coherence of the interaction. If slots are marked in the original, they must be preserved by the worker, which allows us to easily convert the elicited paraphrases to crowd templates. Once a corpus of crowd templates are collected in this fashion, a system developer may filter and decide which to add as viable alternatives to the system's existing list of language generation templates (top path in the pipeline from Figure 1).

We also construct a set of **evaluation HITs** and post them to the crowd to assess the suitability and relative naturalness of the crowd templates (bottom path in the pipeline from Figure 1.) We study how the scores obtained in this crowd-evaluation may be used to help filter the set of new templates that are presented as candidates to the system developer. In the following subsections, we describe each of the pipeline components in detail.

3.1 Dialogue Seed Corpus

We assume as a starting point an existing dialogue system that uses a template-based language generation component. The system uses a set of templates T, which are instantiated with slots filled to generate system phrases. A system turn may contain one or more such phrases connected together. For instance, in the dialogue fragments shown in Figure 2, the template *"Sorry, that was [Place] you wanted, right?"* generates at runtime *"Sorry, that was Ernestine Patrick's office you wanted,*

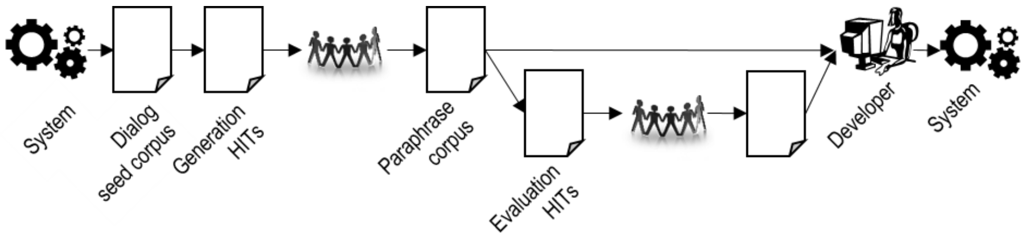


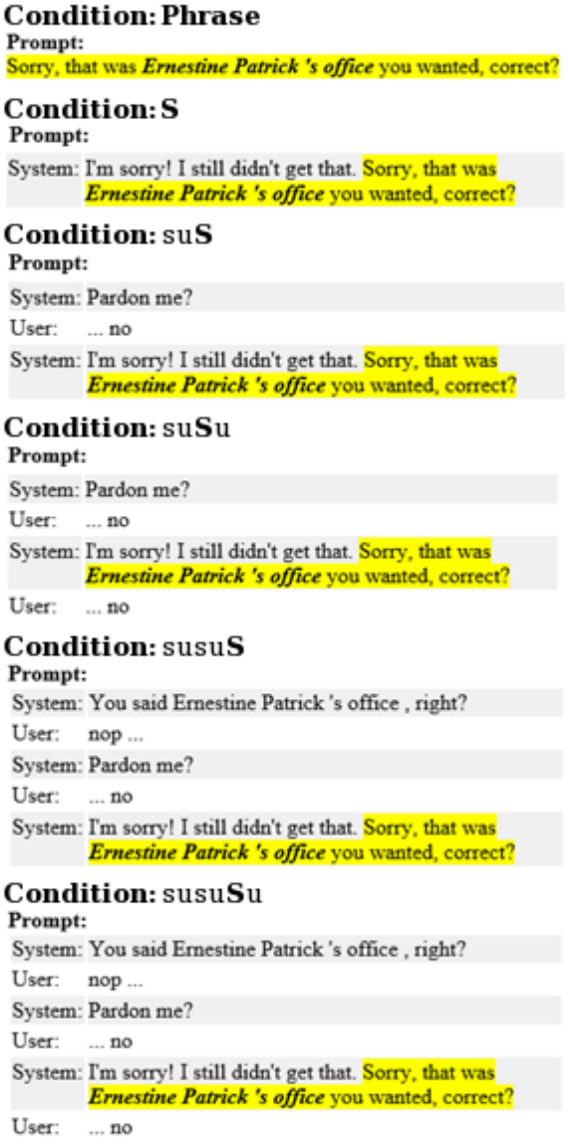
Figure 1: Pipeline for crowd-based development of natural language generation templates.

right?". Statistics on the dialogue seed corpus used in this study are provided in §4.2.

