Formal Issues in Natural Language Generation

Kees van Deemter *, Ehud Reiter *, and Helmut Horacek †

(*) Dept. of Computing Science, University of Aberdeen, UK
(†) Universität des Saarlandes, Saarbrücken, Germany

ABSTRACT:

Natural Language Generation (NLG) can be studied both empirically and formally. In recent times empirical research has tended to dominate the NLG research community, especially with the emergence of corpus and statistical techniques. However, formal research, based on proofs, formal models, and formal linguistics, can also contribute to the NLG research agenda. In this paper we discuss what types of formal research are relevant for NLG, and we introduce the four papers in the Special Issue on Formal Issues in Natural Language Generation of the journal Research on Language and Computation *in the light of this discussion.*

KEYWORDS: *NLG, Meaning, Logical Equivalence, Proof*

1 Introduction

To some readers, the title of this paper, ‘Formal Issues in Natural Language Generation’ (which is also the theme of this Special Issue) might sound logically redundant: Natural Language Generation (NLG) is commonly understood as the pursuit of *computational* models of language production, so surely, NLG must be formal from start to finish; from this perspective, one might be forgiven for thinking that the only issues in NLG are formal ones.

This perspective, however, is probably not the prevalent one in the NLG community. As is true for most other areas of Computational Linguistics, recent years have seen an increasing emphasis on the empirical underpinnings of NLG: linguistic corpora are studied, and experiments with human subjects are performed, to learn how human speakers/writers choose between the many ways in which a given message may be put into words, and the outcomes of this work inform NLG algorithms which aim to make the expressive choices that resemble the choices that humans would make, or the ones most appreciated by human hearers/readers. This new empirical emphasis is of course beneficial, and probably a sign of maturity: the issue of expressive choice, after all, can only come to the fore when generators are able to express their inputs in more than one acceptable way.

This Special Issue, however, is (perhaps unfashionably) concerned with NLG’s other, more formal face. Ideally, we believe that empirical adequacy and formal precision ought to go hand in hand. In many cases, it is precisely when NLG systems become more expressive, and empirically more accurate, that formal methods have most to offer. Consider the sub-area of NLG, for example, where *referring expressions* are generated on the basis of properties in a Knowledge Base shared (i.e. mutually known) by speaker and hearer: as long as the algorithms do little more than conjoin atomic properties, then it is easy to see whether they are logically complete (i.e., whether they generate a distinguishing description whenever such a description exists); but when relations are included (e.g., ‘the bowl on the table’), or negations and disjunctions (e.g., ‘the porcelain dishes and the bowls that are not Chinese’) this becomes harder. When further constraints need to be met, then formal methods come into their own. This is the case, for example, if descriptions are required to be minimal in ‘size’ (e.g., as measured in terms of the number of atomic properties occurring in them). Important claims like the following become nontrivial to prove:

- The algorithm terminates under all inputs
- The algorithm is logically complete
- The algorithm fulfils all constraints
- The algorithm is computationally tractable

We believe that, ultimately, formal issues of this kind deserve a great deal of attention, or else the systems that are built are at risk of failing. (Worse even, they will fail in ways that we are unable to predict, since the limitations of the systems involved will not be understood.) This is not to say that all the papers in this Special Issue address all of these specific topics directly, but by focussing on formalisms whose properties are relatively well understood, we believe they are making significant contributions in this direction.

The history of this Special Issue is as follows. In April 2003, the 9th European Workshop on Natural Language Generation (ENLG-2003) took place in Budapest, organised by the editors of this Special Issue [9]. A new Call for Papers was distributed among those who had been involved with ENLG-2003, asking for submissions on Formal Issues in NLG. Another rigorous review process was then used to select four of the papers that were subsequently submitted, and these papers form the core of this Special Issue. We thank both sets of reviewers for their help.

This Introduction is organised as follows: we start, in section 2, by focussing on formal issues to do with the input to an NLG system or module. Then, in section 3, we discuss some formal aspects of processing (i.e., to do which how a given input is treated by the generator). Finally, in section 4, we highlight some formal issues that have not taken centerstage in the remainder of this Issue, paying particular attention to the role of inference and proofs.

2 Formal Inputs

A prerequisite to formal analysis of any kind is that the inputs to the NLG system be in a well-defined formalism. Although the input to the system can be virtually anything, generators typically start by producing some kind of logical form, through application-specific interpretation of non-linguistic data; this logical form constitutes the input specification for linguistically-oriented processing. If the logic of these specifications is well understood then this makes it possible to formally reason about the inputs; this can in some cases aid the construction of NLG algorithms or systems. Michael Minock's paper, for example, describes an NLG system whose input is a relational database and a query in a language based on relational calculus, both of which are formally defined [7]. Key properties of the query language are known and can be reasoned about, and indeed Minock formally proves theorems about the query language. For example, it is decidable whether there can be an object that satisfies the query, and it is decidable whether one query is a relaxation of another. Properties of this kind are then utilised to detect misconceptions that the user may have concerning the database, and to generate cooperative responses.

