- Absence of unification of development process of natural language interface, thus development process of natural language interface cannot be achieved in parallel, as well as we cannot reuse the already developed component;
- The difficulties in analysis and processing of knowledge representation structure. The knowledge is not stored discretely in the intelligent system; it is interconnected to form a structured data in the system;
- There is no ability to use unified tools to represent various kinds of knowledge (e.g., the linguistic knowledge), which is necessary to provide useful information for NLG;
- Developed problem solvers are not flexible at each stage of NLG, e.g., for generating text in a new language it's difficult to change the already existing components or to extend NLG component to adapt.

The relevance of these problems will be explained in detail by analyzing related work to natural language generation.

C. Analysis of related work for natural language generation

In this paper, we focus on the NLG part of natural language interface. The NLG part can be considered as a separated system, or as a component of user interface. But it's difficult to achieve the development of separated NLG system that can be reused into the component of user interface due to absence of unified principle for user interface design. The NLG part in this paper tends to generate natural language text from structured form, in particular, from knowledge base.

Early in the application domain, the successful NLG system includes the weather reporting, "robojournalist" and so on, which convert the tabular data or data of information box into reasonable natural language text by filling placeholders in a predefined template text [3]. In this situation, the text generated on the base of these predefined template is very simple and inflexible. The rule-based approaches convert data into resulted text by a serious of grammar and heuristic rules. The factors influencing these approaches to generate natural language text are the linguistic rules. For analysis of natural language, linguists proposed many linguistic theories that focus on interpreting linguistic formalism, for example, the dependency grammar is used for the interpretation of the syntactic structure. However these approaches lack supporting of unified basis for representing various linguistic knowledge. Moreover, traditional rule-based approaches focus not on the semantic of natural language text, but rather on the syntax

There is a classic pipeline architecture [4] for NLG. Based on this pipeline, generally, the six tasks are frequently found in NLG system (Fig. 1). In fact, the classic pipeline architecture can be considered as the modular approach to solve the NLG problem. Different modules in the pipeline incorporate different subsets of the tasks described above. However, the biggest problem for the ordering of the modular approach is the generation gap [5] that refers to the mistakes of early tasks in the pipeline passed further downstream.

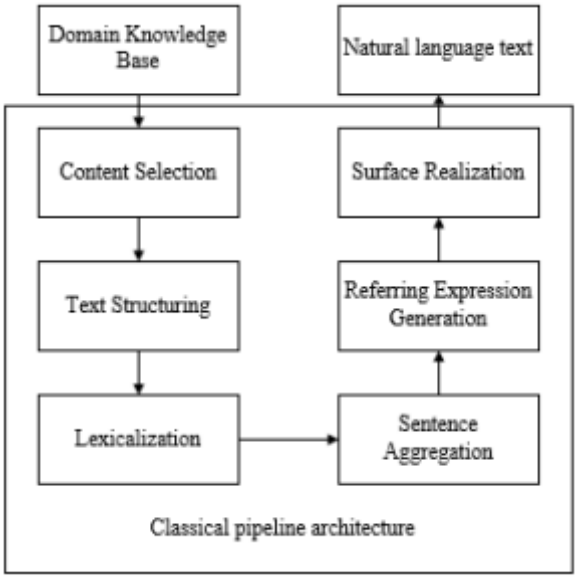

Figure 1: Classic pipeline of natural language generation

The pipeline continues influence on alternative architectures that were proposed in recent years for NLG. The proposed approaches often end up blurring the boundaries between modules. These approaches tend to emphasize statistical methods that is data-driven. From the pipeline perspective, there are three simplified steps:

- content selection;
- content planning;
- surface realization.

In the subtask of SemEval AMR-to-English generation, the abstract AMRs to be converted into syntactic structures by a symbolic generator, then the syntactic structures are linearized with an off-the-shelf tools (e.g. statistical linearizers) [6].

With the advent of deep learning, the most influential architecture for NLG is the Encoder-Decoder [7]. The encoder encodes various kinds of input (e.g., natural language text, structured data, image, video and knowledge base) into a low dimension vector representing the semantic of the input. The decoder generates natural language text from the vector embedding. In practice, the knowledge representation languages like RDF and others are widely used as a kind of input of various modern neural generation models for knowledge-based system constructed by the W3C standards. Currently the W3C standards are widespread used for development of knowledge-based system.