TRANSFORMING INFORMATION IN SPEECH DATABASES INTO KNOWLEDGE

Toomas Altosaar[1], Martti Vainio[2], Matti Karjalainen[1]
[1]Acoustics Laboratory, Helsinki University of Technology
[2]Department of Phonetics, University of Helsinki)

ABSTRACT

Speech databases represent an information source essential for the continued development of spoken language theories and applications. However, due to a lack of standards in data formats and annotation conventions, extracting information from different speech databases and transforming it into generic knowledge-bearing structural frameworks is often difficult. At least two different methods are possible for generating structure. One way is to develop a specific interpreter — consisting of a parser, compiler, and linker — to handle every encountered combination of database format, language, annotation syntax, etc. Although direct and potentially computationally efficient, this solution is expensive and time consuming to implement in systems requiring generic access over many diverse databases. Another method is to apply a knowledge-based approach where rules are used to form structures for speech utterances. This paper describes the speech database interpreters in the QuickSig database environment as well as formulates the requirements of a proposed knowledge-based system.

1. INTRODUCTION

Speech databases provide an essential information source for speech recognition, synthesis, and analysis technologies. Phonetics, which creates a vital infrastructure for many areas of speech research, has also benefited from speech databases by allowing detailed and independent study of similar data. Databases include substantial amounts of recorded speech, parts of which may have associated symbolic annotations. These annotations may consist of, e.g., orthographic, phonemic, phonetic, and prosodic transcriptions, some of which may be time-aligned to the acoustic signal. When properly extracted the information available from speech databases can be transformed into knowledge regarding spoken language by generating a structural framework representation. The latter transformation plays a critical role in the use of speech databases; its quality determines directly how much of the potential information stored in a collection of speech will be available for utilization.

The continuing evolution of information representation paradigms used in speech research has to some extent influenced the format of speech databases thus causing a lack of standards to exist. The ever expanding diversity of file formats, annotation conventions as well as their scope, is emerging as a critical problem since more programming resources must be allocated to gain value from these new databases. For every database consisting of a unique combination of different standards such as phonetic alphabet, language, file format, annotation convention, etc., (often an unfortunately non-independent mixture) a unique and specific interpreter must be written. As many new databases are being published the number of these specialized solutions is increasing making it costly for subscribers to gain intelligent access to the data.

The sooner the lack of a widely accepted standard for speech databases is addressed, the better it will be for the speech community as a whole. Recently, a proposal for a formal framework for linguistic annotation [1] that focuses on logical structure instead of file format has been made. It provides for a framework for constructing, maintaining, and searching linguistic annotations while remaining compatible with existing data structures and storage formats. Existing speech databases of varied formats could be converted to this useful “interlingua” and standardized tools developed to operate on the data.

Generic framework structures enable databases searches to be effectively carried out regardless of the speech database under observation. This paper looks at the specific task of transforming information from a diverse set of currently available speech databases into generic knowledge-bearing structural frameworks within the QuickSig Speech Database environment [2]. Two main approaches can be taken to perform this transformation. As was mentioned above, a specific interpreter for each different database can be used. Being an expensive and specific solution, this approach is only viable if a limited number of databases are under consideration. By employing a knowledge-based system, based on rules and generic interface methods, a higher-level generalization of the same task is possible.

It is well agreed that most published speech databases exist in a so-called “machine-readable” format. However, to be “machine-usable” a substantial amount of human intervention is currently required in order to extract the potential information existing within a certain database. Since none or only a fraction of the detailed meta-knowledge required for interpretation is supplied along with the media a database is distributed on, the researcher or programmer who wishes to access the data must supply the remaining part of this missing meta-knowledge. Even if programs are supplied along side the collection of speech tokens — effectively converting a speech corpus into a speech database — the software may be applicable to only certain hardware and operating system configurations. Rarely do speech databases suggest or contain any structure defining models so as to make the transformation of speech information into knowledge readily possible.

