The Validation of Speech Corpora

Florian Schiel

Angela Baumann, Christoph Draxler,
Tania Ellbogen, Phil Hoole, Alexander Steffen

Version 1.10 : March 21, 2012

Contents

Chapter 1

Introduction

1.1 Summary

This document is the result of a study conducted within the German BITS project in 2002. BITS[1] is an acronym for *BAS[2]: Infrastructures for Technical Speech Processing* and is a 100% publicly funded project devoted to the improvement of the infrastructural situation in *Spoken Language Processing* (SLP) of the German language. One of the sub-projects of BITS aims to come up with a cookbook-like document on the topic of *Speech Corpora Validation.*

Speech Corpus in the scope of this document means a collection of digital recordings of speech created with the aim of exploring the functioning of speech communication, often with respect to certain technical applications like *Automatic Speech Recognition* (ASR), *Speech Synthesis* or *Speaker Verification* etc.

The term *Validation* refers to a process that analyses and documents either a completed speech corpus or a speech corpus that is in the process of being produced with regard to its specifications.

Speech Corpus Validation has several important applications in the field of Spoken Language Processing (SLP):

- **Quality control:** Validation is carried out during or in the last phase of the production of a new speech corpus, either
  - by the producer (*inhouse validation*) or
  - by an independent validation organization (*external validation*)

[1] www.bas.uni-muenchen.de/Forschung/BITS

[2] BAS = Bavarian Archive for Speech Signals.

to ensure certain levels of quality.

- **Controlling:** Validation is carried out by the buyer of a speech corpus to ensure that the speech corpus does meet his/her needs.

- **Improvement:** By validation of existing speech corpora, these corpora may be improved for future re-use.

- **Comparability:** Validation carried out under certain standardized guidelines might lead to a quality grade that simplifies the selection between different existing speech corpora of similar specifications for a certain task.

This document is a cookbook for speech corpus validation. It is the result of the validation experiences gained at the Bavarian Archive for Speech Signals (BAS)[3] in numerous corpus collections.

1.2 Intended audience

This document should act as a guideline for speech corpus validation. It may be used as introductory reading for the newbie or as a reference and/or check list for the experienced scientist/engineer. More specifically it will be most likely used by

1. producers of speech corpora (quality control)

2. institutions that are about to invest in a speech corpus/ speech corpus production and want to perform their own validation

3. institutions that do external validations for other parties

If the validation is not carried out for inhouse purposes, but initiated by an external producer / buyer / client we will refer to this producer / buyer / client as the 'client' for the remainder of this document, whereas the institution that performs the validation is referred to as the 'validator'. The person / institution that actually produces the speech corpus in question will be referred to as the 'producer'. Note that in some cases all three might be the same.

The cookbook is not intended to be used for the quality assessment of speech corpora. If you are interested in this – much more difficult – task, please refer to [2].

[3] www.bas.uni-muenchen.de/Bas

Furthermore, the document does not cover the basic knowledge about *Digital Speech Processing* or even more specialized topics like the above mentioned applications in the field of SLP. We recommend referring to the document *The Production of Speech Corpora* ([5]) for details about best practice in this closely related topic.

At the end of many chapters you will find a check list where all the main points to follow are listed in an abbreviated form. If you do not understand contents of these lists, you may easily find the sections describing the topic in more detail by following the references given to each keyword. All check list (including the chapter 2 of [2]) are summarized in appendix A.

1.3 "Validation" in this document

The term *validation* in the context of spoken language resources (SLR) has slightly different meanings depending on the authors.

Henk van den Heuvel describes the main goals of the validation process of a SLR as

> 1. Checking the SLR against a fixed set of requirements.
> 2. Putting a 'quality stamp' on a SLR as a result of the aforementioned check. If the database passes the check, then we say that is has been "validated".
> 3. The *evaluation* of a SLR in a field test, thus testing the usability of the SLR in an actual application.
> 4. ...
>
> *(e.g. [1], p. 1)*

The *European Language Resources Association (ELRA)* defines the term *validation* as follows:

> "The term 'validation' in ELRA is normally used in reference to the activity of checking the suitability for the market, the adherence to standards, and the quality control of the LR product."
>
> ...
>
> *([3])*

Both sources subsume the *evaluation* of a SLR, that is a *quality assessment* for the usability in an actual application or for the marketability, as an integral part of the validation.

In this document we will concentrate only on the first point in van den Heuvel's list: the **validation against the specification of a SLR** or – if no specification is available – against the documentation.[4]

We agree with Heuvel on the second point that it is essential for the future infrastructure of SLRs to come up with a methodology to assess the quality of a SLR against basic standards ('good practice') to achieve a quality grade of an existing SLR. Please refer to the excellent paper "Validation of Content and Quality of Existing SLR: Overview and Methodology" by H. van den Heuvel et al ([2]) for this topic.

1.4 Terms and definitions

The following is a list of short definitions for technical terms as used throughout this document:

- Speech Corpus = physical time signals, in most cases sound pressure or other measurable time signals, recorded from the act of speaking[5], together with a minimal set of description (annotations, meta data, ...) stored on a digital medium.[6]

- Validation = the (formal) check of a speech corpus with regard to its pre-defined specifications following a documented or standardized procedure and resulting in a validation report and/or a validation quality grade.

[4] We deem the *evaluation* of a SLR a process that can in most cases be carried out only with regard to a certain specific application of the SLR. Therefore we argue that it is very difficult, if not impossible, to evaluate a SLR beforehand and for all thinkable future applications.

For example, the BAS catalogue contains scientific speech corpora that were produced for certain very specific investigation in discourse theory. Since these speech data were produced without any machine readable annotations, an evaluation in the above sense carried out at the time when the SLRs were added to the BAS would have undoubtedly resulted in a very negative verdict: "Not usable for any SLP applications!"

However, it turned out that with today's enhanced indexing techniques these SLRs are very valuable because they contain spontaneous language very close to what is used in normal speech communication. Therefore, engineers now start using these data for their respective applications in Human Computer Interfaces (HCI).

[5] Aside from the speech signal these time signals may include: laryngographic signal, electropalatographic signal, coordinate parameters derived from EMA (Electro Magnetic Articulography), X-ray movie (cineradiography), coordinate parameters derived from X-ray micro beam, air flow, nuclear magnetic resonance imaging, ultrasound imaging etc. In this cook book we will not give any specific instructions on how to use special recording hardware for the listed signals, because this would be far beyond the scope of this book.

[6] For the remainder of this document we will use the term 'corpus' instead of 'speech corpus'.

- Evaluation = a qualitative assessment of a corpus with regard to its usability in a certain task or development scenario or to its market value.

- Specification = the fixed technical description of a speech corpus with regards to all of its features (including annotations, meta data and documentation (see [5], chapter 4 for a detailed discussion of specifications).

- Internal/inhouse validation = validation carried out by the producer of a speech corpus during or after the production.

- External validation = validation carried out by an independent validation institution that is not linked in a any way to the producer of the speech corpus.

- (File) Format = Standardized or specified format of digital data. Either signal data or symbolic data (annotations).

- Annotation = Discrete (categorical) description of a physical signal (coding). Usually consisting of a closed set of symbols and a scheme to link these symbols to either points in time or segments in time.

- Domain = topic, or field of topics, or the situation in which a verbal communication takes place.

- Prompt = A speech item (word, phrase or sentence) presented to a speaker. *Prompt list* or *prompt corpus* is a collection of prompts that define the *spoken content* of the corpus.

- Spoken Content = What was spoken in a speech corpus.

- Meta Data = Data about data. In the context of this book the term meta data is restricted to three types: *recording protocols*, *comments* and *speaker profiles*.

- Codes = categorized data entries, in contrast to free text. If for instance the meta data parameter *place of birth* is restricted to the German states and the category ‘other’, then it is a code. A free comment about a recording success is no code and therefore not machine readable.

1.5 Acknowledgments

The writing of this document was made possible under a grant of the German Ministry for Education and Sciences (BMB+F grant number 01 IV B01) within the *BITS* project.

Angela Baumann and Tania Ellbogen from the BITS team helped tremendously with the research and overall structure of the document. Alexander Steffen was responsible for most of the logistics for this project, did the interviews with external sources and also helped with the overall structure. Christoph Draxler and Phil Hoole tuned the final manuscript to proper English and gave many useful hints.

Aside from the staff members of the *BITS* project and the *Bavarian Archive for Speech Signals (BAS)* we would like to thank Henk van den Heuvel for his valuable contributions.

1.6 Disclaimer

The contents of this document represent the joint knowledge of a group of experts in the field of speech corpora validation. It does not claim to cover all known methods and procedures in this field. The authors do not accept any responsibility for actions caused by readers following the recommendations of this document.

Chapter 2

Main Goal and Overview

The main goal of the following cookbook is to ensure that a speech corpus is usable in the sense that a prospective user may exploit the data of the corpus for his/her purposes without having to fight with technical problems. Most of these problems arise from one of the following basic 'flaws' of the corpus:

- Missing / inconsistent / wrong documentation
- Missing / incomplete / corrupt / wrong / superfluous / inconsistent data files
- Incompatibility between different OS (signals, fonts, control characters)

This list seems to be short but beware: the simple term 'inconsistent documentation' covers a wide range of possible errors.

