Content Planner Construction via Evolutionary Algorithms and a Corpus-based Fitness Function

Pablo A. Duboue
Department of Computer Science
Columbia University
pablo@cs.columbia.edu

Kathleen R. McKeown
Department of Computer Science
Columbia University
kathy@cs.columbia.edu

Abstract

In this paper, we present a novel technique to learn a tree-like structure for a content planner from an aligned corpus of semantic inputs and corresponding, human-produced, outputs. We apply a stochastic search mechanism with a two-level fitness function. As a first stage, we use high level order constraints to quickly discard unpromising planners. As a second stage, alignments between regenerated text and human output are employed. We evaluate our approach by using the existing symbolic planner in our system as a gold standard, obtaining a 66% improvement over a random baseline in just 20 generations of genetic search.

1 Introduction

In a standard generation pipeline (Reiter, 1994), a content planner is responsible for the higher level document structuring and information selection. Any non-trivial multi-sentential/multi-paragraph generator will require a complex content planner, responsible for deciding, for instance, the distribution of the information among the different paragraphs, bulleted lists, and other textual elements. Information-rich inputs require a thorough filtering, resulting in a small amount of the available data being conveyed in the output. Furthermore, the task of building a content planner is normally recognized as tightly coupled with the semantics and idiosyncrasies of each particular domain.

The AI planning community is aware that machine learning techniques can bring a general solution to problems that require customization for every particular instantiation (Minton, 1993). The automatic (or semi-automatic) construction of a complete content planner for unrestricted domains is a highly desirable goal. While there are general tools and techniques to deal with surface realization (Elhadad and Robin, 1996; Lavoie and Rambow, 1997) and sentence planning (Shaw, 1998), the inherent dependency on each domain makes the content planning problem difficult to deal with in a unified framework; it requires sophisticated planning methodologies, for example, DPOCL (Young and Moore, 1994). The main problem is that the space of possible planners is so large. For example, in the experiments reported here, it contains all the possible orderings of 82 units of information.

In this paper, we present a technique for learning the structure of tree-like planners, similar to the one manually built for our MAGIC system (McKeown et al., 1997). The overall architecture for our learning of content planners is shown in Figure 1. As input we utilize an aligned corpus of semantic inputs aligned with human-produced discourse. We also take advantage of the definition of the atomic operators (messages) from our existing system. We learn these tree-like planners by means of a genetic search process. The plan produced as output by such planners is a sequence of semantic structures, defined by the atomic operators. The learning technique is complementary to approaches proposed for generation in summarization (Kan and McKeown, 2002), that utilize semantically annotated text to build content planners.

Our domain is the generation of post cardiac-surgery medical reports or briefings. MAGIC produces such a briefing given the output from inferences computed over raw data collected in the operating room (Jordan et al., 2001). Since we have a fully operational system, it serves as a development environment in which we can experiment with the automatic reproduction of the existing planner. Once the learning system has been fully developed, we can move to other domains and learn new planners. We will also eventually experiment with learning improved versions of the MAGIC planner through evaluation with health care providers.

1.1 Data

The corpus we are using in our experiments consists of the data collected in the evaluation reported in (McKeown et al., 2000). Normal work-flow in the hospital requires a medical specialist to give briefings when the patient arrives in the Intensive Care Unit. In our past evaluation, 23 briefings were collected and transcribed and these were used, along with the admission note, another gold standard, to quantify the quality of MAGIC output (100% precision, 78% recall).

In our work, we align the briefings with the semantic input for the same patient; this input can be used to produce MAGIC output for this patient. In a later stage of the learning process we also align system output with the briefings. An example of the semantic input, system output and the briefing is given in Figure 6. Note that there are quite a few differences among them. In particular, the briefings (c) are normally occurring speech. Aside from being much more colorful than our system output, they also include a considerable amount of information not present in our semantic input. And there is also some information present in our system that is not being said by the doctors. This is because at the time the briefing is given, data such as the name of the patient is available in paper format to the target audience.

