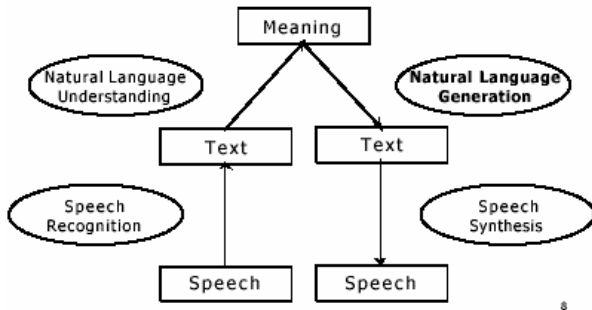
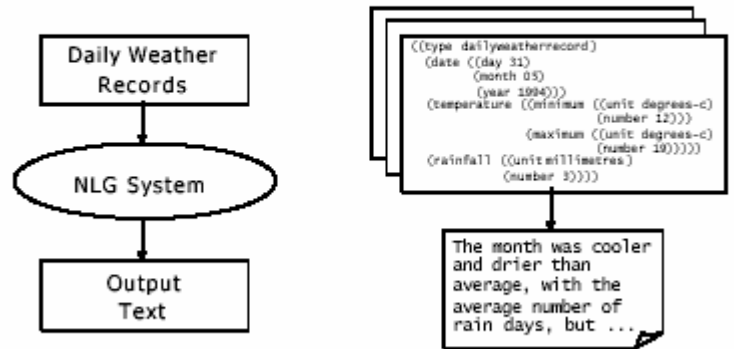


Natural Language Generation

Language Technology



Inputs and Outputs



60

I. Natural Language Generation: a subfield of computational linguistics and artificial intelligence concerned with the construction of computer systems that are able to produce understandable texts in natural languages from some non-linguistic representation of information, such as meteorological maps, airline/railway schedule databases, accounting spreadsheets etc.

non-linguistic input → **NLG system** → output text

When building an NLG system, one may follow the so-called **corpus-based approach**, which means that the builder of the system has to create an initial corpus. Yet, while regular corpora used in natural language analysis consist solely of collections of example texts, NLG corpora are made up of both system inputs and output texts. The output texts in the initial corpora are created by human experts and are a basis upon which the computer generated messages are structured.

II. Applications of NLG:

- i. presenting information in a way which is easy to understand for non-experts
 - a. generating weather forecasts from expert meteorological maps (Fig. 3)
 - b. summarizing statistical data extracted from databases/spreadsheets
 - c. generating medical information based on medical records
 - d. answering questions concerning objects described in some knowledge base
- ii. building authoring aids, that is tools that help people create routine documents, such as business letters and medical records
- iii. NLG systems basically do what an expert human can do as well; therefore, as NLG systems are complex and expensive, their usage is confined to areas where there is need for large amounts of textual messages to be generated in a short time

III. Kinds of information in a typical NLG output:

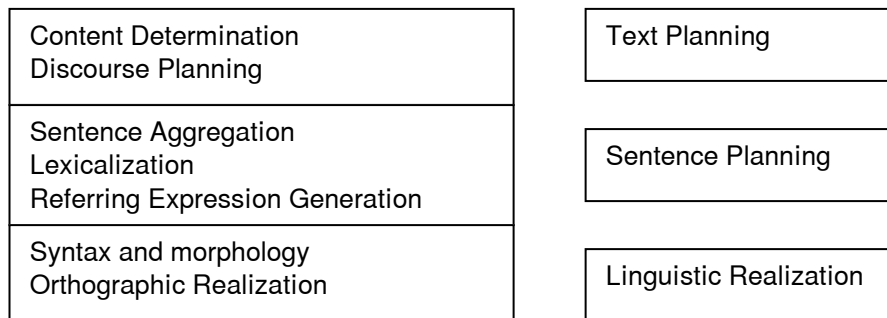
- i. **unchanging text:** always present in the output, e.g. *Thank you for considering rail travel* in Fig. 1
- ii. **directly-available data:** text that presents information which is contained in the input or in a directly accessible database, such as *The next train is the Caledonian Express* in Fig. 1
- iii. **computable data:** text which presents information that can be derived from the input by means of computation or reasoning, e.g. *There are twenty trains each day from Aberdeen to Glasgow* in Fig. 1
- iv. **unavailable data:** text that presents data which is not derivable from the input, e.g. *because of snow on the track near Stirling* in Fig. 1

There are twenty trains each day from Aberdeen to Glasgow. The next train is the Caledonian Express; it leaves Aberdeen at 10am. It is due to arrive in Glasgow at 1pm, but arrival may be slightly delayed because of snow on the track near Stirling.

