NATURAL LANGUAGE GENERATION

D.Thenmozhi

March 31, 2017

Overview

Introduction

- Language generation is the oldest and most pervasive subfield of language processing.
- Natural Language Generation is the automatic generation of a set of natural language sentences for the given non-linguistic inputs

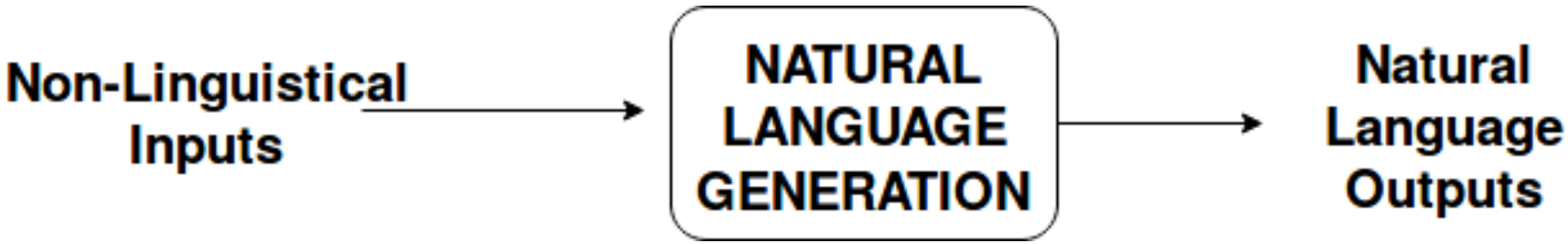


Figure: Natural Language generation

NLG and NLU

- **Natural Language Generation(NLG)** is the inverse of **Natural Language Understanding(NLU)**
- **NLU** - Maps from text to meaning

- **NLG**- Maps from Meaning to

Difference between NLU and NLG

- In NLU, all the linguistic inputs follow the common grammatical rules
- The system must carefully scan the input for alternate interpretations
- The **main concern** is to identify the ambiguity and ill formed sentences

- In NLG, the system addresses different applications with different input specifications
- The non-linguistical inputs are generally unambiguous,well specified and well formed
- The **main concern** is choice of words and discourse structure

Initial approaches to language generation

- **Canned Text** is easy to implement, but is unable to adapt to new situations without the intervention of a programmer. For example, the oldest and most famous C program, the "hello, world" program, is a generation program. It produces useful, literate English in context.
- **Template filling** is more flexible than canned text and has been used in a variety of applications, but is still limited. For example, a form letter with our name carefully inserted in just the right places, along with eloquent appeals for one thing or another.