The aim of the following step-by-step instructions is to detect such 'flaws' in a corpus. The validation procedure results in a *validation report*. This validation report summarizes the findings, both positive and negative with respect to the specifications. For the producer it is a proof that the work has been done properly, for the client it is an (independent) judgment on the technical quality of the corpus.

Since we cannot forsee all possible errors in all possible speech corpora once and for all time, the motivation of the user of this cookbook should be to **find all errors that might hinder the successful usage of the speech corpus**. Therefore we do not recommend following the instructions to the letter but rather seeing them as analogies that have to be adapted for the special needs of the actual corpus at hand.

The remaining document is organized as follows (see fig. 2.1):

The first Chapter 3 'Reference and Check List' describes how to define the reference against which the validation has to be performed. Since you cannot validate without such a reference, this has to be done first. We give hints on how to define the list of check items and how to set up a 'validation contract' with the producer / client.

Chapter 4 'Documentation' provides some help on how to tackle the problem of possible flaws in the documentation of the corpus. Since this is neither a problem of man power nor programming skills, we will simply give you some hints on how to 'see' the documentation with the eyes of a prospective user. This part is traditionally the hardest to perform in-house validations because it requires 'forgetting' everything that has been done during the production of the corpus.

Chapter 5 'Automatic Validation' covers all checks that might be performed automatically on the complete corpus. Since these checks require only programming skills and machine power, they can typically performed by one person or a very small group.

Chapter 6 'Manual Validation' deals with checks that cannot be performed automatically and will therefore most likely be applied only to a selected subset of the corpus. This chapter gives some hints about the selection techniques and describes some basic techniques for manual checking. Typically you will reserve more man power for this part of the process.

The tasks described in chapter 5 and 6 can be carried out in parallel.

Finally, in chapter 7 'Validation Report' we gives a rough structure of what should be contained in the final report.

Note that the result of a validation is not necessarily a perfect corpus. However, a speech corpus with well documented deviations from the specifications is more valuable than a corpus without a validation.

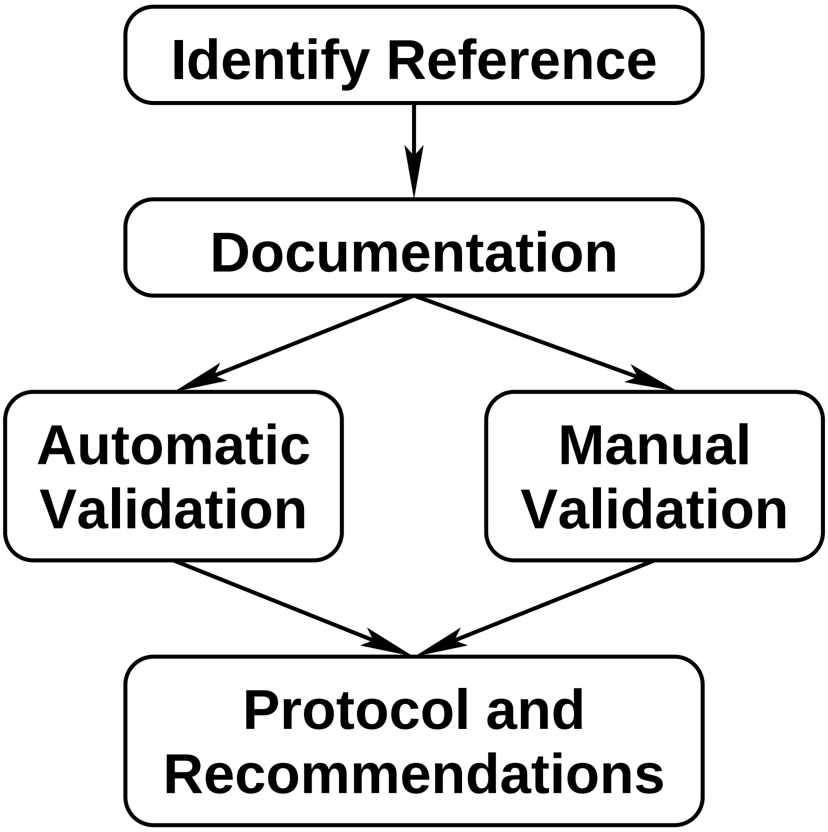

Figure 2.1: Typical task flow in corpus validation

Chapter 3

Reference and Check List

To perform a formal validation as described in this document you need a *reference*, a complete description of the corpus as it should be, and a *validation check list* that describes exactly what has to be checked in the validation.

3.1 How to get the Reference

Typically in the first working step you will browse through the corpus contents and other data provided by the producer / client and find one of the following situations:

- You are provided with a complete specification. Then use it as the reference. Goto the next chapter.
- You are provided with an incomplete (or even non-existent) specification. For example, some properties of the speech corpus like the lexicon are not specified, but you find them in the corpus or in the documentation. Now you have two choices:
  - You are able to fill the gaps in the specification by referring to the documentation of the corpus. Then produce an *extended specification* based on that and proceed with the next chapter.
  - You are not able to fill the gaps in the specification by referring to the documentation of the corpus. Contact the producer or the client and try to clarify the unspecified points[1]. Produce

[1] Typically, these will be the tolerance measures for found errors or deviations from the

an *extended specification* based on that source and write a first chapter for the final validation report listing the missing items in the documentation. Then proceed with the next chapter.

- You are not provided with a specification and the corpus does not have any documentation.[2] Contact the producer / client and state clearly that a validation is not possible in that case.

Beware: It is not wise to use the term *correctness* in all contexts. For instance the reference of a speech corpus should not contain phrases like:
"85% of the phonemic segmentations are *correct*."
The problem with the concept of *phonemic segmentation* (as well as with other linguistic annotations) is that it is hard to agree on what is correct and what is not. Therefore we recommend formulating specifications on items like the *phonemic segmentation* more cautiously, e.g.:
"The *interlabeler agreement* in the phonemic segmentation is at least 85%."
The same is probably true for *the transcript, the prosodic labeling, all kinds of segment boundaries etc.*

3.2 Check for Reference Completeness

To decide whether a reference is satisfactory for a successful validation, check the following points:

- Browse through the speech corpus and compile a 'survey list' in which you note down
  - the names of all directories or directory groups
  - file types (usually by the extension), e.g. `*.wav`, `*.par`, `*.ags`, ...

  Then check if everything found in the corpus is described in the specification. If not, the specification is incomplete.

- Check for basic meta data that must be mentioned to perform a validation[3] such as:

reference. For instance: "The allowed percentage of wrong word labels in the transcript must be less than 2%." In most corpus specifications or documentations there are no numbers concerning the reliability of annotations (SpeechDat being the praiseworthy exception from this rule)

[2] This scenario sounds very unlikely, but it is not: this happened a few times with very old SLRs that were transfered to the BAS.

[3] At least it must be stated that they are 'unspecified' and can therefore be disregarded by the validation process.

- Speakers: number and profile requirements: e.g. gender distribution, age distribution, regional distribution, ...
- Data description: formats, signal specification like sampling rate, word length, S/N ratio, ...
- Contents: Spoken prompts, domains, word distribution, word / utterance count per speaker / domain etc.
- Annotation: formats, numbers, procedures, labeling and segmentation tolerance ...

If in doubt whether a specification is essential, try to mimic a potential user of the SLR and decide whether the specification is needed or not.

The above procedure gives you only hints as to what to look for. In some cases speech corpora are produced with such a special purpose that some of the above listed items may not be important, but others may well be. Therefore we recommend communicating extensively with the producer / client in this phase.

3.3 Validation Check List

The next step after defining the reference is to formulate a description of the validation topics, i.e. what exactly has to be checked in the validation process. This *validation check list* must state precisely what proportion of the corpus must be validated manually. For instance, what percentage of the annotations have to be checked for correctness and what exactly are the references and tolerance measures for being considered ‘correct’. Also, list the required tools to perform the manual validations; these are usually very specialized software tools that should be provided by the producer together with the speech corpus. Both the validator and the producer / client should sign this document for approval.

