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

Pablo A. Duboue and Kathleen R. McKeown

Computer Science Department
Columbia University
in the city of New York

# Motivation

- Importance of Content Planning
  - Tied to Semantics
  - Rich Structure
  - Exponential in Time
  - Lengthy Input
- Importance of Learning for Content Planning
  - Content Planning Requires Customization for Particular Domains
- We Want to Induce a Content Planner
  - From a Corpus of Texts and Semantics

## Structure of this talk

---

1. What we learn
   - *Tree-like planners*

2. How we learned it
   - *Stochastic search using an empirical fitness function*

3. How well the learned planners perform
   - *Experiences developed to test their goodness*

## Our domain: Cardiac surgery briefing (MAGIC)

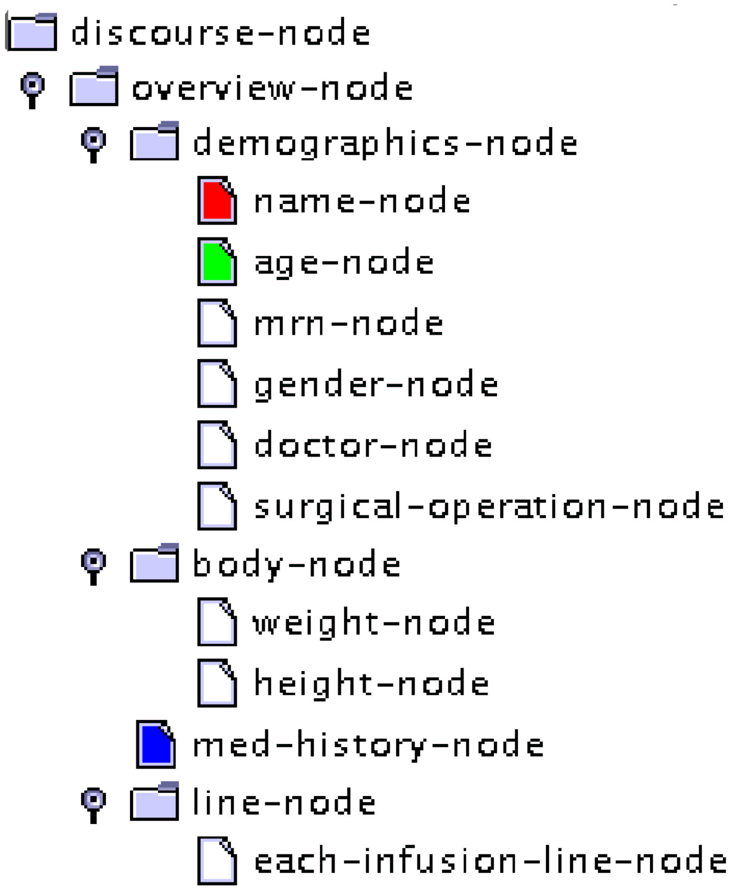
- this is **John Doe** he is **63 years** old 175 centimeters he has a triple vessel coronary artery disease lrd and rca he also has **non-insulin dependent diabetes milletus and hypertension** his symptoms were not just pain , but feeling of tiredness of the chest when he was walking uphill he 's on coumadine , . . . . . .

- mr. **James Smith** . **80 years** old medical history : **high blood pressure , coronary artery disease , status post acute mi , cardiogenic shock , . . . ischemic . . . cabg x 1** . on 11/1 , intra-aortic balloon pump insertion . . . present for preop for cabg x 1 . medication : primacor , heparin preop and imipenem antibiotics . no allergies . under general anesthesia , easy intubation , # 8 tube 8 cc air leaking . easy . . . . . .

# Domain Communication Knowledge

- The structure of the discourse in this type of reports is quite fixed.
- Such rigidity responds not only to logical reasons, but also to an accepted communication pattern in the domain.
- Kittredge, Korelsky and Rambow (1991) defined this type of discourse as rich in Domain Communication Knowledge or DCK.
- DCK-rich discourse is more suitable to be modeled by means of schemas or other structure-strong methodologies.

