

## LOQUENDO ASR

### DRIVING SPEECH RECOGNITION WITH STANDARDS AND TOOLS

#### W3C SEMANTIC INTERPRETATION FOR SPEECH RECOGNITION

Automatic speech recognition is a difficult challenge for any ASR engine, especially if there are no linguistic constraints to help the recognition process. Restricting a set of sentences that can be accepted by the recognition engine in a specific dialog turn is very important to improve recognition results and make speech recognition a usable technology in a wide range of applications.

A widespread way of defining linguistic constraint for speech recognition is to define a speech recognition grammar, which is a means of specifying a fixed series of sentences that will guide the speech recognizer in finding the correct spoken words, which is the goal of speech recognition. Depending on the recognized words, a speech application will carry out automatic or semi-automatic tasks, such as directory assistance, financial transactions, timetable enquires, the list is endless.

##### **Syntactic constraints**

A grammar is an ASCII file which describes the words that can be uttered by the application's users. It is structured into rules composed by words or sequences of words alternating with others. In a rule, it is possible to reference other rules of the same grammar or rules from other grammar files.

The description of these rules are referred to as syntactic constraints because they restrict the syntax of the spoken utterances that will be accepted by the speech recognizer. A number of formalisms are used in grammar writing. The W3C has developed a standard called Speech Recognition Grammar Specification [SRGS]. It describes two formats: a compact one (the ABNF) and an XML one. Another well-known formalism is the Java Speech Grammar Format [JSGF].

##### **Semantic instructions**

In a real-life application scenario, a recognized utterance cannot be used directly: the application may often take advantage of a formatted input to perform transactions or to provide correct information.

Let's suppose we insert a rule to provide the hour of departure in an application.

Different but valid user inputs such as *forty minutes past nine AM* or *twenty to ten in the morning* represent the same time: 9:40AM which is the only necessary information to the application.

It is possible to introduce a number of instructions (called semantic instructions) into a grammar. These are introduced between syntactic constraints, and will be executed during recognition (or immediately after) to format the recognition results. Another possible use of semantic instructions is to perform semantic checks on the results and possibly to complete recognition results with default or derived information.

In a grammar, semantic instructions are inserted into elements that are usually called 'tags'. Each grammar formalism provides a way to insert these semantic tags.

Semantic tags can be associated to words or to grammar sections and they are executed only if words they are associated to are contained into user utterance. There are some rules to follow in order to execute semantic instructions and they depend on the formalism in which these instructions are written.

Like SRGS 1.0 [SRGS], W3C is developing a standard for semantic instructions called "*Semantic Interpretation for Speech Recognition*" [SISR]. It was published as a Last Call Working Draft on November 8, 2004.

Loquendo ASR implements this standard semantic formalism.

## A basic description of SISR

ECMA is a scripting language that is rich in functionality and is capable of producing complex semantic results. Every grammar rule has a single semantic value (which is referred to as *Rule Variable*) that is an ECMAScript value. The value of the Rule Variable is typically assigned by the SI tags within its grammar rule. SI tags also have access to the Rule Variables of any other rules referenced by the current grammar rule and already processed by that point in the utterance. It allows a powerful interaction between intermediate semantic results obtained analyzing different parts of user utterances.

The ECMA scripting language is a standardized version of JavaScript. It is widely used by applications developers as it enables users to script grammars without learning another proprietary language and is extremely powerful, allowing complex processing and data structure definition. Moreover, this standard will allow a better integration with VoiceXML, simplifying semantic results handling.

## LOQUENDO PHONETIC LEARNING TECHNOLOGY

The most critical aspect in creating speech applications based on automatic speech recognition is the need to correctly predict user utterances and to effectively deal with non-native speakers or regional accents. In a directory assistance application, for instance, user formulations for business listings may differ a lot with respect to the system knowledge derived from white and yellow pages. In a very complex application, even if a careful study of user behavior has been carried out, it is impossible to predict exactly how users will formulate their requests. The knowledge provided to the system by *a priori* analysis is very useful in releasing the first version of a speech application, but is nevertheless not enough. The key source of information is the data the system collects from the field itself. The processing of very large amounts of collected data, however, can prove way too costly when performed by human operators.