The proposed methodology does not require transcriptions of user utterances in the dialogue seed corpus; instead, it utilizes the recognition results for each user turn. The primary reason behind this choice is that a dialogue that contains recognized user turns may be more coherent than one that contains transcripts and can be generated automatically, as the dialogue manager generates system responses based on the recognition results. However, turn-overtaking issues and recognition problems sometimes resulted in incoherent dialogue interactions. Improving speech recognition remains an area for future work.

3.2 Generation HITs

We use the dialogue seed corpus to produce generation HITs to elicit paraphrases for system phrases from crowd-workers. In the simplest form, a generation HIT might present a single system phrase to the worker. We hypothesize that the surrounding context may be an important factor in facilitating the construction of appropriate paraphrases, affecting their diversity, naturalness, generalizability, etc.; we therefore investigate the effect of presenting varying amounts of dialogue context to the worker.



Condition: Phrase
Prompt:
Sorry, that was ***Ernestine Patrick 's office*** you wanted, correct?

Condition: S
Prompt:
System: I'm sorry! I still didn't get that. Sorry, that was ***Ernestine Patrick 's office*** you wanted, correct?

Condition: suS
Prompt:
System: Pardon me?
User: ... no
System: I'm sorry! I still didn't get that. Sorry, that was ***Ernestine Patrick 's office*** you wanted, correct?

Condition: suSu
Prompt:
System: Pardon me?
User: ... no
System: I'm sorry! I still didn't get that. Sorry, that was ***Ernestine Patrick 's office*** you wanted, correct?
User: ... no

Condition: susuS
Prompt:
System: You said Ernestine Patrick 's office , right?
User: nop ...
System: Pardon me?
User: ... no
System: I'm sorry! I still didn't get that. Sorry, that was ***Ernestine Patrick 's office*** you wanted, correct?

Condition: susuSu
Prompt:
System: You said Ernestine Patrick 's office , right?
User: nop ...
System: Pardon me?
User: ... no
System: I'm sorry! I still didn't get that. Sorry, that was ***Ernestine Patrick 's office*** you wanted, correct?
User: ... no

Figure 2: Generation HIT excerpts in six different context conditions (w/o instructions, examples).

Specifically, given a system phrase corresponding to a template t instantiated in a dialogue, we investigate six different dialogue context conditions. A phrase in a condition presented to a crowd-worker will be referred to as a **seed**, p. Examples of seeds in each condition are illustrated in Figure 2. In the first condition, denoted $Phrase$, a seed is presented to the worker in isolation. In the second condition, denoted $\mathbf{S}$, the entire system turn containing p is presented to the worker, with p highlighted. In the next 4 conditions, denoted $su\mathbf{S}$, $su\mathbf{S}u$, $susu\mathbf{S}$, $susu\mathbf{S}u$, seeds are presented in increasingly larger contexts including one or two previous system and user turns (denoted with lowercase 's' and 'u' in the encoding above), followed by the system turn $\mathbf{S}$ that contains the highlighted seed p, followed in two conditions ($susu\mathbf{S}u$ and $su\mathbf{S}u$) by another user turn. Not all context conditions are applicable for each instantiated template, e.g., conditions that require previous context, such as $su\mathbf{S}$, cannot be constructed for phrases appearing in the first system turn. We follow a between-subjects design, such that each worker works on only a single condition.