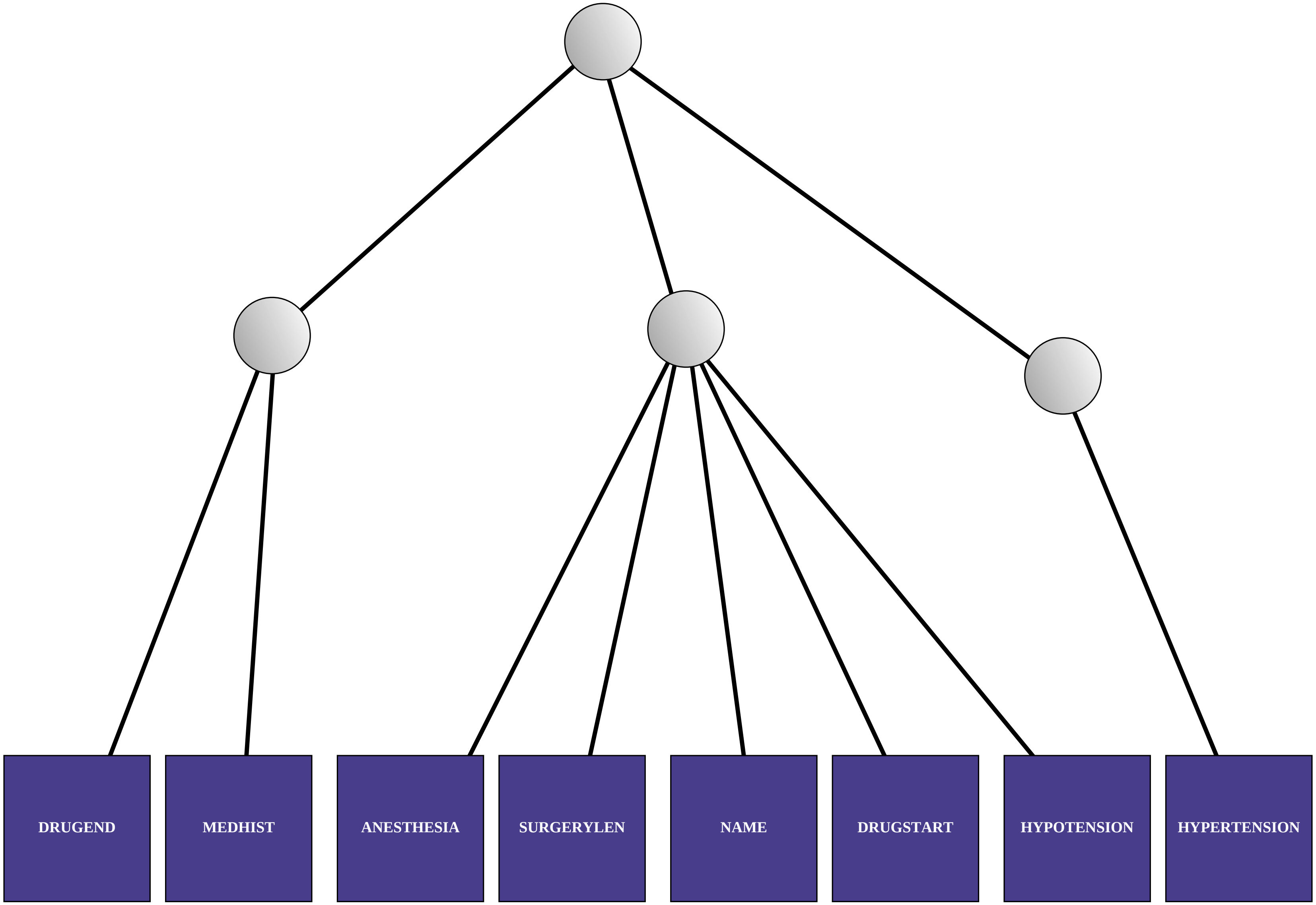
# MAGIC planner: Example




"J. Doe is a seventy-eight year-old male patient of Doctor Smith undergoing aortic valve replacement. He is sixty-six kilograms and one hundred sixty centimeters. His medical history includes allergy to penicillin and congestive heart failure.

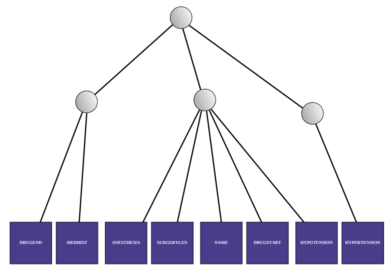
……"

# An arbitrary planner



# A planner in action

planner

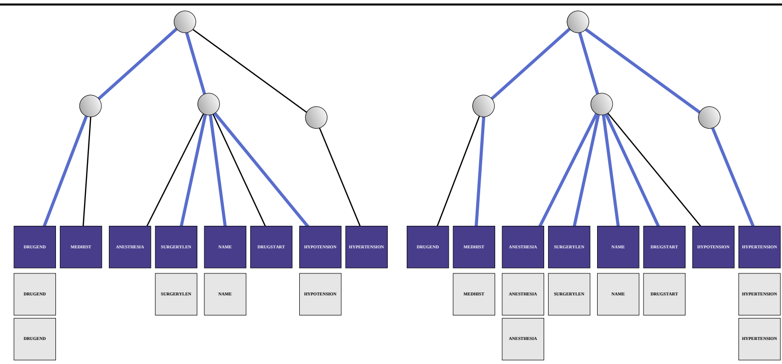



## Semantic input sets **(unordered)**

patient A
drugend-1,drugend-2,hypotension-1,name-1,surgerylen-1

patient B
anesthesia-1,anesthesia-2,drugstart-1,hypertension-1,hypertension-2,medhist-1,name-1surgerylen-1