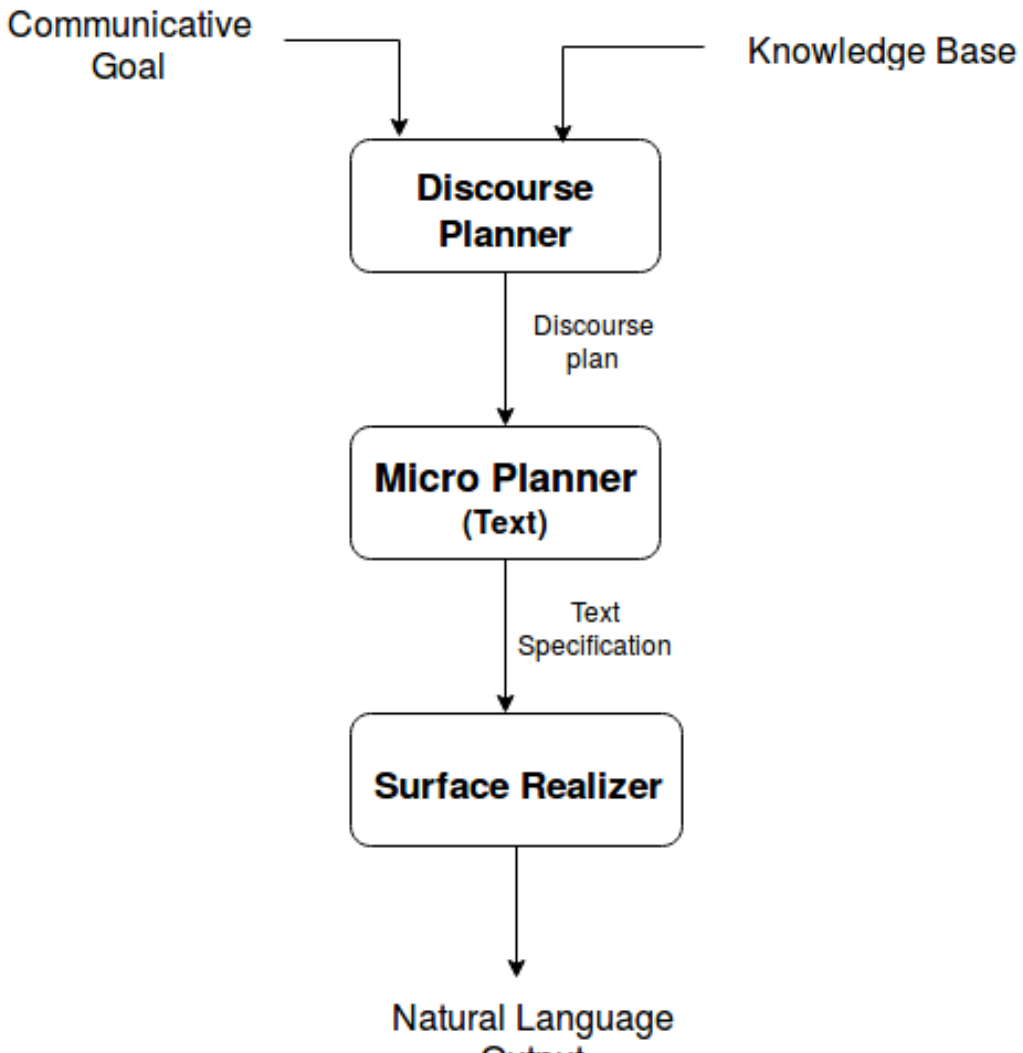
Disadvantages of the initial approaches

- Canned text and Template filling are **not flexible** enough to handle applications with **any realistic variation** in the information being expressed and in the context of its expression.
- Consider the text

> Congratulations, you've just compiled and run a simple C program which means that your environment is configured properly.

- If the readers are experienced systems engineers, then we might choose not to congratulate them on compiling a program because they are well aware about the configurations and coding environments
- Hence, the variations are not produced according to the context.

Architecture of Natural Language Generation



Inputs to the NLG System

- **Communicative Goal** is the overall purpose of the NLG system
- **Knowledge base** represents the data in a non-linguistic manner - one system may use simple tables of numbers, whereas another may use information encoded in some knowledge representation language
- **User Model** is a characterisation of the hearer or intended audience for whom the text is to be generated. It also contains information about the user‘s expertise, task, and preferences.
- **The discourse history** is a model of the previous interactions between the user and the nlg system. A discourse history may be no more than a list of the entities that have been mentioned in the discourse so far

Example : Weather Report Generation

- The **communicative goal** is that of summarising the data for a specified month, which we might represent as SummariseMonth(m).
- The **Knowledge base** represents the weather report in a non-liguistic form like tables.
- The **user model** could be used to determine what information is to be emphasised in the summary. For example, a summary for farmers might emphasise precipitation and temperature, whereas a summary for mariners might emphasise wind conditions.

- The **discourse history** of previous interactions could be used to relate the information in one summary to the information presented in previous summaries. For example, if the user first asks for a summary of the weather in May 1994 and then asks for a summary of the weather in May 1995, the response to the second request could contain explicit comparisons to the weather in May 1994.

Three modules of the NLG architecture

1. **Discourse Planner** - Decides the ordering and structure of the text. It selects the content from the knowledge base and then structures that content appropriately. The output is represented as a tree.
2. **Micro or Text planner**- Decides which words and phrases are used to express the concepts and relationship specified by the discourse planner.
3. **Surface realizer** -Takes the output of the text planner module and generates individual sentences.