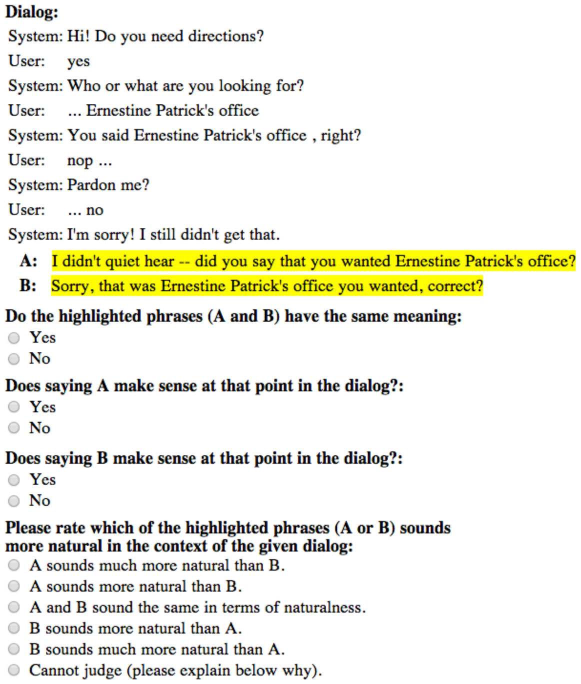
Each generation HIT elicits a paraphrase for a seed. The HIT additionally contains instructions and examples of what workers are expected to do and not to do.[1] We instruct workers to read the dialogue presented and rephrase the highlighted phrase (seed) so as to preserve the meaning and the cohesion of the interaction. To identify slots accurately in the crowd-generated paraphrases, we mark slot values in the given seed with bold italics and instruct workers to keep this portion exactly the same in their paraphrases (see Figure 2). These paraphrases are then turned into **crowd templates** following 3 basic steps: (1) Spelling error correction; (2) Normalization;[2] and (3) Replacing filled slots in the worker's paraphrase with the slot name. We ask workers to provide paraphrases (in English) that differ from the original phrase more substantially than by punctuation changes, and implement controls to ensure that workers enter slot values.

In completing the generation tasks, the crowd produces a corpus of paraphrases, one paraphrase for each seed. For example, *"I apologize, are you looking for Ernestine Patrick's office?"*, is a paraphrase for the highlighted seed shown in Figure 2. As we have asked the workers not to alter slot values, crowd templates can easily be recovered, e.g., *"I apologize, are you looking for [Place]?"*

3.3 Evaluation HITs

A good crowd template must minimally satisfy two criteria: (1) It should maintain the meaning of the original template; and (2) It should sound natural in *any* dialogue context where the original template was used by the dialogue manager, i.e., it should generalize well, beyond the specifics of the dialogue from which it was elicited.

To assess crowd template quality, we construct evaluation HITs for each crowd template. Instantiated versions of the original template and the crowd template are displayed as options A and B (with randomized assignment) and highlighted as part of the entire dialogue in which the original template was used (see Figure 3). In this **in-context (IC)** evaluation HIT, the worker is asked whether the instantiated crowd template has the same meaning as the original, and which is more natural. In addition, because the original dialogues were sometimes incoherent (see §3.1), we also asked the evaluation workers to judge whether the given phrases made sense in the given context.



Dialog:

System: Hi! Do you need directions?
User: yes
System: Who or what are you looking for?
User: ... Ernestine Patrick's office
System: You said Ernestine Patrick's office , right?
User: nop ...
System: Pardon me?
User: ... no
System: I'm sorry! I still didn't get that.
A: I didn't quiet hear -- did you say that you wanted Ernestine Patrick's office?
B: Sorry, that was Ernestine Patrick's office you wanted, correct?

Do the highlighted phrases (A and B) have the same meaning:
○ Yes
○ No

Does saying A make sense at that point in the dialog?:
○ Yes
○ No

Does saying B make sense at that point in the dialog?:
○ Yes
○ No

Please rate which of the highlighted phrases (A or B) sounds more natural in the context of the given dialog:
○ A sounds much more natural than B.
○ A sounds more natural than B.
○ A and B sound the same in terms of naturalness.
○ B sounds more natural than A.
○ B sounds much more natural than A.
○ Cannot judge (please explain below why).

Figure 3: Example evaluation HIT excerpt.

Finally, in order to assess how well the crowd template generalizes across different dialogues, we use a second, **out-of-context (OOC)** evaluation HIT. For each crowd template, we randomly selected a new dialogue where the template t appeared. The out-of-context evaluation HIT presents the instantiated original template and crowd template in this new dialogue. The crowdworkers thus assess the crowd template in a dialogue context different from the one in which it was collected. We describe the evaluation HITs in further detail in §4.