planning

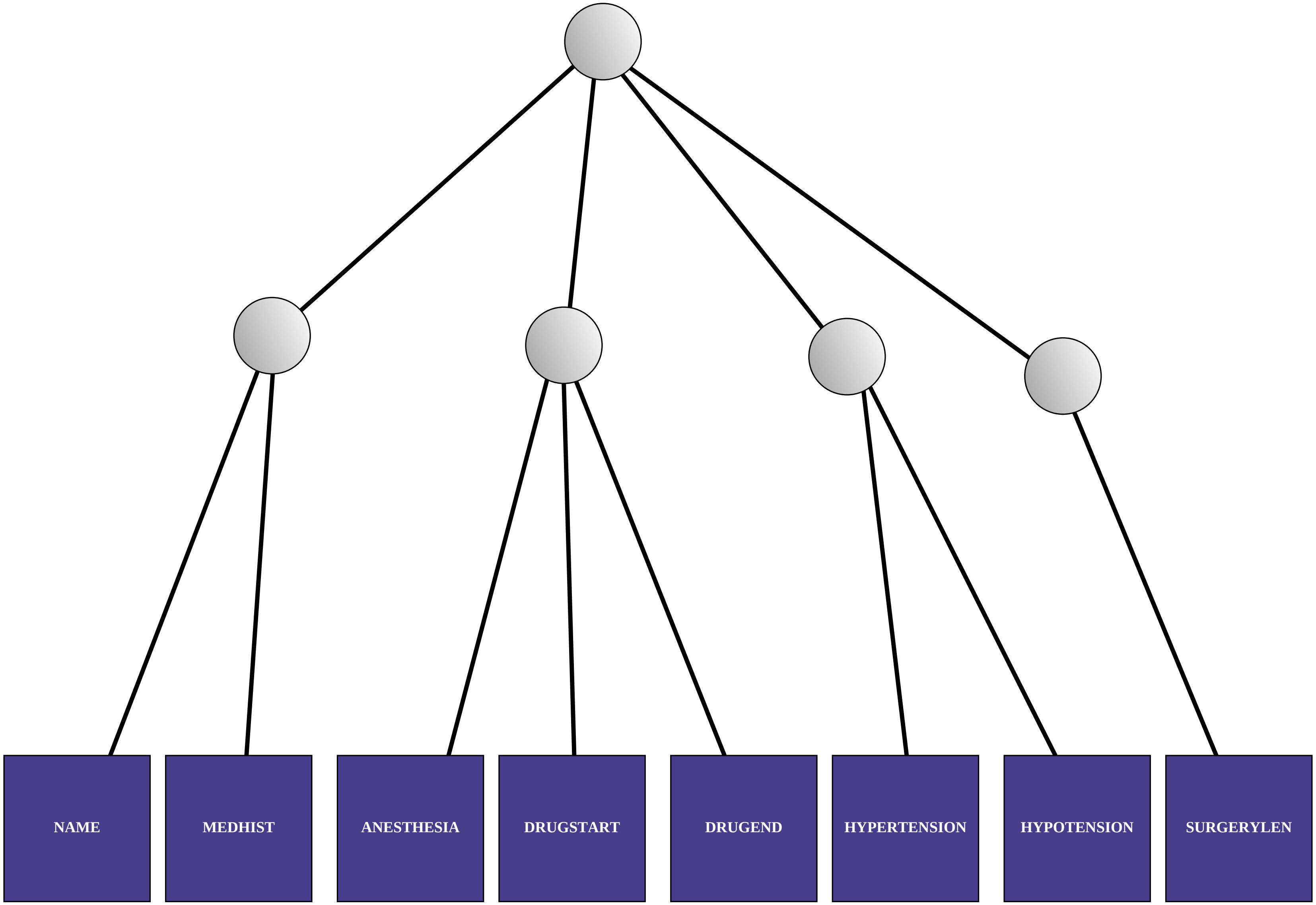



## Plans **(ordered)**

plan for A
drugend-1,drugend-2,surgerylen-1,name-1,hypotension-1

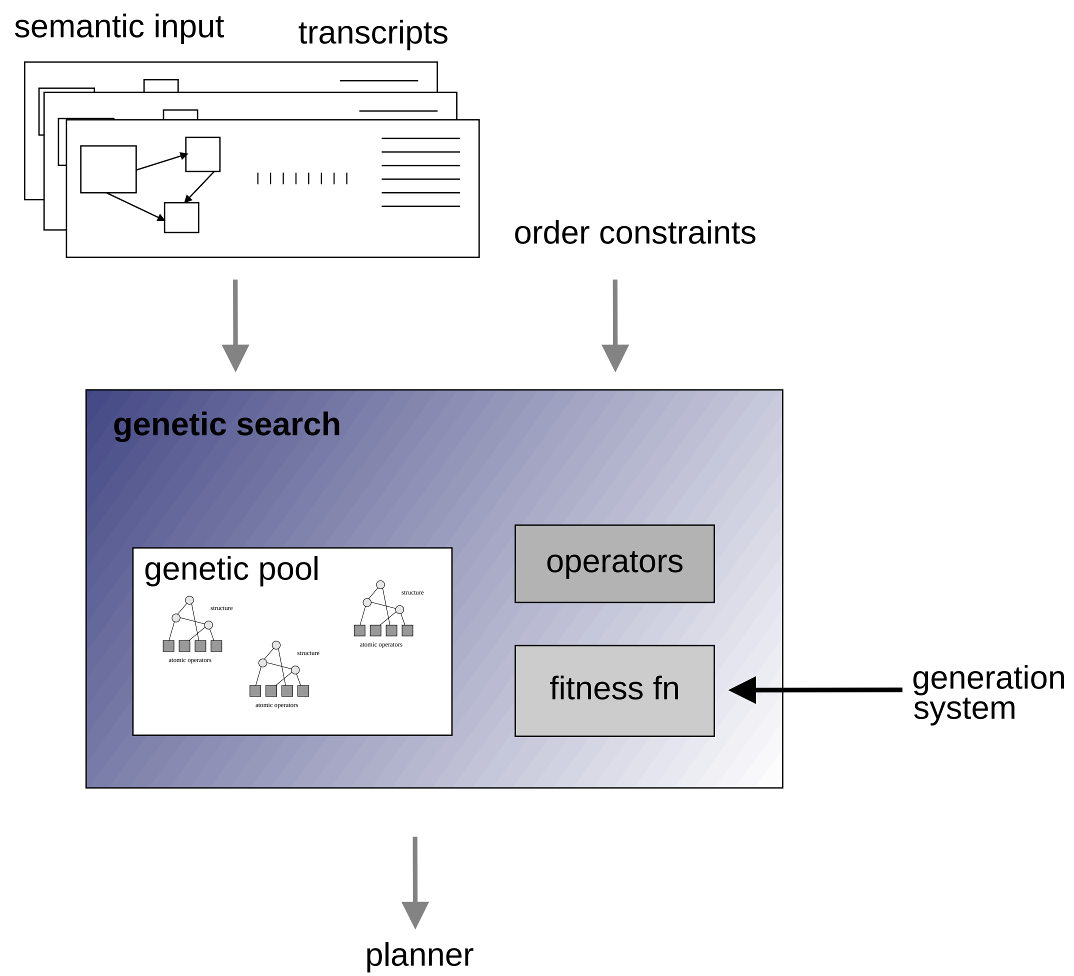
plan for B
medhist-1,anesthesia-1,anesthesia-2,surgerylen-1,name-1,drugstart-1,hypertension-1,hypertension-2

