Breaking Down Barriers:
The Mikrokosmos Generator

Stephen Beale, Evelyne Viegas
and Sergei Nirenburg

Computing Research Laboratory
Box 30001, Dept. CRL
New Mexico State University
Las Cruces, NM 88003-0001 USA
sb, viegas, sergei@crl.nmsu.edu
tel: 505-646-5757 fax: 505-646-6218

Abstract

We argue that the modularization of text generation into separate tasks, as currently practiced, sets up unneeded barriers to the generation task. We propose a new modularity based on natural linguistic phenomena and overview how it is implemented in the Mikrokosmos text generator.

1 Introduction

This paper addresses the area of text generation that has come to be known as sentence planning (Rambow and Koreisky, 1992); (Wanner and Hovy, 1996). Sentence planning involves discourse structuring and marking, sentence boundary planning, clause internal structuring and all of the varied subtasks involved in lexical choice. Conventional wisdom dictates that these complex tasks be modularized and treated separately.

Since the sentence planning tasks listed above are not single-step operations, since they do not have to be performed in strict sequence, and since the planner's operation is non-deterministic, this suggests that each sentence planning task should be implemented by a separate module or by several modules. (Wanner and Hovy, 1996)

In contrast, we argue that such a functional division sets up artificial barriers, with the result that fighting those barriers has become a pervasive topic in the generation literature, a topic that becomes moot when our approach is followed. The very words of the argument quoted above - "not performed in a strict sequence," "non-deterministic" - suggest that this approach is not a natural one. We are certainly not against modularity per se, but our approach modularizes based on the nature of the semantic input and the knowledge sources, and allows an advanced, yet efficient control mechanism to take care of the "complex tasks" involved in sentence planning.

2 The Great Barrier Reek

The primary indication that task-oriented divisions set up artificial barriers is the repeated efforts of researchers to try and breach those barriers. It is recognized that many of the tasks in sentence planning interact and cannot be processed independently. (Meteer, 1992) frames this problem in terms of expressibility. She tackles the problem by adding an intermediate level between components to ensure that what is planned is expressible. (Elhadad et al., 1997) recognizes that constraints on lexical choice come from a wide variety of sources and are multidirectional, making it difficult to determine a systematic ordering in which they should be taken into account. He proposes a backtracking mechanism within his unification framework to overcome the problem. (Rubinoff, 1992) is perhaps the most strongly focused on this issue. He argues that the accepted division into components "ultimately interferes with some of the decisions necessary in the generation process." He utilizes annotations as a feedback mechanism to provide the planning stages with linguistically relevant knowledge. Unfortunately, these solutions, although certainly steps in the right direction (especially Elhadad, in our view), are only perfumes on top of the faulty division of knowledge into functional modules.

Another area of research that belies the unnatural task-based division widely accepted by text generation researchers today is the attempts to control sentence planning tasks using advanced processing architectures such as blackboards. (Nirenburg et al., 1989) and more recently, (Wanner and Hovy, 1996) both argue that the order of sentence planning tasks cannot be pre-determined. Behind this difficulty is the reality that different linguistic phenomena have different, unpredictable requirements. Grammatical, stylistic and collocational constraints combine at unexpected times during the various tasks of sentence planning. Blackboard architectures, theoretically, can be used to allow a certain thread of operation to suspend operation until a needed bit of information is available. Unfortunately, in the best case, such an architecture is inefficient
and difficult to control. In practice, such systems, as admitted by both references above, resort to a “default (processing) sequence for the modules” along with a simplistic truth-maintenance system which ultimately becomes a fail-and-backtrack type of control, completely negating the spirit of the blackboard system. While these shortcomings might eventually be overcome, the fact remains that it was the unnatural division into tasks that necessitated the blackboard processing in the first place.

3 Linguistically- and Ontologically-based Modularization

In contrast to this division of knowledge into tasks such as discourse structuring, clause structuring and lexical choice, the Mikrokosmos project attempts to modularize based on the ontological types and natural linguistic phenomena that serve as inputs to our processing. In semantic analysis (Beale et al., 1995), the natural division is along word types: nouns, verbs, adjectives, etc., and along linguistically-based microtheories\(^1\) such as studies of tense (to discover aspectual components of meaning) and coreference analysis. For generation (Viegas et al., 1997), our inputs are semantic representations, which become the focus of our modularization. The most important types of semantic representations are the ontological categories of EVENTS, OBJECTS and PROPERTIES. In addition to these we have several generation microtheories that deal with issues such as focus and reference.