3.4 Example

As an example appendix C contains the specification for the *WebCommand* speech corpus, while in appendix D you will find the original main corpus documentation file. Based on these data a validator and a client might come up with the following ‘validation contract’ specifying the reference and the validation check list:

Validation Contract

between .... (validator)

and .... (client)

1. The client and the validator agree that the validator will perform a validation of the speech corpus ‘WebCommand’ (corpus) with regard to common rules of best practice in the field of SLR production. The corpus and the software tools that were used for the annotation will be provided by the client (or a third party such as the producer) including all specifications and documentations as being originally delivered to the client.
2. The validation process is based on the specification of the corpus (technical annex ...). The validator will deliver his results in a confidential report to the client. The report should address errors / deviations from the specification in such detail that the client is able to correct these errors (if possible).
3. The validation will cover the following:
    - formal checks for completeness, terminology, readability and parsability of signal files, meta data and annotation files.
    - check for superfluous files in all locations of the SLR.
    - check of the technical specifications of signals files; empty signals; clipped signals; corrupt signal files
    - speaker distribution as stated in the specification. Documented sex checked in 50% randomly selected speakers.
    - completeness of documentation
    - consistency of speech corpus with documentation
    - readability (on Windows and Macintosh) of documentation files

- manual validation of a minimum of 10% randomly selected transcription files including the adherence to the specified prompt texts. Errors to be reported are: typos (the 'Duden' being the reference), mismatch between transcript and the spoken utterance in the recording, wrong noise marker.
- formal check and completeness of the lexicon (coverage), check for mismatch to spelling in the transcripts.
- manual validation of a minimum of 15% randomly selected lexical entries. Errors to be reported are: typos, mismatch to spelling in the transcripts, inconsistent canonical pronunciation.
- readability of distribution media on Windows, Macintosh and Linux

4. Time plan of validation:
   Begin: ....
   Intermediate report: ....
   Final report: ....

5. Compensation
   ....

6. Confidentiality and legal stuff
   ....

Signature Validator Signature Client

Check List — Reference

○ Identify the Reference (p. 15)
○ Check the Reference for completeness (p. 16)
○ Define the Validation Check List (p. 17)
  - What is validated formally and to what extent?
  - What is valdated manually (percentage) and to what extent?
  - What is expected in the validation report?
  - Time schedule?
○ Set up a Validation Contract (p. 17)

Chapter 4

Documentation

Validation of the documentation is simple for an external validator and hard for an internal validator, because the latter knows too much about the corpus. If you act as an internal validator, try to ‘erase’ everything you know about the project and pretend to have just received the corpus and would like to get going with it.

To validate the documentation of the corpus run through the following steps and summarize all missing items and/or deviations from the specifications in the validation report. If possible, list the missing items or numbers in the report to simplify the correction process.

- Check the reference (see chapter 3) for any specifications regarding the documentation.[1]. If you find any, check out whether they have been fulfilled.

- Identify all files that belong to the documentation and try to read them on different OS (in most cases Windows, Macintosh and Linux will suffice) and with standard software (like a text editor or Acrobat). If you find documentation files in other formats than plain ASCII, HTML or Portable Document Format (PDF), report this as not acceptable. If you find documentation files rendered in HTML, try to read them with three different browsers (e.g. Internet Explorer, Mozilla (Netscape) and Opera or `lynx`) on varying platforms (Windows, Linux, Macintosh). You will be surprised how many pages won’t work, especially frame based pages.

[1] Obviously there will be none, if you produced the reference yourself based on the documentation!

Don't even consider proprietary formats like Word or WordPerfect or StarOffice etc. Documentation should never be delivered in such formats. Put a note in the protocol that the producer should convert them into standard formats and re-supply them.

- Go through the 'survey list' you created in the previous chapter (p. 16) and check whether all items appear in the documentation.
- Finally, to check for a minimum standard documentation as expected for a speech corpus, go through the first chapter of [2]. A summary of this list of requirements is given in the following check list.

Administrative Information

○ Contact for requests regarding the corpus
○ Number and type of media
○ Content of each medium
○ Copyright statement and intellectual property rights (IPR)
○ Validation date(s)[2]
○ Validation person(s)/institution(s)[2]

Technical Information

○ Layout of media: file system type and directory structure
○ File nomenclatura: explanation of codes used; no 'white spaces' in file names
○ Formats of signal and annotation files: if non-standard formats are used, a full description is required and tools to convert this format into a standard format
○ Coding: PCM linear, Mu-Law or A-Law; if other codings must be used, they must be fully described
○ Compression: only widely supported compressions (e.g. zip, gzip) should be used
○ Sampling rate: rates others than 8000, 11025, 16000, 22050, 32000, 44100 and 48000 should be reported
○ Valid bits per sample: others than 8, 16 and 24 should be reported
○ Used bytes per sample[2]

[2] Added by the author; in some cases the number of valid bits per sample, e.g. 12, does not fill up a standard word (e.g. 2 bytes). It should then be documented which bits are valid and what values may reside in the remaining invalid bits.

○ Multiplexed signals: exact de-multiplexing algorithm; tools

Database Contents

○ Clearly stated purpose of the recordings
○ Speech type(s): multi-party conversations, human-human dialogues, human-machine dialogues, read sentences, connected and/or isolated digits, isolated words etc.
○ Instruction to speakers (full copy)[2]

Linguistic Contents of Prompted Speech

○ Specification of the individual text items
○ Specification for the prompt sheet design *or*
○ Specification of the design of the speech prompts
○ Example prompt sheet *or*
○ Example sound file from the speech prompting[2]

Linguistic Contents of Non-Prompted Speech

○ Multi-party: Number of speakers, topics, discussed, type of setting (formal/informal)
○ Human-human dialogues: type of dialogue (problem solving, information seeking, chat etc.), relation between speakers, topic(s) discussed, type of setting, scenarios
○ Human-machine dialogues: domain(s), topic(s), dialogue strategy followed by the machine (system driven, mixed initiative), type of system (test, operational service, Wizard-of-Oz[2])

Speaker Information

○ Speaker recruitment strategies
○ Number of speakers
○ Distribution of speakers over sex, age, dialect regions
○ Description/definition of dialect regions

Recording platform and recording conditions

[2] Added by the author.

○ Recording platform
○ Position and type of microphone(s)

  ○ Company name and type id
  ○ Electret, dynamic, condenser
  ○ Directional properties
  ○ Mounting

○ Position of speaker(s) (distance to microphone)
○ Bandwidth (if other than zero to half of sampling rate)
○ Number of channels and channel separation
○ Acoustical environment[2]

... plus for telephone recordings

○ Recording hardware, telephone link (analog, digital)
○ Network from where the call originated
○ Type of handset

... plus for recording in the automobile environment

○ Recording hardware[2]
○ Type of vehicle
○ Average speed of vehicle
○ Status of windows (open/closed)
○ Type of pavement
○ Audio equipment playing during the recording

Annotation (for each of the contained annotations)

○ Unambiguous spelling standard used in annotations
○ Labeling symbols
○ List of non-standard spellings (dialectal variation, names etc.)
○ Distinction of homographs which are not homophones
○ Character set used in annotations
○ Any other language dependent information (such as abbreviations etc.)
○ Annotation manual, guidelines, instructions
○ Description of quality assurance procedures

○ Selection of annotators
○ Training of annotators
○ Annotation tools used

Lexicon

○ Format
○ Text-to-phoneme procedure
○ Explanation or reference to the phoneme set
○ Phonological or higher order phenomena accounted for in the phonemic transcriptions

Statistical Information

○ Frequency of sub-word units: phonemes (diphones, triphones, syllables, ...)
○ Word frequency table

Others

○ Any other essential language-dependent information or convention
○ Indication of how many files were double-checked by the producer together with percentage of detected errors

Chapter 5

Automatic Validation of Data

This working step includes all checks on the corpus data that can be carried out automatically or require some technical background knowledge. Typically this will be done by one person with good programming skills and in parallel to the task described in the next chapter.

The following checklist contains probably more checks than necessary for your particular corpus. If you are sure that a check does not apply for your corpus, simply skip it. On the other hand try to think about checks that might be not included in the following checklist.

In some cases we have included sample scripts written in CSH running under Linux which is fairly readable like a pseudo-code. You can easily transform the code snippets into your preferred script language. We recommend using Perl as a scripting language, but if you love to hack Java, do whatever is fun for you.

Report all performed checks and their findings in the validation report. Describe exactly the testing method and the formulas for resulting numbers, so that the client/producer may reproduce the results if necessary. You may even include the used programs or scripts in the appendix of your report.

○ Media

Check all media for mountability and file system type. Check the mountability on at least three OS: Windows, Macintosh and Linux. Check whether all media contain the same file system type. For instance in plain ISO9660 the characters of the file names appear in capital letters. If there is an additional Rock Ridge Extension in the ISO9660, then on some platforms (for instance UNIX) the file names will appear in small letters. This may cause problems and incompatibilities with tools and scripts.

A good idea is to copy the whole corpus to hard disk — this also simplifies the following checks. If that is not possible (because of the size of the corpus), try to include all the following tests into one script that will then be run over all media of the corpus (e.g. by mounting one CDROM after the other). That way you minimize the handling of CDROMs and tapes to a minimum.

```
#
# Frame to check a large number of CDROMs directly
# In this case 32 volumes of a speech corpus
#

set LOGFILE = Logfile.txt
set VOLCNT = 32
set volnr = 1

if ( ! -e $LOGFILE ) touch $LOGFILE
echo "" >> $LOGFILE
echo "Start validation script at: >> $LOGFILE
date >> $LOGFILE
echo "" >> $LOGFILE
umount /cdrom
while ( $volnr <= $VOLCNT )
  echo ""
  echo "Insert the next CDROM number $volnr and hit RETURN"
  set inp = $<
  mount /cdrom
  if ( $status != 0 ) then
    echo "ERROR: cannot mount CDROM number $volnr - skipping \
        checks" >> $LOGFILE
  else
    echo "CDROM $volnr mounted successfully" >> $LOGFILE
    #
    # Add the checks per volume here
    #
    (
       ...
       ...
    ) >> $LOGFILE
    #
    #
    #
  endif
  @ volnr ++
end
umount /cdrom
echo "" >> $LOGFILE
echo "End validation script at: >> $LOGFILE
date >> $LOGFILE
echo "" >> $LOGFILE
```

○ Completeness

Check for all specified *signal, annotation and meta data files.* Count them and report deviations. Report any other files that are not specified in the reference. Has every signal file the appropriate number of annotation files?