# A better planner

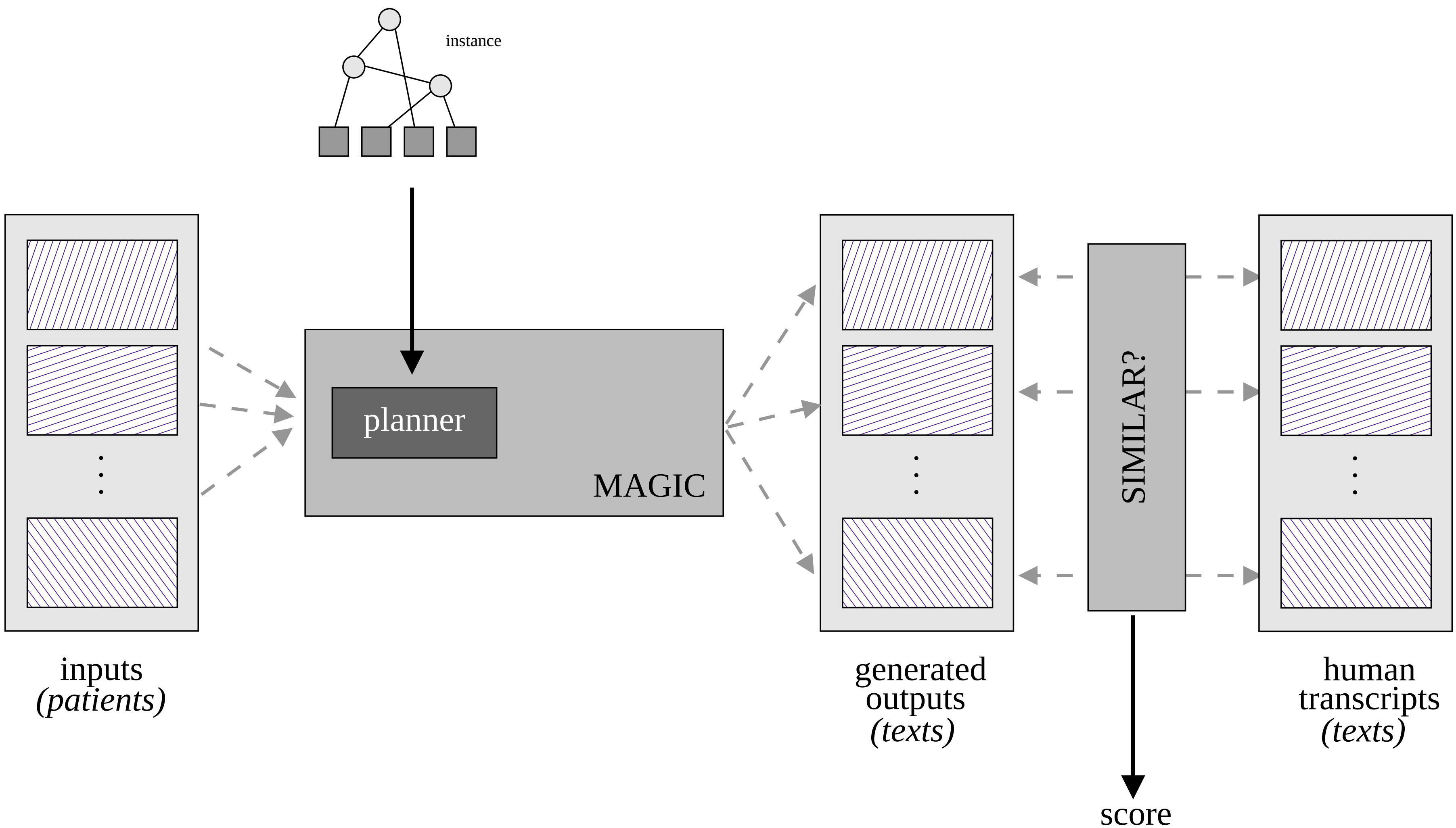


# Learning architecture

- Corpus collected during the evaluation described in McKeown et al. (2001).
- 25 patient data and 25 transcripts.
- In Duboue and McKeown (2000) we mined an annotated version of that corpus to extract ordered constraints between semantic elements.
- We use that corpus again, without annotations.



# How to tell the goodness of a planner



# Genetic Algorithms

- Genetic Algorithms (GAs) are a method to do stochastic search
  - Biological metaphor
  - Have been used in CP as a powerful technique to implement planners (Mellish et al. 1998)
  - Their use in NLP is growing
- They provide a good optimization technique to explore huge search spaces with highly interrelated features.
- We use them to explore the planners' space.

## Genetic Algorithms

- How they work
  - In a genetic search, at all times a **population** of possible solutions, called **chromosomes** is kept.
  - Each chromosome has an **associated fitness value**, indicating is apparent goodness.
  - In each step of the search, or **generation**, a percentage of the worst-fitted chromosomes is discarded.
  - The empty slots are filled by applying **operators**, creating new chromosomes by mixing two existing ones (sexual reproduction) or by making changes in a existing one (mutation).