Thank you for considering rail travel.

Fig. 1. A sample output of a Railway Information System

IV. Six major tasks of an NLG system:



- i. **content determination:** the process of deciding what information should be included in the generated text
- it is based on creating messages, that is data objects used in subsequent language generation; messages consist of **entities** (e.g. specific trains, places), **concepts** (e.g. property of being the next train) and **relations** (e.g. departure as a relation between trains and times); see Fig. 2 for a sample message;

```

message-id: msg01
relation:  IDENTITY
arguments: arg1:NEXT-TRAIN
           arg2:CALEDONIAN-EXPRESS

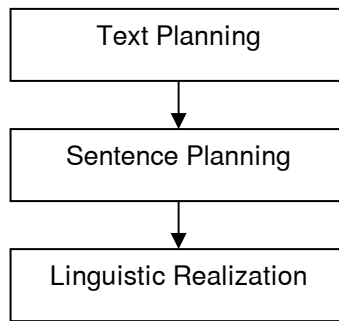
```

Which corresponds to: The next train is the Caledonian Express.

Fig. 2

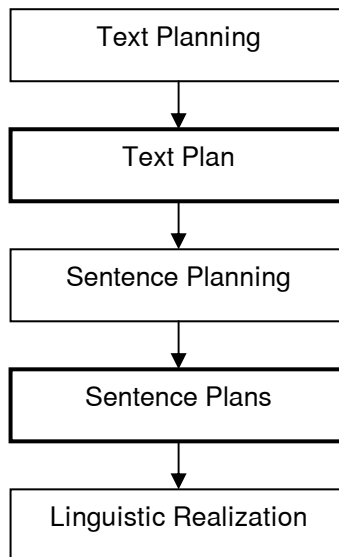
- the choice of messages is often based on an analysis of a corpus of human-generated texts covering the same field (the **corpus-based approach**)
- ii. **discourse planning:** the process of imposing order and structure over the set of messages that are going to be conveyed; it is based on either:
- discourse relations**, such as elaboration, exemplification etc. or
 - schemas**, that is general patterns according to which a text is constructed (Fig. 5).
- iii. **sentence aggregation:** the process of grouping messages together into sentences; it is not always necessary, each message may be presented as a separate sentence, but good aggregation improves the quality of the text; there are several types of aggregation:
- simple conjunctions
 - ellipsis (e.g. *John went to the bank* and *John deposited \$100* may be combined into *John went to the bank and deposited \$100*)
 - set formation (combining messages that are identical except for a single constituent, e.g. *John bought a car*, *John bought a house* and *John bought a computer* may be combined into *John bought a car, a house and a computer*; also, set formation covers instances like replacing *Monday*, *Tuesday*, ..., *Sunday* with *whole week*)
 - embedding (usage of relative clauses)
 - problems arise when the system has to make choices between different possibilities of aggregation – the system builder has to supply it with some rules that govern such choice-making
- iv. **lexicalization:** the process of deciding which words and phrases should be used in order to transform the underlying messages into a readable text; this is the point when pragmatic issues are taken into consideration (e.g. should the text be formal or informal)
- as with aggregation, a poorly lexicalized text may still be understood, but good lexicalization improves the quality and fluency of the text; similarly to aggregation, problems emerge when the system has to make choices between particular words
- v. **referring expression generation:** selecting words and phrases to identify entities (e.g. *Caledonian Express* or *it* or *this train*), generating deictic expressions;
- vi. **linguistic realization:** the process of applying rules of a grammar in order to produce a text which is syntactically, morphologically and orthographically correct; this process may be realized by means of the inverse parsing model, different kinds of grammars or templates

V. NLG architecture: all those tasks are usually combined in some way so as to achieve most efficiency; quite often it is done like this (the so-called **pipelined architecture**):



where the text planning stage consists of content determination and discourse planning, the sentence planning stage of aggregation, lexicalization and referring expressions generation and the linguistic realization stage of syntactic, morphological and orthographical processing.

The partially processed data has to be stored in some way between particular stages, this is done by means of **text plans** and **sentence plans**. We may update the diagram as follows:



Text plans are usually represented as trees whose leaf nodes are particular messages and whose internal nodes show how they are grouped together (Fig. 4). Sentence plans are usually represented by the so-called abstract sentential representations which are similar to syntactic deep structure.

References:

- i. this handout is based on *Building Applied NLG Systems* by E. Reiter and R. Dale, which can be found here: <http://www.csd.abdn.ac.uk/~ereiter/papers/index.html>
- ii. <http://www.cs.columbia.edu/~kathy/> - articles related to NLG
- iii. <http://www.aclweb.org/siggen> - a brief introduction to NLG