Suregen-2: a shell system for the generation of clinical documents

Dirk Hüske-Kraus
University for Health Informatics and Technology Tyrol (UMIT)
Institute for Health Information Systems
postal address: Georgstrasse 11, 32545 Bad Oeynhausen, Germany
Dirk.Hueske-Kraus@suregen.de

Abstract

Suregen-2 applications are intended for use as add-on modules for clinical information systems. Currently, Suregen-2 permits refinement of the predefined medical ontology, specification of text plans and description knowledge for objects of the ontology. It has built-in constructs for referential expressions, aggregation, enumeration and recurrent semantic constellations. A first application built with Suregen-2, which currently supports German only, is in routine use.

1 Introduction

In 2000 we were faced with the situation that our existing systems for generating medical findings had a very high user acceptance but, being programmed conventionally, had become almost unmaintainable. We needed a system linking to existing clinical information systems and generating appropriate (German) text for findings, procedure reports and referral letters. The text should be as close as possible to conventionally dictated text with regard to used lexemes, common formulations, conciseness and even the stylistic preferences of individual physicians. The system should be flexible enough to keep pace with the rapid progress of medical technology in a university hospital. We performed a text-linguistic analysis of medical documents (outlined in (Hüske-Kraus, 2003)) identifying additional requirements for our system: It should incorporate an extendible ontology of clinical medicine, i.e. not only signs, symptoms, diseases and bodily locations but also rather elaborate process structures of clinical actions, and even seemingly trivial items such as syringes, drugs or persons (medical personnel, patients' relatives). It should also be able to handle referential expressions, especially anaphora and deicticals, aggregation of various kinds and ellipsis, and the quite frequent "enumerative expressions". Due to the vital importance of the generated text, the system designer should have maximum control over lexical choice, sentence formation and supra-sentential structures. In order to give a user instant feedback of the resulting text, the generation should be done in under one second.

With these criteria in mind we undertook an extensive review of the literature (see (Hüske-Kraus, 2003)) but, not surprisingly, such a system was nowhere to be found. So we decided to build one in Allegro CommonLisp, using insights and, wherever possible, techniques from the NLG domain and in so doing deliberately valuing practicability over algorithmic elegance or scientific generality.

The Suregen-2 kernel consists of several modules described below, organised in a pipeline architecture (Reiter and Dale, 1997).

2 The ontology

Suregen-2 already has a base ontology of medicine which may be augmented with concepts of the respective application domain. One of the key concepts behind Suregen-2 is the idea of defining description knowledge (via `ToDescribe`) at the class level in the ontology. In an ideal situation it would therefore be sufficient to define the concepts of the domain as subclasses of the Suregen-2 base classes in order to build a new application. In practise, however, it will no doubt be necessary to tailor some of the descriptions to specific application requirements. As the description clauses for a class may contain references to other objects' descriptions (DescribeS), complex texts may be built up from the description of a single object. The text plan could even consist of a single (complex) object's description. Please note that the ontology is not merely a means of organizing description knowledge but also of conveying implicit information used for generation.

3 The generation algorithm

The algorithm essentially substitutes occurrences of DescribeS in the text plan with the corresponding description clauses, then inserts pointers for nominal references, and evaluates the text plan. In the next step calls to Aggregate are evaluated, referential pointers are resolved and word and phrase order is modified. In these successive modifications to the text plan a constituent tree is built up where each node may hold information regarding syntactical form, syntactical role and base lexemes of the subtree.

4 Description templates

The function ToDescribe is used to define description templates for classes. There may be several templates for a given class, varying with regard to the (syntactical) form of generation result, the style to which the description belongs and the facet of the object to be described. Consider the class CAD, inheriting from its superclass SuregenDiseaseEntity the attributes degree and affectedBodyStructures

```
(ToDescribe :a CAD :as :noun
   :use (Case my :degree)
          (:unspecified "CAD")
          (1 "single vessel disease")
          (2 "two vessel disease")
          (3 "three vessel disease")))
```

Wherever a description of an instance of CAD as a noun is now referred to, the above clause (after :use) is inserted. Consider now:

```
(ToDescribe :a CAD :as :NP
   :use (Noun-phrase
      :noun (DescribeS it :as :noun)
      :adjective "coronary"
      :attribute (Parenthesized
   (DescribeS (affectedBodyStructures it)
      :its :stenoses :as :NP)))
```

This illustrates how to describe CAD instances as noun phrases making use of the previous description (as a noun) and the description of the :stenoses-facet of the affected body structures. The :use-clause may contain an arbitrary mix of "canned text" strings, calls to Suregen-2 utilities, descriptions of other objects and calls to morphosyntactic functions. Designed to be configurable by trained end users, Suregen-2 offers elementary grammatical constructs with sensible defaults:

```
(Main-Clause :subject "heart"
   :predicate "to beat")
```

gives "heart beats" since :singular, 3 (:person) :present (:tempus), :active and :indicative are defaults. Suregen-2 offers inflection of lexemes, complex phrases clauses and supraclausal or suprasentential constructs.