2. SPEECH DATABASE ENVIRONMENT

QuickSig [3] is an experimental object-oriented (OO) signal processing system implemented in Common Lisp and CLOS that models signals, speakers, transcriptions, filters, analyzers, displays, etc., as objects. It is readily extendible by the user allowing new signal processing methods to be added incrementally and on-line to the rest of the system. QuickSig is

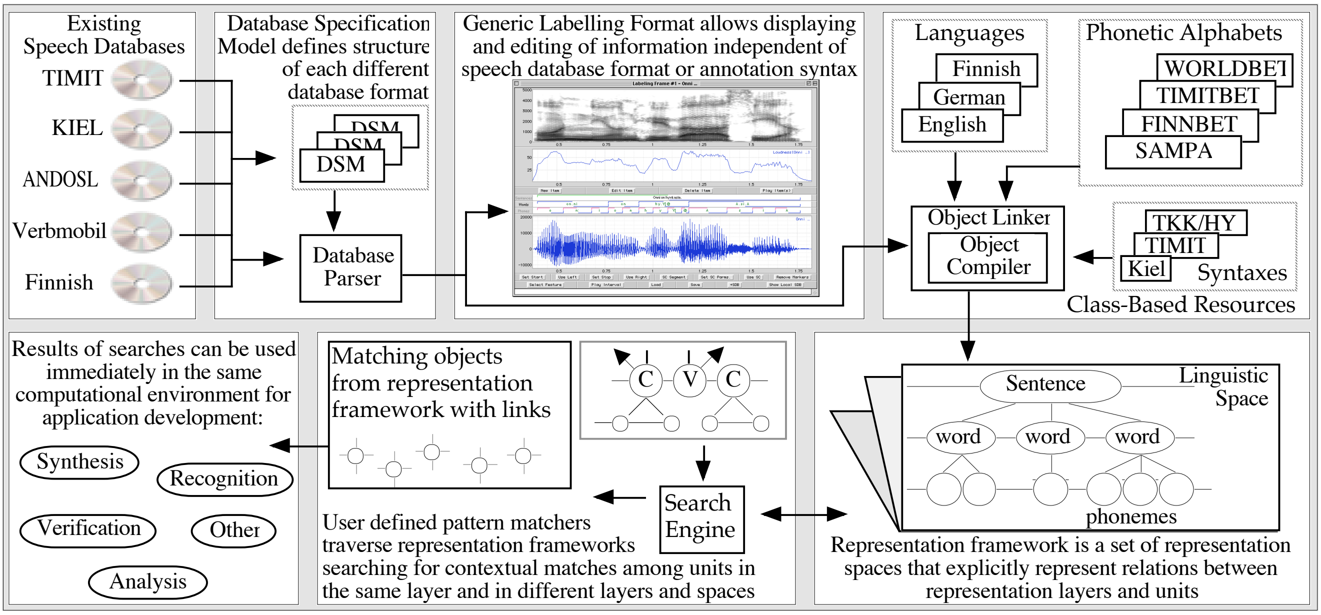

Existing Speech Databases
TIMIT
KIEL
ANDOSL
Verbmobil
Finnish
Database Specification Model defines structure of each different database format
DSM
DSM
DSM
Database Parser
Generic Labelling Format allows displaying and editing of information independent of speech database format or annotation syntax
Languages
Finnish
German
English
Phonetic Alphabets
WORLDBET
TIMITBET
FINNBET
SAMPA
Object Linker
Object Compiler
TKK/HY
TIMIT
Kiel
Syntaxes
Class-Based Resources
Results of searches can be used immediately in the same computational environment for application development:
Synthesis
Recognition
Verification
Other
Analysis
Matching objects from representation framework with links
C
V
C
Search Engine
User defined pattern matchers traverse representation frameworks searching for contextual matches among units in the same layer and in different layers and spaces
Sentence
Linguistic Space
word
word
word
phonemes
Representation framework is a set of representation spaces that explicitly represent relations between representation layers and units

Figure 1. Converting different speech databases into representation frameworks for generic database access.

composed of different packages that are seamlessly integrated and exist concurrently in the same computational environment.

Also modeled are different object types found in speech corpora and databases, e.g., signals and speakers as well as a hierarchy of units such as sentences, words, syllables, phonemes, phones, segments, in linguistic, phonetic, orthographic, and prosodic spaces. Also, implicit knowledge regarding phonetic alphabets, annotations, and syntaxes is supported allowing for speech databases covering different languages and formats to be analyzed in a uniform database access and analysis environment.

2.1. Object Oriented Methodology

The characteristic behavior of different speech units such as acoustic segments, phones, syllables and sentences can be effectively modeled using an object-oriented programming (OOP) approach. The required behavior of a class is obtained by mixing into the class inheritance of these speech units a rich set of speech process and relation modeling classes that are available in the QuickSig environment.

QuickSig supports one-to-one and one-to-many relationships by defining a set of classes for defining inter-object relationships. Instances of classes inheriting these relational classes, such as signals, speakers, and transcriptions, can be linked among themselves. Persistency of objects is attained through QuickSig's object-oriented database (OODB) model. Objects including some of their selected relations are automatically saved to permanent storage and reinitialized at the beginning of a new computational session without requiring user interaction.

