An Overview of Natural Language Generation Systems Evaluation

Feng-Jen Yang, *Member, IAENG*

***Abstract*—The evaluation of natural language generation systems has been an eyes-catching topic of research that is continuously challenging researchers in the effort towards achieving the performance and quality of computational linguistic systems. Although this research topic keeps drawing researchers attention, there is still no agreed benchmark on what should be evaluated and how to evaluate them. This paper analyzed and compiled the evaluations of some well-known natural language generation systems and categorized their evaluation methods according their relevancies and similarities.**

***Index Terms*—Natural Language System Evaluation, Natural Language Processing, Computational Linguistics.**

I. Introduction

THE interactions between a user and a natural language generation system are highly dynamic and volatile. Without a careful planning and handling of discourse structures, it is hard to stay on the conversation focus and the dialogue can easily fall into an open-end discussion between the user and the software system. This idiosyncrasy makes it hard to evaluate the performance and the quality of a natural language generation system. Even the comparison of alternative systems in similar domains is virtually impossible [1]. Nonetheless, the evaluation of natural language systems still plays a critical role in guiding and focusing researches in computational linguistics. It continuously challenges researchers in building quality and performance assured linguistic systems.

In the past three decades, some conferences and workshops, such as Message Understanding Conferences (MUCs), Spoken Language Technology Workshops, Machine Translation Workshops, and ACM SIGMETRIC, have been formed with a certain extent of focus on the evaluation of natural language generation systems. Based these conferences and workshops, the following three aspects of evaluations are recommended to evaluate linguistic systems [2]:

1) Adequacy Evaluation: the fitness of a system to its intended purpose is one of the critical factors in bringing natural language systems to market. For potential users, they have to know if the products on offer in a given application domain are suitable for their particular tasks or not. If so, they have to consider further tradeoffs between fitness and cost and then choose the most suitable one.

2) Diagnostic Evaluation: for systems where the coverage is important, the developers or end-users usually construct a large test suite to cover all of the elementary linguistic phenomena and their important combinations in the input domain. By testing systems with a large test suite, they can generate diagnostic profiles. The typical systems using this evaluation are machine translation and natural language understanding systems.

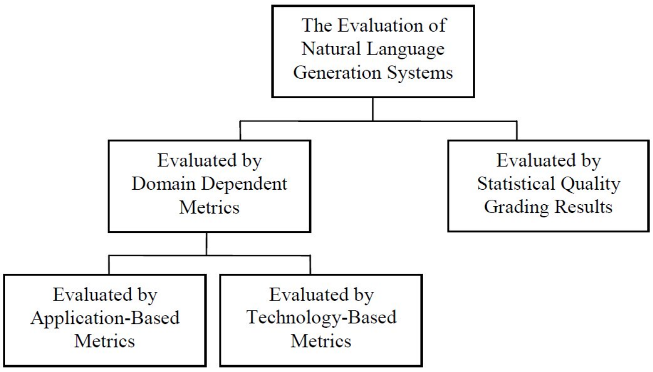
The Evaluation of Natural Language Generation Systems
Evaluated by Domain Dependent Metrics
Evaluated by Statistical Quality Grading Results
Evaluated by Application-Based Metrics
Evaluated by Technology-Based Metrics

Fig. 1. The Categorization of the Natural Language Generation Systems Evaluation

3) Performance Evaluation: Most of the ideas about quantitative performance evaluations are imported from information retrieval. There are three aspects to performance evaluation. The first is Criterion, which addresses what to evaluate such as precision, speed and error rate. The second is Measure, which specifies the property to report in order to get the chosen criterion such as ratio of hits to hits plus misses, seconds to process, and incorrect percentage. The third is Method, which is used to determine the appropriate value for a given measure such as the analysis of system behavior over benchmark tasks. In natural language systems, the approaches provide a useful way for system developers to compare different implementations of a technology or different versions of the same implementation.