1.2 The Current Planner

The planner currently used in the MAGIC system was developed with efficiency in mind, but it lacks flexibility and it is more appropriate for formal speech or text. It has a total of 274 operators; 192 are structure-defining (*discourse* or *topic* levels) and 82 are data defining (*atomic* level) operators[1]. An atomic operator may select, for example, the age of the patient, querying the semantic input for the presence of a given piece of information and instantiating of some semantic structures. Those semantic structures can be as complex as desired, referring to constants, and function invocations. It is also possible for an atomic operator to span several nodes in the output plan if its specified data is multi-valued. During the execution of the planner, the input is then checked for the existence of the datum specified by the operator. If there is data available, the corresponding semantic structures are inserted in the output. The internal nodes, on the other hand, form a tree representing the discourse plan; they provide a structural frame for the placement of the atomic operators. Thus, the execution of the planner involves a traversal of the tree while querying the input and instantiating the necessary nodes.

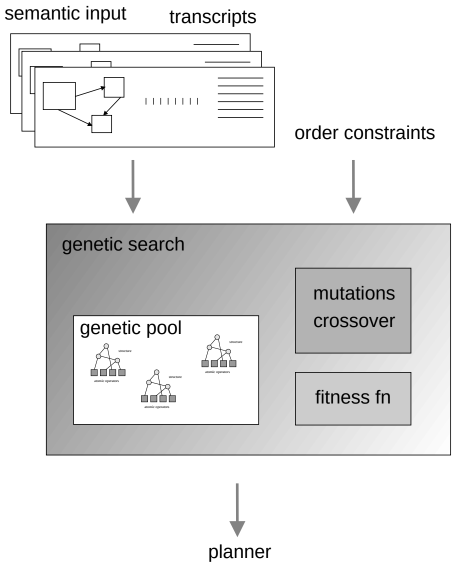

Figure 1: Overall learning architecture

2 Our Approach

Our task is to learn a tree representing a planner that performs as well as the planner developed manually for MAGIC. We explore the large space of possible

[1] equivalent to the notion of **messages** (Reiter and Dale, 2000).

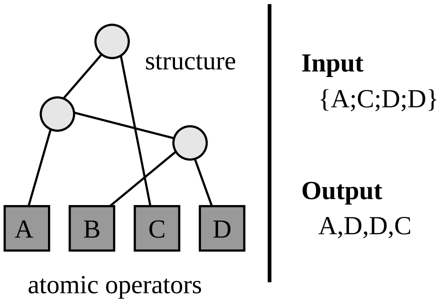

Figure 2: A planner tree-like structure, in our planning formalism, together with an input/output example.

trees by means of evolutionary algorithms. While we use them to *learn* a content planner, they have also proven useful in the past for *implementing* content planners (Mellish et al., 1998). Note that both tasks are different in nature, as ours is done off-line, only once through the life-time of a system, while their use of the GA search will be performed on every execution of the system.

In a genetic search, a population of putative solutions, known as *chromosomes*, is kept. In our case, each chromosome is a tree representing a possible content planner. Figure 2 shows an example planner and how it realizes semantic input when data is missing (B) or duplicated (D). In each cycle, chromosomes are allowed to reproduce themselves, with well-fitted chromosomes reproducing more often. Normally two types of reproductive mechanisms are provided: *mutations* (that produces a new chromosome by modifying an old one) and *cross-over* (that produces a new chromosome by combining two existing ones, its ‘parents’).

Each chromosome has an associated *fitness* value, that specifies how well or promising the chromosome looks. A main contribution of our work is the use of two corpus-based fitness functions, F_C and F_A. We use an approximate evaluation function, F_C that allows us to efficiently determine whether order constraints over plan operators are met in the current chromosome. We use the constraints we acquired on this domain (Duboue and McKeown, 2001),[2] Figure 4). These constraints relate sets of patterns by specifying strict restrictions on their relative placements. Note that a chromosome that violates any of these constraints ought to be considered invalid. However, instead of discarding it completely, we follow Richardson et al. (1989) and provide a penalty function, in order to allow the useful information contained in it to be preserved in future generations.

Once a tree has been evolved so that it conforms to all order constraints, we switch to a computationally intensive fitness function, F_A. In this last stage, we use MAGIC to generate output text using the current chromosome. We then compare that text against the briefing produced by the physician for the same patient. We use alignment to measure how close the two texts are. This procedure is shown in Figure 5. The fitness is then the average of the alignment scores produced for a set of semantic inputs. This approach avoids some of the problems typically found with gold standards. By averaging the fitness function over different semantic inputs, it evaluates the system against different subjects (since each briefing was produced by a different person) in one fell swoop. By capturing similarity in a scalar value (the average itself), it avoids penalizing the system for small discrepancies between system output and gold standard.

