A Comparative Evaluation Methodology for NLG in Interactive Systems

Helen Hastie, Anja Belz
Heriot Watt University, University of Brighton
h.hastie@hw.ac.uk, a.s.belz@brighton.ac.uk

Abstract

Interactive systems have become an increasingly important type of application for deployment of NLG technology over recent years. At present, we do not yet have commonly agreed terminology or methodology for evaluating NLG within interactive systems. In this paper, we take steps towards addressing this gap by presenting a set of principles for designing new evaluations in our comparative evaluation methodology. We start with presenting a categorisation framework, giving an overview of different categories of evaluation measures, in order to provide standard terminology for categorising existing and new evaluation techniques. Background on existing evaluation methodologies for NLG and interactive systems is presented. The comparative evaluation methodology is presented. Finally, a methodology for comparative evaluation of NLG components embedded within interactive systems is presented in terms of the comparative evaluation methodology, using a specific task for illustrative purposes.

1. Introduction

NLG (natural language generation) can sometimes look like the poor cousin of NLA (natural language analysis), especially because two sizeable NLP subfields that once comprised NLG as a subtask (MT and summarisation) for the most part no longer do, and are no longer even seen as overlapping with NLG. The advent of comparative and competitive evaluation has the potential to change the status of NLG, and the next five years will be crucial in this respect. Important trends are likely to be towards reusable plug-and-play system components, and towards applications. Next to data-to-text applications, NLG components embedded in interactive systems are likely to provide some of the most important opportunities for technological advance.

Our aim in this paper is to take a look at existing evaluation methods and shared tasks in NLG and interactive systems, and to use this as the basis for a discussion of how to evaluate NLG within interactive systems, outlining requirements and sketching possible future shared tasks. We begin by presenting an overview of different types of evaluation in a *categorisation framework* that is particularly suited to NLG in the context of interactive systems and that will provide us with a standard terminology to describe specific examples of evaluations in the rest of the paper. We then survey evaluation in NLG for components in isolation, end-to-end NLG systems and NLG components as embedded in interactive systems. Finally, we discuss requirements and methods for evaluating NLG embedded in interactive systems, providing a set of principles for designing new evaluations in a *comparative evaluation methodology* and providing a concrete example of a shared task using this methodology.

2. Categorising different types of evaluations of automatically generated language

Table 1 shows a ***categorisation framework for evaluations of automatically generated language***, incorporating the evaluation types we discuss in this paper, and populated with the systems and shared tasks we survey. The categories reflect *what* is being evaluated (indexing the rows), and *how* it is being evaluated (indexing the columns).

The taxonomy of evaluation measures represented by the columns in Table 1 is a generic one, and the idea is that any given evaluation method can, independently of the context in which it is developed/applied, be assigned to one of the columns. For example, a measure which computes the percentage of times users play a game to completion is an extrinsic measure – more specifically it is a task-success measure – regardless of the type of system that is being evaluated or the purpose of the evaluation.

Designers of evaluation experiments often talk about assessing higher-level ***quality criteria*** such as dialogue performance, dialogue quality, system usability, user satisfaction, efficiency, etc. It does not make sense to pair up individual evaluation measures with single abstract, higher-level quality criteria. Rather, the decision about which measure(s) to use to assess a given quality criterion is part and parcel of the design of evaluations and is very much context-dependent. There is an important distinction between generic types of evaluation measures (columns in Table 1) and evaluation criteria (what the evaluation measures are used as a surrogate measure for).

For comprehensive comparative evaluation of systems or components, multiple evaluation methods, moreover in more than one of the categories of evaluation methods in Table 1 tend to be applied. This has been the case of all NLG shared task evaluations to date, and also of the NIST-run MT and Summarisation evaluations. This approach yields a set of evaluation scores and corresponding ranking of systems for each evaluation measure that is applied, and it frequently happens that different systems come top of the ranking table for different measures.

In some contexts it is desirable that an overall 'best' system should be identified. Deciding on an overall winner on the basis of a single measure is not entirely satisfactory, because it is rare that a single measure can be found that encapsulates all aspects of system quality. This is particularly true of evaluation situations where stakeholders other than system developers (end users, funding bodies seeking to accelerate technological progress, companies that will turn systems into marketable products, etc.) play a role. Here, ***evaluation frameworks*** offer a solution by computing a single overall score on the basis of the separate scores