Types of NLG architecture

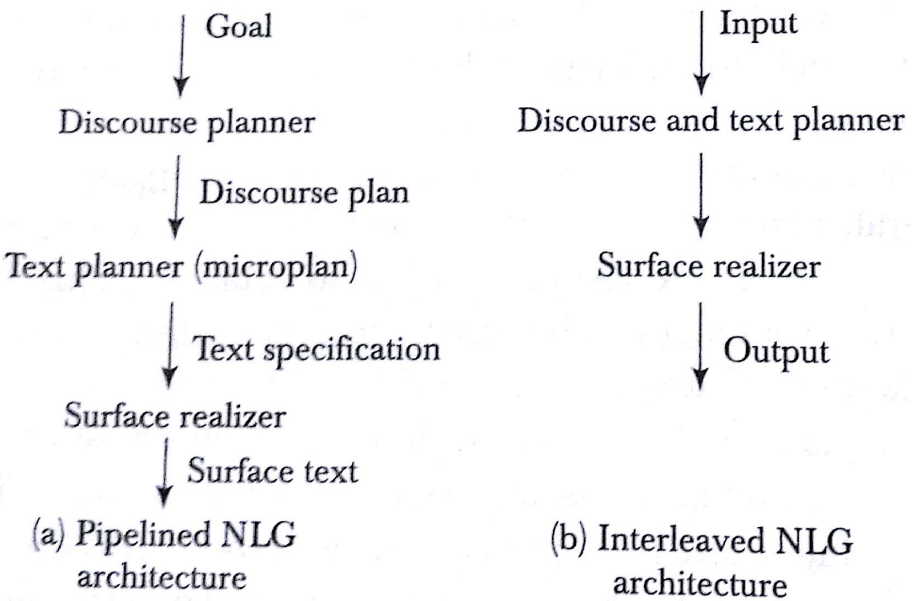


Figure 7.1 Natural language generation architecture

1. Pipelined architecture

- This architecture divides the NLG task into three distinct groups and pass the processed information from one stage into another in a pipelined fashion.
- **Disadv :** This requires that each stage must complete its task before the next stage.
- Consider the example sentences:
  1. He danced with Jaya and she got angry.
  2. He forced Jaya to dance and she got angry
- In the above sentences, the choice of 'dance'or 'forced to dance'can change the referent of 'she '
- In a **pipelined architecture**, commitments are made at each stage and the flow goes in the predefined path. The choices can be changed in an **interleaved architecture**.

2. Interleaved architecture :

- The discourse and text planner stages are merged into a single stage.
- Control and information can be passed back to any task in the two stages.
- Task of generation is completed in the

$$\text{"}planning \rightarrow realisation$$

$$\rightarrow planning \rightarrow \ldots\ldots\text{"}$$

- Example : PAULINE, MUMBLE NLG Systems

3. Integrated Architecture

- In an **Integrated architecture**, the sentences are generated in an interactive environment like a 'blackboard architecture'which can trigger rules or operators.
- In the integrated system, the task can be divided in to sub-tasks at each of the three stages. These orthogonal sub-tasks break the boundaries of the stages and all the stages are integrated into one.

Generation Task 1 : Discourse Planning - Rhetorical Structure Theory

- Rhetorical Structure Theory (RST), is a descriptive theory of text organization based on the relationships that hold between parts of the text. These relationships make the text coherent.
- There are 23 rhetorical relations to describe the structure of the text
- Consider the following two sentences
  1) I love to collect classic automobiles. My favorite car is my 1899 Duryea.
  2) I love to collect classic automobiles. My favorite car is my 1999 Toyota.
- In the first example, the second sentence elaborates the first sentence. Hence the relation is called **Elaboration**.
- In the second example, the second sentence contradicting the