○ File Names

Have the found files the correct file name? Are there mismatches between signal files and annotation files?

For the following example script assume that the signal files are of type `WAV` and stored in groups of 182 each in subdirectories under the main directory `data`. Each subdirectory contains the data of one recording session (001-345) coded into the name of the dir (`SESnum`) as well as into the file name of the signal files (`SESnum_item.wav`). Corresponding annotation files of type `PAR` and `AGS` are stored in the same structure but under the main directory `annot`. Furthermore there has to be a recording protocol (`SESnum.rpr`) in the directory `meta/rpr`.

```
#
# Check for completeness and superfluous files
#

set sesssioncnt = 345
set signalcnt = 182
set datamain = /cdrom/data
set annotmain = /cdrom/annot
set metarpr = /cdrom/meta/rpr
```

```
...

# collect data
cd $datamain
set sessions = 0
set totaldirs = `ls -a | wc -l`
@ totaldir -= 2
foreach ses ( SES[0-9][0-9][0-9] )
  if ( ! -d $annotmain/$ses ) then
    echo "ERROR: missing annotation dir $annotmain/$ses"
    set checkannot = 0
    set totalfilesannot = 0
  else
    set checkannot = 1
    set totalfilesannot = `ls -a $annotmain/$ses | wc -l`
    @ totalfilesannot -= 2
  endif
  if ( ! -e $metarpr/$ses.rpr ) then
    echo "ERROR: missing meta data file $metarpr/$ses.rpr"
  endif
  set files = 0
  set totalfiles = `ls -a | wc -l`
  @ totalfiles -= 2
  foreach file ( $ses/$ses_[0-9][0-9][0-9].wav )
    set basename = ${file:t}
    set basename = ${basename:r}
    if ( $checkannot == 1 ) then
      if ( ! -e $annotmain/$ses/$basename.par ) then
        echo "ERROR: missing annotation file PAR for $file
      else
        @ totalfilesannot --
      endif
      if ( ! -e $annotmain/$ses/$basename.ags ) then
        echo "ERROR: missing annotation file AGS for $file
      else
        @ totalfilesannot --
      endif
      #
      # Add here: Other checks on the annotation files
      #
    endif
    #
    # Add here: Other checks on the signal file
    #
    @ files ++
  end
  if ( $files != $signalcnt ) then
    echo "ERROR: number of signal files in session \
        $ses ($files) not equal $signalcnt"
  else if ( $totalfiles > $files ) then
    echo "ERROR: superfluous or wrongly named files \
      in $datamain/$ses"
  endif
  if ( $totalfilesannot > 0 ) then
    echo "ERROR: superfluous or wrongly named files \
      in $annotmain/$ses"
  endif
  @ sessions ++
end
if ( $sessions != $sessioncnt ) then
  echo "ERROR: number of recording sessions ($sessions) \
      not equal $sessioncnt"
else if ( $totaldirs > $sessions ) then
  echo "ERROR: superfluous or wrongly named \
    directory in $datamain"
endif
```

○ Readability, Empty Files

Are any of the found files empty (zero byte length)? Are they readable?

Add something like the following to the previous piece of code at the comment `Other checks on the signal/annotation file`:

```
...
  if ( -z $file ) then
    echo "ERROR: file $file is empty"
  endif
  cat $file > /dev/null
  if ( $status != 0 ) then
    echo "ERROR: file $file is not readable"
  endif
...
```

Instead of emptiness you may also check for a defined minimum length in signal or annotation files. For instance, if you know that each signal should be at least 1 sec long, the minimum byte length of a `WAV` type sound file with 16kHz sampling rate, 16 bit, mono would be: $44 + 16000 * 2 = 32044$ (headerlength is 44):

```
...
  set length = ‘cat $file | wc -c‘
  if ( $length < 32044 ) then
    echo "Warning: signal file $file is less than 1 sec long"
  endif
...
```

○ Signal format

Do the signal files contain correct standard formats? A good way to test this is `sox`[1] (this test automatically implies readability!).

```
...
  sox -V $file -t raw /dev/null
...
```

This command will issue an error message, if `sox` cannot parse the sound file format (which must be known to sox! Check the man page!). The option `-V` will cause `sox` to print out information about the contents, such as

```
sox: Detected file format type: wav
sox: Chunk fmt
sox: Chunk data
sox: Reading Wave file: MS PCM format, 1 channel, 22050 samp/sec
sox:         44100 byte/sec, 2 block align, 16 bits/samp, 190840 data
bytes
sox: Chunk LIST
sox: Input file /usr/share/gallery/sounds/untie.wav: using
sample rate 22050
       size shorts, encoding signed (2’s complement), 1 channel
sox: Input file /usr/opt/office52/share/gallery/sounds/untie.wav:
comment "1995-04-28
```

You may pipe this output into a script to detect deviations from the expected parameters, for example:

```
...
  sox -V $file -T raw /dev/null | \
```

[1] SOundeXchange http://www.spies.com/Sox/

```
  gawk -v FILE=$file '/Detected file/ { if($NF != "wav" ) \
  { printf("ERROR: file %s is not a valid WAV format\n",\
    FILE) } }'
...
```

○ Annotation, meta data and lexicon file format

Are all annotation and meta data files and the lexicon parsable? If you are lucky, they contain XML with a corresponding DTD or XML scheme description in the documentation; if not, write a simple crude parser to check them. Report any non-parsable formats, because they are essentially not usable.

Do all annotation files, the meta data files and the lexicon have consistent line terminators? DOS requires a combination of CR (Hex 0D) followed by LF (Hex 0A), while UNIX requires only the LF (Hex 0A). Mixed usage of the line terminators may be caused by working on mixed platforms. They may cause problems when parsing the annotation files later.

A simple test for all lines in a annotation file to be DOS-compatible would be[2]:

```
...
  cat $file | tr '\r' '&' | grep -v '&$' > /dev/null
  if ( $status == 0 ) then
    echo "WARNING: $file contains lines not DOS-compatible"
  endif
...
```

To check for UNIX conformity, simply delete the grep option `-v`[3].

○ Annotation and lexicon contents

If not already done in the previous steps[4], write a simple script to extract labels from the annotation files and check them for inconsistencies.

- Cross-check the found labels with the documentation of the labeling. Are all found labels documented? Are there any documented labels not found in the annotations?

[2] In this example the character & must not be contained in the annotation files; in case it does, choose another character that does not.

[3] DOS-compatible text files are preferable, because UNIX usually has no trouble processing them.

[4] Beware: a XML parser using a DTD cannot check for correct label categories etc., because a DTD describes only the syntax of a XML document, but is not powerful enough for lexical analysis of semantics.

- Report any digits or numerals that are not written in their full orthographic form.

- Report any punctuation used in the annotations. There shouldn't be any except in cases where they are separated from other items by white space and have a special meaning (for instance prosodic).

- Report any words that are written with an initial capital because they are at the beginning of a sentence.

- Cross-check all words extracted from the transcripts with the spelling in the orthographic part of the lexicon.

Also you might check the timing information in label files for overlapping segments or gaps between segments, if this should not happen according to your reference.

○ Character code checks

If not already covered by the last step, check all annotation files of one type for the number of used character codes and compare them to other annotation files, to the lexicon and statistic files (if any). If there are any deviations, list them in the report.

If all or parts of the annotations are rendered in HTML, also check for the consistent usage of named entities.

○ Cross checks of meta information

Cross-check meta information in signal files, meta files and annotation files. For example, the speaker ID could be contained in a NIST SPHERE signal file header, in a SAM label file and in a meta data file describing the speaker characteristics. If they are not all the same for the same recording item, something is wrong. Report deviations caused by upper and lowercase spellings. Do signal files and annotation files have exactly the same 'length'[5]?

These checks can be integrated into the other checks on signal and annotation files by simply looking up the corresponding meta data files.

[5] that is: the length of of the recorded speech signal vs. the total length as reported in the corresponding annotation files, e.g. the last boundary of the last segment.

○ Cross checks of summary listings

If the speech corpus contains summarizing tables that list the signal / annotation files (very often together with a pointer to the medium where the files are stored), check consistency of these summaries both ways: all existing files must be listed in the summary, all files listed in the summary must exist.

○ Tools, software

Install tools and software that come with the corpus exactly as directed in the documentation. If the software is not explicitly restricted to certain platforms, try Windows, Macintosh and Linux (for instance for Perl or Tcl/Tk scripts). Report any installation/usage problems.