		HOW			
	Evaluation mode:	**Intrinsic**		**Extrinsic**	
	Measure type:	**Output quality**	**User like**	**User task success**	**System purpose success**
WHAT	**NLG components in isolation**	TUNA, GREC, SR'11, Prodigy-METEO	?	GREC-MSR, TUNA-AS, TUNA-REG	?
	End-to-end NLG systems	GREC-Full, SumTime-METEO, QG'10, BT-Nurse, HOO	BT-Nurse	SumTime-METEO	STOP
	Embedded NLG components	SkillSum, GIVE, SPOT, M-PIRO, COMIC, GRUVE, CLASSiC	SkillSum, GIVE, ILEX, M-PIRO, CLASSiC, PARLANCE, GRUVE	SkillSum, GIVE, ILEX, M-PIRO, CLASSiC, GRUVE	SkillSum, ILEX, GIVE

Table 1: Overview of the system evaluations (represented by name of project or shared task) we survey in this paper, categorised according to *what* is being evaluated (components in isolation, end-to-end systems and embedded components), and *how* it is evaluated (evaluation mode and type of measure).

produced by multiple individual measures.

For example, the overall goal of a dialogue system evaluation may be to assess 'Dialogue Performance'. The evaluation designers may decide that Dialogue Performance is comprised of the quality criteria Dialogue Quality, User Satisfaction and Usability. They select several intrinsic output quality measures to capture Dialogue Quality, several user-like measures in exit questionnaires to capture User Satisfaction, and several task-success measures to capture Usability. Finally, a method is chosen for computing a single score from the measures; this could be as simple as computing the (weighted) average of the measures, or more complex, involving e.g. methods from decision theory.

3. Evaluation and Shared Tasks in NLG

Present day NLG is a very different field from what it was seven years ago before the advent of the first NLG shared task. One of the most important changes over the past seven years has been the introduction of systematic comparative evaluation methodologies.

In this section, we provide a survey of evaluation in NLG, grouped into three categories (as indicated in the rows in Table 1): (a) NLG system components (which implement an NLG subtask) that are evaluated in isolation, e.g. the REG, TUNA, GREC, and Surface Realisation shared tasks; (b) end-to-end NLG systems evaluated as a black box, e.g. the METEO data and task, the HOO and Question Generation Shared Tasks; and (c) NLG systems that are components in an embedding interactive system and where evaluation is at the level of the embedding system, e.g. COMIC, ILEX, M-PIRO, SPOT, SkillSum, CLASSIC, PARLANCE and GIVE.

3.1. NLG Components in Isolation

NLG today is a very different field from what it was in 2005. One of the most important changes over the past few years has been the introduction of systematic comparative evaluation methodologies. *Competetive* comparative evaluation soon followed: the 2007-09 Referring Expression Generation (REG) competitions were the first NLG shared tasks and were organised against a background of growing interest in empirically evaluating REG algorithms, which had hitherto often been justified on the basis of theoretical and psycholinguistic principles, but lacked a sound empirical grounding. Early empirical evaluations of REG algorithms either used existing corpora such as COCONUT (Jordan and Walker, 2005, Gupta and Stent, 2005), or constructed new datasets based on human experiments (Viethen and Dale, 2006, Gatt et al., 2007, van der Sluis et al., 2007). These evaluations focused on classic approaches to the REG problem, such as Dale's Full Brevity and Greedy algorithms (Dale, 1989), and Dale and Reiter's Incremental Algorithm (Dale and Reiter, 1995). In studies by Gupta and Stent (2005) and Jordan and Walker (2005), these algorithms were evaluated in a dialogue context and were extended with novel features to handle dialogue and/or compared to new frameworks such as Jordan's *Intentional Influences* model (Jordan, 2000).

The TUNA shared tasks were run between 2007 and 2009 and focused on referring expressions generation (REG). They were based on the TUNA Corpus (van Deemter et al., 2012), and focused on the generation of full, identifying, definite noun phrases in visual domains where the entities are either furniture items or people. There were three TUNA shared tasks: TUNA-AS (focusing on attribute selection), TUNA-R (focusing on referring expression realisation), and TUNA-REG (combining attribute selection and realisation).

The GREC-MSR (Main Subject Reference) challenge used the GREC Corpus (version 2.0) which consists of roughly 2,000 introductory sections in Wikipedia articles (Belz et al., 2009). In each text, three broad categories of Main Subject Reference (MSR) were annotated (13,000 REs in total). A variant of this task, GREC-NEG, used 1,100 annotated introduction sections from articles about people on Wikipedia. For both these shared tasks, the objective was to select one of the REs in a given list, for each mention in each text in the test sets; in GREC-MSR, systems generated REs for a single referent (the main subject), and a set of grammatically correct possible choices was provided for each mention; in GREC-NEG, systems

generated REs for all people in a text, and a single list of possible REs was provided for each person. Finally, the task in GREC-NER was a straightforward combined Named-Entity Recognition and coreference resolution task, restricted to people entities. Systems inserted RE tags and coreference IDs around recognised mentions. The aim was to match the "gold-standard" tags in the GREC-People data.