3.4 Developer Filtering

While a crowd-based evaluation can provide insights into the quality of the crowd templates, ultimately, whether or not a template is appropriate for use in the dialogue system depends on many other factors (e.g., register, style, expectations, system goals, etc.). The last step in the proposed methodology is therefore a manual inspection of the crowd templates by a system developer, who assesses which are acceptable for use in the system without changes.

[1] Instructions available at m-mitchell.com/corpora.html.

[2] We normalize capitalization, and add punctuation identical to the seed when no punctuation was provided.

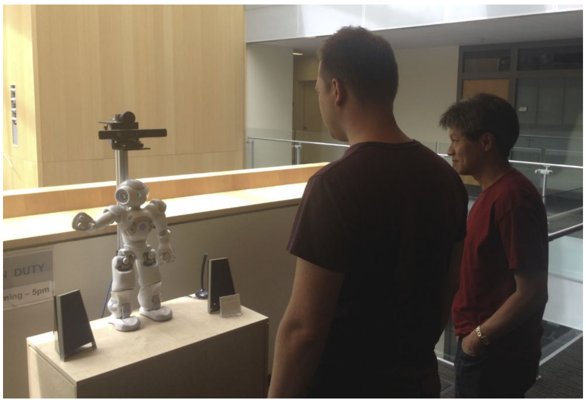


Figure 4: Directions Robot system.

4 Experiments and Results

We now describe our experiments and results. We aim to discover whether *there is an effect of the amount of surrounding context on perceived crowd template naturalness*. We additionally explore whether the crowd template retains the meaning of the original template, whether they both make sense in the given context, and the diversity of the templates that the crowd produced for each template type. We report results when the templates are instantiated *in-context*, in the original dialogue; and *out-of-context*, in a new dialogue. We first describe the experimental test-bed and the corpora used and collected below.

4.1 Experimental Platform

The test-bed for our experiments is Directions Robot, a situated dialogue system that provides directions to peoples' offices, conference rooms, and other locations in our building (Bohus et al., 2014). The system couples a Nao humanoid robot with a software infrastructure for multi-modal, physically situated dialogue (Bohus and Horvitz, 2009) and has been deployed for several months in an open space, in front of the elevator bank on the 3rd floor of our building (see Figure 4). While some of the interactions are need-based, e.g., visitors coming to the building for meetings, many are also driven by curiosity about the robot.

The Directions Robot utilizes rule-based natural language generation, with one component for giving directions based on computed paths, and another component with 38 templates for the rest of the dialogue. Our experimentation focuses on these 38 templates. As the example shown in Figure 2 illustrates, slots are dynamically filled in at run-time, based on the dialogue history.

	Crowd Generation				**Crowd Eval.**	
Cond.	# Gen HITs (× 3)	# *w*	Time/ HIT (sec)	# Uniq. Para.	# Eval HITs (× 5)	Time/ HIT (sec)
Phrase	767	26	34.7	1181	1126	29.4
S	860	28	30.8	1330	1260	39.2
suS	541	26	33.3	1019	772	30.5
suSu	265	24	38.8	531	392	32.6
susuS	360	24	41.0	745	572	32.3
susuSu	296	28	42.9	602	440	34.4
Total	3089	-	-	5408	4562	-
Average	-	26	36.9	-	-	33.1

Table 1: Statistics for the crowd-based generation and evaluation processes. Each generation HIT was seen by 3 unique workers and each evaluation HIT was seen by 5 unique workers. *#w* represents number of workers. For evaluation, *#w* = 231.

We conducted our experiments on a general-purpose crowdsourcing marketplace, the Universal Human Relevance System (UHRS).[3] The marketplace connects human intelligence tasks with a large population of workers across the globe. It provides controls for selecting the country of residence and native languages for workers, and for limiting the maximum number of tasks that can be done by a single worker.