Check List — Automatic Validation

○ Media	(p. 27)
○ Completeness	(p. 29)
○ File naming	(p. 29)
○ Readability, Empty Files	(p. 31)
○ Signal Format	(p. 32)
○ Annotation, Meta data, Lexicon Format	(p. 33)
○ Parse Annotation for Content	(p. 33)
○ Character codes	(p. 34)
○ Cross Checks on Meta Data	(p. 34)
○ Cross Checks on Summary Listings	(p. 35)
○ Tools, Software	(p. 35)

Chapter 6

Manual Validation

This chapter describes those validations of the corpus data that cannot be performed automatically and therefore require considerable efforts in terms of manpower. Typically, this will be carried out by a group of selected validators – usually with special skills and native speakers of the corpus language – under the supervision of a person responsible for the logistics. This supervisor must be able to

- select (maybe also train) the validators
- organize the selection of validation material
- organize the distribution of work load to the validators
- collect and check the results
- do quality control (e.g. by taking random sample checks)
- calculate the results (some basic statistics)

Since the variety of annotations is large, we cannot give detailed advice on how to validate all the different annotations schemes. Instead we will concentrate on some practical hints that will most likely be useful in all kinds of manual corpus validations.

6.1 Manual Validation Contents

Refer to your validation check list (see chapter 3) for the items of the corpus that have to be validated manually. Sometimes the exact contents to be validated are not given in the contract. Here are some typical contents that are usually subject of a manual validation:

> **Transcript:** Spelling based on a standard reference, use of capital letters, mismatches with spelling used in the prompt text / lexicon / annotation files, mismatch to the recording, wrong usage of markers etc.
>
> **Labeling/Tagging:** Wrong usage of labels, extra or missing labels.
>
> **Segmentation:** Deviation of segment boundaries / points in time of more than a defined threshold.
>
> **Lexicon:** Spelling based on a standard reference, use of capital letters, wrong canonical pronunciation as given in a standard reference.[1]
>
> **Meta data:** wrong sex of speaker, wrong dialect class (difficult).

Before you get started with the manual validation set up a list of possible errors being checked for and document these in the validation protocol.

6.2 Selection of Validation Data

In most cases manual validation will not concern the whole speech corpus. Typically, a fixed proportion of the annotation and meta data will be randomly selected for manual validation. The proportion is chosen so that the sample is "representative for the speech corpus". Actually, nobody exactly knows what that means. In practice, the proportion is set to an amount that can be treated by the validator without causing undue costs: 5-20% for smaller corpora (10000-100000 recorded items), 1-2% for very large corpora (>100000 recorded items).

You may use a truly random process (e.g. shuffled cards or dice) to produce random numbers. Use of a pseudo-random sequence, which can be generated by most programming languages, is easier.

Beware: We found that some programming languages actually generate the identical pseudo-random sequence every time the program or script is

[1] If there is no reference available or the reference does not give specific rules for the canonical pronunciation, check for consistency. For example, morphs that occur in more than one word should always be transcribed in the same way.

executed if the random number generator is not properly seeded. A good random number generator is for instance used in the gawk programming language.

The following example gawk script selects a random sequence of 40 session numbers from a corpus session range between 150 and 350. Since the random generator is seeded with the actual system time, it will generate a different sequence every new second. It also keeps track of the already selected numbers and will not produce the same session number twice:

```
BEGIN {
        srand()     # seeding the random number generator
        i = 1
        while(i<=40)
        {
          flag = 1
          while ( flag == 1 )
          {
            random = int(rand() * 200) + 150
            flag = 0
            for ( j in randarr )
              if ( randarr[j] == random ) flag = 1
          }
          randarr[i] = random
          printf("%03d ",randarr[i])
          i ++
        }
        printf("\n")
      }
```

In most cases the selection process not only involves random sequences but also a number of other constraints. For instance: equal distribution between sexes, certain proportions of special features within the corpus etc. There are several ways to implement such constraints on a random selection. The brute force approach is to run the random sequencer repeatedly until the resulting sample meets the required constraints.

Document the resulting data sample and your method for creating it in the validation report.

6.3 Validation Method

Once you have selected the validation data, you'll have to decide which validation method to follow. Basically there are four different methods (probably more):

- **One-pass Check**
  All selected annotations are presented together with the raw data to the validator and the validator decides for certain discrete categories, for instance *phonemic label correct / wrong* or *deviation less than 10msec / deviation between 20 and 25 msec / deviation larger than 25 msec* or *gender correct / gender wrong*[2] etc.

- **Multiple-pass Check**
  The same as the one-pass check but performed by a group of independent validators. The results of these checks have to be summarized accordingly. Very often concurring judgments across the group of validators are considered to be correct, while non-concurring judgments are double-checked by the supervisor who then decides, or – in the extreme – all non-concurring judgments are considered as errors. Needless to say, the multiple-pass check requires more man power.

- **One-pass Re-annotation**
  The annotation of the selected validation data is repeated under exactly the same conditions as during the speech corpus production by a different annotator (= validator). Then the results of both are compared to calculate deviations.[3]

  This method is considered to be 'more objective' because the validator is not biased by the results of the original annotation. This is certainly true for problematic linguistic items like 'phonemic categories' or 'segment boundaries'. However, this method has also its drawbacks: It is very hard to reconstruct the exact labeling conditions by the validator. You'll need at least one validator in your group that has the skills to do the annotation on the same level as the producer (or even better) and a supervisor who is able to ensure the quality of the annotation. Also, in most cases the inevitable discussion about "what is to be considered correct" ensues between the producer and the validator of the speech corpus.

[2] in case of meta data to be validated

[3] *Beware:* The innocent term "calculate deviations" may hold a bunch of systematic problems, especially with regards to segmental boundaries. Please refer also to the remarks about the term 'correctness' in chapter 3 (p. 16).

We consider this method to be very effective for the validation of annotations or meta data where the nature of the categorical system is well established and the validator has no problems to justify his/her annotations. These are typically: *the gender of a speaker, the transcript, dialog act or word segmentation* and *simple linguistic and noise tagging.* The following data types are considered to be problematic for re-annotation: *phonemic, prosodic, syllable, morph segmentation, dialect, age, phrase accent and boundaries.*

- **Multi-pass Re-annotation**
  Of course, as with the one-pass check, the method of re-annotation may also be extended to multiple-pass re-annotations. To make things short: everything gets much more complicated. In most cases this method is considered to be way too costly anyway.

6.4 Manual Validation Tools

In some cases the original transcription or annotation tools are part of the speech corpus and we recommend using them for the manual validation to maximize the coherence with the original data. However, in most cases the validator will have to fall back on his own validation tools.

For the validation of the transcript you may use a simple text viewer / editor together with a general purpose sound editor, e.g. Praat[4], Cool Edit etc. The sound editor should be capably loading the original signal files of the corpus and replaying selected portions of the signal. Do not use software that plays only the total utterance.

The validation of annotations usually requires specialized tools which should be provided by the producer of the corpus. If this is not the case, we recommend Praat as a very good tool for basic segmentations, prosodic tagging etc.

6.5 Logistics

With a re-annotation scheme you have to take care that the resulting files of your validators can be automatically compared to the original annotation files. Include all re-annotations into the validation report package.

Even if you are not using a re-annotation technique, we recommend that the validators create a copy of each validated transcript or annotation file

[4] General phonetic tool developed by Paul Boersma at the University of Amsterdam, www.praat.org

and mark the errors found in the file in a way that allows a later automatic extraction of the errors. For example, if the following is a piece of phonemic segmentation from the corpus

```
SAP: 2343 16574 h
SAP: 18917 9780 OY
SAP: 28697 2376 d
SAP: 31073 3289 @
```

and the validator checking this data decides that the phoneme category /d/ is wrong, he adds a special marked line into his copy of the annotation file like:

```
SAP: 2343 16574 h
SAP: 18917 9780 OY
SAP: 28697 2376 d
ANN_ERROR: SAP:    28697   2376 t
SAP: 31073 3289 @
```

This way the validator can provide detailed information about the errors to the client / producer, which is often required in the validation contract.

Only employ validators that are native speakers of the corpus language. If you are working with a group of validators, try to achieve the same level of expertise for all of them. For instance, if you are validating the phonemic segmentation of speech signals, only hire well trained phoneticians and let them participate in a special training to make sure everybody has the same conception of the potential errors found in the data.

Define an error scheme for each type of annotation, i.e. a closed set of error types together with their description and examples. Test the scheme on a small scale set of data before the whole group of validators starts working.

For larger validation groups use a database system to keep track of already validated data. Use some kind of server/client architecture to automatically deal out data that are not validated yet and to collect the results. A simple and very effective tool to achieve this is the WWWTranscribe tool. See appendix B for a short description of WWWTranscribe and how to get it.

Check List — Manual Validation

○ Organize the validator group and training (p. 37)
○ Define the validation contents (p. 37)
○ Select the data sample (p. 38)
○ Decide on the validation method (p. 40)
- One-pass Check
- Multiple-pass Check
- One-pass Re-annotation
- Multi-pass Re-annotation

○ Select and test the validation tools (p. 41)
○ Organize Logistics (p. 41)
- Checking method: create copies with special markers
- Recruit only native speakers of same expertise level
- Define error schemes
- Database, server / client architecture

Chapter 7

Validation Report

The validation report summarizes all validation results, gives recommendations for fixing errors and/or improving the overall quality of the speech corpus and gives an executive summary.
As a rough structure the validation report should contain:

- An executive summary listing the most prominent results of the validation.