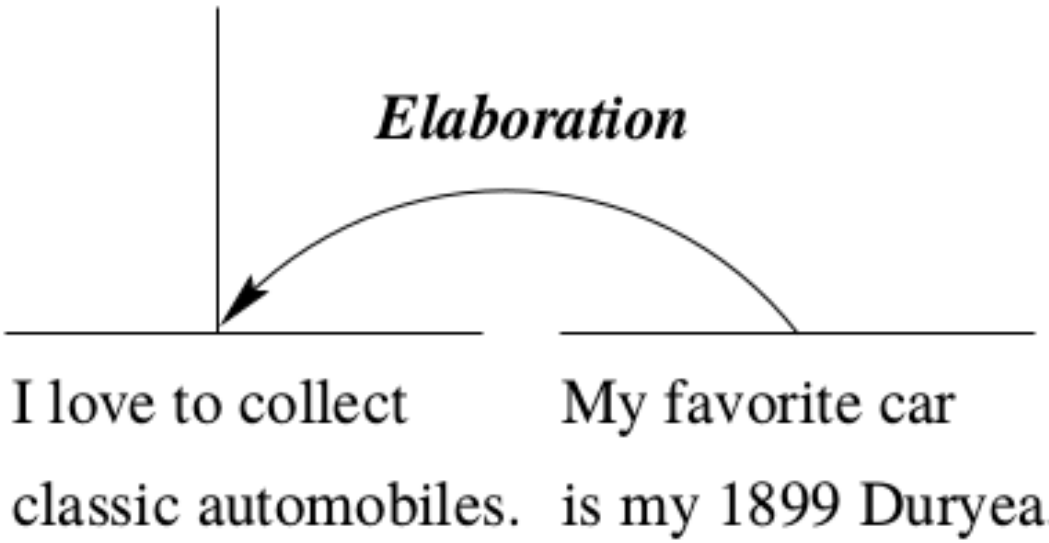
RST -continuation

- The first sentence forms the main segment and is called the central segment or **nuclei**
- The other following sentences are called the peripheral segments or **Satellite**
  Some of the rhetorical relations are
- Elaboration
- Contrast
- Condition
- Purpose
- Sequence (Multi Nuclear)
- Result

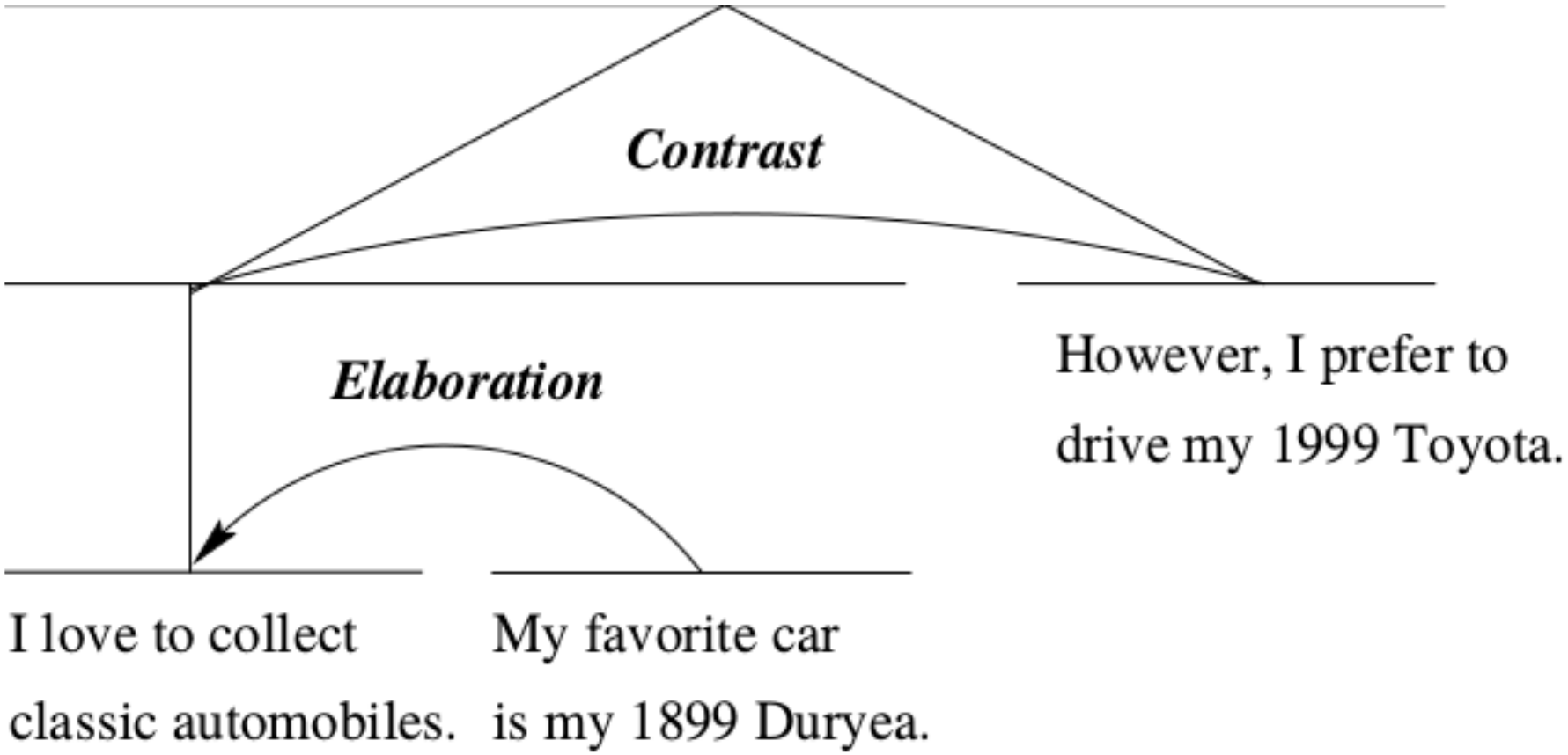
Diagrammatic representation of RST relations

