Infrastructure for Collaborative Annotation of Speech

Mickel Grönroos and Manne Miettinen

The Finnish IT center for science CSC
P.O. Box 405, FIN-02101 ESPOO, Finland
{Mickel.Gronroos, Manne.Miettinen}@csc.fi

Abstract

Vast amounts of digital language data (primary data) and increasingly complex linguistic annotations (secondary data) are being created around the world with accelerating speed. There is a real risk of losing much of this data unless the compilers of language resources (primary and secondary data) and creators of tools start to pay more attention to the reusability of the resources and the interoperability of the tools. In this poster we report our effort to create best practices for the creation and dissemination of reusable speech resources in Finland. Our suggested solution allows *collaborative annotation*, which means that researchers in different sites can work on the same speech data, adding different kinds of linguistic annotation and share their work with other researchers.

Reuse is Still a Problem

The reusability and interoperability of language resources has been a recurring theme in language resources related conferences and workshops in recent years. In short, the reuse of existing language data and linguistic annotation is currently too cumbersome and time consuming if it is at all possible.

In the long term the reusability problem will hopefully be solved by the emergence of international standards. As has been reported in the previous LREC conference there is a working committee of the International Standards Organization (ISO TC 37 SC4) working on standards for language resources[1].

One cannot, however, put resource creation on hold while an international standard is in the making. In the short term, time is best used in developing competing tools, services and best practices that are *reusable*, *extendible*, *interoperable* and *portable*. Efforts to create best practices should also give important feedback to the standardization process.

By reusable we mean that the language resource must outlast the project where the resource was created and be usable as it is for different purposes by different users in different environments. By extendible we mean that one must be able to add new types of linguistic annotation to the language data without breaking the existing language resource, e.g. the tools must still work. By interoperable we mean that the language resource can be processed with several contemporary tools. By portable we mean that the language resource must be convertible to other formats, e.g. a future standard, without loss of information. (See Bird & Simmons, 2003, for a related discussion.)

Best Practices for Speech Resources

In 2002, the Finnish IT center for science CSC, the University of Helsinki, Helsinki University of Technology and the University of Turku began working on the project "Integrated resources for speech technology and spoken language research in Finland" funded by the Academy of Finland. The main objective of the project was to create a maximally reusable language resource and tools. In practice, this meant creating a speech corpus (recordings and annotations) with editing and search tools that would suit the very different needs of phoneticians, speech technologists and conversation analysts.

The main contributions of CSC in this three-year project are:

1. An annotation model with an annotation vocabulary and an exchange format
2. An annotation editor for speech corpora
3. A storage, discovery and delivery service for speech resources (speech corpora of recordings and annotations)

Annotation Model

The design process of the exchange format and annotation editor led to the creation of an annotation model that consists of an annotation vocabulary, defining what can be annotated, and an exchange format, defining how the annotations are stored and distributed.

The annotation model must accommodate two rather contradictory objectives. On the one hand we want the users to create consistent and explicit annotations. This calls for the vocabulary to be well defined and well documented to avoid terminology mix-up. On the other hand we cannot restrict the research agenda in beforehand -- the users need to be able to annotate whatever they want. Therefore the annotation vocabulary must be extendible, but in a controlled way.

RDF (Resource Description Framework) and RDF Schema caught our attention when we were looking for technology that would closely reflect the conceptual model and annotation vocabulary that were drafted in the first stage of the project[2]. It seems that RDF has been considered for similar purposes also by others (Ide & Romary, 2003).

RDF Schema is used for defining vocabularies by creating classes and properties, much like class definitions in object-oriented programming. RDF Schema has a built-in

[1] http://www.tc37sc4.org/

[2] http://www.helsinki.fi/~lennes/sapuhe/sanasto.html (in Finnish)

mechanism for inheritance, which allows hierarchical organization of vocabulary items.

