SURFACE FORM TO SENTENCE PLANS: A METHOD FOR ENGLISH-TO-ENGLISH TRANSLATION

Steven Banks and Chrysanne DiMarco

Department of Computer Science, University of Waterloo, Waterloo, Ontario, N2L 4G1, Canada.

Natural language generation (NLG) systems aim at the production of fluent and expressive text from an internal computational representation. This representation can be as simple as canned text, where the internal representation *is* the text, or templates that use databases to fill in the blanks. More-powerful NLG systems use highly abstract and linguistically complex internal representations to allow the creation of highly varied texts from the same internal representation. But, to be able to create such complex texts, the user must first provide the NLG system with the input representation of the desired content. So in this way, natural language generation systems, although potentially powerful, suffer from the compromise between power and accessibility. A tool that could simplify the construction of input specifications for generation systems could thereby open powerful NLG resources to a wider, less specialized, audience – this is the problem we address in the development of a tool for the automated translation of English surface form to input representations for natural language generator systems.

Key words: natural language generation, translation

1. INTRODUCTION

1.1. The starting point: Reusable text as input for natural language generation

Natural language generation (NLG), like any other highly complex task, suffers from the need to make a compromise between power and accessibility. As NLG systems gain in robustness and versatility, the background and knowledge needed to learn to use them limits the range of their use. One of the most difficult problems in natural language generation is creating an input specification for a generation system that is reasonably easy to learn and use, and will produce high-quality, expressive text. People have tried to avoid the problem by using canned text as input or finessed the problem by using templates, but in order to achieve expressive language, most solutions have relied on input specifications that are based on concepts in an underlying deep-semantic knowledge base. But while the degree of expressivity of language generated from a knowledge base can be high, the degree of difficulty in composing the appropriate input to the generator is also very high.

We have taken an approach to generation that tries to take advantage of the ease of use of canned, or reusable, texts while still being able to generate flexible and high-quality text by working from a deeper level of representation. We aim to reduce the difficulty inherent in generation by moving the problem from one of authoring and using deep knowledge bases to authoring and transforming surface text. In our approach, the users of a generation system specify reusable surface text that will then be automatically or semi-automatically converted to a deeper level of representation, in our case, to the deep-syntactic, *sentence-plan* level. This deeper level of linguistic description can then be modified and used as an input specification to a natural language generation system.

In our earlier work on the HealthDoc Project (DiMarco, Hirst, Wanner, and Wilkinson 1995; Wanner and Hovy 1996; Hovy and Wanner 1996; DiMarco and Foster 1997; DiMarco, Hirst, and Hovy 1997; Hirst, DiMarco, Hovy, and Parsons 1997), we developed natural language generation systems that could produce health-education and patient-information material tailored to the individual patient. HealthDoc takes as its starting-point the paradigm of *'generation by selection and repair'*, that is, the premise that new texts can be created from existing texts and sentence-plan templates by a process of selection and re-assembly, and then 'repaired' to restore coherence and cohesion. But, in order to be able to repair the sentence-plan level of representation, the original English text must first be converted, in effect, *translated* to sentence-plan notation — a task that is tedious and difficult if done manually for any but sparse amounts of text. We therefore propose a method for the automated translation of surface text into sentence-plan form that attempts to 'reduce, recycle, reuse' whenever possible, that is, to put to new use pre-existing linguistic resources, both theoretical and computational. Over the course of this work, a representative suite of test sentences has been compiled and processed using an implementation of this method.

1.2. The role of stylistic analysis

Mapping from an English sentence to its sentence-plan representation is inherently difficult because relatively small differences in the surface English text can mean consequent large variations in the underlying sentence plan needed to generate that text. The main problem in trying to automate the process of producing a sentence-plan notation from a surface form is that we have to deal with a plethora of details about the surface structure, many of which have no bearing on the form that the sentence plan will take. It would be helpful therefore to have an intermediate stage that abstracts from the details of the syntactic analysis to the larger characteristic structures within the sentence that do define the form of the corresponding sentence plan. In our system, we have chosen to use *stylistic abstraction* as this intermediate stage.