Another NLG subtask for which a shared-task competition has been run is Surface Realisation; the aim of the SR'11 Shared Task was to enable a direct comparison of independently developed surface realisers by developing a "common-ground" input representation, which could be used to generate realisations by all participating systems (Belz et al., 2011). In fact, the organisers created two different input representations, one shallow, one deep, in order to enable more teams to participate. SR'11 systems mapped from the shallow (or deep) inputs to word strings that are readable, clear and as similar as possible in meaning to the original corpus sentences.

3.2. End-to-end NLG systems

End-to-end NLG systems that have been evaluated through shared tasks fall into two categories: "Data-to-Text" and "Text-to-Text" Generation. For data-to-text evaluation, the METEO data is, as far as we aware, the only resource that has been used by multiple research groups to develop directly comparable systems outside the context of NLG shared tasks (Reiter et al., 2005, Belz, 2007, Reiter and Belz, 2009, Angeli et al., 2010, Langner, 2010). The Prodigy-METEO corpus (Belz, 2007) contains just a.m. wind forecasts and corresponding input data vectors, whereas the original SumTime-METEO corpus (Sripada et al., 2002) contains complete weather forecasts and the complete original weather data files.

With regard to text-to-text applications, these include machine translation, summarisation, paraphrasing and text improvement. All of these involve (possibly shallow) analysis of the input text followed by (possibly shallow) synthesis of the output text. Many approaches do not comprise a clearly identifiable NLG subtask. Three text-to-text shared tasks have so far been organised by the NLG community: the GREC-Full shared task, Helping Our Own (HOO) and the Question Generation (QG) Task. The immediate motivating application context for the GREC-Full Task is the improvement of referential clarity and coherence in extractive summaries and multiple edited texts (such as Wikipedia articles) by regeneration of Referring Expressions contained in them. The HOO Task is an automated text correction task focused on correcting mistakes by non-native speakers of English in English academic papers in the NLP domain. A pilot version of this task was run in 2011 (Dale and Kilgarriff, 2011), followed by the HOO Prepositions and Determiners Task in 2012. Rus et al. (2011) report on the QG task which included two tasks: Question Generation from sentences and Question Generation from paragraphs. Input data sources for both tasks were Wikipedia, OpenLearn, and Yahoo!Answer.

3.3. NLG as an Embedded Component

In this section we discuss evaluations from several different projects (part of) whose aim was to develop and evaluate NLG components as part of wider system. These include COMIC (Foster and White, 2005), ILEX (Cox et al., 1999), M-PIRO (Androutsopoulos et al., 2005), SPOT (Rambow et al., 2001), SkillSum (Williams and Reiter, 2008), CLASSiC (Janarthanam et al., 2011) and PARLANCE (Dethlefs et al., 2013). All of these applications are interactive to some degree, and some of the NLG components take aspects of the interaction into account when generating outputs. For example, the ILEX system varies descriptions depending on browsing history and descriptions can refer back to previously viewed items (Cox et al., 1999).

Evaluations are at the system level comparing systems that differ only in the NLG module, so they count as embedded module evaluations in the terminology of Table 1. For example in the CLASSiC project, two end-to-end systems were compared where only the NLG modules differed. It was found that the NLG module using a trained information presentation strategy significantly improved dialogue task success for real users, over using a system with an NLG module which uses conventional, hand-coded presentation methods (Rieser et al., 2014). Where this is an example of improvement observed in *extrinsic user task success measures* another study in the same project showed similar results for perceived task success but also an improvement in *intrinsic user like quality measures* by showing that User Satisfaction was significantly increased using a data-driven approach to Temporal Expression Generation compared to a rule-based system, again as an embedded module in an interactive system (Janarthanam et al., 2011).

3.3.1. Shared Tasks for Embedded NLG Components

The main shared task for this category is GIVE where participants implement NLG modules that generate instructions to guide a human user in solving a treasure-hunt task in a virtual 3D world in real time. In evaluations, the NLG modules are plugged into the same interactive virtual environment, and the performance and experience of users is evaluated.

The first edition of the GIVE Task was limited in that only discrete steps of the same size, and rotation in one of four directions, were possible in the virtual environments. As a result, the NLG task was easier than intended (as noted in the results report (Koller et al., 2010), one of the best systems generated instructions of the type *Move three steps forward*). In the second edition of the task, users were able to navigate the virtual environment freely, turning at any angle and moving any distance at a time. In this subsection, we provide an overview of this second version of GIVE.