2.2. Representation of Speech Knowledge

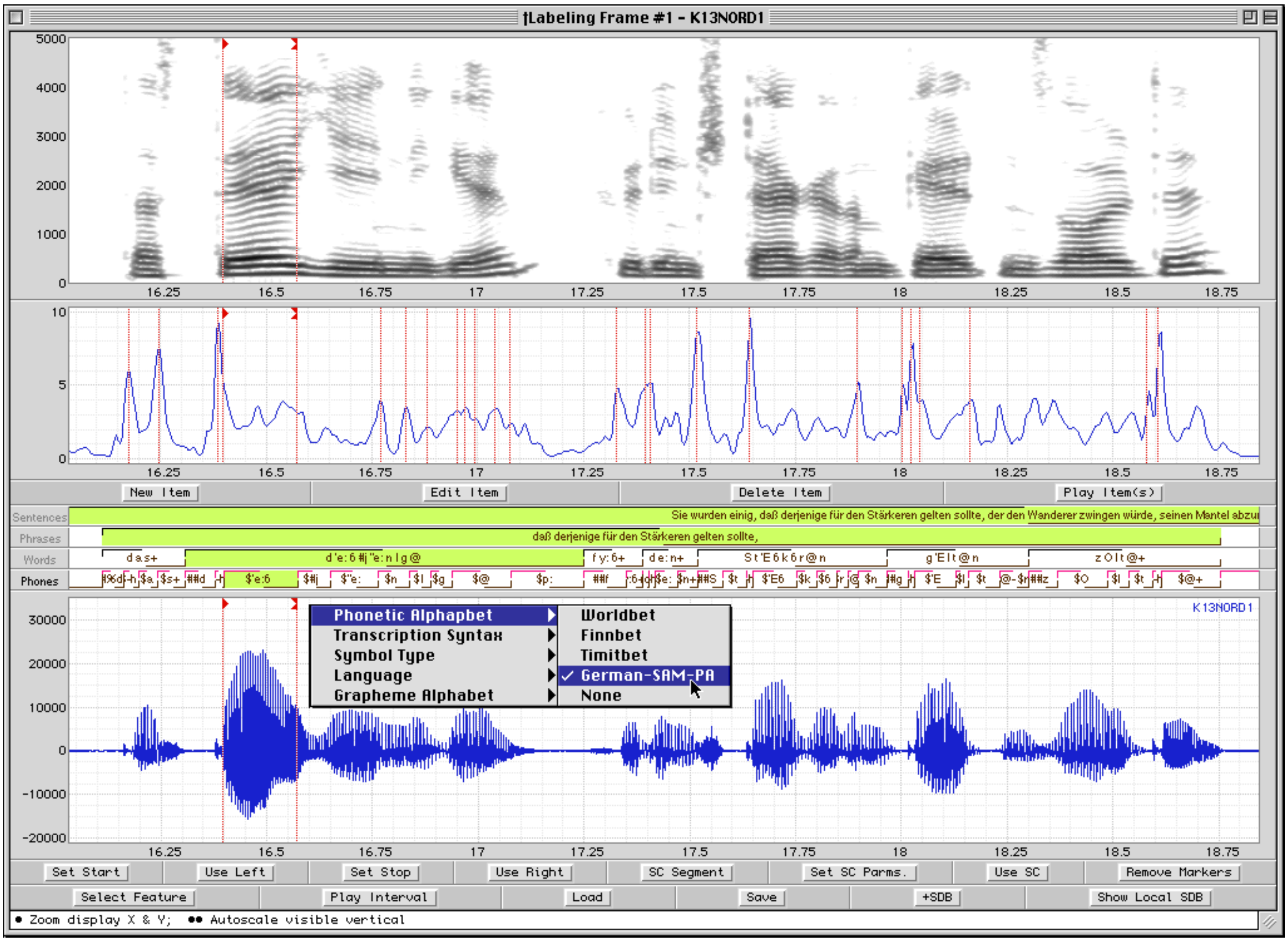
A knowledge representation method that associates features with nodes representing objects is referred to as a *frame*. In addition to frames, QuickSig represents speech knowledge through *semantic nets* that are networks of nodes representing speech objects connected by links that describe the relationships between nodes.