1.3. The role of subsumption

Although a stylistic analysis of the input sentence gives a means of targetting the larger stylistic structures in a sentence that are relevant to its sentence-plan composition, what is still needed is a way of mapping from the stylistic description of the sentence to its sentence-plan representation. To do this, we use subsumption classification based on a taxonomy of stylistic categories, thereby allowing us to perform a finely-grained decomposition of a sentence into a set of stylistic components, use those components to map into a corresponding taxonomy of sentence-plan templates, and then build up the overall sentence plan from these templates.

2. STYLISTIC FORM AS AN INTERMEDIATE REPRESENTATION

2.1. ASSET: A stylistic analyzer

During the HealthDoc project, we manually composed about 100 pages of sentence-plan expressions in Sentence Plan Language (SPL) (The Penman SPL Guide 1991) for English texts and during this process we noted many correspondences between our stylistic method of analysis, based on DiMarco and Hirst's (1993) stylistic grammar, and the resulting SPL expressions. As a consequence, in this research we have used ASSET (Hoyt 1993), the stylistic analyzer that implements DiMarco and Hirst's grammar, as the basis for an English-to-SPL translation system.

ASSET, or Analyzing the Style of SEnTences (Hoyt 1993), was developed at the University of Waterloo as an implementation of the computational theory of style originally developed by DiMarco (1990) and refined several times thereafter (DiMarco and Hirst 1993; Green 1993; Hoyt 1993; Mah 1994). ASSET requires a syntactic parser to provide its input and when ASSET was originally developed, PUNDIT, or Prolog UNDerstander of Integrated Text (Dahl 1992), was chosen to fill this role due to its fairly large syntactic coverage and its comprehensive treatment of conjunctions.

ASSET, when given a properly formatted syntactic parse of a sentence, will return an analysis of its stylistic structure. This process is divided into several steps, each with its own module corresponding to different aspects of the underlying stylistic grammar. In our English-to-sentence-plan processing, we mainly use the level of an 'abstract-element' analysis from ASSET's output, which picks out the patterns of linear ordering and hierarchical structure that define a sentence's characteristic stylistic structure. The abstract elements are based on the notions of stylistic *concord* and *discord*, where concord means a stylistic construction that conforms to the norm for a given genre, and discord, not necessarily a bad thing, means a construction deviating from normal usage. Through patterns of conformance to, and deviation from, normal usage, a sentence's linear ordering and hierarchical structure produces stylistic effects related to symmetry, parallelism, position, and nesting of structures within the sentence. It is those effects that DiMarco and Hirst characterized in their stylistic grammar and that we use to map to corresponding structures within a sentence-plan representation.

The abstract stylistic elements that we use most often are as follows:

Homopoise: A sentence with interclausal coordination of syntactically similar components.

A homopoisal effect indicates that there is a concordant parallellism within the sentence that must be indicated within the sentence plan. In the example below, there is a coordination of two top-level clauses:

(1) High blood pressure may contribute to diabetic retinopathy and smoking can aggravate the condition.

In the corresponding sentence plan shown in figure 1, the parallelism is indicated by a conjunction between the *domain*, a 'major' clause, and the *range*, also a major clause.

Sentences that are more stylistically complex can have their balance interrupted or perturbed by a *heteropoisal* ("different weight") component:

Heteropoise: A sentence in which one or more parenthetical components are syntactically 'detached' and dissimilar from the other components at the same level in the parse tree.

In the example below, there is an interrupting noun phrase, *a disease of the retina*, in the main clause:

(2) High blood pressure may contribute to diabetic retinopathy, a disease of the retina, and smoking can cause the condition to worsen.

In the corresponding sentence plan shown in figure 2, the heteropoisal noun phrase is handled as a 'restatement' (RESTATE-00034).

Most of the sentences we studied were *centroschematic*:

Centroschematic: A sentence with a central, dominant clause with one or more of the following optional features: complex phrasal subordination, initial dependent clauses, terminal dependent clauses.

