

In the Encyclopedia of Linguistics, 2 vols., Philipp Strazny (ed.), Fitzroy Dearborn, New York, 2005

Computational Linguistics – Generation

David D. McDonald

Generation is the process by which thought is rendered into language. Within computational linguistics, it is referred to as ‘natural language generation’ (NLG) to help distinguish it from Chomsky’s generative grammar and to contrast with natural language understanding. It is the study of how actual speakers, people or computers, construct utterances in actual contexts – the situations that motivate them to speak. As such, NLG is part of the larger field of Cognitive Science, where it is also referred to as ‘production’, particularly by psycholinguists.

The first NLG systems were developed in the 1950s as part of machine translation systems. The field gained maturity in the 1980s, with its own series of workshops and conferences and its own unique problems. As gauged by membership in the special interest group SIGGEN (see www.aclweb.org) there are several hundred people today actively pursuing research on generation.

NLG takes its methodology from Artificial Intelligence. To study a cognitive capability you design and implement computer programs that attempt to replicate it: in this instance to produce fluent utterances for a purpose. To do this the program (‘generator’) starts with a body of data or information. This might be the daily movement of a stock market index (Kukich 1988) or numerical data about temperature and wind patterns (Goldberg et al. 1994). In cases like these the first thing the generator must do is analyze the data to determine both what information it contains (e.g. what is particularly salient: have winds increased or diminished) and what concepts – ultimately what words or phrases – could be used to communicate that information.

There is a consensus among NLG researchers that generation involves three broad-brush components: (1) determining and organizing the information content to be expressed; (2) ‘microplanning’, where the sentential and referential structures are determined; and (3) ‘surface realization’ where the text plan is processed by a grammar to construct the sequence of syntactically and morphologically appropriate word forms, which are then rendered on some output medium, typically formatted text displays or web pages. (For details of alternative NLG architectures see McDonald 2000 or Reiter and Dale 2000.)

Surface realization is the most advanced of these three components since it draws on the well-established knowledge of grammar in linguistics and computational linguistics as a whole. Virtually every kind of grammar that linguists have developed has been applied to NLG including some that are relatively unknown in the wider community such as Melcuk’s Meaning-Text Theory and Halliday’s Systemic Functional Grammar (SFL). Interest in these theories of grammar stems from the need to reason about the alternative choices that are available, the functions they perform, and their consequences for other choices later on. For example, saying “*the house is red*” vs. “*the red house*” will express the same fact about the house, but with differences in emphasis and in what parts of the sentence they leave open for other information to fill. Surface realization is also the only component within the architecture of NLG systems that is sufficiently mature for ‘plug and play’ reusable components to have emerged, notably KPML (Bateman 1997) and FUF/SURGE (Elhadad and Robin in press) both of which use SFL.

The state of the art in NLG is measured by a combination of the fluency of the texts it is possible to produce and comparative difficulty of adapting to new subjects or genres.

Consider this example, an automatically generated recipe for butter bean soup (Dale 1992, pg. 14).

“Soak, drain and rinse the butter beans. Peel and chop the onion. Peel and chop the potato. Scrape and chop the carrots. Slice the celery. Melt the butter. Add the vegetables. Sauté them. Add the butter beans, the stock and the milk. Simmer. Liquidize the soup. Stir in the cream. Add the seasonings. Reheat.”

Notice how this text is tailored to its genre. All the sentences are imperatives; objects can be omitted when they are obvious (“*Reheat* ___”), and the sentences are simple and short. One of the problems in microplanning is how to formulate this ‘tactical’ knowledge about a genre’s preferred constructions in such a way that it can be deployed by other NLG systems producing texts about a different subject, especially when they employ different processing methodologies. We know how to do this with grammars, but not with knowledge of how to balance the consequences of alternatives when a text is composed.

This class of problems, unique to generation, is further illustrated with the example below, which was produced by Robin’s STREAK system (1993). It is an example of the best that can be done today as it is indistinguishable what a human sports journalist would produce as a capsule summary of a basketball game.

“Dallas, TX – Charles Barkley matched his season record with 42 points Friday night as the Phoenix Suns routed the Dallas Mavericks 123 - 97.”

STREAK uses a architecture based on revisions to an initial draft, where it continually looks for opportunities to incorporate historical knowledge into a skeleton of reported facts. In this instance, for example, there is the fact that the Mavericks are on a long losing streak. It is not possible to add the number of loses to the sentence given its present structure, however this generator has extensive tactical knowledge about the choices available to it and knows that if it uses an alternative way of phrasing the fact that the Suns lost, one that reifies the loss as a noun, it can then incorporate the number of losses by modifying the noun with the count. Underlines indicate the text that has changed:

“. . . the Phoenix Suns handled the Dallas Mavericks their 27th defeat in a row at home 123 - 97.”

The focus of ongoing research is in text planning problems such as this illustrates; in the extension of established capabilities to larger texts (as this is written the limit is multiple page, individually tailored instruction or advice pamphlets); and in integration with other modalities such as real-time graphics and the production of speech with appropriate prosodics.

References

- Bateman, John, “Enabling technology for multilingual natural language generation”, *Natural Language Engineering* (3)15-35 (1997)
- Dale, Robert, *Generating Referring Expressions*, Cambridge, Massachusetts: MIT Press, 1992
- Elhadad, Michael, Jacques Robin, SURGE: a Comprehensive Plug-in Syntactic Realization Component for Text Generation, *Computational Linguistics*, in press, 2001
- Goldberg, Eli, Norbert Driedger, and Richard Kittredge, “Using natural language processing to produce weather forecasts,” *IEEE Expert* 9(2) (1994)
- Kukich, Karen “Fluency in Natural Language Reports,” in *Natural Language Generation Systems*, edited by David McDonald and Leonard Bolc, New York: Springer-Verlag, 1988

- Levelt, Willem J.M., *Speaking: From Intention to Articulation*, Cambridge, Massachusetts: MIT Press, 1989
- McDonald, David, "Natural Language Generation," in *Handbook of Natural Language Processing*, edited by Robert Dale, Herman Moisi, and Harold Somers, New York: Marcel Decker, 2000
- Reiter, Ehud, and Robert Dale, *Building Natural Language Generation Systems*, Cambridge, U.K.: Cambridge University Press, 2000
- Robin, Jacques, "A revision-based generation architecture for reporting facts in their historical context," in *New Concepts in Natural Language Generation: Planning, Realization and Systems*, edited by Helmut Horacek and Michael Zock, London: Pinter, 1993