Elaboration

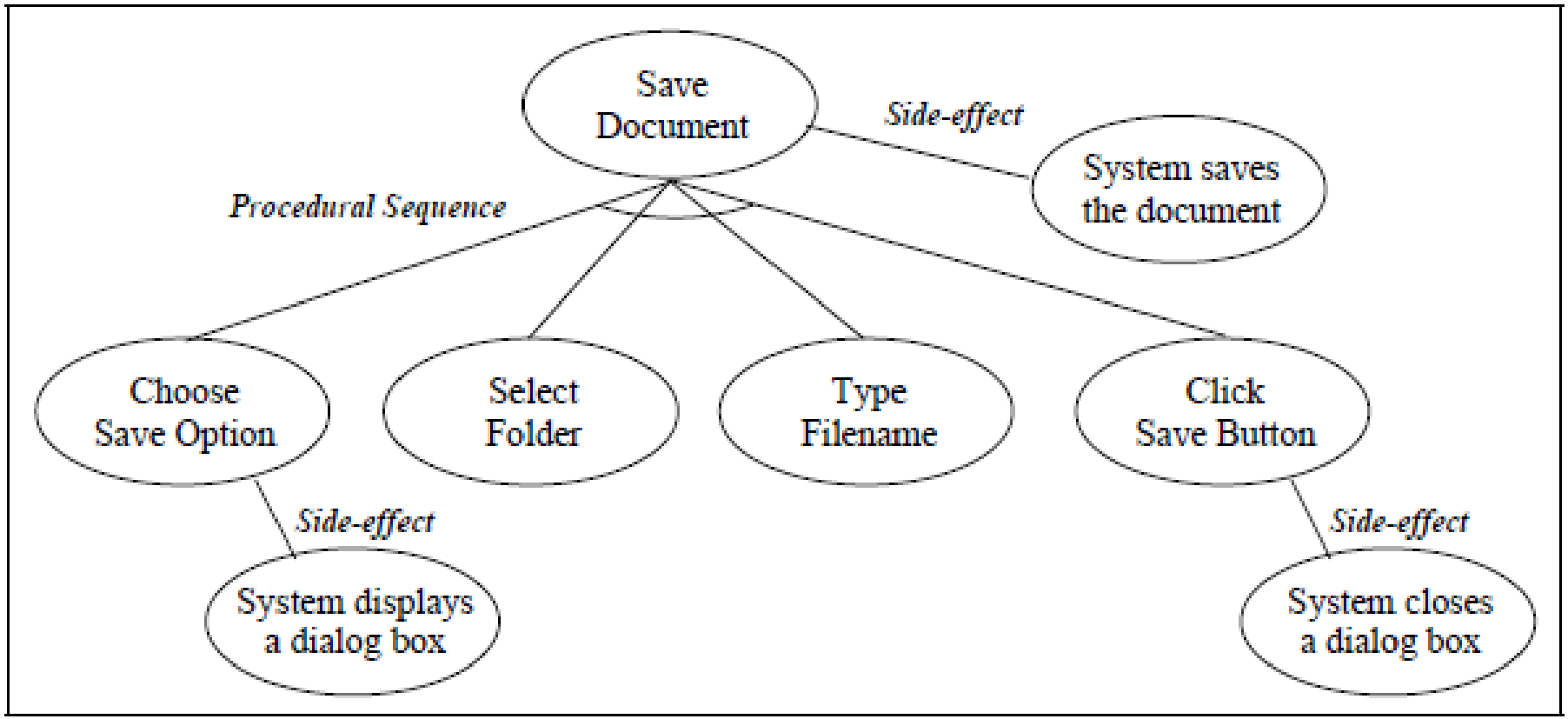
RST relations are typically graphed as follows:



Elaboration and contrast



RST representing the action of Saving a document



Generation Task 2 : Micro planning

- The micro planning is concerned about
  - How information should be grouped into sentences.
  - What lexical items should be used
- The input to a micro planner is macro plan produced by the discourse planner.
- The micro planner has to carry out the detailed planning - called as micro planning.
- Three tasks are involved in micro planning
  1. Sentence Aggregation
  2. Lexicalization
  3. Referring Expression Generation

Micro Planner - contd.