In this case, both the stylistic analysis and the sentence-plan structure are built up through linear ordering and hierarchical nesting — for example, the following sentence is centroschematic:

```
;
;***Process Complete
;***SPL : (|i|COMPLETE-00001 / CONJUNCTION
;          :DOMAIN |i|MAJOR-00002
;          :RANGE  |i|MAJOR-00003)
;***Pieces :
;(|i|NG-00023 / ABSTRACTION
;  :LEX SMOKING :DETERMINER ZERO)
;(|i|NG-00029 / ABSTRACTION
;  :LEX CONDITION :DETERMINER THE)
;(|i|MAJOR-00003 / DISPOSITIVE-MATERIAL-ACTION
;  :LEX AGGRAVATE :TENSE PRESENT :MODALITY CAN
;  :ACTOR |i|NG-00023 :ACTEE |i|NG-00029)
;(|i|ADJECTIVAL-00014 / SCALABLE-QUALITY
;  :LEX HIGH)
;(|i|NOUN-00015 / SUBSTANCE
;  :LEX BLOOD)
;(|i|NG-00006 / ABSTRACTION
;  :LEX PRESSURE :DETERMINER ZERO
;  :CLASS-ASCRIPTION |i|NOUN-00015
;  :SCALED-COMPARISON |i|ADJECTIVAL-00014)
;(|i|NG-00017 / ABSTRACTION
;  :LEX RETINOPATHY :DETERMINER ZERO
;:PROVENANCE-PROPERTY-ASCRIPTION
;                 |i|ADJECTIVAL-00020)
;(|i|ADJECTIVAL-00020 / QUALITY
;  :LEX DIABETIC)
;(|i|MAJOR-00002 / NONDIRECTED-ACTION
;  :LEX CONTRIBUTE :TENSE PRESENT
;  :DESTINATION |i|NG-00017 :MODALITY MAY
;  :ACTOR |i|NG-00006)
```

Figure 1. Sentence plan for sample homopoisal sentence

(3) The pancreas is a large gland located behind the stomach that produces the hormone insulin.

In the corresponding sentence plan shown in figure 3, the sentence's characteristic structure of linear ordering and postmodification is mirrored: the postmodifying nonfinite clause *located behind the stomach* is indicated as a 'process' (NF-CLAUSE-00019) attached to the noun *gland* (NG-00011), while the postmodifying relative clause *that produces the hormone insulin* (REL-CLAUSE-0020) is also indicated as a process attached to the noun *gland*.

Throughout our study, our concern has been to break down the process of sentence-plan construction into small steps, relying on the stages of analysis that have gone before to direct and constrain the sentence-plan–building process. As we describe in the following section, the processes of stylistic analysis and sentence-plan construction are then orchestrated through the key stage of pattern classification based on subsumption of stylistic components and matching sentence-plan templates.

```
;***Process Complete
;***SPL : (|i|COMPLETE-00001 / CONJUNCTION
;           :DOMAIN |i|MAJOR-00002
;           :RANGE  |i|MAJOR-00003)
;***Pieces :
;(|i|NG-00037 / ABSTRACTION
;  :LEX SMOKING :DETERMINER ZERO)
;(|i|NF-CLAUSE-00048 / NONDIRECTED-ACTION
;  :LEX WORSEN :TENSE PRESENT
;  :ACTOR |i|NG-00043)
;(|i|NG-00043 / ABSTRACTION
;  :LEX CONDITION :DETERMINER THE
;  :PROCESS |i|NF-CLAUSE-00048)
;(|i|MAJOR-00003 / CREATIVE-MATERIAL-ACTION
;  :LEX CAUSE
;  :TENSE PRESENT :MODALITY CAN
;  :ACTOR |i|NG-00037 :ACTEE |i|NG-00043)
;(|i|ADJECTIVAL-00014 / SCALABLE-QUALITY
;  :LEX HIGH)
;(|i|NOUN-00015 / SUBSTANCE
;  :LEX BLOOD)
;(|i|NG-00006 / ABSTRACTION
;  :LEX PRESSURE :DETERMINER ZERO
;  :CLASS-ASCRIPTION |i|NOUN-00015
;  :SCALED-COMPARISON |i|ADJECTIVAL-00014)
;(|i|NG-00017 / ABSTRACTION
;  :LEX RETINOPATHY :DETERMINER ZERO
;  :PROVENANCE-PROPERTY-ASCRIPTION
;                  |i|ADJECTIVAL-00021)
;(|i|RESTATE-00034 / RESTATEMENT-COMMAS
;  :DOMAIN |i|NG-00017
;  :RANGE  |i|NG-00023)
;(|i|NG-00030 / OBJECT
;  :LEX RETINA :DETERMINER THE)
;(|i|NG-00023 / ABSTRACTION
;  :LEX DISEASE :DETERMINER A
;  :PART-OF |i|NG-00030)
;(|i|ADJECTIVAL-00021 / QUALITY
;  :LEX DIABETIC)
;(|i|MAJOR-00002 / NONDIRECTED-ACTION
;  :LEX CONTRIBUTE :TENSE PRESENT
;  :DESTINATION |i|RESTATE-00034
;  :MODALITY MAY :ACTOR |i|NG-00006)
```