Annotation Vocabulary
We used RDF Schema to define an annotation vocabulary with a basic set of annotatable items[3]. At the top of the hierarchy, there is an abstract AnnotationUnit class. All units that can be annotated in our annotation model are derived ultimately from this abstract class. For example, to annotate creaky or laryngealized voice, one uses the CreakyVoice class that is a subclass of VoiceQuality, which in turn inherits Speech that finally inherits the abstract AnnotationUnit class. Because the subclass inherits the properties of its ancestor, it is easy to create specializations of existing classes, as in Figure 1.

```
<rdfs:Class rdf:ID="NounPhrase">
  <rdfs:subClassOf rdf:resource="#Chunk" />
  <rdfs:label>Noun Phrase</rdfs:label>
  <rdfs:comment>
    A constituent of a sentence that consists
    of a noun and any modifiers it may have,
    a noun clause, or a word, such as a pronoun,
    that takes the place of a noun.
    (Source: Collins English Dictionary)
  </rdfs:comment>
</rdfs:Class>
```

Figure 1. The simplest case of inheritance. A definition of a NounPhrase class that inherits all properties of the Chunk class.

Due to the fact that annotation units are defined as classes, an annotation unit is not merely a label assigned to a stretch of time. Instead they can have an arbitrary number of properties, such as language, creation stamp, sound source, and creator etc., in addition to the usual start and end time and general-purpose label. Annotation is therefore not just labeling, but creating possible complex records of annotation. See Figure 2 for an example.

Exchange Format
The exchange format we use is RDF in XML syntax, where a set of instances (annotation units) of the same class is stored in one file (an annotation tier). Figure 2 shows one instance of a Word.

```
<Word rdf:ID="Word-1">
  <label>myrsky</label>
  <language>fi-FI</language>
  <pos>N</pos>
  <case>Nom</case>
  <number>SG</case>
  <englishGloss>storm</englishGloss>
  <start>0.88</start>
  <end>1.44</end>
  <status>0</status>
  <soundSource rdf:resource="#Speaker-1"/>
  <creationStamp>
    2004-02-26 10:25:14 ling
  </creationStamp>
</Word>
```

Figure 2: An example of an annotation unit with several editable properties.

[3] http://www.csc.fi/kielipankki/puhe/schemas/official/annotation/coreUnits.rdfs

RDF/XML is an open and well-supported standard with a simple data model. Several parsers are freely available for many platforms and many programming languages, e.g. the rdflib[4] parser for Python that we use.

A disadvantage of RDF compared to a custom XML Schema is that RDF is designed for resource *description*, not validation. A statement in RDF is merely a claim; there is no checking that the claim is actually valid. This means that it is up to the tools reading and writing RDF to interpret and validate the statements against the RDF Schema.

Annotation Editor

In order to make annotating easy according to the annotation model, CSC developed an annotation editor called the *Puh Editor.* The editor acts as the interface between the annotator and the annotation model. It is freely available on the web[5].

The starting point was to reuse existing open source tools and components by adding support for our annotation model. We examined several available open source tools (e.g. Praat, QuickSig, Transcriber, AGAPPS and TASX-Annotator) and finally decided to create our own graphical user interface using the same low-level components as AGAPPS.

The Python programming language, the Snack sound library[6] and the Wsurf sound visualization widget[7] (Sjölander & Beskow 2000, Sjölander 2002) made it possible to create a multi-platform GUI application with one man-year of effort.

The editor secures the consistency of the annotation by checking that the annotation follows the definitions in the annotation vocabulary. As a simplified example, the editor will not allow the user to insert the string 'Noun' in the part-of-speech property of a Word unit if the vocabulary requires the value to be one of 'N', 'V' and 'A'.

The Puh Editor allows the researcher to:

1. Add and edit annotation of recordings
2. Define annotation units
3. Create recordings from signal files by assigning metadata
4. Publish annotation to other researchers
5. Download annotation made by other researchers
6. Write his own plug-ins in Python for programmatically manipulating annotation
7. Import annotation from Praat TextGrids and plain text files

[4] http://rdflib.net/
[5] http://www.csc.fi/kielipankki/puhe/
[6] http://www.speech.kth.se/snack/
[7] http://www.speech.kth.se/wavesurfer/