## Genetic Algorithms: Characterization

- A GA then is defined by:
  - Chromosomes
  - Fitness function
  - Operators

# Genetic Algorithms: Characterization

- A GA then is defined by:
  - Chromosomes $\rightarrow$ planner trees
  - Fitness function
  - Operators

## Genetic Algorithms: Characterization

- A GA then is defined by:
  - Chromosomes $\rightarrow$ planner trees
  - Fitness function $\rightarrow$ alignment, orderings
  - Operators

## Genetic Algorithms: Characterization

- A GA then is defined by:
  - Chromosomes $\rightarrow$ planner trees
  - Fitness function $\rightarrow$ alignment, orderings
  - Operators $\rightarrow$ generation motivated

# Operators

- Our operators are motivated by the generation problem at hand.
- One reproductive operator (cross over).
  - Mixes two chromosomes by selecting top level subtrees from each one.
- Three asexual operators (mutations), given a random internal node:
  - Node Insertion: move some randomly selected children to a newly created subnode.
  - Node Deletion: absorb one of its children.
  - Shuffle: randomize the order of its children.
- New instances are created by insertion and shuffling.

# Fitness function

$$\boxed{if(F_C < 0)F_C, F_A}$$

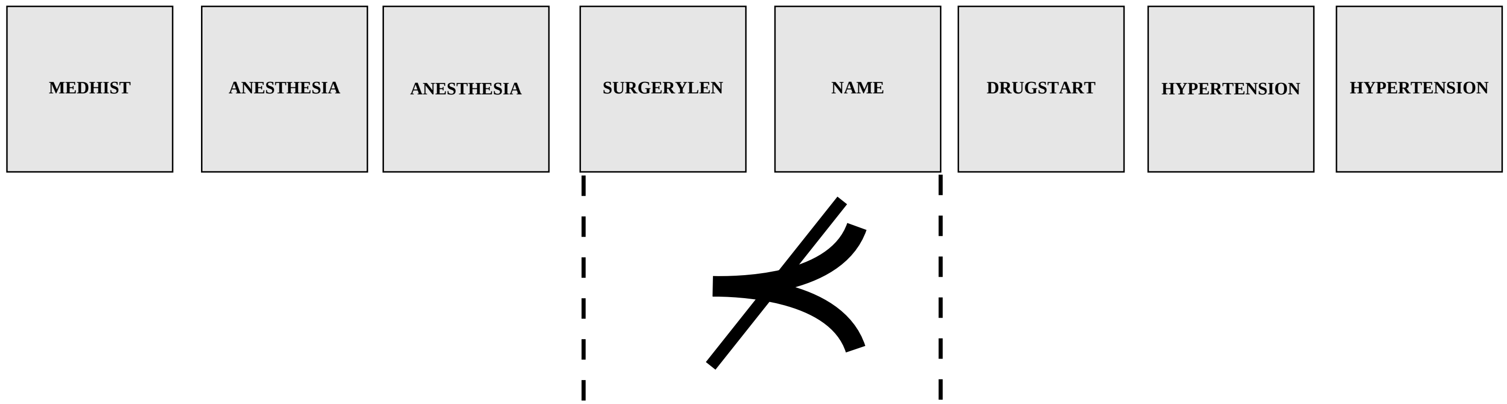
- We replace a computationally expensive fitness function with a faster but approximate one
  - in order to speed up the early stages of the population

- For the first stages, we use a constraint-based function, $F_C$, using the ordering constraints mined by Duboue and McKeown (2001).

- If the results returned by it are good enough, we turn to a full-fledged alignment-based function $F_A$.

## $F_C$: Constraint-based fitness function

- This function works as follows:
  - given a set of semantic inputs;
  - the chromosome is used to generate corresponding plans;
  - then order constraints are checked for validity.




- The actual elements being compared include higher level constructions.