The task for GIVE-2 NLG systems is to generate, in real time, natural-language instructions that enable users to successfully complete the treasure hunt. The embedding environment consists of multiple rooms containing objects (as landmarks), various buttons (with one of several functions, including opening the treasure safe, triggering an alarm, and no function), and floor tiles that trigger an alarm. The possible outcomes of a single game are "success" (treasure found), "alarm" and "user leaves".

From the point of view of the NLG system, the interaction between it and the embedding GIVE-2 system looks as follows: The NLG system receives a message every 200ms containing the position and orientation of the user and the list of currently visible objects. Furthermore, the NLG system receives a message every time the user manipulates an object. The NLG system can, at any point, request from the GIVE-2 system a plan of actions to get the user from his/her current position to the treasure. It is up to the NLG system to decide when to request plans. So for any single instance of generating an instruction, the NLG module has the following available to it as input:

- A representation of the world the user is in;
- Object manipulations by the user so far;
- Current position and orientation of the user;
- Current list of visible objects; and
- Current plan of actions to reach treasure.

Certain properties of the embedding GIVE-2 system affect the kind of instructions that the NLG system should generate. For example, the GIVE-2 system displays the instructions for a fixed length of time, so if an instruction is too long, the user cannot read all of it. Furthermore, as the user is free to move at any point (including while the next instruction is being generated), it is possible for the next instruction to be out of date by the time it reaches the user, and they may therefore not be able to understand it (e.g. if objects which the NLG system assumes are visible, are no longer visible from the user's new position).

Non-NLG tasks that the NLG system needs to perform include deciding when to request a new plan of actions and what to do with the information about objects manipulated by the user. Content-selection tasks performed by the system include deciding which subset of actions to select to be covered by the next instruction, and selecting objects from the list of visible objects (or even the world representation) to refer to as landmarks. Most participating NLG systems perform discourse-planning tasks such as aggregating actions and objects. Other typical NLG tasks participating systems could (but do not have to) perform include referring expression generation, lexicalisation, and surface realisation.

Participating NLG systems were evaluated in terms of the following *extrinsic user-task success measures* and *intrinsic, automatic output-quality measures* that were computed by the GIVE-2 organisers (apart from the first measure, these were computed on successfully completed games only):

- Task success rate: percentage of games in which user found trophy;
- Task duration: mean number of seconds from start of game to trophy retrieval;
- Distance covered by user: mean number of distance units of the virtual environment covered between start of game and retrieval of trophy;
- Actions required: mean number of object manipulation actions in games;
- Instructions required: mean number of instructions produced by the NLG module in games; and
- Instruction length: mean number of words in instructions.

The GIVE-2 organisers also obtained ratings for a further twenty-two *intrinsic user-like measures* from players via an exit questionnaire filled in at the end of each game. Questions ranged from the perceived clarity and readability of instructions to how enjoyable the game was. If one deemed entertainment the purpose of the GIVE-2 environment, some of these measures could be classified as *system-purpose success measures*. Some measures aim to assess the quality of part of the system output (the instructions), while most of the measures are more obviously *user like measures*.

While NLG module developers receive feedback (in the form of exit questionnaires filled in after test games) on how highly users rate the performance of their module, they do not receive any feedback about how to improve their instructions. Such feedback could be provided in the form of model instructions. Human-authored model instructions could conceivably be created, perhaps for a set of specific individual game situations, against which modules could be automatically measured, or which could be used to train modules.

The status of some of the automatic measures mentioned above is unclear. To count as output quality measures, it would have to be clear which end of a scale corresponds to higher quality, which is not the case for instruction length, and possibly some of the other measures.

While GIVE-2 is an interactive system, it is not linguistically interactive like the dialogue systems we focus on in the next section. The system outputs (a) continuously updated visualisations of the world the user is moving through, and (b) verbal instructions to the user, at points determined by the NLG module. The user does not directly input any information, but can move through a given virtual world and manipulate some of the objects in it, both of which can affect system behaviour. The input can be assumed to be reliable and the NLG task can be viewed as data-to-text generation, except that strategies have to be developed for what to do if the user strays from the intended path. All in all, this is a relatively simple interactive set-up, making the GIVE tasks ideal first shared tasks on NLG embedded within an interactive system.

A more recent shared task proposal for NLG embedded systems is GRUVE (Janarthanam and Lemon, 2012). The envisaged task is a variant of direction generation similar to the GIVE challenge, but in GRUVE the system has varying certainty of input data. This challenge is also more interactive than the GIVE challenge with users able to interact with the system using an Interaction Panel thus modifying the behaviour of the system through interaction as well as through movement in the world.