4.2 Crowd-based Generation

Dialogue seed corpus We used 167 dialogues collected with the robot over a period of one week (5 business days) as the dialogue seed corpus. The number of turns in these dialogues (including system and user) ranges from 1 to 41, with a mean of 10 turns. 30 of the 38 templates (79%) appeared in this corpus.

Generation HITs We used the dialogue seed corpus to construct generation HITs, as described in §3.2. In a pilot study, we found that for every 10 instances of a template submitted to the crowd, we received approximately 6 unique paraphrases in return, with slightly different ratios for each of the six conditions. We used the ratios observed for each condition in the pilot study to down-sample the number of instances we created for each template seen more than 10 times in the corpus. The total number of generation HITs resulting for each condition is shown in Table 1.

Crowd generation process Statistics on crowd generation are shown in Table 1. Each worker could complete at most 1/6 of the total HITs for that condition. We paid 3 cents for each genera-

[3] This is a Microsoft-internal crowdsourcing platform.

tion HIT, and each HIT was completed by 3 unique workers. From this set, we removed corrupt responses, and all paraphrases for a generation HIT where at least one of the 3 workers did not correctly write the slot values. This yielded a total of 9123 paraphrases, with 5408 unique paraphrases.

4.3 Crowd-based Evaluation

Evaluation HITs To keep the crowd evaluation tractable, we randomly sampled 25% of the paraphrases generated for all conditions to produce evaluation HITs. We excluded paraphrases from seeds that did not receive paraphrases from all 3 workers or were missing required slots. As discussed in §3, paraphrases were converted to crowd templates, and each crowd template was instantiated in the original dialogue, *in-context* (IC) and in a randomly selected *out-of-context* (OOC) dialogue. The OOC templates were instantiated with slots relevant to the chosen dialogue. This process yielded 2281 paraphrases, placed into each of the two contexts.

Crowd evaluation process As discussed in §3.3, instantiated templates (crowd and original) were displayed as options A and B, with randomized assignment (see Figure 3). Workers were asked to judge whether the original and the crowd template had the same meaning, and whether they made sense in the dialogue context. Workers then rated which was more natural on a 5-point ordinal scale ranging from -2 to 2, where a -2 rating marked that the original was much more natural than the crowd template. Statistics on the judgments collected in the evaluation HITs are shown in Table 1. Workers were paid 7 cents for each HIT. Each worker could complete at most 5% of all HITs, and each HIT was completed by 5 unique workers.

Outlier elimination One challenge with crowd-sourced evaluations is noise introduced by spammers. While questions with known answers may be used to detect spammers in objective tasks, the subjective nature of our evaluation tasks makes this difficult: a worker who does not agree with the majority may simply have different opinions about the paraphrase meaning or naturalness. Instead of spam detection, we therefore seek to identify and eliminate outliers; in addition, as previously discussed, each HIT was performed by 5 workers, in an effort to increase robustness.

We focused attention on workers who performed at least 20 HITs (151 of 230 workers, covering 98% of the total number of HITs). Since we randomized the A/B assignment of instantiated original templates and crowd templates, we expect to see a symmetric distribution over the relative naturalness scores of all judgments produced by a worker. To identify workers violating this expectation, we computed a score that reflected the symmetry of the histogram of the naturalness votes for each worker. We considered as outliers 6 workers that were more than z=1.96 standard deviations away from the mean on this metric (corresponding to a 95% confidence interval). Secondly, we computed a score that reflected the percentage of tasks where a worker was in a minority, i.e., had the single opposing vote to the other workers on the *same meaning* question. We eliminated 4 workers, who fell in the top 97.5 percentile of this distribution. We corroborated these analyses with a visual inspection of scatterplots showing these two metrics against the number of tasks performed by each judge.[4] As one worker failed on both criteria, overall, 9 workers (covering 9% of all judgements) were considered outliers and their responses were excluded.

4.4 Crowd Evaluation Results

Meaning and Sense Across conditions, we find that most crowd templates are evaluated as having the *same meaning* as the original and *making sense* by the majority of workers. Evaluation percentages are shown in Table 2, and are around 90% across the board. This suggests that in most cases, the generation task yields crowd templates that meet the goal of preserving the meaning of the original template.