# $F_A$: Alignment-based fitness function

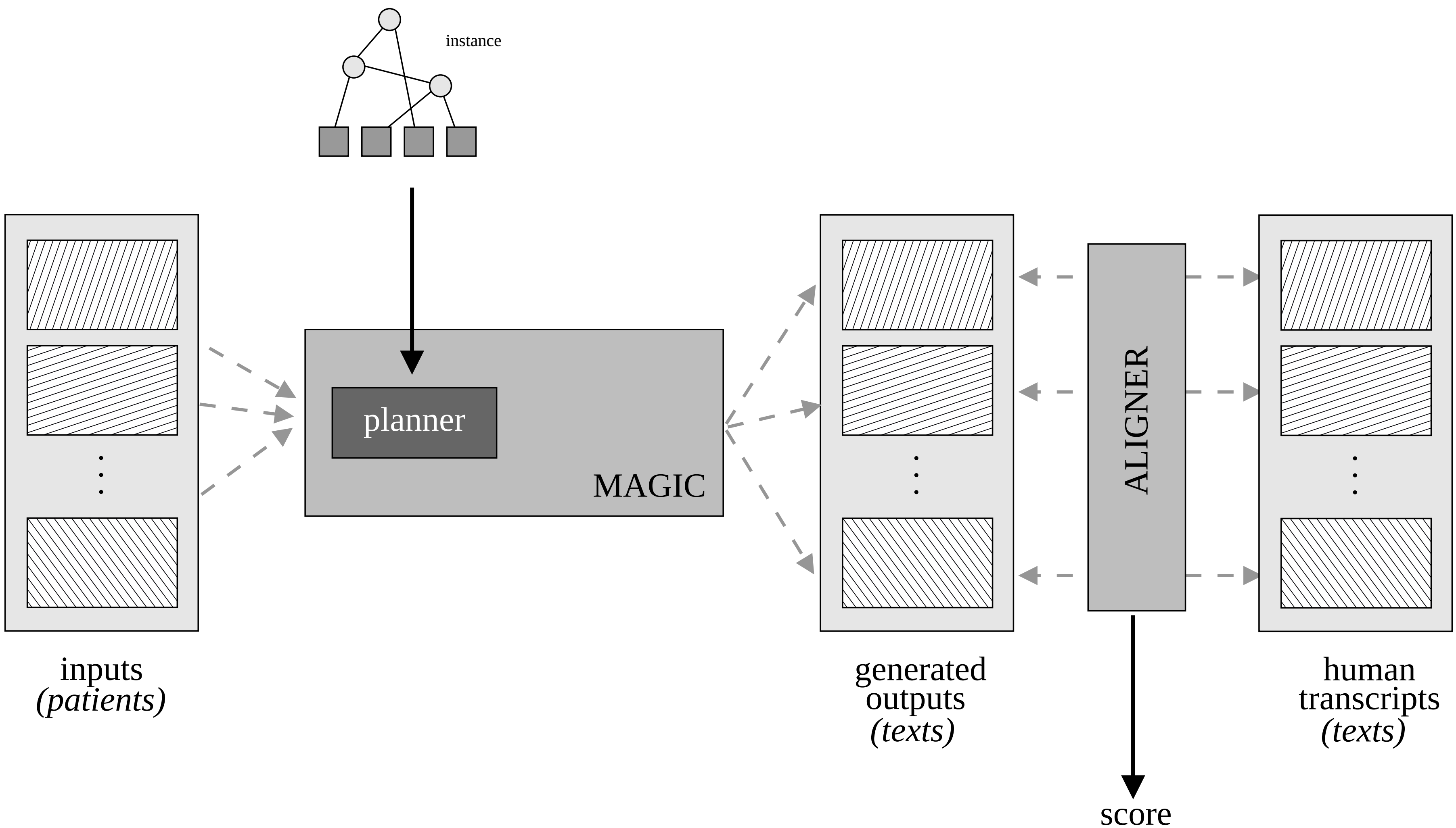
- Our alignment-based fitness function takes a chromosome an plug it inside our MAGIC system, replacing its hand-built planner

- The modified generation system is used with the 25 semantic inputs to generate 25 reports

- Each of these reports is measured for similarity against the transcript of the human report given at the time the semantic input was acquired

- The similarity scores are then averaged to result on the fitness value for the chromosome.

# $F_A$: Alignment-based fitness function



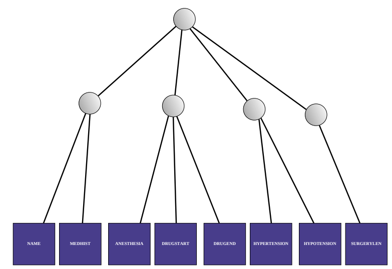
# $F_A$: An example (Inputs)

- Three inputs **(unordered)**
  1. drugend-1,drugend-2,hypotension-1,name-1,surgerylen-1
  2. drugend-1,hypertension-1,name-1,surgerylen-1
  3. drugstart-1,medhistory,name-1,surgerylen-1

- Transcripts **(human produced)**
  1. John Doe had one episode of hypotension. By the end of the surgery he received Dopamine and Levophed. His total bypass time was 2h and 30m.
  2. Jane Doe had one episode of hypertension and by the end of case she received Vecuronium. She was in surgery for 1h and 15m.
  3. James Smith is a diabetic patient that underwent a 3h surgery. We gave him Dopamine after induction.

# $F_A$: An example (Planning)

- Using the improved planner,




- Generated plans

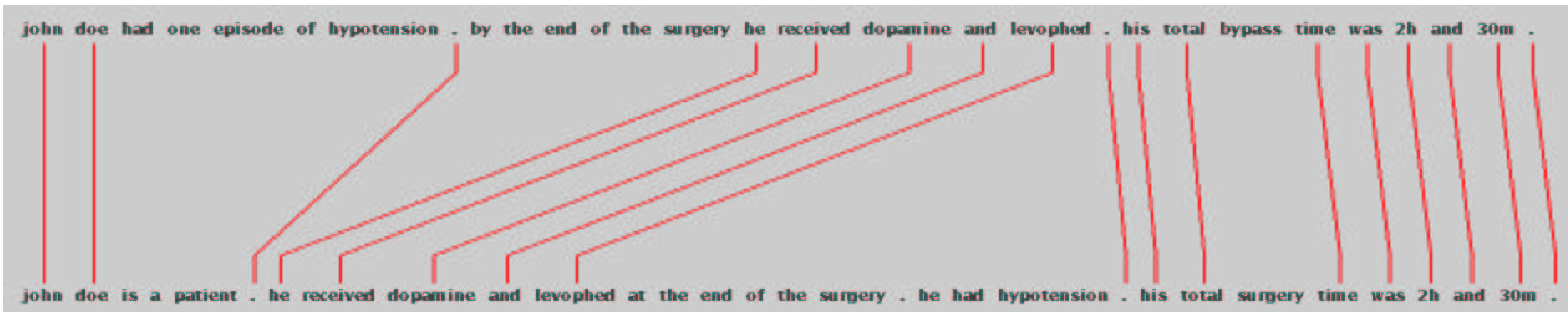
  1. **name-1**,drugend-1,drugend-2,hypotension-1,surgerylen-1
  2. name-1,drugend-1,**hypertension-1**,surgerylen-1
  3. name-1,**medhistory**,drugstart-1,surgerylen-1

- Output from generator

  1. **John Doe is a patient.** He received Dopamine and Levophed at the end of the surgery. He had hypotension. His total surgery time was 2h and 30m.
  2. Jane Doe is a patient. She received Vecuronium at the end of the surgery. **She had hypertension.** Her total surgery time was 1h and 15m.
  3. James Smith is a patient. **He has a past medical history of diabetes.** He received Dopamine at the start of the surgery. His total surgery time was 3h.