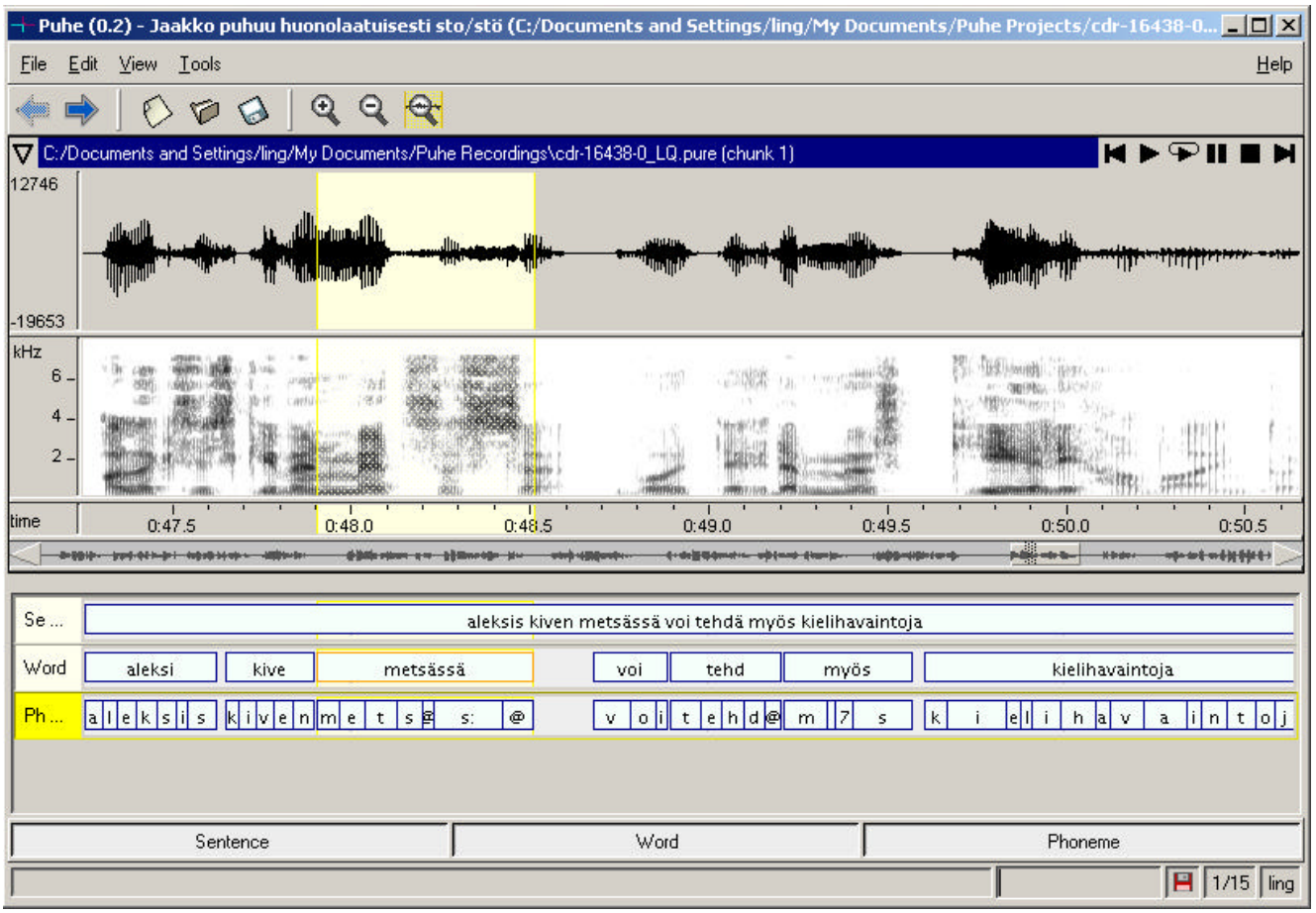

Figure 3. Puh Editor main window with annotation of the classes Sentence, Word and Phoneme.

Storage, Discovery and Delivery

The nexus of the speech resource service is the Language Bank of Finland maintained by CSC. The Language Bank takes care of storage, distribution, backups, security, rights management and documentation of the recordings and the annotation. The recordings are stored on a limited access file server. Published annotation tiers are freely available on the web.

A web service has been implemented to support discovery and distribution[8] of the recordings available at the Language Bank. Download access is restricted to formally registered users that have signed an end-user agreement. This secures the rights of the data providers.