1. **Sentence Aggregation :**
   - It decides how messages in a discourse are grouped together into sentences.
   - Consider the sentences .
     1. Ram sang a song.
     2. The song was good.
     3. Some people did not like it

     The three sentences can be combined into a single sentence like Ram sang a good song but some people did not like it
2. **Lexicalization :** It is the process of choosing appropriate words or phrases to realise concepts that appear in the message.

$$\text{"}did\ not\ like\text{"} \longrightarrow \text{"}dislike\text{"}$$

Micro Planner - contd.

3. Referring Expression Generation : Task of determining appropriate words or phrases to identify domain entities.
In some cases, this task requires the consideration of contextual factors.

One of the most popular abstract representation language is the **sentence planning language .**

SPL for "Ram sang a song"

(S1
: subject (Ram)
: process (sing)
: object (song)
: tense (past)
)

Generation Task 3 : Surface Realization

- The surface realization component produces ordered sequences of words as constrained by the contents of a lexicon and grammar.
- It takes as input sentence-sized chunks of the discourse specification.
- Two approaches used are:
  1. Systemic Grammar
  2. Functional Unification Grammar.

Example :

The system will save the document.

Systemic Grammar

–> Systemic grammars represent sentences as collections of functions and maintain rules for mapping these functions onto explicit grammatical forms.

–> Systemic sentence analyses organize the functions being expressed in multiple layers.

	The system	*will*	*save*	*the document*
Mood	subject	finite	predicator	object
Transitivity	actor	process		goal
Theme	theme	rheme		

Figure: Layers in Systemic Grammar Representation of a senetence

Systemic Grammar - contd.

–> **Mood layer** indicates a simple declarative structure with subject, finite (auxiliary), predicator (verb) and object.

–> **The transitivity layer** indicates that the system is the actor, or doer, of the process of saving, and that the goal, or object acted upon, is the document

–> **The theme layer** indicates that the system is the theme, or focus of attention, of the sentence

Systemic Grammar - contd.

- ★ A systemic grammar is capable of **building a sentence structure**.
- ★ The grammar is represented using a directed, acyclic, and/or graph called a **system network**.
- ★ A systemic grammar uses **realization statements** to map from the STATEMENTS features specified in the grammar (e.g., Indicative, Declarative) to syntactic form.
- ★ Each feature in the network can have a set of realization statements specifying constraints on the final form of the expression.

Systemic grammar - contd .