Naturalness To evaluate whether the amount of surrounding context has an effect on the perceived naturalness of a paraphrase relative to the original phrase, we use a Kruskal-Wallis (KW) test on the mean scores for each of the paraphrases, setting our significance level to .05. A Kruskal-Wallis test is a non-parametric test useful for significance testing when the independent variable is categorical and the data is not assumed to be normally distributed. We find that there is an effect of condition on the relative naturalness score (KW chi-squared = 15.9156, df = 5, p = 0.007) when crowd

[4] Scatterplots available at m-mitchell.com/corpora.html.

	Crowd Evaluation								**Developer Evaluation**		
Cond.	% Same Meaning		% Makes Sense		Avg. Relative Naturalness		Avg. D-score		% Dev. Accepted		Avg. D-score
	IC	**OOC**	**IC**	**OOC**	**IC**	**OOC**	**IC**	**OOC**	**All**	**Seen>1**	
Phrase	92	91	90	90	-.54 (.66)	-.50 (.61)	.67	.67	37	67	.30
S	91	89	88	88	-.50 (.65)	-.47 (.66)	.68	.64	35	53	.29
suS	84	87	85	87	**-.37** (.65)	**-.37** (.61)	.70	.70	40	63	.41
suSu	88	85	**95**	88	-.48 (.62)	-.43 (.61)	.76	.71	38	50	.39
susuS	**94**	**94**	91	**94**	-.43 (.70)	-.39 (.67)	**.81**	**.80**	38	**78**	.34
susuSu	91	89	92	86	-.40 (.61)	-.38 (.66)	.73	.74	**45**	67	**.42**

Table 2: % same meaning, % makes sense, and average relative naturalness (standard deviation in parentheses), measured in-context (IC) and out-of-context (OOC); crowd-based and developer-based diversity score (D-score); developer acceptance rate computed over all templates, and those seen more than once. The $susuS$ condition yields the most diverse templates using crowd-based metrics; removing templates seen once in the evaluation corpus, this condition has the highest acceptance in the developer evaluation.

templates are evaluated in-context, but not out-of-context (KW chi-squared = 9.4102, df = 5, p-value = 0.09378). Average relative naturalness scores in each condition are shown in Table 2.

Diversity We also assess the diversity of the templates elicited from the crowd, based on the evaluation set. Specifically, we calculate a diversity score (D-score) for each template type t. We calculate this score as the number of unique crowd template types for t voted to make sense and have the same meaning as the original by the majority, divided by the total number of seeds for t with evaluated crowd templates. More formally, let P be the original template instantiations that have evaluated crowd templates, M the set of unique crowd template types voted as having the *same meaning* as the original template by the majority of workers, and S the set of unique crowd template types voted as *making sense* in the dialogue by the majority of workers. Then:

$$\text{D-score}(t) = \frac{|M \cap S|}{|P|}$$

The average diversity scores across all templates for each condition are shown in Table 2. We find the templates that yield the most diverse crowd templates include `WL_Retry` *"Where are you trying to get to in this building?"* and `OK_Help`, *"Okay, I think I can help you with that"*, which have a diversity rating of 1.0 in several conditions: for each template instance we instantiate (i.e., each generation HIT), we get a new, unique crowd template back. Example crowd templates for the `OK_Help` category include *"I believe I can help you find that"* and *"I can help you ok"*. The templates with the least diversity are those for `Hi`, which has a D-score around 0.2 in the S and $Phrase$ conditions.

4.5 Developer Acceptability Results

For the set of crowd templates used in the crowd-based evaluation process, one of the system developers[5] provided binary judgments on whether each template could be added (without making any changes) to the system or not. The developer had access to the original template, extensive knowledge about the system and domain, and the way in which each of these templates are used.

Results indicate that the developer retained 487 of the 1493 unique crowd templates that were used in crowd-evaluation (33%). A breakdown of this acceptance rate by condition is shown in Table 2. When we eliminate templates seen only once in the evaluation corpus, acceptability increases, at the expense of recall. We additionally calculate a diversity score from those templates accepted by the developer, which is simply the number of crowd template types accepted by the developer, divided by the total number of seeds used to elicit the crowd templates in the developer's evaluation, for each template type t.

