Feature grammar systems. Incremental maintenance of indexes to digital media warehouses

Windhouwer, M.A.

Chapter 8

Conclusion and Future Work

If SETI@home works, for example, we'll need libraries for communicating with aliens. Unless of course they are sufficiently advanced that they already communicate in XML.

Paul Graham – *The Hundred-Year Language*

This thesis describes many formal, architectural and implementation aspects of the Acoi system. They have provided the author with a wide scale of research topics and challenges during the past years. This final chapter concludes the description of this system by providing a look backward, into the past, and a look forward, into the future.

8.1 Conclusion

The aim of the Acoi system was to implement support for the complete life cycle of a DMW (*Digital Media Warehouse*), *i. e.* creation, storage and maintenance, by various interpreters of one declarative description. The model underlying this description would have to support these key requirements: (1) providing context for (possible) bridging of the semantic gap, (2) allowing ambiguous interpretations, (3) describing both contextual and output/input dependencies, (4) give enough context for incremental maintenance, and (5) keep the input specification of algorithms generic enough to enable and promote reuse. As all these requirements involve some form of context grammars, as known from formal language theory, were considered a good starting point.

Chapter 2 provided the formal basis for the system. Mildly context-sensitive feature grammar systems allow the embedding of annotation extraction algorithms, *i. e.* feature detector functions, on a low level inside the grammar formalism, while underspecifying their context. The feature grammar language of Chapter 3 introduced a more natural notation of this formalism. The next three chapters described the various interpretors, the FDE (*Feature Detector Engine*), the database schema and the

FDS (*Feature Detector Scheduler*), needed for the various life cycle stages. These core chapters of the thesis address the key requirements using formal language theory. This is mainly visible in the design of the FDE, which is directly related to practical algorithms from the computer science research fields of natural language processing and compiler technology. Where needed proper extensions of this theory and related practice have been defined.

The three case studies from Chapter 7 gave an impression of the practical impact of the Acoi system. Formal language theory in the form of feature grammar systems, with its focus on extending and in the same time limiting context-sensitivity, appears to be indeed well suited to meet the identified requirements of a DMW annotation subsystem. This, as this thesis already shows, makes a plethora of formal techniques and practical experience available to this application domain. Future experiments with a mature FDS implementation and further evaluation of the complete system may add additional support for this conclusion.

8.2 Future Work

The various components of the Acoi system provide a basis for the annotation subsystem of a DMW. However, there is always room for improvement, as has been indicated in various places throughout the thesis. The next sections will revisit these areas of future work and also describe some additional ones.

8.2.1 Feature Grammar Systems

The description of feature grammar systems is directly based on the formal theories of cooperating distributed grammar systems and regulated rewriting. Contributions of this thesis to these theories are lPC (*(left) path-controlled*) grammars (see Section 2.2.3). Intuition tells that these grammars are as powerful as conditional grammars, however, this should backed up by a formal proof.

Detector functions can produce CS (*Context-Sensitive*) sentences, however, the feature grammar components are CF (*Context-Free*). In a future version of Acoi the CF components may be replaced by CS or mildly CS components, *e.g.* lPC (*left Path-Controlled*) components. This will also have to be reflected in the feature grammar language. In the case of lPC components a regular path expression could be associated to arbitrary non-terminals instead of only to detector symbols.

8.2.2 Feature Grammar Language

The core feature grammar language directly relates to the underlying feature grammar system. The extensions to the language provide shortcuts to the developer. One of tasks of these extensions is to keep the grammar semantically rich and to avoid clutter with additional symbols which are only there to steer the extraction process. Ideally these additional symbols are all anonymous so they are transparent to both the developer and the user. Additional language constructs may proof to be convenient, *e. g.* directly embedding of anonymous whitebox detectors in the production rules. Also the use of symbol or symbol specific scripts, in the vain of attribute grammars, for the propagation of confidence values may proof an interesting addition.

8.2.3 Feature Detector Engine

The current FDE implementation is based on a depth-first top-down parsing algorithm: exhaustive backtracking. Experiments with other parsing algorithms, and thus with other moments of control transfer between components, can provide alternatives for this algorithm. Another interesting experiment will be the use of a breadth-first algorithm to add some form of parallelism to the FDE. These alternative algorithms can also be realized by translating a specific feature grammar system into a specification for a coordination system, *e. g.* a T script for ToolBus.

A detector function can depend on some external functionality, which is (temporarily) unavailable. Adding an *unknown* result state next to *success* and *failure* may enable the FDE to proceed validation partially and to return later to the detector to retry its execution.

8.2.4 Feature Database

As already stated in Chapter 5 the current storage scheme can be replaced by another, probably better suited, XML storage scheme. The query process of the feature database is also still in its infancy. For example further experiments with the usage of the confidence values of detectors as input to a probabilistic reasoning scheme or a ranking formula will be useful. This is directly related to a proper use of the, possible, various alternative interpretations of one multimedia object.

At the moment all annotations are extracted during the building of the index. However, some annotations or features may be more dynamic, *i. e.* they are computed on demand and are not persistent. These derived annotations and features may be used to capture properties of the whole index at a specific moment in time, *i. e.* the moment of query execution. The impact of these type of symbols on the analysis of the dependency by the FDS will have to be investigated.

8.2.5 Feature Detector Scheduler

The implementation of the FDS has been sketched and some core components have been prototyped. Future work certainly contains an actual implementation of the complete FDS, accompanied by further experiments in the domain of, for example, the WWW multimedia search engine.

8.2.6 Digital Media Warehouses

The case studies of Chapter 7 gave insight or hints in how to embed the system in a complete DMW system. But in most cases the Acoi system functioned as a blackbox. Future experiments could involve a further integration of the various system components. The manual annotation part is one of the first components which comes to mind. How well will the Acoi system cope with manual detectors and semi-automatic annotation extraction? If this turns out favorable for the Acoi system it will implement a complete annotation subsystem.

Comparison with the execution characteristics of other systems may become possible by implementing a common task, and may give insight in both modeling power and performance of the competing systems.

8.3 Discussion

This thesis touched upon one of the key research challenges of a DMW: multimedia annotation extraction and (incremental) maintenance. Although there are still many open issues to be resolved, the Acoi system, with feature grammar systems as basis, rallied formal language theory and practice to meet this challenge with success.