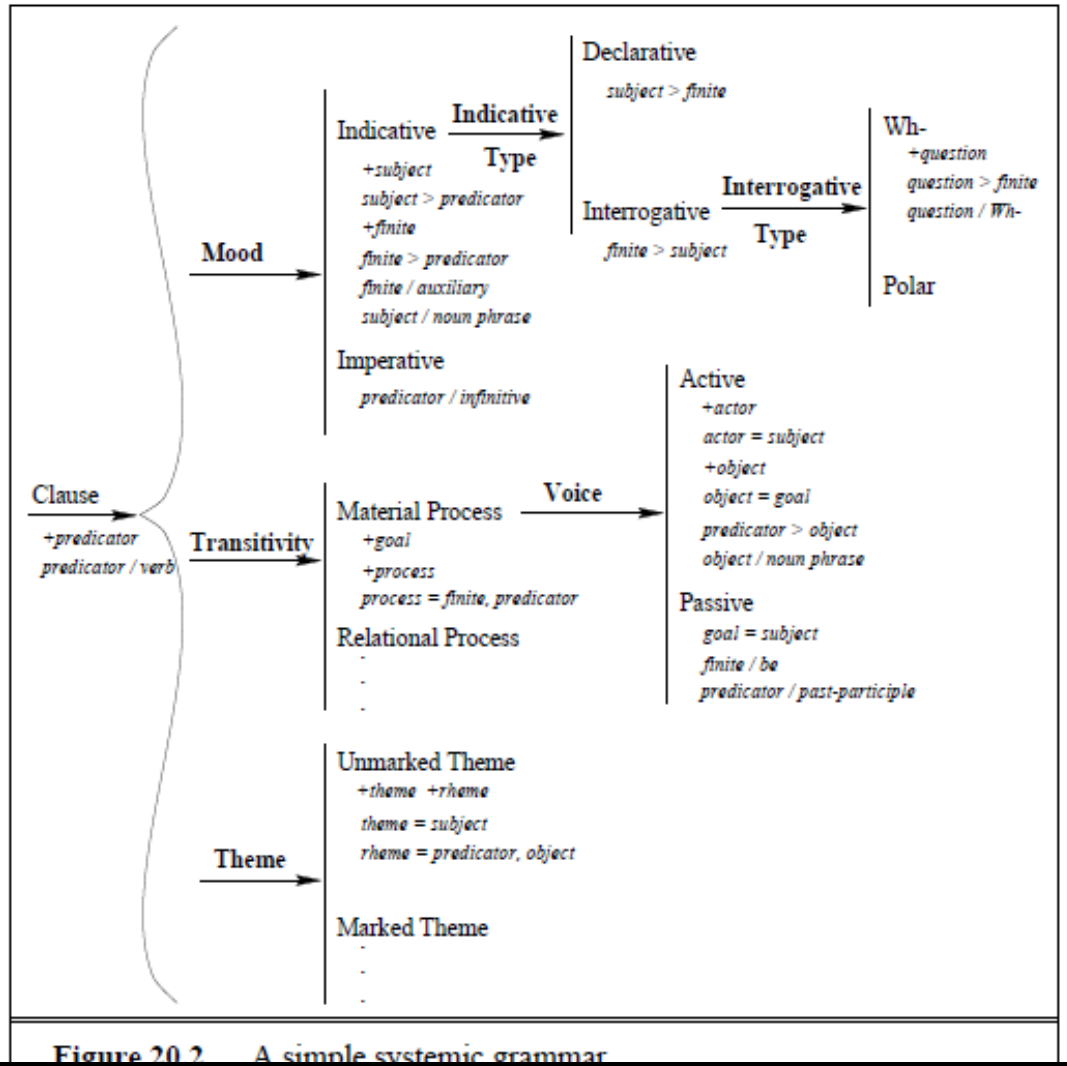


Figure 20.2 A simple systemic grammar

Functional Unification Grammar

- Functional Unification Grammar **uses unification** to manipulate and reason about **feature structures**.
- The basic idea is to build the generation grammar as a feature structure with lists of potential alternations, and then to unify this grammar with an input specification built using the same sort of feature structure.
- This process produces full feature structure which can then be linearized to form sentence output.

FUG - contd.

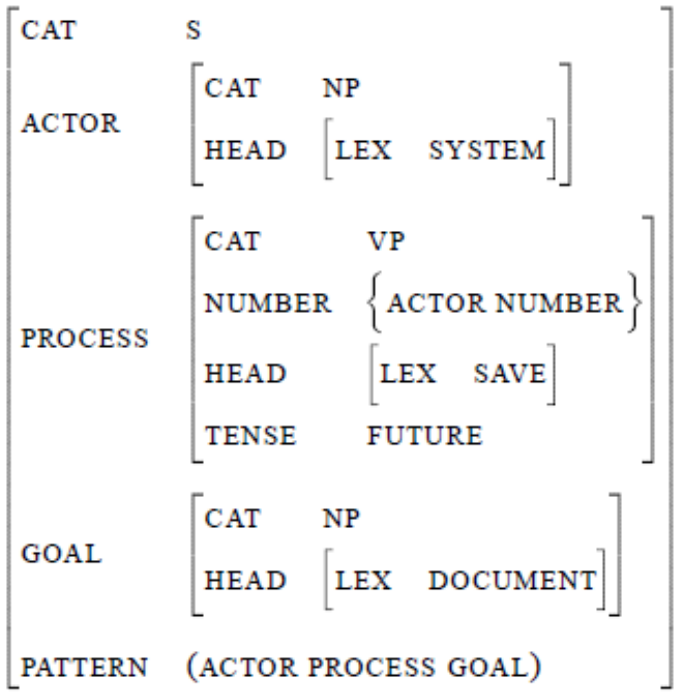


Figure: Functional Unification Grammar for : **"System will save the document"**

Applications of NLG

The NLG Systems have been used to provide natural language interfaces to many databases like :

- Airline schedule Database
- Accounting databases
- Expert system Knowledge Base

Here are some applications of NLG:

1. NLG is also used to **summarise statistical data** , graphical and speech data.
2. It is used in **WeatherReporter System** to create multi-sentence weather summaries from meteorological data.
3. Maybury (1995) used NLG to **create event data**.
4. NLG **produces answers to questions** about an object described in a knowledge base.