The developer evaluation revealed a wide range of reasons for excluding crowd templates. Some of the most common were lack of grammaticality, length (some paraphrases were too long/short), stylistic mismatch with the system, and incorrect punctuation. Other reasons included register issues, e.g., too casual/presumptive/impolite, issues of specificity, e.g., template was too general, and issues of incompatibility with the dialogue state and turn construction process. Overall, the developer interview highlighted very specific system

[5] The developer was not an author of this paper.

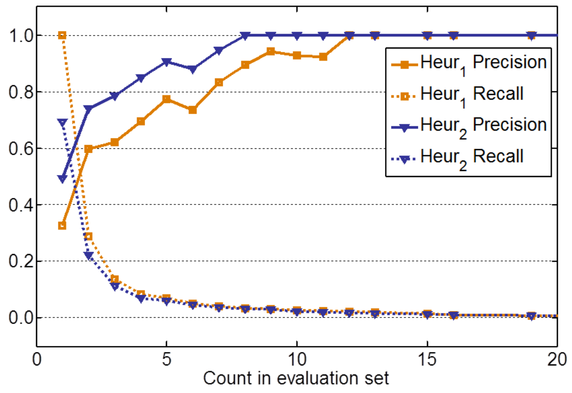


Figure 5: Precision and recall for heuristics.

and domain knowledge in the selection process.

4.6 Crowd-based Evaluation and Developer Acceptability

We now turn to an investigation of whether statistics from the crowd-based generation and evaluation processes can be used to automatically filter crowd templates. Specifically, we look at two heuristics, with results plotted in Figure 5. These heuristics are applied across the evaluation corpus, collating data from all conditions. The first heuristic, Heur$_1$, uses a simple threshold on the number of times a crowd template occurred in the evaluation corpus.[6] We hypothesize that more frequent paraphrases are more likely to be acceptable to the developer, and in fact, as we increase the frequency threshold, precision increases and recall decreases.

The second heuristic, Heur$_2$, combines the threshold on counts with additional scores collected in the out-of-context crowd-evaluation: It only considers templates with an aggregated judgment on the *same meaning* question greater than 50% (i.e., the majority of the crowd thought the paraphrase had the same meaning as the original), and with an aggregated relative naturalness score above the overall mean. As Figure 5 illustrates, different tradeoffs between precision and recall can be achieved via these heuristics, and by varying the count threshold.

These results indicate that developer filtering remains a necessary step for adding new dialogue system templates, as the filtering process cannot yet be replaced by the crowd-evaluation. This is not surprising since the evaluation HITs did not express all the different factors that we found the developer took into account when selecting templates, such as style decisions and how phrases are combined in the system to form a dialogue. Future work may consider expanding evaluation HITs to reflect some of these aspects. By using signals acquired through crowd generation and evaluation, we should be able to reduce the load for the developer by presenting a smaller and more precise candidate list at the expense of reductions in recall.

5 Discussion

We proposed and investigated a methodology for developing a corpus of natural language generation templates for a spoken dialogue system via crowdsourcing. We investigated the effect of the context we provided to the workers on the perceived meaning, naturalness, and diversity of the alternates obtained, and evaluated the acceptability of these alternates from a system developer viewpoint.

Our results show that the crowd is able to provide suitable and diverse paraphrases within this methodology, which can then be converted into crowd templates. However, more work is necessary before elicited crowd templates can be plugged directly into a system.

In future work, we hope to continue this process and investigate using features from the crowd and judgments from system developers in a machine learning paradigm to automatically identify crowd templates that can be directly added to the dialogue system. We would also like to extend beyond paraphrasing single templates to entire system turns. With appropriate controls and feature weighting, we may be able to further expand dialogue capabilities using the combined knowledge of the crowd. We expect that by eliciting language templates from multiple people, as opposed to a few developers, the approach may help converge towards a more natural distribution of alternative phrasings in a dialogue. Finally, future work should also investigate the end-to-end effects of introducing crowd elicited templates on the interactions with the user.