II. The Categorization of Evaluation Methods

So far, there is no established standard or benchmark for the evaluation of natural language generation systems. All of the workshops and conferences have just reiterated the importance of evaluation, but failed to reach an agreement on what should be evaluated and how to evaluate them. Although several evaluation methods have been developed in the past three decades, most of them are quite domain dependent and hard to be generalized. For the purpose of reaching a more conclusive categorization, I have analyzed and compared the evaluation methods that were applied in several well-known natural language generation systems and have been able to categorize them into two major categories and two subcategories within the first major category as shown in Fig 1.

The first major category of methods evaluates systems by using domain dependent metrics that can be further subdivided into two subcategories, namely the subcategory

Manuscript received July 10, 2015; revised July 22, 2015.

F. Yang is with the Department of Computer Science and Information Technology, Florida Polytechnic University, FL 33805, USA, e-mail: fyang@flpoly.org.

that evaluates systems by using application-based metrics and the subcategory that evaluates systems by using technology-based metrics. The second major category of methods evaluates systems by using statistical quality results. The qualities of systems are then quantified by the the means and standard deviations. Some evaluation examples of these categories and subcategories are illustrated in the following sections.

III. The Evaluation by Using Domain Dependent Metrics

Generally speaking, the domain dependent evaluations metrics are defined by the project team members and tend to be more suitable for performance-oriented systems in which the throughput of dialogue or text generation is usually a major design criterion. Based on the perspectives from which the throughput is evaluated this category can be further divided into the subcategory of application-based throughput evaluation and technology-based throughput evaluation.

A. The Evaluation by Using Application-Based Metrics

Two good example systems that were evaluated by using application-based metrics are the JUPITER system [3] and the EAGLE project [1].

1) The JUPITER System: The JUPITER system is a telephone-based conversational system used to provide world-wide weather information over the telephone [3]. In the JUPITER domain, the research group proposed the following suite of metrics to evaluate the systems performance in understanding and generating the spoken dialogue between a user and the system [4]:

1) Word/sentence accuracy: this metric is used in evaluating the Speech Recognizer.

2) Parse coverage: this metric is used in evaluating the Parser.

3 Phrase comparisons: this metric is used in the evaluation of Content Understanding and Generation.

4) Understanding score: this metric is used in the evaluation of the Recognizer, Parser and Discourse Planning.

5) Static database assessment: this metric is used in the evaluation of Understanding, Discourse planning, Dialogue, Database Access and Generation.

6) Log file evaluation: this metric is used in the evaluation of Recognition, Understanding, Discourse Planning, Dialogue, Database Access and Generation.

On one hand, this suite of metrics provides a good assessment of the system behavior by examining each query/response pair. On the other hand it also examines the behavior of each part of the system and shows how well each performs separately.

2) The EAGLE Project: The EAGLE project was launched to coordinate the European efforts of both academic and industrial participants toward the creation of de facto standards for corpora, lexicons, speech data, evaluations, and formalisms. As a part of the work of the EAGLE project, the research group proposes a simple and practical reporting framework for spoken dialogue systems. This approach defines three sets of parameters and specifies the range of their possible values [1].

The first set belongs to system metrics that are used to characterize the basic features of the spoken dialogue system to be evaluated, such as:

1) Input type: this parameter characterizes the way users dialogue is input to the system. The possible values are Speech, Text, Pulse and Other.

2) Input vocabulary: the systems overall vocabulary size should be indicated.

3) Input perplexity: the perplexity is a doubt while recognizing the input. This parameter lists the average perplexity of the recognition vocabulary.

4) Output type: this parameter characterizes the systems output to the user. The possible values are Speech, Text and Other.

5) Dialogue type: this parameter indicates the level of dialogue complexity supported by the system. The possible values are Menu, System-Led and Mixed-Initiative.

The second set belongs to test conditions that are used to characterize the basic features of the evaluation exercise, such as:

1) Type of users: this parameter characterizes the kind of users. The possible values are Project, those who involved in designing or building the system, Expert, those who are familiar with the domain and Nave, those who are totally unfamiliar with the domain.

2) Number of users: in general the significance of the results increases with sample size, but counting only the number of dialogues is not an adequate sampling technique. It is important to understand whether the corpus is provided by many people or by a small number of people. This parameter indicates the number of users.