In general, then, we modularize based on the types of inputs we expect, not on the types of processing we need to perform. Each module can perform any task. For instance, EVENTS and PROPERTIES both set up clause and sentence structures as well as contribute to lexical choice, as will be shown below. Interactions and constraints flow freely among the modules and are processed by the control mechanism. It is interesting to note that one outcome of this division of labor is that the bulk of our knowledge is resident in the lexicon, both for analysis (where the lexicon is indexed on words) and generation (where the same lexicon can be indexed on concepts). This has greatly simplified knowledge acquisition in general (Nirenburg et al., 1996) and made it easier to adapt analysis knowledge sources to generation (Viegas and Beale, 1996) as well as converting knowledge sources for one language to another.

Below we sketch out some examples of how this type of organization works. We begin by describing the main types of lexicon entries with the goal of demonstrating how each can perform various generations tasks. We then describe how lexicon entries are combined to create options for the generator. We also discuss a few of the heuristics used to choose between these options. The following section then gives a brief overview of how the control architecture efficiently processes these locally created plans to obtain a globally optimal output.

3.1 Types of Lexicon Entries

In generation, our input is semantic representations. The generation lexicon is thus indexed on semantic concepts. The top level division of concepts in our ontology is into OBJECTS, EVENTS and PROPERTIES. Please refer to (Mahesh and Nirenburg, 1995) for details on the Mikrokosmos ontology.

3.1.1 Objects

Objects, whether simple (one-to-one mapping between concept and lexeme) (Figure 1), or complex (involving some properties) (Figure 2), are always lexicalized into Nouns. Refer to (Viegas et al., 1998) for the different types of mappings between concepts and lexemes.

![Figure 1: Object Lexeme Simple Mapping](image1)

![Figure 2: Object Lexeme Complex Mapping](image2)

3.1.2 Events

Events can be lexicalized into simple Nouns (Figure 3) and into Verbs or Nominalized Verbs (complex Noun) (Figure 4). This built-in transcategoriality (where an event can potentially be

\(^1\)Microtheories are simply any collection of knowledge, declarative or procedural, aimed at a particular language phenomena.

\(^2\)Results of microtheories are combined here as well but will not be discussed.
lexicalized as either a Noun or a Verb) is a very natural aspect of our approach to lexicon design, which allows for paraphrases in generation.

![Event Simple Noun Mapping](image)

Figure 3: Event Simple Noun Mapping

![Event Verb - Complex Noun Mappings](image)

Figure 4: Event Verb - Complex Noun Mappings

(Figure 4) shows partial entries for the verb *explode* which is in fact used as an inchoative *The bomb exploded,* and a nominalized event *explosion* with 2 obliques (in reality optional) *the explosion of the mine (caused)* by....

3.1.3 Properties

Properties are perhaps the most interesting of the types discussed here because they are so flexible. They can be realized as adjectives, relative clauses, complex noun phrases and complete sentences. Often a PROPERTY is included in the definition of another OBJECT or EVENT, such as in Figure 2. CASE-ROLE-RELATIONS typically are consumed by the EVENT entry, except in the case of nominalizations such as “explosion,” where only the EVENT itself is realized. In such a case additional case role relations may have to be planned separately: “The explosion of the building.” DISCOURSE-RELATIONS contribute to setting up sentence boundaries, sentence ordering and pronominalization. Figure 5 summarizes some of these options. Each of these entries would be indexed in the lexicon on the PROPERTY involved.

3.2 Developing the options

Each lexicon entry sets up an option for realizing a part of the input semantics. To these we

5RELATION and ATTRIBUTE are the subtypes of PROPERTY.

![Realizing Properties](image)

Figure 5: Realizing Properties

![Example Semantic Representation](image)

Figure 6: Example Semantic Representation

also add options set up by microtheories, which we will not discuss here due to space limitations. Two important functions of the text generation system are to determine how well each of the options fit given the input semantics, and how each of the options can potentially interact with other options. The Mikrokosmos semantic matcher performs these functions.

(Beale and Viegas, 1996) discuss this process in more detail; here we give a simple example to highlight some of the issues. Consider the input semantic representation in Figure 6. Assume there are three lexicon entries that “match” the concept ACQUIRE as shown in Figure 7. The first of these entries produces a fairly generic word “acquire” and expects to match only an AGENT and THEME. The second entry is more specific, producing the English verb “procure” along with a prepositional phrase. This prepositional phrase realizes the INSTRUMENT of the event. The
constant filler values
* HUMAN (age 35) in input vs.
  HUMAN (age 30) in lexicon
- concept fillers
  * HUMAN (origin FRANCE) vs.
  HUMAN (origin EUROPE)

More generally, the semantic matcher sets up the basic mechanism to also deal with language gaps such as lexical divergences or semantic mismatches, (Viegas, 1997).