- A short introduction to the speech corpus stating who produced it when and for what purpose.

- The results of the validation of the corpus documentation.

- All results of the automatic validation steps as listed in the validation contract, together with the methodology by which the results were achieved. If you list figures (percentage of errors), also give the appropriate confidence intervals[1]. Long listings of erroneous files should be put into the appendix.

- All results of the manual validation together with a description of the techniques, the selection scheme[2], the statistics used and a profile of the participating validators. Again, don't forget to give confidence intervals for the results obtained.

- A list of the tools and programs used.

- Other relevant observations outside the required validation steps.

[1] which heavily depend on the number of samples checked.

[2] List the selected files in the appendix.

- Comments on how the quality of the corpus may be improved and what could be done better in future corpus productions.

- In the appendix: reference or corpus specification or validation contract on which this validation report is based on, listings of errors.

Try to prioritize the results and distinguish clearly between errors that can be and errors that cannot be repaired.

If you are performing regular validations on several releases of the same speech corpus, it might be a good idea to include a table summarizing the results of the actual and previous validations.

As an example you will find the validation report on the speech corpus WebCommand in appendix E.

Bibliography

[1] H. van den Heuvel (2000): The Art of Validation, The ELRA Newsletter, Vol. 5(4), pp. 4-6.

[2] H. van den Heuvel, L. Boves, E. Sanders (2000): Validation of Content and Quality of Existing SLR: Overview and Methodology. ELRA/9901/VAL-1 Deliverable 1.1, Jan 2000.

[3] ELRA Validation Activities: *http:/www.elda.fr/valida.html*

[4] Dafydd Gibbon, Roger Moore, Richard Winski, eds. (1997). Handbook of Standards and Resources for Spoken Language Systems. Mouton de Gruyter, Berlin New York.

[5] Florian Schiel et al (2002): The Production of Speech Corpora, to appear.

[6] P. Baker, L. Burnard, A. McEnery, A. Wilson (1997): An Analytic Framework for the Validation of Language Corpora.

Appendix A

Summary of Check Lists

Reference and Validation

○ Identify the Reference (p. 15)
○ Check the Reference for completeness (p. 16)
○ Define the Validation Check List (p. 17)
 - What is validated formally and to what extent?
 - What is valdated manually (percentage) and to what extent?
 - What is expected in the validation report?
 - Time schedule?
○ Set up a Validation Contract (p. 17)

Documentation [1]

Check the provided documentation files for the following items:

Administrative Information

○ Contact for requests regarding the corpus
○ Number and type of media
○ Content of each medium
○ Copyright statement and intellectual property rights (IPR)
○ Validation date(s)[2]

[1] This list was compiled from [2], Chapter 1.

○ Validation person(s)/institution(s)[2]

Technical Information

○ Layout of media: file system type and directory structure
○ File nomenclatura: explanation of codes used; no 'white spaces' in file names
○ Formats of signal and annotation files: if non-standard formats are used, a full description is required and tools to convert this format into a standard format
○ Coding: PCM linear, Mu-Law or A-Law; if other codings must be used, they must be fully described
○ Compression: only widely supported compressions (e.g. zip, gzip) should be used
○ Sampling rate: rates others than 8000, 11025, 16000, 22050, 32000, 44100 and 48000 should be reported
○ Valid bits per sample: others than 8, 16 and 24 should be reported
○ Used bytes per sample[2]
○ Multiplexed signals: exact de-multiplexing algorithm; tools

Database Contents

○ Clearly stated purpose of the recordings
○ Speech type(s): multi-party conversations, human-human dialogues, human-machine dialogues, read sentences, connected and/or isolated digits, isolated words etc.
○ Instruction to speakers (full copy)[2]

Linguistic Contents of Prompted Speech

○ Specification of the individual text items
○ Specification for the prompt sheet design *or*
○ Specification of the design of the speech prompts
○ Example prompt sheet *or*
○ Example sound file from the speech prompting[2]

[2] Added by the author; in some cases the number of valid bits per sample, e.g. 12, does not fill up a standard word (e.g. 2 bytes). It should then be documented which bits are valid and what values may reside in the remaining invalid bits.

[2] Added by the author.

Linguistic Contents of Non-Prompted Speech

○ Multi-party: Number of speakers, topics, discussed, type of setting (formal/informal)
○ Human-human dialogues: type of dialogue (problem solving, information seeking, chat etc.), relation between speakers, topic(s) discussed, type of setting, scenarios
○ Human-machine dialogues: domain(s), topic(s), dialogue strategy followed by the machine (system driven, mixed initiative), type of system (test, operational service, Wizard-of-Oz[2])

Speaker Information

○ Speaker recruitment strategies
○ Number of speakers
○ Distribution of speakers over sex, age, dialect regions
○ Description/definition of dialect regions

Recording platform and recording conditions

○ Recording platform
○ Position and type of microphone(s)

  ○ Company name and type id
  ○ Electret, dynamic, condenser
  ○ Directional properties
  ○ Mounting

○ Position of speaker(s) (distance to microphone)
○ Bandwidth (if other than zero to half of sampling rate)
○ Number of channels and channel separation
○ Acoustical environment[2]

... plus for telephone recordings

○ Recording hardware, telephone link (analog, digital)
○ Network from where the call originated
○ Type of handset

... plus for recording in the automobile environment

○ Recording hardware[2]
○ Type of vehicle
○ Average speed of vehicle
○ Status of windows (open/closed)
○ Type of pavement
○ Audio equipment playing during the recording

Annotation (for each of the contained annotations)

○ Unambiguous spelling standard used in annotations
○ Labeling symbols
○ List of non-standard spellings (dialectal variation, names etc.)
○ Distinction of homographs which are not homophones
○ Character set used in annotations
○ Any other language dependent information (such as abbreviations etc.)
○ Annotation manual, guidelines, instructions
○ Description of quality assurance procedures
○ Selection of annotators
○ Training of annotators
○ Annotation tools used

Lexicon

○ Format
○ Text-to-phoneme procedure
○ Explanation or reference to the phoneme set
○ Phonological or higher order phenomena accounted for in the phonemic transcriptions

Statistical Information

○ Frequency of sub-word units: phonemes (diphones, triphones, syllables, ...)
○ Word frequency table

Others

○ Any other essential language-dependent information or convention
○ Indication of how many files were double-checked by the producer together with percentage of detected errors

Automatic Validation

○ Media (p. 27)
○ Completeness (p. 29)
○ File naming (p. 29)
○ Readability, Empty Files (p. 31)
○ Signal Format (p. 32)
○ Annotation, Meta data, Lexicon Format (p. 33)
○ Parse Annotation for Content (p. 33)
○ Character codes (p. 34)
○ Cross Checks on Meta Data (p. 34)
○ Cross Checks on Summary Listings (p. 35)
○ Tools, Software (p. 35)

Manual Validation

○ Organize the validator group and training (p. 37)
○ Define the validation contents (p. 37)
○ Select the data sample (p. 38)
○ Decide on the validation method (p. 40)
- One-pass Check
- Multiple-pass Check
- One-pass Re-annotation
- Multi-pass Re-annotation
○ Select and test the validation tools (p. 41)
○ Organize Logistics (p. 41)
- Checking method: create copies with special markers
- Recruit only native speakers of same expertise level
- Define error schemes
- Database, server / client architecture

Validation Report

○ Executive summary, overall result (one sentence)
○ List of all checks, results, methodology (error listings in appendix)
○ List of the used tools and programs
○ Manual validation techniques, selection, statistics

○ Other relevant observations
○ Comments

Appendix B

WWWTranscribe

WWWTranscribe is a tool for the annotation of audio signals via the WWW. It features an oscillogram display of the speech signal, audio output, editing buttons that simplify the task of annotating the signal, and a formal consistency checker for the annotations. WWWTranscribe was developed at the Bavarian Archive for Speech Signals (BAS)[1] within the SpeechDat project. Currently[2], it supports orthographic transcriptions according to the SpeechDat guidelines; other annotation systems can be added simply by extending the annotation object class hierarchy.
WWWTranscribe is implemented in Java using only the standard JDK classes to guarantee platform independence.

In WWWTranscribe, the transcriber logs in and enters the ID of the session to be transcribed. A session consists of a number of recordings, each containing a single utterance corresponding to a prompt in the interview. Once a recording is selected, the transcription page is displayed. It contains a single output button with a speaker icon, a signal display, transcription and comment text fields, an assessment menu, and save and clear buttons (see figure B.1). A click on the speaker button outputs the speech signal auditorily. For read items, the original text of the prompt sheet is displayed in the transcription field, for spontaneous speech this field is initially empty. Any text in the transcription field can be edited. The buttons below the transcription field perform some basic conversation tasks on the text in the transcription field, e.g.:

- text to lower or upper case

[1] Contact Dr. Chr. Draxler, draxler@bas.uni-muenchen.de, for more information regarding WWWTranscribe.