A central web repository for sharing annotation tiers in RDF/XML format is also available. The Puh Editor integrates with the repository so that published annotation tiers can be downloaded directly from the editor. The editor can also publish annotation tiers created by a user to the web repository. The automated sharing architecture is the core of collaborative annotation.

8 http://www.csc.fi/kielipankki/aineistot/naytaPuheAineistot.phtml (in Finnish and Swedish only)

Last but not least, plug-ins to the Puh Editor can be distributed automatically to all users by publishing the plug-in code in the web repository. Plug-ins are revised and published by the administrators at the Language Bank in order to secure that malicious code is not distributed.

Collaborative Annotation in Practice: A Use Case

The following is a simple use case of collaborative annotation:

A researcher with privileges to use the speech resources at the Language Bank of Finland is interested in annotating the word order in radio news broadcasts with the aim of extracting frequency information. He uses the web service to locate those recording files that contain radio news broadcasts and downloads the files to his own computer using his web browser. He then launches the Puh Editor and starts a new annotation project with the first of the downloaded recordings as primary data. The speech signal in the recording is visualized as a waveform and spectrogram (by default).

The researcher asks the Puh editor to check what annotation already exists for the recording he is annotating. If other researchers have already been working on the same recording and decided to share their work, their annotation will be available for download.

Lets assume that some sentence annotation already exists and the researcher decides to import it from the web repository. The annotation appears in the Puh Editor as an annotation tier, where each sentence is clearly marked as an annotation unit.

Unfortunately, the simple units of the class Sentence are not good enough for the researcher, as there is no word order property available. Therefore, the researcher uses the Puh Editor to define a new annotation unit, SentenceWithWordOrder, which is inherited from Sentence and distinguished from it by having an additional property, wordOrder, that is allowed to take any of the following values: “SVO”, “SOV”, “VOS”, “VSO”, “OSV” and “OVS”.

When the new unit is defined, the researcher adds a new annotation tier to the project accepting SentenceWithWordOrder units. He then copies all sentence units to the newly created annotation tier. The Puh editor will ask the researcher if he wants to convert the Sentence units on the clipboard to SentenceWithWordOrder units. As the researcher accepts that, the sentences appear on the new tier as SentenceWithWordOrder units.

To begin annotating the word order of sentences, the researcher tells the Puh Editor that the wordOrder property of the SentenceWithWordOrder units on the annotation tier will be annotated next. When he double-clicks the first sentence, a pull down-menu appears with the valid alternatives (“SVO, “SOV” etc.). The researcher chooses the correct alternative of these if possible. At any time he can naturally play the time-span of the sentence in the recording and hear the newscaster speak.

When the researcher is finished with his annotation he uses the publishing function in the Puh Editor to publish the SentenceWithWordOrder tier to the web repository at the Language Bank of Finland. The annotation and the definition of the SentenceWithWordOrder unit are uploaded and by the next morning they are available to all other researchers using the speech resource.

Finally, the user wants to calculate the frequencies of the word order in the sentences in the recording. Luckily, there is a plug-in suitable for this task, that the researcher can use to generate a frequency table and store it as a file on disk.

References

Altosaar, T. et al. (2003). Designing a Finnish Multimodal Speech Database System. In Proceedings of the 15th International Congress of Phonetic Sciences (pp. 1369--1372), Barcelona, Spain.

Bird, S. & Simmons, G. (2003). Seven Dimensions of Portability for Language Documentation and Description. In Language 79:3 (pp. 557--582).

Ide, N. & Romary, L. (2003). Outline of the International Standard Linguistic Annotation Framework. In the Proceedings of ACL'03 Workshop on Linguistic Annotation: Getting the Model Right (pp. 1--5), Sapporo, Japan.

Sjölander, K. and Beskow, J. (2000). Wavesurfer -- An Open Source Speech Tool. In Proceedings of the 6th International Conference on Spoken Language Processing (464--467), Beijing, China.

Sjölander, K. (2002). Recent Developments Regarding the WaveSurfer Speech Tool. In Speech, Music and Hearing Quarterly Progress and Status Report 44 (pp. 53--56), Stockholm, Sweden.