Another example is Michael White's paper [13]. White focuses on linguistic realisation, the last stage of the NLG process. Studying linguistic realisation is difficult because realisation relies for its input on a large number of modules earlier in the pipeline. Realisation can only get off the ground once it allows itself to make highly detailed assumptions about its input. White's approach works in two stages: First, he defines a fragment of Hybrid Logic Dependency Semantics (HLDS, based on [2]); then he defines a translation function from HLDS to expressions of Discourse Representation Theory (DRT). Since the semantics of DRT expressions is well-understood, in terms of their truth and in terms of the relation of logical consequence between them (e.g. [8]), this indirectly establishes that HLDS itself is also well understood.

3 Formal Processing

Formal techniques and analysis can also be applied to the processing performed by the NLG system. In particular, we can use formal declarative representations to drive and control what the NLG system does; and the NLG system can incorporate or integrate with a formal reasoning engine, such as a theorem prover.

Advaith Siddharthan's paper illustrates the first idea, of declarative specifications of functionality [11]. Siddharthan's goal is to re-express texts using simple and easy to understand language, in part to make them more accessible to readers with limited literacy. In order to do this. Siddharthan's system needs knowledge about what transformations reduce the linguistic complexity of texts, and where possible he encodes this knowledge declaratively, as soft and hard constraints. One advantage of such formal declarative representations is that they make it easier to relate what Siddharthan's system does to formal linguistic and psycholinguistic theories, such as centering.

Helmut Horacek's paper illustrates the second idea, of integrating an NLG system with a formal reasoning engine [6]. In Horacek's case the engine is a theorem prover which provides the input to Horacek's system, which generates an English presentation of a proof created by the theorem prover. Horacek's paper makes clear that presenting and explaining a formal artefact (such as a proof) in English is not a straightforward task: a considerable amount of presentation optimisation and massaging is needed to make the proof understandable to readers.

A difficult theoretical issue to do with processing is whether and how to let a generator respect logical equivalences between different inputs. Many NLG systems use logic-based input languages, which can encode the same meaning in different ways; for example $dog(X) \wedge run(X)$ versus $run(X) \wedge dog(X)$. Should an NLG attempt to always generate the same output (or the same sets of possible outputs) from logically equivalent inputs, as would be dictated by a strict view of what these inputs mean? This is not easy to do, especially given that determining logical equivalence is computationally undecidable in many cases ([1], [10]). This issue, which is always lurking in the background of papers on NLG, is specifically discussed in the contributions by Minock and Horacek.

4 Other Aspects of Formal NLG

The four papers in this special issue do not illustrate all aspects of formal NLG (one would need many more than four papers for this!). In this section we briefly discuss some of these other aspects of formal NLG, focussing in particular on some issues that have not been stressed in the literature as much as we believe they deserve.

Ultimately, formal models and methods should allow us to mathematically prove properties of NLG systems. Proofs can be about functionality; that is, we can try to prove that the output of an NLG algorithm is guaranteed to satisfy some properties or obey some constraints, provided that the input to the algorithm satisfies some properties or constraints. There is of course a large community of theoretical computer science researchers working on proving things about computational algorithms and models (e.g. [4]). An example of proofs in NLG is [12], which proved that an algorithm for generating certain types of referring expressions would successfully find an appropriate referring expression, provided that it was possible to construct such a referring expression (which is a constraint on the input to the algorithms). Without a proof of this kind, a generator could suddenly (and unnecessarily) fail in a situation that the designer had not anticipated.

Proofs can also be about non-functional properties, such as computational complexity; that is, we can try to prove that a particular algorithm always runs fairly quickly, or conversely that we expect it to be very slow in the worst case. Proofs of this kind can also be made about models; that is, one can try to show that any algorithm that attempts to generate a text which meets some conditions will probably take a lot of computational time in the worst case. For example, Dale and Reiter showed that a certain formulation of the problem of generating referring expressions was NP-complete, and hence any algorithm which generates appropriate referring expressions under this model will probably have very poor worst-case performance [3]. Insights of this kind are vital, for example, if we are to obtain an understanding of the practical applicability of the algorithm (e.g., in a real-time setting), but also of the extent to which the behaviour of the algorithm resembles that of humans when they speak or write.