[2] Oct 2002.

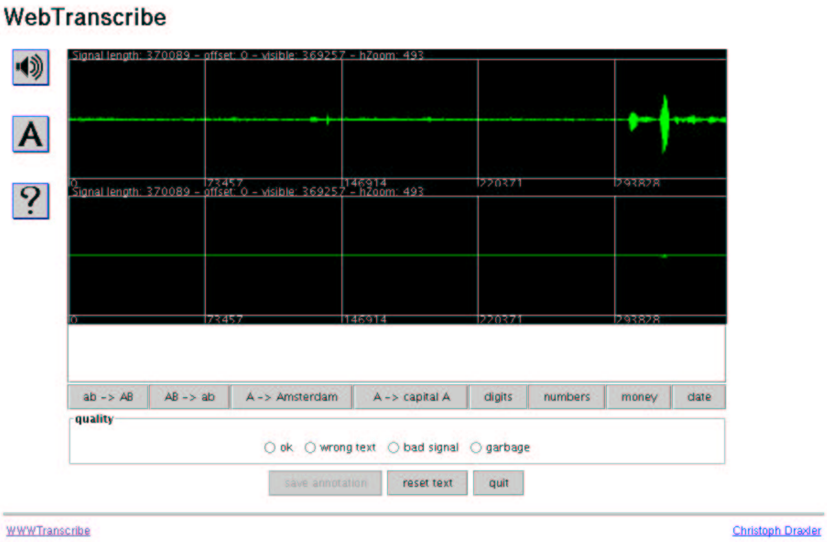

Figure B.1: Transcription page of WWWTranscribe

- digit sequences to orthographic digit or number strings
- money amounts and date expressions to orthographic strings

The Assessment pop-up-menu allows the transcriber to select general noise markers. Comments on the recording, e.g. on the quality of the speech or the signal, may be entered into the comment field. The Save button saves the transcription to the file system at the server site in the SpeechDat SAM database exchange format.

WWWTranscribe performs an automatic consistency check on the annotation text so that only formally valid annotations are entered into the annotation database.

At the BAS WWWTranscribe has been successfully used for a wide range of transcription, tagging, validation and evaluation tasks. WWWTranscribe is currently being packaged for public distribution[3].

[3] See www.bas.uni-muenchen.de/Forschung/BITS for updated information about the availability of WWWTranscribe.

Appendix C

WebCommand – Specification

Speaker Profiles	Speakers are native speakers of British English or French and at least 18 years old. Gender distribution is 50:50, all dialects allowed, education level not specified
Number of Speakers	At least 40 speakers had to be recorded, 20 for British English and 20 for French. The number of male and female speakers had to be preferably equal in every language.
Contents:	The contents of the corpus were specified by the client in form of an ASCII command list. The text corpus was fixed – that is all speakers recorded in one recording room spoke the same corpus of 135 command words. There are in total four text corpora: one for each of the two recording environments (see below) in the languages British English and French.
- Vocabulary	English: 163 words; French: 188 words
- Domain	Control commands and names
- Phonologic Distribution	No distribution specified
Speaking Style:	
- Read Speech	+

Recording Setup:	On-site Recording
- Acoustical environment	Each speaker is to be recorded on-site in two different recording rooms P and S on different days. The acoustical background consisted only of the hum of the recording device which was a regular Macintosh Desktop PC approx. 50 cm from the head of the speaker. The PCs were rated to be rather silent.
- Script	Speakers have to read from display in their native language
- Background noise	no artificial background noise specified
- Microphones	The speaker wears a ear-free headset Beyerdynamik NEM 192; a second Beyerdynamik MCE 10 is mounted on the upper left corner of a dummy laptop case that the user holds with both hands on his/her lap to simulate free speaking.
Technical Specifications:	
- Sampling Rate	22050 Hz
- Sample Type and Width	Sample Type: linear, not compressed.
- Number of Channels	Two channels recording: left channel: Beyerdynamik NEM 192; right channel:Beyerdynamik MCE 10.
- Signal File Format	File format: WAV stereo (RIFF)
- Annotation File Format	SAM annotation files according to SpeechDat specifications and a summarized annotation table for each recording block.
- Meta Data File Format	Table SPEAKER.TBL give a mapping of 4-digit speaker id to sex, age and mother tongue. Table SESSION.TBL contains a mapping of 4-digit session id to speaker id, place of recording, microphone types, channel mapping, environment. The file SUMMARY.TXT contains the SpeechDat conform summary of recordings: for each recording session all individual recordings are listed in the line. If a recording is missing, a '-' in listed instead of the three-digit prompt number.
- Lexicon Format	Two-column ASCII file: orthography and pronunciation coded in SAM-PA

Corpus Structure:	
- Structure	Recordings are stored in separate subdirectories for each combination of recording environment and language. The corpus contains 47 complete sessions (130 recordings per session). Care is taken that each speaker is recorded in complete sessions in each of the two recording rooms. Additional incomplete recording sessions are collected in the directories NOT_USED_FR (4 sessions) and NOT_USED_EN (7 sessions) respectively. Signal data are stored on DVD; a separate CDROM contains documentation, annotation files and pronunciation dictionaries.
- Terminology	Session names are coded as `SES#***` where `#` codes the combination of environment and language and `***` encodes the session number, e.g. SES6013 is the 13th recording session of a French speaker in room P. A mapping from speaker IDs to sessions, as well as the speaker profile can be found in the file `SESSION.TBL`. A recording file name is encoded as `Q1#***YYYY.WAV` where `YYYY` denotes the number of the text prompt (000-129) e.g. Q16013051.WAV contains the two microphone signals in a WAV stereo file of the 52nd prompt of the 13th recording session of French speakers in room P. The channel assignment for the microphones is stored in the file `SESSION.TBL`.
- Distribution Media	The corpus consists of two DVD-5 with a total size of 7.5 GByte plus a CD-ROM with the label files and documentation. On one DVD the data of the British speakers are stored; on the second DVD the data of the French speakers.

Release Plan	06.05.02 : Start of project, delivery of the prompts for both languages by ordering company. 01.07.02 : Database British English will be delivered to ordering company. 15.07.02 : Database British English will be delivered to ordering company. The client agrees that the corpus is offered to third parties via the national catalogue of the BAS and the international catalogue of the European Language Resource Association (ELRA) after a blocking period of one year. If the ELDA acts as a broker to deliver the corpus to a third party, ELDA earns a commission of 20% of the agreed royalties. A discount for research and for members of the ELRA is not provided.
Documentation	`REPORT.TXT:` main documentation including copyrights, history and error log (see section D for a complete listing) `SAMEXPORT.TXT:` summary of annotation `SESSION.TBL:` recording protocol: mapping of 4-digit session id to speaker id, place of recording, date of recording, microphone types, channel mapping, environment `SPEAKER.TBL:` speaker protocol: mapping of 4-digit speaker id to sex, age and mother tongue Documentation of SpeechDat annotation guidelines and format and pictures from the recording setup

Appendix D

WebCommand – Main Documentation

```
        _/_/_/_/         _/_/         _/_/_/_/
       _/      _/       _/ _/        _/      _/
      _/      _/       _/  _/       _/
     _/      _/       _/   _/       _/
    _/_/_/_/         _/_/_/_/        _/_/_/
   _/      _/       _/     _/             _/
  _/      _/       _/      _/             _/
 _/      _/       _/       _/    _/      _/
_/_/_/_/         _/        _/     _/_/_/_/
```

BAVARIAN ARCHIVE FOR SPEECH SIGNALS

University of Munich, Institut of Phonetics
Schellingstr. 3/II, 80799 Munich, Germany
bas@bas.uni-muenchen.de

```
----------------------------------------------------------------------



WEBCOMMAND 1.1 - on-site recordings for webpad voice control

----------------------------------------------------------------------

This is the documentation for the WEBCOMMAND database created in
Jun - Aug 2002 as a subcontract to Siemens Company.

WEBCOMMAND contains recording sessions of native speakers of
France and Great Britain. All speakers read a list of 130 prompts from
a screen. They are recorded with two microphones: a high quality headset
and a high quality microphone fixed to a 'webpad' hold on the lap.


------------------- Contents of this file ------------------------

   DVD directory structure
   Recording situation
   Naming conventions
   Signal file formats
   Transcription and error markers
   Annotation format
   Known errors
   History


----------------- DVD directory structure -------------------------

The corpus consists of two DVD-5 with a total size of 7.5 GByte plus a
CD-ROM with the label files and documentation ('DOCCDROM').

On one DVD (Webcommand_EN, #1) the british speakers are stored; on the
second DVD (Webcommand_FR, #2) the french speakers.

Recordings are situated in the 'BLOCK' directories:

BLOCK40  :  british, room P, 26 sessions
BLOCK50  :  british, room S, 26 sessions
BLOCK60  :  french, room P, 21 sessions
BLOCK70  :  french, room S, 22 sessions

The corpus contains 47 complete sessions (130 recordings per session).
Care is taken that each speaker is recorded in complete sessions
in each of the two recording rooms.
Additional incomplete recording sessions (speakers did not record a second
session, or corrupted sessions) are collected in the directories NOT_USED_FR
(4 sessions) and NOT_USED_EN (7 sessions) respectively.

The CDROM 'DOCCDROM' contains additional documents about the
```