3) Number of dialogues: this parameter records the number of dialogues in the tested corpus. A dialogue is defined as a continuous session of interaction with the system.

4) Number of tasks: this parameter records the number of tasks in the evaluation exercise.

The third set belongs to test results that are used to characterize the basic features of the systems performance collected during the evaluation exercise, such as:

1) Average turns per dialogue: this parameter records the total number of system and user turns in the tested corpus divided by the number of dialogues in the corpus.

2) Average dialogue duration: this parameter is used to describe the average dialogue duration, starting from the beginning of the first utterance to the end of the last utterance.

3) Average turn delay: this parameter is used to describe the average time taken by the system to respond to a user input.

4) Dialogue success rate: this parameter is used to describe the percentage of all dialogues in the corpus where the system either succeeds in correctly satisfying all the users requests or it correctly identifies the fact that the requested tasks cannot be performed.

5) Task success rate: this parameter is used to describe the percentage of all tasks in the corpus where the system either succeeds in correctly satisfying the users tasks or it correctly identifies the fact that the tasks cannot be satisfied.

6 Crash rate: this parameter records the percentage of all dialogues in the corpus where the system fails to complete a dialogue in a coherent manner.

An especially important feature of EAGLEs evaluation worthy of notice here is that it takes the users views and needs into account. This kind of attention has seldom been paid by other systems to the users satisfaction.

B. The Evaluation by Using Technology-Based Metrics

A good example the was evaluated by using technology-based metrics is the TRAIN-96 System [5].

1) The TRAIN-96 System: The TRAINS-96 system was constructed from the TRAINS-95 system by adding distances and times to the train route, allowing users to modify routes, adding robust rules in the parser to prevent incorrect understanding, and adding a template-based post-parser module to handle more domain-specific and less well-defined examples [5].

During the formal evaluation of the TRAINS-95 system, two parameters, time to task completion and quality of the solution, were used to evaluate the general criteria from the task-based perspective. The quality of the solution was measured by whether the stated goals for a task were met, and if so, how much time was taken to complete.

The evaluation of the TRAINS-96 system involved sixteen subjects in a one-hour session with the TRAINS system. Of the sixteen subjects, three were recent college graduates, two were high school students and seven were undergraduates. None of them had experience with the TRAINS systems before. The evaluation used the same five tasks used in evaluating the TRAINS-95 system plus a sixth task for data collection. Each of the first five tasks comes with its own restrictions to simulate different scenarios. In the sixth task the user was given seven trains at different cities and asked to move as many trains as possible to a same destination. After each task the subject was asked to complete a questionnaire and see if the subject had difficulty in completing the task. If so, what caused the difficulty? After completing the final task, the subject completed a more general questionnaire allowing the subject to comment on the system in general [5].

The results of task performance are:

1) Tasks with robustness: the time to completion in four out of five tasks is lower and the length of the route is longer in four out of five tasks.

2 Tasks with speech feedback: the time to completion in four out of five tasks is lower and the length of route is less in three out of five tasks.

3 Tasks with combinations: in two of the tasks the time to completion is lowest with robustness but not speech feedback. In another two the time to completion is lowest with robustness and speech feedback. Overall the best time to completion is obtained when both robustness and speech feedback are used.

The subject questionnaire responses show:

1) With robustness: subjects are less likely to blame the route planner for difficulties.

2) With speech feedback: subjects are more likely to blame the natural language parts of the system for difficulties.

3 Overall: subjects are less likely to blame the route planner then to blame the language understanding parts of the system.

This evaluation showed some preliminary results indicating performance differences with and without the robust parsing rules and speech feedback. The results did match their hypotheses but the small sample size also caused a large variance. An experiment like this should be performed with more subjects.

IV. The Evaluation by Using Statistical Quality Results

For systems in which the quality of system generated text or dialogue is a major design criterion, the evaluation by using statistically results is more suitable. In this evaluation method, the quality of system generated texts or dialogues are graded by their intended users, and then using the means and standard deviations to quantify the quality of a natural language generation systems. The rational of this method is that people tend to agree on what is a good dialogue or text and what is a bad dialogue or text, even if they are not be able to articulate what is good and bad clearly. Two good examples of using this evaluation method is the EBMT project [6].