One final formal issue which we shall mention has to do with proofs in a different way, because they involve the inferences that the hearer or reader of a text might make. Some of these inferences are logical deductions similar to those performed by a theorem prover, and these inferences can be relevant, for example, to predict whether a 'linguistic' ambiguity in a generated sentence can be disambiguated by a reader/hearer who has certain background knowledge. Other, equally important inferences are triggered by pragmatic rules such as conversational implicature [5]. For example, instead of letting a generator say *Please close the window*, one might let it say *It is cold here*. Conversely, a generator whose input would be directly expressed as *It is cold here* should ideally be aware that this can be taken as a request to close the window. To what degree is it possible and sensible for an NLG system to predict and also exploit these inferences? This is an issue that is much underrepresented in the field. A reasonable inferential competence on behalf of NLG systems is especially important in connection with inference-rich discourse, or when the inferences are incorrect and could mislead the reader, or in applications where politeness is a requirement.

5 Conclusion

Computational Linguistics (perhaps fortunately) does not have a rigourous *canon* concerning the way in which research must be conducted or papers structured. (Compare, for example, experimental psychology, where there exists broad concensus concerning the types of experiments that are done, the way in which the data are analysed, and the manner in which the results are reported.) However, over the past 5-10 years there has been a growing emphasis on empirical methods, typically based on corpora or experiments with human subjects. Empirical approaches to NLG of course have many strengths and are extremely valuable, but at the same time we believe that Computational Linguistics would suffer if formal approaches were completely abandoned in the rush towards empirical techniques and practical applications.

We would like to advocate that increased empirical awareness goes hand in hand with a renewed *formal* awareness, which focusses on the types of concerns that were outlined at the start of this Introduction. While it would be unrealistic to expect every paper in NLG to make both formal and empirical contributions, we hope that all NLG researchers are at least aware of the formal perspective on NLG, and that the papers in this special issue will help to illustrate to all members of the NLG community examples of modern formal NLG research. We hope that this Special Issue will also help to reinforce emerging connections between computational linguistics (more specifically on NLG) on the one hand, and formal/computational semantics (one of the target audiences of Research on Language and Computation), on the other.

References

[1] D.E.Appelt 1987. Bidirectional grammar and the design of natural language generation systems. In Procs. of TINLAP-3. Las Cruces, NM. pp. 185-191. Reprinted (1989) in Y.Wilks (Ed.) *Theoretical Issues in Natural Language Processing*, pp. 199-205. Lawrence Erlbaum Associates.

[2] J.Baldridge and Kruijff,G.-J. 2002. Multi-modal combinatory categorial grammar. In Proc. of 10th Annual Meeting of the European Assoc. for Computational Linguistics.

[3] R.Dale and Reiter, E. 1995. Computational Interpretations of the Gricean Maxims in the generation of referring expressions. Cognitive Science 19: 233-263.

[4] M.R.Garey and Johnson, D.S. 1979. *Computers and Intractability: A guide to the Theory of NP-Completeness.* Freeman and Company, New York.

[5] P.Grice 1975. Logic and Conversation. In: P.Cole and J.Morgan (Eds.) *Syntax and Semantics* (3): Speech Acts. new York, Academic Press, pp. 41-58.

[6] H.Horacek 2004. Handling dependencies in reorganizing content specifications - A case study of case analysis. *Research on Language and Computation* (This issue.)

[7] M.Minock 2004. Modular generation of relational query paraphrases. *Research on Language and Computation* (This issue.)

[8] R.Muskens, J.van Benthem, and Visser, A. 1997. Dynamics. In J.van Benthem and A.ter Meulen, Handbook of Logic and Language. Elsevier and The MIT Press.

[9] Reiter, E., Horacek, H., van Deemter, K. 2003. Proceedings of the 9th European Workshop on Nat-

ural Language Generation (ENLG-2003), Budapest. In accociation with the 11th Conference of the European Chapter of the Association for Computational Linguistics, EACL-2003.

[10] S.Shieber 1994. The problem of logical-form equivalence. Computational Linguistics, 19(1): 179-190.

[11] A.Siddharthan 2004. Syntactic simplification and text cohesion. *Research on Language and Computation* (This issue.)

[12] K. van Deemter 2002. Generating Referring Expressions: Boolean Extensions of the Incremental Algorithm *Computational Linguistics* **28** (1), pp. 37-52. March 2002.

[13] M.White 2004. Efficient realization of coordinate structures in combinatory categorial grammar. *Research on Language and Computation* (This issue.)

Received November 2004