Figure 2. Sentence plan for sample heteropoisal sentence

3. LOOM: THE INFERENCE ENGINE FOR PUTTING THE PIECES TOGETHER

3.1. Why Loom was chosen

Although there are several different systems that together form the English-to-sentence-plan transformation mechanism, it is the Loom knowledge representation system (Brill 1993)

```
;***Process Complete
;***SPL : (|i|MAJOR-00001 / ASCRIPTION
;           :LEX IS :TENSE PRESENT
;           :DOMAIN |i|NG-00004
;           :RANGE |i|NG-00011)
;***Pieces :
;(|i|NG-00011 / OBJECT
;  :LEX GLAND :DETERMINER A
;  :PROCESS |i|NF-CLAUSE-00019
;  :PROCESS |i|REL-CLAUSE-00020
;  :SIZE-PROPERTY-ASCRIPTION
;             |i|ADJECTIVAL-00018)
;(|i|NF-CLAUSE-00019 / DIRECTED-ACTION
;  :LEX LOCATE :TENSE PRESENT
;  :BEHIND |i|NG-00040
;  :REDUCED :ACTEE |i|NG-00011)
;(|i|NG-00040 / OBJECT
;  :LEX STOMACH :DETERMINER THE)
;(|i|REL-CLAUSE-00020 / CREATIVE-MATERIAL-ACTION
;  :LEX PRODUCE :TENSE PRESENT
;  :ACTOR |i|NG-00011 :ACTEE |i|RESTATE-00048)
;(|i|NG-00038 / OBJECT
;  :LEX HORMONE :DETERMINER THE)
;(|i|RESTATE-00048 / RESTATEMENT
;  :DOMAIN |i|NG-00038
;  :RANGE |i|NG-00039)
;(|i|NG-00039 / SUBSTANCE
;  :LEX INSULIN :NUMBER MASS)
;(|i|ADJECTIVAL-00018 / QUALITY
;  :LEX LARGE)
;(|i|NG-00004 / OBJECT
;  :LEX PANCREAS :DETERMINER THE)
```

Figure 3. Sentence plan for sample centroschematic sentence

that is the real powerhouse: we use Loom's knowledge-based inferencing, which is based on subsumption, to classify the pieces of the stylistic parse and to match these pieces to a library of sentence-plan templates. In this section, we will give an overview of the features of Loom that we use in the sentence-plan 'classification engine'.

3.2. An overview of Loom

Loom is a highly expressive programming language and environment that is based on the KL-ONE (Brachman 1985) family of languages. As with other languages within the KL-ONE family, Loom features automatic classification to compute subsumption relationships between the concepts defined within a knowledge base. This capability was key in its selection for this research. Another attractive feature was the versatile interaction of Loom *methods*, *actions*, and *production rules*. These two features are known respectively as the *modelling* and *behaviour languages*.

```
(defconcept Postmod-Type
  :is (:one-of 'PP 'EXEM 'NP 'REL-CLAUSE 'NF-CLAUSE 'NONE))
```

Figure 4. Sample Loom concept defining postmodification types

The Loom *modelling language* is used to define the conceptual components relevant to the working domain. Classes of objects are defined, as well as the relations used to connect these classes and to form the subsumption taxonomy within the knowledge base. To construct the Loom model for subsequent classification based on subsumption, the definitions in the ASSET stylistic grammar were transformed into corresponding Loom concept definitions, An example of a Loom concept based on the ASSET grammar is given in figure 4. In the example, the concept of postmodification is defined as being either a prepositional phrase, exemplification, noun phrase, relative clause, nonfinite clause, or null postmodification.