For computing each pairwise alignment, we use a global alignment[3] with affine gap penalty, the Needleman–Wunsch algorithm, as defined by Durbin et al. (1998). These alignments do not allow flipping (i.e., when aligning A–B–C and C–B–A they will align both Bs but neither A nor C[4]) and capture the notion of ordering more appropriately for our needs. We adapted their algorithm by using the information content[5] of words, as measured in a 1M-token corpus of related discourse, to estimate the goodness of substituting one word by another.

An important point to note here is that both F_C and F_A are data-dependent, as they analyze the goodness or badness of *output* plans, i.e., sequences of instantiated atomic operators. They require running the planner multiple times in order to do the

[2] In particular, we set $fitness = -1 * N$, where N is the number of violated constraints over on the training set.

[3] We employ global alignments because we are comparing two discourses derived from identical semantic input.

[4] or they will align only the As or the Cs, depending on the score of aligning correctly any of them.

[5] as computed by Pan and McKeown (1999).

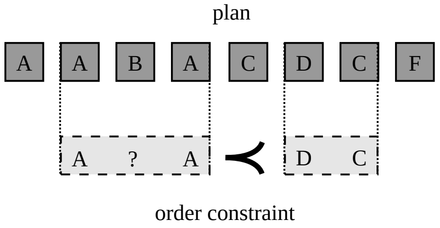

Figure 4: Fitness function: Constraints.

evaluation. We do this because, for one instance, as the planning process may delete (because there is no data available) or duplicate nodes (because of multi-valued data).

An advantage of F_C is that it can be tested on a much wider range of semantic inputs than it is trained on[6].

2.1 Operations over chromosomes

We define three mutation operators and one cross-over operation. The mutations include node insertion, which picks an internal node at random and moves a subset of its children to a newly created subnode, and node deletion, which randomly picks an internal node different from the root and removes it by making its parent absorb its children. Both operators are order-preserving. To include order variations, a shuffle mutation is provided that randomly picks an internal node and randomizes the order of its children. The cross-over operation is sketched in Figure 3. We choose an uniform cross-over instead of a single or double point one following Syswerda (1989).

3 Experiment results

The framework described in the paper was implemented as follows: We employed a population of 2000 chromosomes, discarding 25% of the worse-fitted ones in each cycle. The vacant places were filled with 40% chromosomes generated by mutation and 60% by cross-over. The mutation operator was applied with a 40% probability of performing a node insertion or deletion and 60% chance of choosing a shuffle mutation. The population was started from a chromosome with one root node connected to a random ordering of the 82 operators and then nodes were inserted and shuffled 40 times.[7]

The search algorithm was executed by 20 generations in 8d 14h (total CPU time, using about 20 machines in parallel to compute the verbalizations[8]). The best chromosome obtained was compared with the current planner in an intrinsic evaluation, described below.

Given the size of both planners, an automatic evaluation process was needed. We use a metric that captures the structural similarities between the two planners. In our metric, we recorded for each pair of atomic operators, the list of internal nodes that **dominates** both of them. This information was used to build a matrix of counts with the size of such lists (for example, in the original planner "age-node" and "name-node" are both dominated by "demographics-node", "overview-node" and "discourse-node"; therefore, their entry in the table is the size of that list, i.e., 3). Given the fact that both the MAGIC planner and our learned planners have the same set of atomic operators, it is possible to score their similarity by subtracting the associated matrices and then computing the average of the absolute values of this difference matrix. Lower values will indicate closer similarity, with a perfect match receiving the value of 0. Applying this metric to the MAGIC planner and our best planner we obtained the score 1.16. We compare this score against 2000 random planners from the initial population of the genetic search. Taking the average of their scores we obtain 3.08.

Finally, if we compare our learned planner against the random ones we obtain 2.92. We clearly improve over this baseline. An example of the learned planner is seen in Figure 6 (d).

4 Related work

In recent years, there has been a surge of interest in empirical methods applied to natural language generation (Columbia, 2001). Some work on content planning also deals with constructing plans from semantically annotated input. Kan and McKeown (2002) use an n-gram model for ordering constraints. The approach is complemen-

[6] not implemented in the current set of experiments.

[7] this figure was picked to obtain trees with $height \approx 4$.