2.3. Generic Database Access

Figure 1 shows the different processing stages used in QuickSig to process speech corpora. Information is transformed into knowledge through the use of representation frameworks that enable generic database access. Starting from the top-left and moving towards the right, each database first has a hand-coded database specification model (DSM) drawn up for it. A DSM includes part of the missing meta-knowledge that was refereed to earlier by indicating to the database parser how a database is structured as well as what information it contains. For example, information regarding directories, file naming and extension conventions, as well as waveform information description formats, e.g., AIFF, WAV, NIST headers, etc., are indicated. Also, symbolic information describing the type and format for each unit of information is defined. The parser, using specification information supplied by a specific DSM, segments the annotation data into generic labeling frame (GLF) objects that include this reinstated data and type information. GLF objects can be presented in a labeling frame for viewing and editing by the user, if necessary. GLF objects are also given knowledge of their language, phonetic alphabet, transcription style, if applicable, and temporal interval, if determinable. Figure 2 shows a labeling frame for an utterance within the Kiel Corpus of Read Speech where the latter GLF object attributes can be read and modified if desired.

GLF objects are then fed to an interpreter composed of an object compiler and linker that utilize a set of class-based resources, visualized in the top-right corner of figure 1. These resources cover different fields such as language, phonetic alphabet, annotation syntax, etc., thus enabling the object compiler and linker to be void of domain specific knowledge. Section 3 describes two different approaches to realizing the interpreter.

Once an utterance has been interpreted, compiled speech units such as phones, words, phrases, etc., will have been placed into a representation framework. Representation frameworks are



Sentences: Sie wurden einig, daß derjenige für den Stärkeren gelten sollte, der den Wanderer zwingen würde, seinen Mantel abzu

Phrases: daß derjenige für den Stärkeren gelten sollte,

Figure 2. A QuickSig labeling frame for an utterance within the Kiel Corpus. Annotation levels for sentence, phrase, word, and phone are displayed and others added if desired, e.g., syllable, prosody, etc. Items from the database have been parsed into an intermediate generic labeling format (GLF) form that has additional information associated with each object such as phonetic alphabet, syntax, symbol type, language, and grapheme alphabet. This knowledge is used in later interpretation stages.

composed of a set of representation spaces, e.g., acoustic, phonetic, linguistic, etc. Each space in turn may have associated with it a set of discrete or continuous representation layers indicating structural hierarchy, e.g., dominance relationships. Layers are made up of a set of representation units that were compiled and linked by the previous stage. Finally, units are linked to one another by vertical, horizontal, and cross-representation-space bi-directional links that include type information. Thus, units within a representation framework contain domain specific knowledge of their "genetic" makeup (through class inheritance) as well as the relationships they share with their local environment. Also, every unit can reach any other unit within an utterance. For example, a phone unit existing within the phonetic space can access its neighboring phones and knows which syllable or word it belongs to. Objects within some space can be linked to corresponding objects in other spaces, e.g., orthography and prosody. For example, if a phone is linked to its related phoneme(s) in the linguistic space it can deduce whether it is a realization of a phoneme or an insertion, e.g., a schwa insertion. Likewise, phonemes know if they have been deleted in actual speech by noting that they are missing a realization in the phonetic space. Phones and phonemes have knowledge of their distinctive features through a comprehensive set of feature-mixin classes [2]. Modeling shared phones along word boundaries, e.g., geminates, as well as componential residues where a deleted phone may still have an influence, is also supported.

During the database access phase, the explicit links that exist between the speech representation units are utilized. Their relationships, e.g., previous phone, syllables in word, etc., are all available for use by a search engine given a structure and user defined pattern to be matched, i.e., a search function. Complex searches can be performed over the framework representations revealing desired contexts. Since speech is represented in a generic manner using the frameworks, searches can be performed over a single utterance, a subset of a database, an entire database, or concurrently over several databases covering different languages, annotated in differing phonetic alphabets or conventions.

The result of a search is a set of actual objects that exist within the frameworks. Since all links are available for use the speech researcher can, e.g., immediately listen, display, or manipulate matching contexts and perform analyses on them, all in the same computational environment. Finally, the results of a search can be applied to a problem area under study, e.g., training a speech synthesizer, analyzing the performance of a speech recognizer, etc.

3. TRANSFORMATION METHODS

At least two different methods are available to implement the transformation of generic labeling frame (GLF) objects into knowledge-bearing representation framework structures. The first method is to have specific interpreters available for every unique database that is required for analysis. The second method is to control the interpretation and structure generation process with a rule-based system where generalizations can be made.