4. Evaluation and Shared Tasks for Interactive Systems

Comparative evaluations of end-to-end interactive systems have typically been in the form of task-based experiments where paid participants perform a task in the laboratory and then fill out a questionnaire, yielding a range of *intrinsic user like measures*. However, it can be argued that lab-based evaluations are inefficient, and involve unrealistic scenarios and user goals. Studies have shown that dialogue quality measures vary significantly depending on whether the pool of callers are real users or paid subjects (Ai et al., 2008).

Here, we discuss spoken dialogue system evaluations in terms of realism and level of environment control. The scenario with the least control but perhaps the most realism is where the evaluation is 'in the wild', for example, booking real flights for the DARPA challenge (Walker et al., 2002) and the Let's Go! shared task (Black et al., 2011). The latter involves real users with real goals, which in the case of Let's Go is to get bus information.

Moving the same type of experiment out of the real world and into the lab comes with a further decrease in realism but an increase in control.

Crowdsourcing platforms such as Amazon Mechanical Turk (AMT) provide an alternative environment to traditional lab-based evaluation. With AMT, participants are paid a small amount of money to perform short tasks. The key advantage of crowdsourcing is the ease of recruitment. With regard to evaluating NLG systems that are embedded, task based experiments have been shown to be effectively evaluated using AMT. For example, Jurčiček et al. (2011) describe a telephone infrastructure for interacting with the system and a web interface for feedback collection. Crowd sourcing platforms provide a simple and cost effective way of evaluating interactive systems and in particular output of NLG systems. The main drawback is that it is hard to monitor the cooperativeness and accuracy of the participants, although initial results from evaluations of participant accuracy are encouraging (Yang et al., 2010, Jurčiček et al., 2011).

Changing the experiment from one in which the user interacts directly with the interactive system to an "overhearer-style" method in which the user observes (part of) an interaction (Stent et al., 2004, Dethlefs et al., 2014) gives even greater control, but less realism as well as higher costs.

Wizard of Oz (WoZ) experiments, where a human generates system responses, provide a powerful mechanism for testing certain NLG strategies, such as different referring expression generation (Janarthanam and Lemon, 2010) and Information Presentation strategies (Liu et al., 2009). The major drawback of WoZ experiments is that WoZ environments are costly to build, and running experiments and training the wizards is very time-consuming. In addition, there may be discrepancies between the different styles of wizards; this is particularly prevalent in NLG WoZ studies. Despite these drawbacks, an important benefit of WoZ experiments is that the human-generated system turns can be treated as model outputs, and having model outputs that facilitate NLG strategy optimisation (Liu et al., 2009, Rieser and Lemon, 2008) is very valuable.

Finally, user simulations are an important component of fine-tuning statistical systems where one can explore a large variety of user responses to different NLG generated utterances (Rieser and Lemon, 2010). However, evaluations using user simulations must be combined with evaluations with real human users.

In order to leverage a greater variety of testbeds, such as evaluations in the wild, some mechanism for system comparison is necessary. Evaluation frameworks for interactive systems go some way towards providing these comparative measures. These frameworks include PARADISE (Walker et al., 2000), Möller and Ward (Möller and Ward, 2008), SERVQUAL (Hartikainen et al., 2004) and Contender (Suendermann et al., 2010) and enable systems to be compared and ranked using single system-level scores derived from multiple evaluation measures that, in combination, capture relevant aspects of system quality.

5. A Methodology for Comparative Evaluation of NLG Components in Interactive Systems

In Table 2, we provide a generic overview of the steps involved in designing an **evaluation model**, i.e. a complete specification of the goals and implementation of a specific evaluation. Note that the evaluation frameworks mentioned above could all be described in these terms.

Point 1 in the table expresses that in order to be able to evaluate multiple versions of a component, its task specification has to be clear enough for developers to work from, and it has to be architecturally separate enough to be plug-and-playable (so that different components can be plugged in and evaluated under the same conditions). Overall evaluative goals (2a) tend to be too abstract to be directly assessed by a set of measures, and it is often found helpful to break goals down into more fine-grained quality criteria (2b). This process could result in a hierarchical structure of more than two levels, with some of the quality criteria decomposed into further quality criteria. For example, in the PARADISE model (Walker et al., 2000), the overall evaluative goal of user satisfaction is taken to be composed of quality criteria of low cost and task success, with low cost further composed of efficiency and quality.

However this is done, at the leaves of the resulting (possibly flat) tree, evaluation criteria have to be paired with individual evaluation measures which return a single score for each system (2c). If multiple measures are used, then a composition model can be created, defining how to combine them into a single score and/or to reflect their relative importance (2d), corresponding e.g. to the value model in SERVQUAL.