5 Semantic functions

In the building of the first tentative applications with Suregen-2 it became obvious that certain constellations tend to reoccur frequently in the intended domains. For instance, the quantification of a medical parameter is very common and so is the description of a certain pathological condition at a certain body location. To avoid defining structurally identical descriptions of, say, body temperature, heart rate or LVEDD, Suregen-2 offers the possibility of defining the constellation "parameter with value, dimension and unit" as a semantic function, a kind of "macro"-construct." Unlike other approaches (e.g. TECHDOC (Stede, 1999)) these functions do not operate on relations specified declaratively in a formalism such as LOOM (MacGregor and Bates, 1987) but operate on instances' attributes. This surely trades flexibility and expressivity of a standard formalism for simplicity and performance.

6 Aggregation

As has already been observed (Shaw, 1998a), the sublanguage of clinical medicine uses much aggregation. Suregen-2 therefore supports two types of aggregation, conjunction reduction ("segregatory coordination" (Shaw, 1998b)) and conceptual aggregation (Wilkinson, 1995). The former is mainly used to aggregate noun phrases with identical or coreferential components. Conceptual aggregation, in our case paraphrased as "finding a single term for a collection of terms", is necessarily connected with relations holding in the domain. The most common case of conceptual aggregation is that from subparts of the body to the encompassing body structure. Unfortunately aggregation along relations in the ontology is not trivial, since not all relations are transitive along the part-of relation (Bernauer, 1996). Moreover, speech habits in medicine permit aggregation even where it is unjustified by part-whole reasoning. To account for this, Suregen-2 offers the possibility of defining "aggregator objects", in the above example an object which performs the possible aggregations given the presence or absence of a certain condition for the left and right atria and ventricles. The necessity of defining ad-hoc aggregator objects is somewhat unsatisfactory but it is due to idiosyncrasies of the medical sublanguage rather than to the chosen approach.

7 Lexical choice

The problem of lexical choice, for which there are plenty of approaches (Cahill, 1999; Edmonds, 1997; Elhadad, 1996; Nogier and Zock, 1992), has more been circumnavigated than solved in Suregen-2. Firstly, there are many proper names in the Suregen-2 target application domain, standardized in the UMLS (McCray and Nelson, 1995) and a large body of names and acronyms common to a medical sub-specialty. Together with the lexemes which are "hard coded" in the `DescribeS`-clauses all these have one-to-one associations of concepts to lexemes. Additionally, using a nanofunctionality of lexical semantics, Suregen-2 is able to deal with synonyms, antonyms and hyperonyms, permitting the aggregation of two adjacent NPs such as "Einbringen des Kameratrokars" and "Einführen der Optik", with different but synonymous heads.

The process for the production of referential expressions is based on a quite simple algorithm which is nevertheless sufficient for clinical documents. Due to the rather clear-cut thematic-rhematic structure, simple pronouns and an occasional reference using a hyperonym for the object in focus is appropriate. Lexical choice is also involved in the generation of "preferred terms". Declarative statements such as

```
(AssertThat :a "inflammation"
   :which (Is-Located o-appendix)
   :is-called "appendicitis")
```

allow for the selection of common terms.

8 Evaluation

For a shell system such as Suregen-2, evaluation has two different meanings. The first would address the question of how easy it is to build an NLG application using Suregen-2. The second would focus on a prototypical Suregen-2 application considering text quality. We did not perform any formal evaluation in either direction, but a few remarks can nevertheless be made.

8.1 Suregen-2 as an NLG shell

Although development of Suregen-2 and the building of the cardiology application were intricately connected, it is possible to estimate the efforts for designing an application in, say, gastroenterology to be approximately within one person year. Of course, each new application will add functionality to the generic body of Suregen-2, for instance new classes such as "ulcer" or "endoscopy" or even new semantic functions, such that subsequent applications may benefit from earlier work. The wish for individual text for each physician may, in the reverse extreme, multiply the efforts required. Having programmed several (conventional) systems with text generation functionality we can say that Suregen-2 means less effort and better text quality. It is, however, still an open question whether this benefit will be accessible to other developers as well.