3.1. Specific Interpreter

For speech databases currently handled by the QuickSig speech database environment, such as ANDOSL, FINNISH, KIEL, TIMIT, and Verbmobil, unique solutions for the interpretation stage have been designed and implemented. Through evolution some domain specific knowledge has been removed from the compiler and linker and has been distilled in the form of useful resources. Phonetic alphabets, languages, syntaxes, hierarchical order, linking strategies, and state-machines for parsing sequences of annotation symbols, are examples of such orthogonal knowledge resources. However, by the use of purely procedure-oriented methods, extraction of all domain specific knowledge from the compiler and linker has been found to be difficult. Simply stated, every time a new speech database format has been introduced changes have had to be made to the interpretation stage.

3.2. Knowledge-Based Approach

Since the development of a unique interpreter for every different database is time consuming and prone to errors — and therefore costly — we are currently evaluating a knowledge-based approach to control the interpretation stage. From the experience gained from working with specific interpreters we are forming a skeletal knowledge engineering language that will be void of all domain specific knowledge. By using rule-based methods instead of purely procedural ones, additional generalization will be possible. We base the feasibility of developing a knowledge-based interpreter on the following arguments:

The development of a knowledge-based system for interpreting the contents of diverse speech databases should be *possible* since:

- the task does not require commonsense, i.e., unlimited reasoning
- the methods required to solve the problem can be articulated
- genuine experts exist
- agreed solutions exist
- task is not too difficult
- task is not poorly understood

This development is *justifiable* since:

- the task solution has a high potential savings in engineering costs
- human expertise is scarce, i.e., few want to deal with this problem
- the expertise is needed in many locations, e.g., in every speech workstation

And finally, the use of a knowledge-based system is *appropriate* for this problem since:

- *nature*: the problem requires both symbolic manipulation as well as heuristic solutions
- *complexity*: the task is not too easy
- *scope*: the task has practical value and is of manageable size

Figure 3 shows the proposed internal structure of the interpreter based on a knowledge-based approach. If successful, this will replace the top-right frame of figure 1 and should simplify the task of incorporating new databases into the generic database access environment.

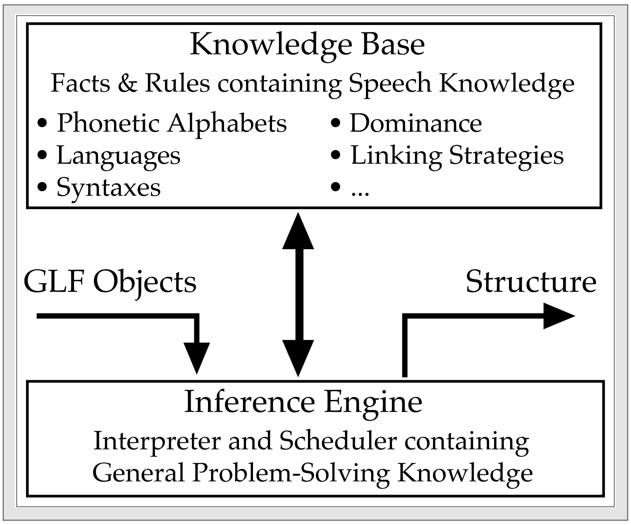


Figure 3. Diagram of the proposed knowledge-based interpreter.

4. CONCLUSION

The use of database specification models (DSM) has been found to be useful since they form an abstraction level that hides much of the physical structure of a database from later processing stages. Hopefully in the future, DSMs will be specified in some general markup language and included with speech databases in a standardized location on the distribution media or network. Likewise, databases should also supply the knowledge-based resources, i.e., infrastructure, that are required for their interpretation. The latter could also include information to specify the form of some standard structure(s).

Without the above-mentioned items, machine-readable speech databases will not easily be “machine-usable” nor will the transformation of information to knowledge be a straightforward task. Only when this missing meta-knowledge is included with collections of annotated speech, will speech corpora truly become databases where speech is generically accessible.

REFERENCES

1. Bird, S. and Liberman, M., A Formal Framework for Linguistic Annotation. [http://xxx.lanl.gov/abs/cs.CL/9903003] March 1999.
2. Altosaar T., Karjalainen M., Vainio M., A Multilingual Phonetic Representation and Analysis System for Different Speech Databases. Proc. of ICSLP 96, Philadelphia 1996.
3. Karjalainen M., Altosaar T., Alku P., QuickSig-An Object-Oriented Signal Processing Environment. Proc. Of ICASSP-88, New York 1988.