To solve this issue, Loquendo has developed a new technology which automatically analyzes application data to detect the most significant weaknesses. Loquendo ASR Phonetic Learning technology supports speech application developers in improving application performance. It addresses two main issues, which aim to improve the effectiveness of the grammar recognition objects:

- Automatic discovery of pronunciation variants for the vocabulary words
- Clustering of frequent unforeseen linguistic formulations

### Loquendo Phonetic Learning Algorithms

Today, speech applications are mainly based on recognition grammars. A recognition grammar is a formalism that allows application designers to specify what a user can say. The recognized sequence of words must be foreseen within the grammar. However, a user might utter a sentence that is not completely covered by the grammar. Moreover, the speaker's pronunciation may not be completely covered by the system's phonetic knowledge, mainly in the case of foreign words. When these phenomena become frequent, they become a serious application issue. How can one detect them? The application log data contains all the information related to a single recognition interaction, including the recognized words, their confidence values, the audio signal and the phonetic transcriptions decoded through a telephone network. It is quite likely that a poor confidence score triggers a number of mismatched conditions. Different repetitions of the same utterances should provide similar phonetic transcriptions.

Loquendo phonetic learning technology is capable of finding clusters characterized by phonetically similar utterances, providing a phonetic transcription for the most populated clusters.

Loquendo phonetic learning technology uses the same approach when searching for pronunciation variants of vocabulary words. In this case, the cluster is made up of automatically derived phonetic transcriptions related to a given recognized word. Only repetitions with medium / high confidence score values are taken into account in order to obtain high reliability on the recognition results. Additional phonetic transcriptions can be found and added to the regular ones.

In order to use recognition data collected from the field, Loquendo ASR's log mode automatically saves recognition information.

Typically, a voice application may need more than a single recognition server to be available to users. The Loquendo ASR log will be produced on a recognition server basis, but it is advisable to merge all the collected information, because the phonetic learning algorithms require a large number of data to increase the statistical significance of the produced results. The Loquendo LogCollector tool gets log information data from different recognition servers, producing a unique local database that is usable for phonetic learning purposes.

The final step in phonetic learning analysis, is the use of data saved by the LogCollector software inside the Learner tool. The Learner tool produces a hypothesis related to pronunciation variants or unforeseen formulations. The application developer should check the produced hypothesis and accept or refuse them.

### **The Loquendo phonetic learning experience**

Loquendo has applied phonetic learning technology to a fully-automated directory service developed by Loquendo for Telecom Italia. This service has been live since 2000. Given a Name and Address, it returns a Phone Number from a database of 25 million Italian subscribers. The automatic system can fully automate both business and residential requests. All calls are routed to the automatic system. Calls that cannot be automated are routed to human operators, together with transcribed data.

Directory Assistance for business listings is a challenging task: one of its main problems is that customers formulate their requests for the same listing with great variability. Since it is difficult to reliably predict user formulations *a priori*, Loquendo has studied a procedure for detecting, user formulations that were not foreseen by the designers from the field data itself. These formulations can be added, as variants, to the denominations already included in the system in order to reduce failures. In particular, it is fundamental to detect new formulations for frequently requested listings, for example "nicknames" of hospitals or other public services, or user requests for the phone number of a popular TV talk show, that of course do not appear in the directory listings. Loquendo's approach is based on partitioning the field data into phonetically similar clusters, from which new user formulations can be derived.

The results of the experiments on a very large number of calls that the system was unable to serve automatically, are very positive indeed.

The technology has allowed considerable improvement in terms of linguistic coverage, reducing the mismatch among the most frequently asked business listings and the system's knowledge. The system is now operational, and allows periodical updates to the formulation variants system for each town.