6. A Specific Evaluation Model for Comparative Evaluation of NLG Modules in Interactive Systems

In this section, we take the generic model described above and apply it to a specific evaluation for an NLG module embedded in an interactive system. The overall evaluative

	Objective/Task	Example
1. System Design	Achieve clear separation in terms of task specification and software architecture between components that will be evaluated separately and the remaining system.	Component to be evaluated: NLG module; embedding system: flight information system; NLG module performs all and only NLG tasks; module-system interaction specified by API.
2. Evaluation Design	a. Specify the overall evaluative goal of the evaluation, identifying both the system and/or module that is to be evaluated, as well as which (subset of) its properties are to be evaluated.	Goal: to determine which of a set of alternative NLG modules performs better when embedded in a given flight information system; properties to be evaluated: module outputs and contribution to user satisfaction.
	b. Specify the quality criteria which the property to be evaluated breaks down into.	Module output quality: • Linguistic quality of NLG module outputs; • Impact on task effectiveness and efficiency. User satisfaction: • Impact on user satisfaction.
	c. For each quality criterion, select the evaluation measures that will be used to assess it.	Impact on user satisfaction: design questionnaire incorporating *user like measures* in the form of questions such as *Would you be willing to pay to use this system?*
	d. Create a composition model describing how the scores produced by the selected set of evaluation measures are to be combined.	Compute a weight function over the set of scores from the different evaluation measures, such as their weighted sum or weighted mean.

Table 2: Steps in designing an evaluation model: a specification of the goals and implementation of a specific evaluation.

goal (2a) depends on who the stakeholders in the evaluation are, and on how the results will be used. In the present context, we are assuming a developers' perspective, and that we wish to evaluate (just) the NLG component, not the rest of the system. For this last reason, a small number of frozen versions of the embedding system suffice (although it cannot be ruled out that using other versions may produce different results).

To achieve encapsulation of NLG tasks, the embedding system should make available all the required input information, including some or all of the following types, depending on how tasks have been divided between NLG module and embedding system:

- Representations of the user's interaction with the system so far (including e.g. user actions so far, previous dialogue turns, common ground, etc.);
- Representation of the virtual or real environment the user is in if any;
- Representation of current user state (including e.g. most recent user dialogue act, current position and orientation of user, etc.);
- Representation of current and/or remaining options (plans) for how to achieve goal of interaction (e.g. booking a ticket, locating an item, etc.); and
- User model.

The system should send requests to the module to produce a system turn, i.e. unlike in the GIVE-2 Task, all interaction management is performed outside of the NLG module.

A good general-purpose evaluative goal that embodies the developers' perspective is to assess NLG module performance. The following quality criteria can be seen as contributing to this property (also given are the evaluation measure *types* that can be used to assess the criteria):

1. Context-independent intrinsic quality of NLG module outputs: *intrinsic output-quality measures*, both automatic and human-assessed, e.g. measures of grammaticality, fluency and clarity;
2. Context-dependent intrinsic quality of NLG module outputs: *intrinsic output-quality measures*, both automatic and human-assessed, e.g. measures of appropriateness, clarity and efficiency;
3. Contribution to user satisfaction with system: *intrinsic user-like measures*, e.g. asking users questions;
4. Contribution to task effectiveness and efficiency: *extrinsic user task success measures*, e.g. average number of system and user turns in successful interactions, percentage of interactions with a successful task outcome, distance travelled in a virtual world, number of restarts, etc.; and
5. Contribution to system purpose success: *extrinsic system purpose-success measures*; highly system-dependent, but tends to involve both quantitative measures and asking users in some way whether they feel that the system accomplishes its purpose.

7. Conclusion

In this paper, we started with an overview of different categories of evaluation measures, in order to provide a standard terminology for categorising existing and new evaluation techniques. This was followed with some background on existing evaluation methodologies in the NLG and interactive systems fields separately, after which we moved on to presenting a methodology for the evaluation of NLG components embedded within interactive systems, using a specific task as an example.

Interactive systems have become an increasingly important type of application for deployment of NLG technology over recent years. At present, we do not yet have a

commonly agreed terminology or methodology for evaluating NLG within interactive systems. We have attempted to take a step towards addressing this shortfall in this paper.

Acknowledgements

The research leading to this work was funded by the EPSRC (UK) under three consecutive grants (F059760, G03995X, Ho32886), and by the EC FP7 programme FP7/2011-14 under grant agreement no. 287615 (PARLANCE).