[8] each regeneration involves complex unification processes timing 31' on average in a PIII 1Ghz

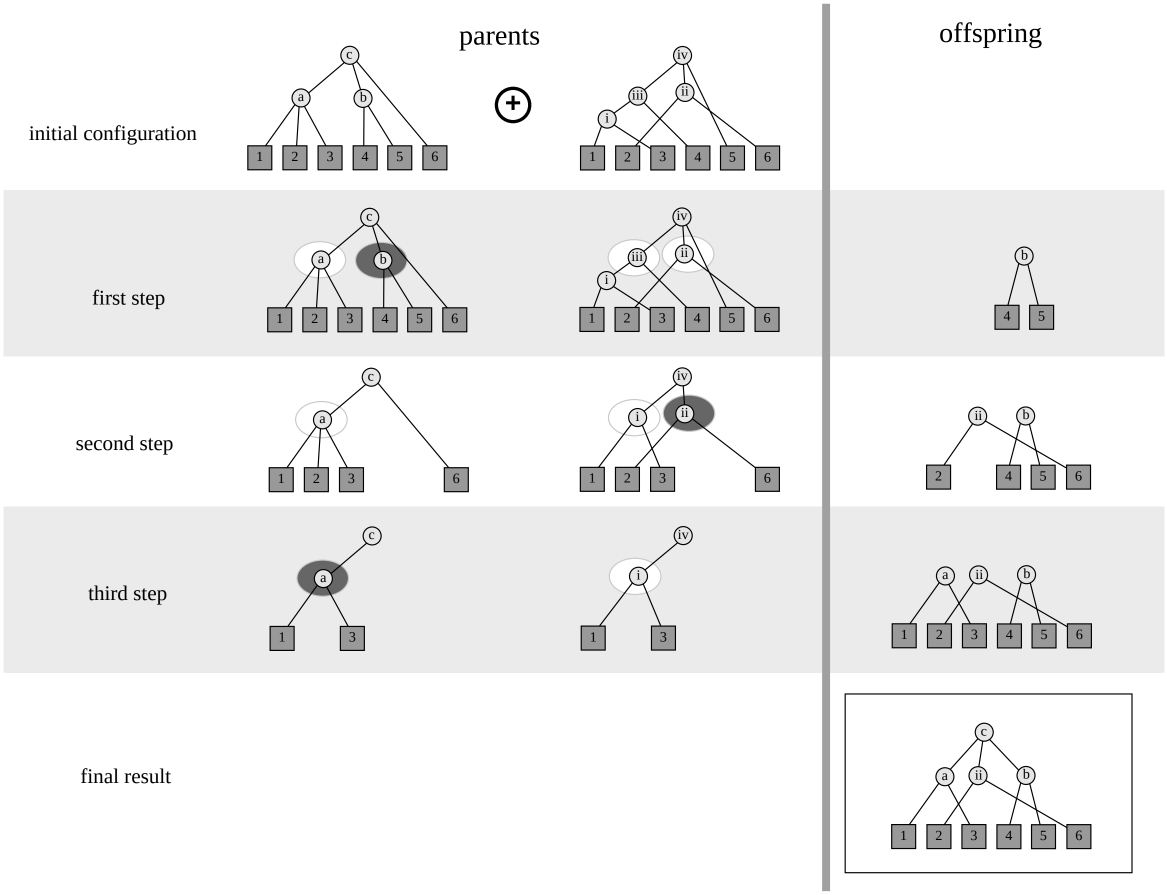

Figure 3: Cross over between chromosomes. Two trees are merged by picking subtrees of them (trees rooted at the dark circled nodes in 1st and 2nd step). Once a tree is moved to the offspring, all its leaves (and connecting nodes) are removed from both trees (note that node 6 is removed from both trees in step 3). The process continues until the parents are empty. This algorithm was chosen to maximize the amount of structure from the parents preserved in the offspring, its inspired on Bickel and Bickel (1987). Note that the first parent represents the ordering 123456, the second parent, 134265 while the offspring is 132645.