## $F_A$: An example (Alignments)

- Alignment (first patient)




- This alignment produce an score that is then averaged over the different patients.

# Alignments

- Pairwise alignments computed using the Needleman–Wunsch algorithm, as defined by Durbin et al. (1998).
- A dynamic programming-based algorithm that computes global alignments.
- Using an affine gap penalty.
- These alignments do not allow flipping:

$$A\text{–}B\text{–}C$$
$$C\text{–}B\text{–}A$$

*will only recover the alignment of $B$.*

- They capture the notion of ordering more appropriately for our needs.
- Algorithm adapted to use the information content of words:
  - measured in a 1M-token corpus of related discourse.
  - estimates the goodness of substituting one word by another.

# Experiments

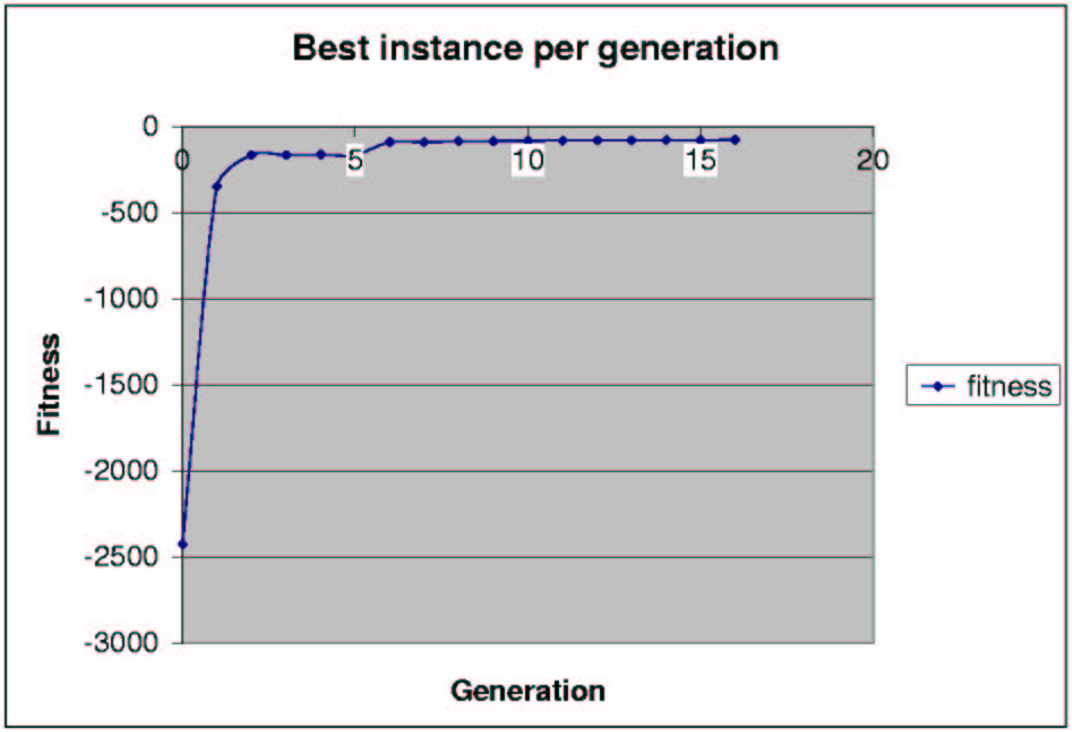
- To evaluate the ideas explained before we performed a series of experiments, using 25 data/text pairs from the MAGIC system evaluation (McKeown et al. 2001).

- A population of 2000 chromosomes was kept, discarding the 25% worst fitted chromosomes in each cycle.

- This population was growth for an average of 16 generations, in ten independent experiments.