3.3 Linguistically-based preferences

So far, we have discussed the main types of concept - lexeme mappings we have in our lexicons, where we saw that our transcategorial approach provided a good basis for paraphrases. Then, we discussed the main types of matchings between the input semantics and the lexicon entries along with the way entries will interact. We now turn to the options which arise locally through lexicon entries. Finally, we will show how to make decisions on the basis of lexicon-based preferences and combination-based preferences at the intra- and inter-sentential levels. Among the following options, which are local to lexicon entries (whether simple or complex, the latter for idioms for instance), the generator will be provided some preferences, such as prefer collocations heavy smoker over the decomposition a person who smokes a lot. Some preferences are stronger than others, such as grammatical constraints provided by lexicon entries as illustrated later.

Options at the Lexicon Level. Preferences arise at various levels we illustrate below:

1. Lexical:
   dog → canine

2. Lexical and Syntactic Level:
   The bomb exploded at 3am → *Something exploded the bomb at 3am

3. Collocational:
   heavy smoker → *big smoker
   set the table → *put the table

4. Stylistic:
   pal → friend
   explain → explicate

5. Pronominalization:
   John → he

6. Definite Description:
   I saw a snake on the path. → The snake stared at me...
(7) Lexical Anaphora:
Buy 1 pound of carrots, potatoes and flank beef. Peel the carrots and the potatoes... → Peel the vegetables...
They built a new house. They made it in no time

(8) Grammatical Constraint:
a- I promised Marie to go →
b- * I told Marie to go for sure
c- * I told Marie (that) I would go for sure

(9) grammatical ellipsis:
John ate a candy and Paula ate a candy. → John ate a candy and Paula did too.

(10) stylistic - genre
assertive-act
a- announce formality high - frequency 5
b- state formality high - frequency 3
c- say simplicity high - frequency 8

In the next paragraph, we give some examples of preferences used in combining entries at the sentence level: lexicon-based and combination-based.

Intra-sentential Preferences The generator must be given some preferences on the above options in order to help it make decisions. For instance, the generator will prefer to always use grammatical constraints as in (8), where to keep the same meaning as in (8-a), one should select (8-c) with a COMP and not (8-b) with the XCOMP. The generator will also prefer collocations, as said previously. Some other preferences, still lexicon based, will be provided by the stylistic and frequency features of an entry, such as preferring (10-a) for a formal text over (10-b) which is less frequent or over (10-c) which is not formal.

Finally, the generator must allow for dealing with combination-based preferences or in other words, looking at interactions globally and not only locally. For instance, the global stylistics of a text might be different from the local stylistic of lexemes looked at individually. So, assuming we want a formal text, the generator must keep track of how many very formal lexemes we have used, as we would not want to end up with a strange text. Some other global preferences will be goal driven, such as maximizing the coreference possibilities.

Inter-sentential Preferences Here we illustrate some preferences to help the generator make decisions on where to “break” the lexico-semantics of the semantic input into different sentences. For instance, in absence of any other preference, it is not natural to split a sentence on an entity slot, such as in I drive a car. The car has a color and a cost. The color is purple and the cost is unaffordable. It is better to perform an entity split, that is only breaking on the entities, such as in I drive an unaffordable purple car.

4 Efficient Processing

The Mikrokosmos project utilizes an efficient, constraint-directed control architecture named Hunter-Gatherer (HG). (Beale et al., 1996) overviews how it enables semantic analysis to be performed in near linear-time. Its use in generation is quite similar. (Beale, 1997) describes the architecture in detail.

HG uses knowledge of constraints to partition the input problem into smaller, relatively independent subproblems which can be processed in isolation. Solution synthesis techniques are then used to combine sets of optimized solutions to subproblems into global solutions. HG is able to guarantee optimal solutions (optimal as far as the internal scoring mechanisms are concerned) and can do so in near linear time for many types of problems, including semantic analysis and, we expect, text generation.

The following sums up the advantages HG provides for text generation:

• Allows “exhaustive” look at local combinations.

• No need to make early decisions.

• Allows interacting constraints, constraints at any level, while still utilizing modular, declarative knowledge.

• Guarantees optimal answers (as measured by preferences).

• No separate microtheory needed to generate sentence-level structure.

• Speed

5 Summary

We have argued that modularizing knowledge sources based on functional tasks such as discourse structuring, clause structuring and lexical choice creates unnecessary barriers for the generation process. Such processes are not independent, cannot be ordered ahead of time, and generate constraints which act in unpredictable ways. The Mikrokosmos project divides up the world differently. We

These constraints can be semantic, syntactic, collocation, etc.
create a modularization based on natural linguistic phenomena. Each module can perform any of the tasks mentioned above and can freely interact and constrain any other module. Not only has this organization removed many of the barriers inherent in the older approach, but it allows for an efficient control architecture and has simplified knowledge acquisition.

References