References

Ai, H., Raux, A., Bohus, D., Eskenzai, M., and Litman, D. (2008). Comparing spoken dialog corpora collected with recruited subjects versus real users. In *Proceedings of the SIGdial Workshop on Discourse and Dialogue (SIGDIAL)*.

Androutsopoulos, I., Kallonis, S., and Karkaletsis, V. (2005). Exploiting OWL ontologies in the multilingual generation of object description. In *Proceedings of the European Workshop on Natural Language Generation (ENLG)*.

Angeli, G., Liang, P., and Klein, D. (2010). A simple domain-independent probabilistic approach to generation. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

Belz, A. (2007). Probabilistic generation of weather forecast texts. In *Proceedings of Human Language Technologies: The Meeting of the North American Chapter of the Association for Computational Linguistics (HLT-NAACL)*.

Belz, A., Kow, E., Viethen, J., and Gatt, A. (2009). The GREC main subject reference generation challenge 2009: overview and evaluation results. In *Proceedings of the Workshop on Language Generation and Summarisation*.

Belz, A., White, M., Espinosa, D., Kow, E., Hogan, D., and Stent, A. (2011). The first surface realisation shared task: Overview and evaluation results. In *Proceedings of the European Workshop on Natural Language Generation (ENLG)*.

Black, A. W., Burger, S., Conkie, A., Hastie, H., Keizer, S., Lemon, O., Merigaud, N., Parent, G., Schubiner, G., Thomson, B., Williams, J. D., Yu, K., Young, S., and Eskenazi, M. (2011). Spoken dialog challenge 2010: comparison of live and control test results. In *Proceedings of the SIGdial Workshop on Discourse and Dialogue (SIGDIAL)*.

Cox, R., O'Donnell, M., and Oberlander, J. (1999). Dynamic versus static hypermedia in museum education: An evaluation of ILEX, the intelligent labelling explorer. In *Proceedings of the Conference on Artificial Intelligence in Education (AIED)*.

Dale, R. (1989). Cooking up referring expressions. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL)*.

Dale, R. and Kilgarriff, A. (2011). Helping our own: The HOO 2011 pilot shared task. In *Proceedings of the European Workshop on Natural Language Generation (ENLG)*.

Dale, R. and Reiter, E. (1995). Computational interpretation of the Gricean maxims in the generation of referring expressions. *Cognitive Science*, 19(2):233–263.

Dethlefs, N., Cuayahuitl, H., Hastie, H., Rieser, V., and Lemon, O. (2014). Cluster-based prediction of user ratings for stylistic surface realisation. In *Proceedings of the Meeting of the European Chapter of the Association for Computational Linguistics (EACL)*.

Dethlefs, N., Hastie, H., Cuayáhuitl, H., and Lemon, O. (2013). Conditional random fields for responsive surface realisation using global features. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL)*.

Foster, M. E. and White, M. (2005). Assessing the impact of adaptive generation in the COMIC multimodal dialogue system. In *Proceedings of the Workshop on Knowledge and Reasoning in Practical Dialogue Systems (KRPDS)*.

Gatt, A., van der Sluis, I., and van Deemter, K. (2007). Evaluating algorithms for the generation of referring expressions using a balanced corpus. In *Proceedings of the European Workshop on Natural Language Generation (ENLG)*.

Gupta, S. and Stent, A. (2005). Automatic evaluation of referring expression generation using corpora. In *Proceedings of the Workshop on Using Corpora for Natural Language Generation*.

Hartikainen, M., Salonen, E.-P., and Turunen, M. (2004). Subjective evaluation of spoken dialogue systems using SERVQUAL method. In *Proceedings of the International Conference on Spoken Language Processing (ICSLP)*.

Janarthanam, S., Hastie, H., Lemon, O., and Liu, X. (2011). The day after the day after tomorrow? a machine learning approach to adaptive temporal expression generation: training and evaluation with real users. In *Proceedings of the SIGdial Workshop on Discourse and Dialogue (SIGDIAL)*.

Janarthanam, S. and Lemon, O. (2010). Learning to adapt to unknown users: referring expression generation in spoken dialogue systems. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL)*.

Janarthanam, S. and Lemon, O. (2012). A web-based evaluation framework for spatial instruction-giving systems. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL)*.

Jordan, P. W. (2000). Can nominal expressions achieve multiple goals?: an empirical study. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL)*.

Jordan, P. W. and Walker, M. A. (2005). Learning content selection rules for generating object descriptions in dialogue. *Journal of Artificial Intelligence Research*, 24(1):157–194.

Jurčíček, F., Keizer, S., Gašić, M., Mairesse, F., Thomson, B., Yu, K., and Young, S. (2011). Real user evaluation of spoken dialogue systems using Amazon mechanical turk. In *Proceedings of the International Conference on Spoken Language Processing (ICSLP)*.