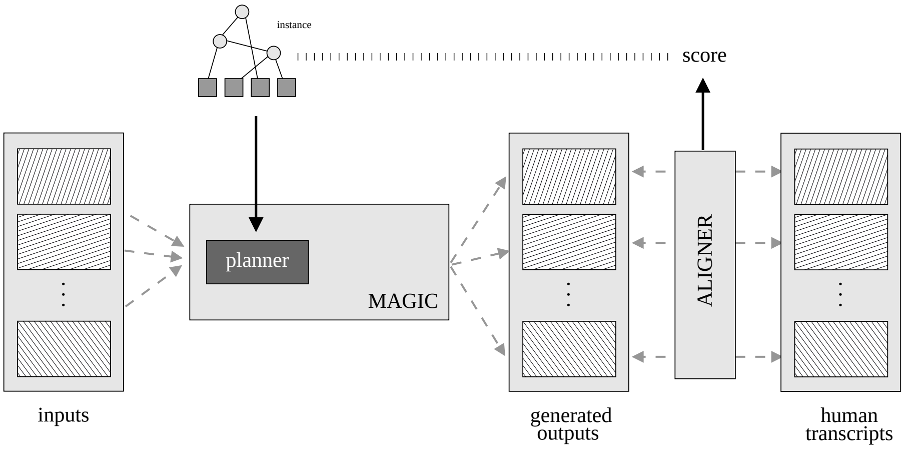

Figure 5: Fitness function: Alignment architecture.

`(patient-info-12865, c-patient, (a-age, age-12865), (a-name, name-12865), (a-gender, gender-12865), (a-birth-date, ...), ..., (r-receive-blood-product, received-BloodProduct1-12865), ...)`
`(age-12865, c-measurement, (a-value, 38), (a-unit, "year"))` *maps to sentence 1 (b)*
`(ht-12865, c-measurement, (a-value, 175), (a-unitm "centimeter"))` *maps to sentence 1 (b)*
`(name-12865, c-name, (a-first-name, "John"), (a-last-name, "Doe"))` *maps to sentence 1 (b)*
. . .
`(received-BloodProduct1-12865, c-receive-blood-product, (r-arg2, BloodProcut1-12865), (a-dosage, Measure-BloodProduct1-12865))` *maps to sentence 5 to last (b)*
`(BloodProduct1-12865, c-blood-product, (a-name, ‘‘Cell Savers’’))` *maps to sentence 5 to last (b)*
`(Measure-BloodProduct1-12865, c-measurement, (a-value, 3.0), (a-unit, ‘‘unit’’))` *maps to sentence 5 to last (b)*
. . .

(a)

John Doe is a 41 year-old male patient of Dr. Smith undergoing mitral valve repair. His weight is 92 kilograms and his height 175 centimeters. Drips in protocol concentrations include Dobutamine, Nitroglycerine and Levophed. He received 1000 mg of Vancomycin and 160 mg of Gentamicin for antibiotics. Around induction, he was anesthetized with 130.0 mg of Rocuronium, 11.0 mg of Etomidate, 500.0 mcg of Fentanyl and 1.0 mg of Midazolam. Before start of bypass , he had hypotension, at start of bypass, alkalosis, before coming off bypass, bradycardia and after coming off bypass, hypotension and relative-anemia. He received three units of cell savers. His total cross clamp time was 2.0 hour 1.0 minute. His total bypass time was 2.0 hour 33.0 minutes. His pre-op cardiac output was 4.13. Cardiac output immediately off was 4.73 .

(b)

Approximately 175-cm gentleman. History of rheumatic fever and polio. He is nonambulatory but can move his legs. History of acute renal insufficiency with a hematocrit of 1.4. History of mixed mr/ms lesion, tricuspid regurg and ai. Decreased right and left sided function, 4 chamber dilatation. Tricuspid repair with the ringand mvr with a st. jude’s valve. History of pulmonary hypertension with a baseline of 90/40 catheter. He was on heparin nph preop. No allergies. Feed and . . . . lines were extubated he was on bypass approximately 2.5 hours. His ischemic time was 2 hours and 2 minutes. No problems. He came off on dobutamine because of poor function. No problems post-bypass. Maintained on levo, nitro and dobutamine at 4.5 mcg per kilo. Got vancomycin and gentamicin at 9 o’clock, standard iv anesthetics. He received a liter of albumin, 3 units of cell saver, no exogenous blood. Last po2 was 453, potassium of 4.6, hematocrit of 26, before getting any blood gas. His cardiac output with his chest closed

(c)

The patient is male. He had an easy intubation. Before coming off bypass, he had bradycardia. Drips in protocol concentrations include Dobutamine, Nitroglycerine and Levophed. At start of bypass, he had alkalosis. After coming off bypass, he had relative-anemia. Around induction, he was anesthetized with 130.0 mg of Rocuronium, 11.0 mg of Etomidate, 500.0 mcg of Fentanyl and 1.0 mg of Midazolam. His weight is 92 kilograms and his pre-op cardiac output 4.13.

