Towards the Evaluation of Referring Expression Generation

Jette Viethen

Centre for Language Technology
Macquarie University
Sydney, Australia
{jviethen,rdale}@ics.mq.edu.au

Robert Dale

Motivation

Many research communities within Natural Language Processing have adopted common evaluation metrics (e.g. Bleu, Rouge) and shared evaluation tasks (e.g. DUC, TREC) to compare systems.

For Natural Language Generation (NLG) systems no such comparative metrics or tasks exist.

The NLG community is currently engaged in discussion as to whether and how to introduce shared evaluation.

Referring expression generation with its widely agreed problem definition is a good candidate for piloting shared task evaluation in NLG.

We conducted an evaluation experiment (Viethen and Dale 2006) to investigate the issues arising in evaluating referring expression generation (REG) algorithms against a corpus of natural data.

The Algorithms

We chose the following three algorithms for our evaluation experiment:

Full Brevity (FB) — Dale 1989

- Aims at finding a minimal referring expression.
- In each step, it selects the property that eliminates *most* distractors.

Incremental Algorithm (IA) — Dale and Reiter 1995

- Permits limited redundancy.
- In each step, it selects the next property from a predefined ordered set, if it eliminates *any* distractors.

Relational Algorithm (REL) — Dale and Haddock 1991

- Aims at finding a minimal referring expression like FB.
- Uses a constraint network to allow use of relations between objects.

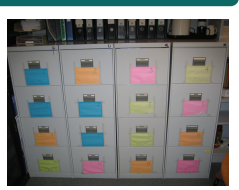
An Evaluation Experiment

The Domain

A grid of 4 × 4 filing cabinet drawers.

Each drawer has a number between 1 and 16.

4 drawers each are blue, yellow, pink and orange.



The Task

Given the number of a drawer, describe it to an onlooker without mentioning any of the numbers.

The Data Set

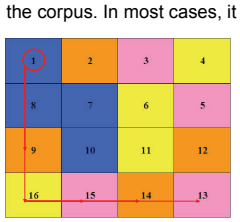
- 118 human-produced referring expressions, of which:
  - 89 are minimal descriptions.
  - 29 are descriptions using redundancy.
  - 15 are descriptions using relations between drawers.

Can the algorithms replicate the corpus data?

- Coverage

Algorithm \ Description type	Overall	Minimal	Redundant
FB	79.6% (103)	100% (79)	31.0% (24)
IA	95.1% (103)	100% (79)	82.8% (24)
REL	0% (118)	0% (89)	0% (29)

- FB and IA achieve full coverage of the minimal, non-relational referring expressions in the data set.
- IA also performs very well with respect to redundant, non-relational expressions.
- However, REL does not generate any of the referring expressions from the corpus. In most cases, it does a "corner run":



- Describing Drawer1
- REL's output:

  the object above the object
  above the object above the object
  left of the object left of the object
  left of an object

- Distinguishing! But useful?

For more detailed information on the results of the experiment, see (Viethen & Dale 2006).

Issue #1: Deciding on Input Representation

> Natural Language *Understanding* is like counting from 1 to infinity,
> Natural Language *Generation* is like counting from infinity to 1.
> — YORICK WILKS

Unlike in NLU tasks, the input for NLG systems is *not* well-defined.

For our experiment, we had to take a number of decisions about the implementation of object properties in our domain.

For example: "The drawer in the top right corner"

- A relation between the drawer and the corner?
- A position property with the value 'corner'?
- Inferred from the position information 'top left'?

We chose a representation knowing how the algorithms work.

In REG and most other NLG tasks, the design of the underlying knowledge base and that of the algorithm are tightly intertwined. This poses a great problem for comparative evaluation.

Issue #2: Dealing with Determinism

> NLG algorithms are deterministic — natural language is not!

For one object, a given REG algorithm will always deliver the same referring expression that is deemed most appropriate.

However, the one 'best' referring expression for an object does not exist. Even the same person will describe the same drawer in different ways on different occasions. For example:

"The orange drawer in the first row."
"The topmost orange drawer."
"The orange drawer in the top row, second from the left."

Therefore, no corpus of natural language samples can be guaranteed to contain all the possible 'right' expressions. This means that we cannot penalise an algorithm if its output does not appear in the corpus.

Comparison against a gold standard might not work for NLG.

Issue #3: Measuring Performance

Most REG and NLG systems use certain parameters to model preferences in different domains or other environmental factors.

IA explicitly encodes a *preference ordering* in which it considers the properties. FB and REL both require a decision when several properties eliminate the same amount of distractors and a preference ordering is a straightforward solution for this.

Preference ordering	*Output of IA for Drawer1*
(row column colour corner) →	"The drawer in the first row, first column."
(colour column row corner) →	"The blue drawer in the first column, first row."

We aggregated the results for all preference orderings to achieve a fairer comparison of the algorithm output to the data given by the human participants.

Allowing multiple results lets us examine whether the systems are capable of producing the human data under any circumstances.

However, it is then unclear what conventional numeric metrics, such as precision and recall, exactly mean.

Also, the evaluation still hinges on the quality of the gold standard.

Conclusions

Evaluation in NLG is a hard problem with a multitude of complex issues.

The inherently different nature of NLG and NLU tasks means that there is no simple way to adopt evaluation models from NLU.

Before jumping on the evaluation bandwagon, the NLG community needs to devote a lot of effort to gain a better understanding of these issues:

- How can we agree on input representations for a common task?
- What counts as a gold standard when there is no one correct answer?
- How do we interpret the numeric measures?

References

Dale, R. (1989), Cooking up referring expressions. In *Proceedings of the 27th ACL*, 68–75, Vancouver, British Columbia.

Dale, R. and Haddock, N. (1991), Generating referring expressions involving relations. In *Proceedings of the 5th EACL*, 161–166, Berlin, Germany.

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

Viethen, J. and Dale, R. (2006). Algorithms for generating referring expressions: Do they do what people do? In *Proceedings of the Fourth International Natural Language Generation Conference*, 63–70, Sydney, Australia.