Koller, A., Striegnitz, K., Gargett, A., Byron, D., Cassell, J., Dale, R., Moore, J., and Oberlander, J. (2010). Report on the second NLG challenge on generating instructions in virtual environments (GIVE-2). In *Proceedings of the International Workshop on Natural Language Generation (INLG)*.

Langner, B. (2010). *Data-driven natural language generation: Making machines talk like humans using natural corpora*. PhD thesis, Language Technologies Institute, School of Computer Science, Carnegie Mellon University.

Liu, X., Rieser, V., and Lemon, O. (2009). A Wizard-of-Oz interface to study information presentation strategies for spoken dialogue systems. In *Proceedings of the Europe-Asia Spoken Dialogue Systems Technology Workshop*.

Möller, S. and Ward, N. G. (2008). A framework for model-based evaluation of spoken dialog systems. In *Proceedings of the SIGdial Workshop on Discourse and Dialogue (SIGDIAL)*.

Rambow, O., Rogati, M., and Walker, M. A. (2001). Evaluating a trainable sentence planner for a spoken dialogue system. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL)*.

Reiter, E. and Belz, A. (2009). An investigation into the validity of some metrics for automatically evaluating NLG systems. *Computational Linguistics*, 35(4):529–558.

Reiter, E., Sripada, S., Hunter, J., and Davy, I. (2005). Choosing words in computer-generated weather forecasts. *Artificial Intelligence*, 167:137–169.

Rieser, V. and Lemon, O. (2008). Learning effective multimodal dialogue strategies from Wizard-of-Oz data: Bootstrapping and evaluation. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics: Human Language Technologies (ACL-HLT)*.

Rieser, V. and Lemon, O. (2010). Natural language generation as planning under uncertainty for spoken dialogue systems. In Krahmer, E. and Theune, M., editors, *Empirical methods in natural language generation*, pages 105–120. Springer-Verlag, Berlin, Heidelberg.

Rieser, V., Lemon, O., and Keizer, S. (2014). Natural language generation as incremental planning under uncertainty: Adaptive information presentation for statistical dialogue systems. *ACM Transactions on Speech and Language Processing*. In press.

Rus, V., Piwek, P., Stoyanchev, S., Wyse, B., Lintean, M., and Moldovan, C. (2011). Question generation shared task and evaluation challenge: Status report. In *Proceedings of the European Workshop on Natural Language Generation (ENLG)*.

Sripada, S. G., Reiter, E., Hunter, J., and Yu, J. (2002). SUMTIME-METEO: A parallel corpus of naturally occurring forecast texts and weather data. Technical Report AUCS/TR0201, Computing Science Department, University of Aberdeen.

Stent, A., Prasad, R., and Walker, M. A. (2004). Trainable sentence planning for complex information presentation in spoken dialog systems. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL)*.

Suendermann, D., Liscombe, J., and Pieraccini, R. (2010). Contender. In *Proceedings of the Spoken Language Technology Conference (SLT)*.

van Deemter, K., Gatt, A., van der Sluis, I., and Power, R. (2012). Generation of referring expressions: Assessing the incremental algorithm. *Cognitive Science*, 36(5):99–836.

van der Sluis, I., Gatt, A., and van Deemter, K. (2007). Evaluating algorithms for the generation of referring expressions: Going beyond toy domains. In *Proceedings of the International Conference on Recent Advances in Natural Language Processing (RANLP)*.

Viethen, J. and Dale, R. (2006). Algorithms for generating referring expressions: do they do what people do? In *Proceedings of the International Workshop on Natural Language Generation (INLG)*.

Walker, M., Rudnicky, A. I., Aberdeen, J., Bratt, E. O., Garofolo, J., Hastie, H., Le, A., Pellom, B., Potamianos, A., Passonneau, R., Prasad, R., Roukos, S., Sanders, G., Seneff, S., and Stallard, D. (2002). Darpa Communicator evaluation: Progress from 2000 to 2001. In *Proceedings of the International Conference on Spoken Language Processing (ICSLP)*.

Walker, M. A., Kamm, C., and Litman, D. (2000). Towards developing general models of usability with PARADISE. *Natural Language Engineering*, 6(3–4):363–377.

Williams, S. and Reiter, E. (2008). Generating basic skills reports for low-skilled readers. *Natural Language Engineering*, 14(4):495–525.

Yang, Z., Li, B., Zhu, Y., King, I., Levow, G., and Meng, H. (2010). Collection of user judgments on spoken dialog system with crowdsourcing. In *Proceedings of the Spoken Language Technology Conference (SLT)*.