(d)

Figure 6: Examples. (a) Semantic input excerpt. (b) MAGIC output. (c) Physician briefing. (d) Learned planner output.

tary and suitable for scenarios where semantic annotation is an inexpensive task, such as in the automatic summarization tasks presented in that paper. Also working on order constraints for summarization, Barzilay et al. (2002) collect a corpus of ordering preferences among subjects and use them to estimate a preferred ordering.

Evolutionary algorithms were also employed be Mellish et al. (1998) for content planning purposes. While their intention was to push stochastic search as a feasible method for implementing a content planner and we pursue the automatic construction of the planner itself, both systems produce a tree as output. Our system, however, uses a corpus-based fitness function, while they use a rhetorically-motivated heuristic function with hand-tuned parameters.

Our approach is similar to techniques employed in evolutionary algorithms to implement general purpose planners,such as SYNERGY (Muslea, 1997); or to induce grammars (Smith and Witten, 1996). In general, all these approaches are deeply tied to Genetic Programming (Koza, 1994), that deals with the issue of how to let a computer program itself.

Our two-level fitness-function employs a lower-order function for the initial approximation of solutions in a process similar to the one taken by Haupt (1995) in a very different domain. This technique is appropriate for dealing with expensive fitness functions.

Finally, our approach with respect to human-produced discourse as gold standard is similar to Papinini et al. (2001) as it avoids adhering to the particularities of one specific person or discourse.

5 Conclusions and Further Work

The task of learning a general content planner such as Moore and Paris (1992) is well beyond the state of the art. We have identified reduced content planning tasks that are feasible for learning. In learning for traditional AI planning, e.g., learning of planning operators (García-Martínez and Borrajo, 1997), the focus is on **reduced** planning environments. The kind of discourse targeted by our current techniques has been identified in the past as rich in domain communication knowledge (Kittredge et al., 1991). We identify such discourse as non-trivial scenarios in which investigation of learning in content planners is feasible.

In searching for an appropriate solution, a function that enables the computer tell from one solution to another is always needed. A second contribution of this paper falls in our proposed fitness function, motivated by generation issues behind the content planning in generation. By means of a powerful 2-level fitness function we obtain a 66% improvement over a random baseline (the one we started from) in just 20 generations.

The problem with costly functions is always execution time. By using F_C, the constraint-based fitness function, we speed up the process, computing only 55K regenerations from a total of 187.5K. Our proposed cross-over and mutation operators are also well-suited for the task and are the result of our analysis of the generation domain.

Moreover, our technique achieves results without the need of extensive semantic annotation nor large sized input.

We are interested in extending this work by migrating it to other domains. We also plan to investigate the quality of the obtained planners head to head with our existing system. We want to see if it is possible to improve the existing planner (extensive evaluation with human subjects is required), as the newly obtained output resembles more normally occurring discourse in the domain.

Aside from the structure discovery, techniques are required for automatically detecting and building **messages** (Reiter and Dale, 2000). This task has been reported as extremely costly in the past (Kukich, 1983). We believe there are good chances of combining techniques such as (Duboue and McKeown, 2001) and (Barzilay and Lee, 2002) to semi-automate its creation process.

References

Regina Barzilay and Lillian Lee. 2002. Bootstrapping lexical choice via multiple-sequence alignment. In *EMNLP-2002*, Philadelphia, PA.

Regina Barzilay, Noemie Elhadad, and Katheleen McKeown. 2002. Inferring strategies for sentence ordering in multidocument news summarization. *JAIR*.

A.S. Bickel and R.W. Bickel. 1987. Tree structured in genetic algorithms. In L. Davis, editor, *Genetic Algorithms and Simulated Annealing*. Pittman.

NLP Group Columbia. 2001. Columbia statistical generation day. `http://www.cs.columbia.edu/nlp/sgd/`.

Pablo A. Duboue and Kathleen R. McKeown. 2001. Empirically estimating order constraints for content planning in generation. In *Proceedings of the 39th Annual Meeting of the Association for Computational Linguistics (ACL-EACL 2001)*, Toulouse, France, July.

Richard Durbin, Sean Eddy, Anders Krogh, and Graeme Mitchison, 1998. *Biological sequence analysis*, pages 17–28. Cambridge Univeristy Press.

Michael Elhadad and Jacques Robin. 1996. An overview of surge. Technical report, Dept. of Mathematics and C.S., Ben Gurion University, Beer Sheva, Israel.