The Loom *behaviour language* is used to define procedural components that manipulate the working domain defined by the modelling language. These procedures are implemented in the form of *production rules*, *actions*, and *methods*. A Loom *production* is a rule that is invoked when a certain user-proscribed event is detected within the knowledge base. This invocation can trigger the calling of one or more *actions*, or even the running of an external Lisp program.

Once the production rule has passed its arguments to the specified Loom action, it is the action that determines how to choose the correct Loom *method* that will actually perform the action. From the list of applicable methods, which specify the many ways in which an action can be implemented, Loom will pick the one most closely related to the arguments passed to the action.

3.3. The Loom sentence-plan classification engine

Loom's property of allowing many methods to be defined for the same action, coupled with its subsumption and automatic classification capabilities, made it easy to model our sentence-plan classification problem in abstract terms. Actions that must be performed repetitively on many classes of objects are defined generically, while the class-dependent differences are left to the methods. The separation of the specification for a production-rule firing from the decision of how to choose between multiple responses to a rule invocation creates a modular problem-solving structure. And the subsumption-based power of Loom was just what was needed to build a classifier that would operate on the pieces of the stylistic parse, organized into an inheritance hierarchy, and put them together from a matching taxonomy of sentence-plan templates.

4. A PROCESS OF DECOMPOSITION AND RECOMPOSITION

4.1. Putting it all back together

The process of building the SPL forms that correspond to the original English input can be described as one of *decomposition* of the input sentence and *recomposition* from SPL templates. The translation process splits a sentence into manageable fragments according to a stylistic analysis, determines the SPL forms for these fragments, then rebuilds the sentence while constructing larger and larger SPL expressions. The entire process consists of only five Loom actions and the five Loom production rules that trigger them, but dozens of Loom methods were coded to implement those actions. And even though the bulk of the code is made up of methods, the interaction between the production rules and the actions can be quite complex.

4.2. Decomposing into syntactic fragments

The Loom model contains a generic concept *Fragment*, which subsumes every concept in the grammar. Every instance that is created to represent a constituent of the input sentence is a specialized *Fragment*, from the full input sentence, down to each individual word. As each new *Fragment* is produced, it must be identified in the grammar and classified so that it can be properly processed. This classification determines whether the *Fragment* is a *DecomposableFragment* or a *BaseFragment*.

4.3. Building up from SPL basics

Only a small class of *Fragments* are *BaseFragments*, so most *Fragments* need to be decomposed — this means that the *Fragment* is still too large and general for the translation process to use a simple form in the library of existing SPL templates. The process of breaking a large decomposable *Fragment* into more manageable pieces creates a series of *NewFragments*, each of which must be 'filled' with the appropriate SPL form.

As each *NewFragment* is created, it will be detected by the Loom classification engine, filled, and decomposed itself. Then the children of the *NewFragment* will be filled and decomposed, and so on, creating a large tree structure representing the sentence at varying levels of syntactic analysis. At some point, a *DecomposableFragment* will yield a *Fragment* that can be classified as a *BaseFragment*. *BaseFragments* are not decomposed — instead, their SPL form is simply retrieved from the library of simple templates and passed back up the tree to their parent *Fragments*.

4.4. Recomposing the pieces

As the *BaseFragments* are being processed and their SPL templates are being passed up the tree, some of the smaller *DecomposableFragments* located near the bottom of the tree will now be passed all the information needed to construct their own SPL templates. This iterative process continues until a *Fragment* is flagged as 'done', i.e., its SPL form has been completely constructed. Then, its larger parent *Fragment* is retrieved and a check is made to see whether or not the rest of the parent's children have also been flagged as done. If so, then the SPL expression for the parent *Fragment* has been completely recomposed. A final 'clean-up' of the reconstructed SPL form is then performed and the translation process is now complete.