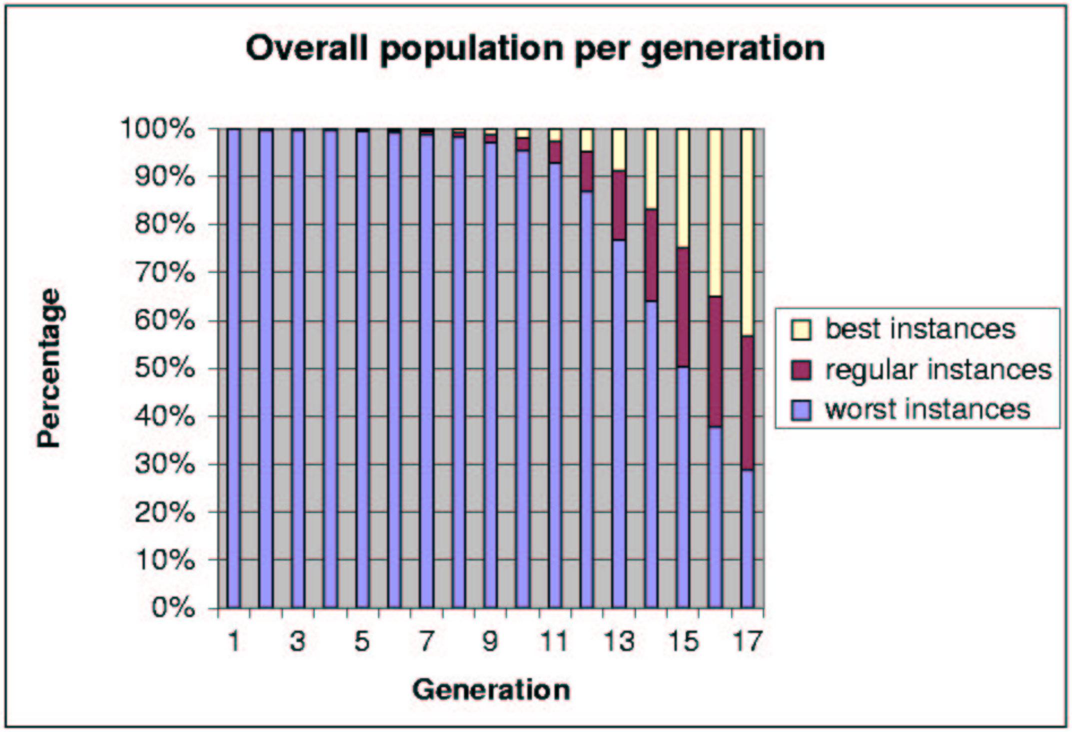
# Learning curve: Best Chromosome

- The learning process can be seen as we follow the evolution of the fitness value of the best instance in the population.



# Learning curve: Overall Population

- Another way to appreciate the learning process is to take a look at the goodness of the population as a whole, on different generations.



## Metric$_1$ (structural)

- This metric computes how similar a planner is to a gold standard by looking at them as trees.
- Both planners have the same leaf-set, while their internal nodes are arbitrary.
- We count the number of common ancestors between each pair of nodes and record this in a matrix corresponding to each planner.
- The average difference between the two matrices reflect the level of similarity between the two structures. A value of 0 implies perfect match.
- This metric does not capture ordering

# Metric$_1$: Number of common ancestors (Example)

Plan I

| | A | D | E | H | I | N | M | S |
|---|---|---|---|---|---|---|---|---|
| anesthesia A | 2 | 1 | 2 | 1 | 2 | 2 | 1 | 2 |
| drugend D | 1 | 2 | 1 | 1 | 1 | 1 | 2 | 1 |
| drugstartE | 2 | 1 | 2 | 1 | 2 | 2 | 1 | 2 |
| hypertension H | 1 | 1 | 1 | 2 | 1 | 1 | 1 | 1 |
| hypotension I | 2 | 1 | 2 | 1 | 2 | 2 | 1 | 2 |
| name N | 2 | 1 | 2 | 1 | 2 | 2 | 1 | 2 |
| medhist M | 1 | 2 | 1 | 1 | 1 | 1 | 2 | 1 |
| surgerylen S | 2 | 1 | 2 | 1 | 2 | 2 | 1 | 2 |

Gold Standard

| | A | D | E | H | I | N | M | S |
|---|---|---|---|---|---|---|---|---|
| A | 2 | 2 | 2 | 1 | 1 | 1 | 1 | 1 |
| D | 2 | 2 | 2 | 1 | 1 | 1 | 1 | 1 |
| E | 2 | 2 | 2 | 1 | 1 | 1 | 1 | 1 |
| H | 1 | 1 | 1 | 2 | 2 | 1 | 1 | 1 |
| I | 1 | 1 | 1 | 2 | 2 | 1 | 1 | 1 |
| N | 1 | 1 | 1 | 1 | 1 | 2 | 2 | 1 |
| M | 1 | 1 | 1 | 1 | 1 | 2 | 2 | 1 |
| S | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 2 |

Difference

| | A | D | E | H | I | N | M | S |
|---|---|---|---|---|---|---|---|---|
| A | 0 | 1 | 0 | 0 | 1 | 1 | 0 | 1 |
| D | 1 | 0 | 1 | 0 | 0 | 0 | 1 | 0 |
| E | 0 | 1 | 0 | 0 | 1 | 1 | 0 | 1 |
| H | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 |
| I | 1 | 0 | 1 | 1 | 0 | 1 | 0 | 1 |
| N | 1 | 0 | 1 | 0 | 1 | 0 | 1 | 1 |
| M | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 0 |
| S | 1 | 0 | 1 | 0 | 1 | 1 | 0 | 0 |

Difference matrix average (metric$_1$): $\frac{32}{64} = 0.5$

## Metric$_2$ (linear)

- To measure ordering behavior, we align the output of our gold standard planner (MAGIC), against the output of the evaluated planner.

- We average this over a set of semantic inputs (different from the ones used for learning).

# Results

| System | $\text{Metric}_1$ | $\text{Metric}_2$ |
|---|---|---|
| **learned planners** | 1.82 | 45.25 |
| **baseline** | 3.06 | 2.54 |

- As baseline, we use the initial population of the ten runs (20K randomly built planners in total).
- The MAGIC planner was used as our gold standard
  - It has been previously evaluated by domain experts as highly accurate.
  - It was not involved in any part of the our learning process.

# Conclusion

- We developed a fitness function for content planners
  - Based on alignments between generated text and human text
  - Speed-up by using order constraints
- A suitable planner representation
  - It can be learned.
  - It has been used to solve problems in real domains
- Genetic Algorithms are a useful tool for learning content planners.
- Future work:
  - Characterize the domains where this technique is applicable
  - Improve and analyze the quality of the learned plans