8.2 Evaluation of an application

Integration of the cardiology application into our hospital information system was seamless, robust and performant. With regard to text quality a few options seem feasible:

With a modified CLOZE procedure (where persons are asked to fill in artificial gaps with missing morphemes, lexemes or even entire phrases (Taylor, 1953)) it would be possible to measure the accuracy of, and time used for this process, giving estimations of correctness for lexical choice and inflection, as well as aggregate measures of the coherence/cohesion of the generated text. Another approach would be “reverse entry”. Here a physician, having read a Suregen-2 generated text, would re-enter the information into the system which had generated the text in the first place. The corresponding states of the ontologies’ instances, O_1 and O_2, could then be compared, giving a measure of how accurately the output text reflected O_1. Moreover, the time required by the physician to re-enter the information would hint at the “communicative efficiency” of the output.

We did neither of these, relying for the time being on the rather informal method of displaying generated text as a response to each physician’s mouse click in the structured data entry dialog. The following sample generated by the cardiology application (Hüske-Kraus, 2002) may illustrate the text quality:

“Frau Test arbeite als IT-Leiterin, momentan ist sie arbeitsunfähig. Aktuell gibt die Patientin gelegentliche Übelkeit an. Sie klagt über gürtelförmige, retrosternale Schmerzen, die in den linken Arm ausstrahlen, formal CCS III. Diese Beschwerden treten unter Ruhebedingungen und bei Belastungen auf. […] Gelegentlich sowohl prätibiale als auch Knöchel- und Sakralödeme überwiegend links.”

The text in this example is syntactically correct, uses “standard” terms like “CCS”, referential expressions and ellipsis. It contains an enumerative expression, in turn containing two “preferred terms” (“Knöchelödem” and “Sakralödem”), sorted in a way that enables the first of the two to be hyphenated in the correct manner for compound nouns in German. In short, it does not in any way look “computer-generated”, which our hospital would regard as unacceptable.

9 Conclusions

The main motive for developing Suregen-2 was to arrive at an operational system. Concededly this entailed some decisions in favor of simple, but workable solutions rather than those with more elegance or generality. With regard to referential expressions the approach chosen was sufficient; with respect to lexical choice reliance on one-to-one associations of concepts to lexemes may prove too rigid. The results we have obtained so far - with regard to coverage of linguistic phenomena in the medical sublanguage, flexibility and performance of the generation component - seem to warrant the statement not only that the architectural design of Suregen-2 is well adapted to the needs of medical document generation, but also that further efforts, especially to refine the ontology and the generic semantic functions, will be worthwhile.

References

Bernauer, J., 1996. Analysis of part-whole relation an subsumption in the medical domain. DATA KNOWL ENG, 20: 405-415.

Cahill, L., 1999. Lexicalisation in applied NLG systems. ITRI-99-04, Information Technology Research Institute, Brighton.

Edmonds, P., 1997. Choosing the Word Most Typical in Context Using a Lexical Co-occurence Network. In: Estival, Lavelli, Netter and Pianesi (Eds), 35th Annual Meeting of the Association of Computational Linguistics, Madrid, pp. 507-509.

Elhadad, M., 1996. Lexical Choice for Complex Noun Phrases. Machine Translation, 11: 159-184.

Hüske-Kraus, D., 2002. Suregen-2. www.suregen.de

Hüske-Kraus, D., 2003. Text Generation in Clinical Medicine - a Review. METHODS INF MED, 1.

MacGregor, R. and Bates, R., 1987. The LOOM representation language. ISI/RS-87-188, USC Information Sciences Institute.

McCray, A.T. and Nelson, S.J., 1995. The representation of meaning in the UMLS. Methods of Information in Medicine, 34 (1-2): 193-201.

Nogier, J.-F. and Zock, M., 1992. Lexical choice as pattern matching, Knowledge Based Systems, 5 (3), pp. 200-212.

Reiter, E. and Dale, R., 2000. Building Natural Language Generation Systems, Cambridge: Cambridge University Press.

Shaw, J., 1998a. Clause Aggregation Using Linguistic Knowledge, 9th International Workshop on Natural Language Generation, Niagara-on-the-Lake, Canada.

Shaw, J., 1998b. Segregatory coordination and Ellipsis in Text Generation, 36th Annual Meeting of the Association for Computational Linguistics and the 17th International Conference on Computational linguistics, pp. 1220-1226.

Stede, M., 1999. Lexical semantics and knowledge representation in multilingual text generation. Kluwer Academic Publishers.

Taylor, W.L., 1953. Cloze Procedure: A new tool for measuring readability. JOURNALISM QUART, 30 (4): 415-433.

Wilkinson, J., 1995. Aggregation in Natural Language Generation: Another look. M. Sc. Thesis, University of Waterloo, Canada, Waterloo.