A. The EBMT Project

System generated examples are commonly used by many natural language generation systems to help users understand the context. As a part of the research on the EBMT project, the language technology research group at Carnegie Mellon University looked into the issues of presenting examples in a useful and effective form using integrated descriptions of text and examples. They identified several critical heuristics in terms of understanding descriptions containing examples. They are descriptions with and without examples, positioning the example, presentation of different example types, complexity and number of examples, and presentation orders of examples. A further verification was shown in an empirical evaluation to see how each heuristic can help in gaining a better understanding of tutorial context [6].

The experiment was conducted by presenting different tutorial descriptions to different groups of participants. Each description takes a heuristic into account, while other descriptions disregard that heuristic on purpose. After reading these descriptions in a limited time, the participants were asked to answer a set of questions designed to measure how much a heuristic can help improve the understanding of that tutorial description. The evaluation showed the following results:

1) Descriptions with and without examples: the usefulness of examples in tutoring context is almost indubitable. The group given a description without examples made between four and eleven mistakes out of twenty one questions with an average of six mistakes. The other group, the group given descriptions containing examples, made between zero and four mistakes out of twenty questions with an average of two mistakes. The result shows that the inclusion of examples does help in understanding a concept.

2) Positioning the example: it is important for examples to be placed in appropriate places whether before the text, within the text or after the text. In the group given interleaved examples, only one person made a mistake out of ten questions. In the group with examples after the description five participants made an average of three mistakes. In the group with the examples before the description, six participants made an average of three mistakes. The participants showed that the best placement for examples is immediately following the point they are supposed to illustrate.

3 Presentation of different example types: the research group categorized the variation of examples in three dimensions, their polarity with respect to the definition they

accompany, the text type for which they are generated, and the knowledge type of which they happen to be instances. The polarity of an example can be positive, negative or anomalous. Anomalous examples are defined as including instances that are examples not covered by the definition. In this experiment, they only consider the difference of presenting anomalous examples together with and apart from the normal examples. In the group given a description with unmarked anomalous examples, all participants got all questions wrong. In the group given a description with marked anomalous examples, only two out the six people got questions wrong. Therefore, it is important to separate anomalous examples from others and present them explicitly.

4) Complexity and number of examples: the complexity and number are two factors working together to help understand a concept. Two descriptions with the same number but different complexity of examples or the same complexity but different number of examples may lead to different extents of understanding. To see the difference, two experiments were conducted. The first experiment tested both the complexity and number of examples. Its results show that in the group given a description with three simple examples, all participants got all ten questions right. In the group given a description with three complex examples, the participants made an average of two mistakes out of ten questions. In the group given a description with only the last example, the participants made an average of 3.25 mistakes out of ten questions. The second experiment was designed to measure the number of examples required. The results showed that giving the participants more than enough examples did not raise the success rate significantly.

5) Presentation orders of examples: it is important that related examples appear in an appropriate sequence. The generation of examples has to take into account associated information such as prompts, background information, and contrasting information. A different sequence of examples will result in a different sequence of associated information. The results show that the group given a description with ordered examples made an average of two mistakes out of ten questions. The group given descriptions with unordered examples made an average of six mistakes out of ten questions. This shows that the ordering of examples is an important factor ensuring the coherence and usefulness of the overall description.

This evaluation leads to a very good reflection of how closely machine-generated descriptions can be matched to texts made by humans. Especially, the idea of testing the efficiency of each heuristic in increasing the users comprehension of a concept is applicable to the evaluation of the turn planner, since the turn planner considers several issues in improving the fluency and coherence of our machine dialogue. It would be useful to conduct an evaluation to see how much the machine dialogue has improved with turn planning.