Acknowledgments

Thanks to members of the ASI group, Chit W. Saw, Jason Williams, and anonymous reviewers for help and feedback with this research.

[6] Since the evaluation corpus randomly sampled 25% of the generation HITs output, this is a proxy for the frequency with which that template was generated by the crowd.

References

D. Bohus and E. Horvitz. 2009. Dialog in the open world: Platform and applications. *Proceedings of ICMI'2009*.

Dan Bohus, C. W. Saw, and Eric Horvitz. 2014. Directions robot: In-the-wild experiences and lessons learned. *Proceedings of AAMAS'2014*.

Martin Potthast Burrows, Steven and Benno Stein. 2013. Paraphrase acquisition via crowdsourcing and machine learning. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 43.

Olivia Buzek, Philip Resnik, and Benjamin B. Bederson. 2010. Error driven paraphrase annotation using mechanical turk. *Proceedings of the NAACL HLT 2010 Workshop on Creating Speech and Language Data with Amazon's Mechanical Turk*.

Chris Callison-Burch and Mark Dredze. 2010. Creating speech and language data with amazon's mechanical turk. *Proceedings of the NAACL HLT 2010 Workshop on Creating Speech and Language Data with Amazon's Mechanical Turk*.

David L. Chen and William B. Dolan. 2011. Collecting highly parallel data for paraphrase evaluation. *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies-Volume 1*.

Michael Denkowski and Alon Lavie. 2010. Exploring normalization techniques for human judgments of machine translation adequacy collected using amazon mechanical turk. *Proceedings of the NAACL HLT 2010 Workshop on Creating Speech and Language Data with Amazon's Mechanical Turk*.

Matthai Philipose Han, Seungyeop and Yun-Cheng Ju. 2013. Nlify: lightweight spoken natural language interfaces via exhaustive paraphrasing. *Proceedings of the 2013 ACM international joint conference on Pervasive and ubiquitous computing*.

Filip Jurčíček, Simon Keizer, Milica Gašić, Frančois Mairesse, Blaise Thomson, Kai Yu, and Steve Young. 2011. Real user evaluation of spoken dialogue systems using amazon mechanical turk. *Proceedings of INTERSPEECH*, 11.

Walter S. Lasecki, Rachel Wesley, Jeffrey Nichols, Anand Kulkarni, James F. Allen, and Jeffrey P. Bigham. 2013. Chorus: a crowd-powered conversational assistant. *Proceedings of the 26th annual ACM symposium on User interface software and technology*.

Sean Liu, Stephanie Seneff, and James Glass. 2010. A collective data generation method for speech language models. *Spoken Language Technology Workshop (SLT), IEEE*.

François Mairesse, Milica Gašić, Filip Jurčíček, Simon Keizer, Blaise Thomson, Kai Yu, and Steve Young. 2010. Phrase-based statistical language generation using graphical models and active learning. *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*.

Matteo Negri, Yashar Mehdad, Alessandro Marchetti, Danilo Giampiccolo, and Luisa Bentivogli. 2012. Chinese whispers: Cooperative paraphrase acquisition. *Proceedings of LREC*.

Gabriel Parent and Maxine Eskenazi. 2010. Toward better crowdsourced transcription: Transcription of a year of the let's go bus information system data. *Spoken Language Technology Workshop (SLT), IEEE*.

Martin Tschirsich and Gerold Hintz. 2013. Leveraging crowdsourcing for paraphrase recognition. *LAW VII & ID*, 205.

William Yang Wang, Dan Bohus, Ece Kamar, and Eric Horvitz. 2012. Crowdsourcing the acquisition of natural language corpora: Methods and observations. *Spoken Language Technology Workshop (SLT), IEEE*.

W. Ward and B. Pellom. 1999. The cu communicator system. *Proceedings of IEEE ASRU*.

Omar F. Zaidan and Chris Callison-Burch. 2011. Crowdsourcing translation: Professional quality from non-professionals. *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies-Volume 1*.