4.5. Testing the translation process

We tested our translation process on a set of about 30 representative sentences for which we had handwritten SPL forms that had been successfully run through the Penman language generation system. In all cases, it was shown that with our system the sentences could successfully be broken down into 'base case' pieces with trivial SPL templates and then built back up by examining larger and larger pieces of the sentence to yield the same SPL as for the manually constructed versions. The ASSET stylistic grammar and its subsuming implementation in Loom provided an elegant structure in which to specify the methods that carried out the necessary actions. The stylistic differentiation also allowed many different types of sentences to be processed, from basic sentences with minimal modification, to large compound sentences with varying forms of clauses and modification.

5. CONCLUSION

The intent of this research has been to take steps toward, and explore the possibilities and limitations of, the automatic translation of natural language input into a deep-syntactic, sentence-plan, representation. Although it has been clear since the beginning that a fully automatic system was impossible, as ambiguous sentences require clarification from the user before the process even begins if the correct sentence plan is to be produced, our initial results support the validity of this approach. In the longer-term, we would like to incorporate the results of this research into developing more-capable authoring tools for natural language generation systems.

ACKNOWLEDGEMENTS

The HealthDoc Project was supported by a grant from Technology Ontario, administered by the Information Technology Research Centre.

REFERENCES

Brachman, R.J. and J.G. Schmolze. An overview of the KL-ONE knowledge representation system. Cognitive Science, **9**(2):171–216, 1985.

Brill, D. Loom Reference Manual, Version 2.0. Information Sciences Institute, University of Southern California, 1993.

Dahl, D. PUNDIT - Natural Language Interfaces. Proceedings of Logic Programming in Action. Heidelberg, Germany, September 1992.

DiMarco, C. Computational stylistics for natural language translation. Ph.D. thesis, Department of Computer Science, University of Toronto, 1990.

DiMarco, C., and G. Hirst. A computational theory of goal-directed style in syntax. Computational Linguistics, **19**(3):451–499, 1993.

DiMarco, C., and M.E. Foster. The automated generation of Web documents that are tailored to the individual reader. Proceedings, AAAI Spring Symposium on Natural Language Processing on the World Wide Web. Stanford University, March 1997.

DiMarco, C., G. Hirst, and E. Hovy. Generation by selection and repair as a method for adapting text for the individual reader. Proceedings, Workshop on Flexible Hypertext, Eighth ACM International Hypertext Conference. Southampton UK, April 1997.

DiMarco, C., G. Hirst, L. Wanner, and J. Wilkinson. HealthDoc: Customizing patient information and health education by medical condition and personal characteristics. Workshop on Artificial Intelligence in Patient Education. Glasgow, August 1995.

Green, S. A functional theory of style for natural language generation. Master's thesis, Department of Computer Science, University of Waterloo, 1992.

Hirst G., C. DiMarco, E. Hovy, and K. Parsons. Authoring and generating health-education documents that are tailored to the needs of the individual patient. Proceedings, Sixth International Conference on User Modeling. Sardinia, Italy, June 1997.

Hovy, E. and L. Wanner. Managing sentence planning requirements. Proceedings, ECAI-96 Workshop on Gaps and Bridges: New Directions in Planning and Natural Language Generation. Budapest, August 1996.

Hoyt, P. A goal-directed functionally-based stylistic analyzer. Master's thesis, Department of Computer Science, University of Waterloo, 1993.

Jakeway, B. SPLAT: A sentence plan authoring tool for natural language generation. Master's thesis, Department of Computer Science, University of Waterloo, 1995.

Mah, K. Comparative stylistics in an integrated machine translation system. Master's thesis, Department of Computer Science, University of Waterloo, 1991.

The Penman SPL Guide. Penman Natural Language Generation Group, Information Sciences Institute, University of Southern California, 1991.

Wanner, L. and E. Hovy. The HealthDoc sentence planner. Proceedings of the Eighth International Workshop on Natural Language Generation. Brighton, UK, June 1996.