Ramón García-Martínez and Daniel Borrajo. 1997. Planning, learning and executing in autonomous systems. In Steel and Alami, editors, *Recent Advances in AI Planning*, pages 208–220. Springer.

L.R. Haupt. 1995. Optimization of highly aperiodic conduction grids. In *11th Annual Review of Progress in Applied Comp. Electromag. Conf.*, Monterrey, CA.

Desmond Jordan, Kathleen McKeown, K.J. Concepcion, S.K. Feiner, and V. Hatzivassiloglou. 2001. Generation and evaluation of intraoperative inferences for automated health care briefings on patient status after bypass surgery. *J. Am. Med. Inform. Assoc.*, 8:267–280.

Min-Yen Kan and Kathleen R. McKeown. 2002. Corpus-trained text generation for summarization. In *Proceedings of INLG-2002*, Ramapo Mountains, NY.

Richard Kittredge, Tanya Korelsky, and Owen Rambow. 1991. On the need for domain communication language. *Computational Intelligence*, 7(4):305–314.

J. Koza. 1994. *Genetic Programming II*. MIT Press.

Karen Kukich. 1983. Knowledge-based report generations: A technique for automatically generating natural language reports from databases. In *Sixth ACM SIGIR Conference*, pages 246–250, Bethesda, MA.

Benoit Lavoie and Owen Rambow. 1997. A fast and portable realizer for text generation systems. In *Proc. of ANLP'97*, Washington, DC.

Katheleen McKeown, Shimei Pan, James Shaw, Jordan D., and Barry A. 1997. Language generation for multimedia healthcare briefings. In *Proc. of ANLP'97*.

Kathleen McKeown, Desmond Jordan, Steven Feiner, J. Shaw, E. Chen, S. Ahmad, A. Kushniruk, and V. Patel. 2000. A study of communication in the cardiac surgery intensive care unit and its implications for automated briefing. In *Proc. of the AMIA 2000*.

Chris Mellish, Alistair Knott, Jon Oberlander, and Mick O'Donnell. 1998. Experiments using stochastic search for text planning. In *Proc. of the 9th International Workshop on Natural Language Generation*, pages 98–107, Niagra-on-the-Lake, Ontario, Canada.

Steven Minton, editor. 1993. *Machine learning methods for planning*. Morgan Kaufmann series in machine learning. M. Kaufmann, San Mateo, CA.

Johanna D. Moore and Cecile L. Paris. 1992. Planning text for advisory dialogues: Capturing intentional and rhetorical information. *Computational Linguistics*, 19(4):651–695.

Ian Muslea. 1997. A general-purpose AI planning system based on the genetic programming paradigm. In *Late Breaking Papers at GP-97*, pages 157–164.

Shimei Pan and Katheleen McKeown. 1999. Word informativeness and automatic pitch accent modeling. In *Proc. of EMNLP/VLC'99*, College Park, MD.

Kishore Papinini, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2001. Bleu: a method for automatic evaluation of machine translation. Technical report, IBM.

Ehud Reiter and Robert Dale, 2000. *Building Natural Language Generation Systems*, pages 61–63. Cambridge University Press.

Ehud Reiter. 1994. Has a consensus nlg architecture appeared and is it psychologically plausible? In *Proc. of 7th IWNLG*, pages 163–170.

J.T. Richardson, M.R. Palmer, G. Liepins, and M. Hilliard. 1989. Some guidelines for genetic algorithms with penalty functions. In J.D. Schaffer, editor, *Proc. of the Third Intl. Conf. in Genetic Algorithms*, pages 191–197. Morgan Kaufmann, Los Altos, CA.

James Shaw. 1998. Clause aggregation using linguistic knowledge. In *Proc. of 9th International Workshop on Natural Language Generation*, pages 138–147.

Tony C. Smith and Ian H. Witten. 1996. Learning language using genetic algorithms. In Riloff and Scheler, editors, *Connectionist, statistical, and symbolic approaches to learning for natural language processing*, pages 133–145. Springer.

Gilbert Syswerda. 1989. Uniform crossover in genetic algorithms. In J.D. Schaffer, editor, *Proc. of the Third Intl. Conf. in Genetic Algorithms*, pages 2–9. Morgan Kaufmann, Los Altos, CA.

Michael R. Young and Johanna D. Moore. 1994. DPOCL: A principled approach to discourse planning. In *Proc. of 7th IWNLG*, Kennebunkport, ME.