V. Further Studies

The methodologies of evaluating natural language systems are still evolving. The aforementioned categorization and examples are taken form well-established systems in the literacy of computation linguistics for the sake of providing more persuasive and better representative illustrations of the evaluation methods for natural language systems. Although there is still no standardized benchmarking that most linguistic systems can follow, some other expositions and arguments are worth of reading while choosing the evaluation methods of computational linguistic related systems, such as the task-oriented evaluations of natural language systems [7], the performance evaluations of natural language systems [8], the quantitative evaluation of a large-scale natural language system [9], the need of alignment between system responses and answer key entries in an information extraction system [10], and the black-box and glass-box evaluations of natural language processing systems [11], the evaluation for a natural language-based tutoring system [12], and the evaluation for the pipeline architectures in natural language dialogue systems [13].

References

[1] N.M. Fraser, ”Spoken Dialogue System Evaluation: A First Framework for Reporting Results,” in *Proceedings of the 5th European Conference on Speech Communication and Technology*, Rhodes, Greece, pp. 1907-1910, 1997.

[2] L. Hirschman and H. Thompson, ”Overview of Evaluation in Speech and Natural Language Processing,” in Varile, G., Zampolli, A., Cole, R., Mariani, J., Uszkoreit, H., Zaenen, A., and Zue, V. (editors), *Survey of the State of the Art in Human Language Technology*, Chapter 13.1, Cambridge, UK: Cambridge University Press, 1997.

[3] V. Zhu, S. Seneff, J. Glass, L. Hetherington, E. Hurley, H. Meng, C Pao, J. Polifroni, R. Scholming, and P. Schmid, ”From Interface to Content: Translingual Access and Delivery of On-line Information,” in *Proceedings of the 5th European Conference on Speech Communication and Technology*, Rhodes, Greece, pp. 2227-2230, 1997.

[4] J. Polifroni, S. Seneff, J. Glass, and T.J. Hazen, ”Evaluation Methodology for a Telephone-Based Conversation System,” in *Proceedings of the 1st International Conference on Language Resources and Evaluation*, Paris, France: European Language Resources Association, pp. 43-49, 1998.

[5] A. Stent and J. Allen, ”TRAINS-96 System Evaluation,” *TRAINS Technical Note 97-1*, Department of Computer Science, University of Rochester, Rochester, NY, USA, 1997

[6] V.O. Mittal, *Generating Natural Language Descriptions with Integrated Text and Examples*, Mahwah, NJ: Lawrence Erlbaum Associates, USA, 1999.

[7] B. Sundheim, ”Plans for a task-oriented evaluation of natural language understanding systems,” in *Proceedings of the workshop on Speech and Natural Language (HLT ’89)*, p.p. 197-202, 1989.

[8] G. Guida and G. Mauri, ”Formal Basis for Performance Evaluation of Natural Language Understanding Systems,” *Computational Linguistics*, Volume 10, Issue 1, pp. 15-30, 1984

[9] C. Samuelsson and M. Rayner, ”Quantitative Evaluation of Explanation-Based Learning as an Optimization Tool for a large-scale natural language system,” in *Proceedings of the 12th international joint conference on Artificial intelligence (IJCAI ’91)*, Volume 2, pp. 609-615, 1991.

[10] A. Kehler, J. Bear, and D. Applet, ”The Need for Accurate Alignment in Natural Language System Evaluation,” *Computational Linguistics*, Volume 27, Issue 2, pp. 231-248, 2001.

[11] M. Palmer and T. Finin, ”Workshop on the Evaluation of Natural Language Processing Systems,” *Computational Linguistics*, Volume 16, Issue 3, pp. 175-181, 1990.

[12] M. Chi, K. VanLehn, D., Litman, and P. Jordan, ”An Evaluation of Pedagogical Tutorial Tactics for a Natural Language Tutoring System: a Reinforcement Learning Approach,” *International Journal of Artificial Intelligence in Education*, Volume 21 Issue 1-2, pp. 83-113, 2011.

[13] E. Margaretha and D. DeVault, ”An Approach to the Automated Evaluation of Pipeline Architectures in Natural Language Dialogue Systems,” in *Proceedings of the SIGDIAL 2011 Conference, (SIGDIAL ’11)*, PP. 279-285, 2011.