corpus recording and annotation as well as pronunciation dictionaries:

```
PRON_FR.LEX     : Pronunciation dictionary, SAM-PA, french
PRON_EN.LEX     : Pronunciation dictionary, SAM-PA, english
TRANSCRP.PDF    : description of rules and conventions of SpeechDat
                  transcription (German)
TRANSCRP_EN.PDF : description of rules and conventions of SpeechDat
                  transcription (English)
PICS/           : Pictures of the recording setup
BLOCK##/        : SAM annotation files to recording block ##
REPORT.TXT      : this file
SAMEXPORT.TXT   : condensed summary of all SAM label files in one table
SUMMARY.TXT     : SpeechDat conform summary of recordings: foreach recording
                  session all individual recordings are listed in one line.
                  If a recording is missing, a '-' is listed instead of the
                  three-digit prompt number.
SPEAKER.TBL     : mapping of 4-digit speaker id to sex, age and mother tongue
SESSION.TBL     : mapping of 4-digit session id to speaker id, place of
                  recording, date of recording, microphone types, channel
                  mapping, environment
```

---------------- Recording Situation -------------------------

Each speaker (complete sessions only!) was recorded in two different recording rooms P and S on different days. Each session consists of 130 prompts as given in the prompt lists doc/PROMPTS*.
The speaker wears a ear-free headset Beyerdynamik NEM 192; the second mic is a Beyerdynamik MCE 10 mounted on the upper left corner of a dummy laptop case that the user holds with both hands on his/her lap.

The recording setup is documented with photos in the directory PICS.

During the recording the user does not have to use the keyboard or the mouse. The acoustical environment of both rooms is quiet office environment. There is only one computer (Mac desktop mounted in front of the speaker); no other noise sources. The signal of the microphones is amplified by a Beyerdynamik MV 100 amplifier: headset mic + 20 dB, webpad mic + 20 dB and then connected to the standard Mic input of the recording Mac.
Each session starts with a short instruction of the speaker, then the microphones are mounted by the supervisor and a short training session (not recorded) of 5 prompts is performed. Then the supervisor leaves the room for the rset of the session. The prompting and recording runs automatically; for each prompt a fixed time slot of 5.7 sec was recorded. The timing is controlled by a 'red light' control: a red light indicates not to speak, the yellow light indicates to get ready and then together with the green light the prompt is displayed and the speaker reads from the sreen. After the fixed recording time the red light comes again and the cycle starts anew.

```
Recording specs:

Minimum speakers per language               20
Minimum speakers per sex                    20
Recording sessions per speaker              2
Prompts per session:                        130 (000-129)
Length per prompt:                          5.7 sec
Sampling rate:                              22050 Hz
Bits per sample:                            16
File format:                                WAV stereo
Head set:               Beyerdynamik NEM 192, left channel
Webpad mic:             Beyerdynamik MCE 10,  right channel
Amplifier:              Beyerdynamik MV 100, set to +20dB, LF Cut off


----------------------- Naming conventions ----------------------


Session names are coded as follows:

SES####    where #### denotes the session number


Session numbers starting with ’4’ : british speaker, room P
Session numbers starting with ’5’ : british speaker, room S
Session numbers starting with ’6’ : french speaker, room P
Session numbers starting with ’7’ : french speaker, room S

e.g. SES6013 is the 13th recording session of a french speaker in
room P.

A mapping from speaker IDs to sessions, as well as the speaker profile
can be found in the file TABLE/SESSION.TBL

Each recording file is named as follows:

Q1####%%%.WAV     where: #### denotes the session number
                         %%%  denotes the prompt number (000-129)


e.g. Q16013051.WAV contains the two microphone signals in a WAV stereo
file of the 52nd prompt of the 13th recording session of french speakers
in room P. The channel assignment for the microphones is stored in the
file TABLE/SESSION.TBL



------------------------- Signal file formats ----------------------

All recording files are stored in WAV standard format.
See specs aboce for details.
```

Transcription and error markers

All recordings were annotated according to SpeechDat conventions.
See the document doc/TRANSCRP.PDF for details about this.

The transcription files (SAM label format) are stored
on a separate CD-ROM in a file system hierarchy that mirrors
that of the signal files, i.e.\ BLOCKxx/SESxxxx.

The same information is also stored in a semicolon delimited text file
SAMEXPORT.TXT.

The SAM label names are the following (this is also the field
order of SAMEXPORT.TXt):

LHD	SAM Header specification
DBN	database name
SES	session number
CMT	comment
SRC	name of signal source file
DIR	directory path of signal file
CCD	corpus code of signal file
BEG	begin recording
END	end recording (in samples)
REP	recording place
RED	recording date
RET	recording time
CMT	comment
SAM	sample rate
SNB	sample number of bytes
SFB	byte order
QNT	quantization
NCH	number of channels
CMT	comment
SCD	speaker code
SEX	speaker gender
AGE	speaker age
ACC	speaker accent
CMT	comment
MIP	microphone position
MIT	microphone type
ENV	environment
CMT	comment
LBD	label file body
LBR	prompt text
LBO	transcription of utterance
ELF	end of label file

e.g.

```
LHD: SAM 6.0
DBN: Siemens WebCommand Database
SES: 6005
CMT: *** Recording data ***
SRC: Q16005004.WAV
DIR: BLOCK60/SES6005
CCD: 004
BEG: 0
END: 126064
REP: University of Munich, Phonetics Institute
RED: 04.07.2002
RET: 13:54:42
CMT: *** Signal data ***
SAM: 22054
SNB: 2
SBF: lo_hi
QNT: PCM
NCH: 2
CMT: *** Speaker data ***
SCD: 1005
SEX: F
AGE: 23
ACC: FR
CMT: *** Environment data ***
MIP: HEADSET=RIGHT, WEBPAD=LEFT
MIT: HEADSET=BEYERDYNAMIC_NEM_192,WEBPAD=BEYERDYNAMIC_MCE_10
ENV: P-ROOM
CMT: *** Label file body ***
LBD:
LBR: 0,126064,,,,appeler Nicolas Moulin
LBO: 0,63032,126064,appeler Nicolas Moulin
ELF:

-------------------------------------------------------------------------

Known errors


Remark: The subdirectories NOT_USED_* contain sessions that are incomplete,
either because speakers were not recorded a second time, or because signal
files were corrupted.

-------------------------------------------------------------------------

History

01.06.02 : start of recording
20.07.02 : start of validation
01.08.02 : end of recording
08.08.02 : end of validation
09.08.02 : delivery date 1.0
19.08.02 : delivery date 1.1 (update of DOCCDROM only)
```

Appendix E

WebCommand – Validation Report

Summary

The speech corpus WebCommand has been validated against the specified checks as given in the validation contract (see annex) as well as against general principles of good practice. The validation covered completeness, formal checks and manual checks of selected subsamples. The overall quality of the corpus is good and there should be no problem in using the corpus for the intended and other applications. Some flaws in the corpus documentation may be corrected without much effort.

Introduction

This document summarizes the results of an inhouse validation of the speech corpus *WebCommand*[1]. WebCommand was produced by the Bavarian Archive for Speech Signals (BAS) in the year 2002 as a contractor to Siemens AG, Munich. The aim of the corpus was to record application-specific commands in British English and French by native speakers in a quiet office environment. The aimed application is the control of a so called WebPad (a laptop without keyboard) used for surfing the internet and some other proprietary services. The spoken texts were prompted on screen and recorded with two different microphones and in two different rooms. The data were transcribed using SpeechDat conventions. Also a canonical pronunciation

[1] For the original corpus specification and documentation of WebCommand see appendices C and D.

dictionary with all spoken words was included in the corpus.

Validation Results

The following list contains all validation steps as specified in the validation contract[2] together with the methodology and the results.

- Completeness, file naming, readability
  - Signal files
    The corpus is divided into complete and incomplete recording sessions. The complete part contains more than the required number of recording sessions and meets the minimum numbers per language (20) and per gender (20). Each session dir of the complete part contains exactly 130 recording files (WAV stereo) as stated in the specs. The file names of the signals files meet the documented specs.
  - Meta data files
    The recording conditions are summarized for all recording sessions in the file SESSION.TBL; a SpeechDat-compatible version of this table is stored in the file SUMMARY.TXT, which also contains markers for each individual recording item. Both files contain consistent data and the data is compatible with found sessions dirs. Speaker profiles are stored in the file SPEAKER.TBL which covers all speakers of the corpus.
  - Annotation files
    All annotation files are stored on a separate CDROM and in SpeechDat-compatible SAM format. Every signal file has a corresponding SAM label file. The file naming is consistent with the file naming of signal files.

  All checked files were readable.
  Status completeness: ok.

- Superfluous files
  No superfluous files were found in the corpus.
  Status superfluous files: ok

- Signal files
  All signal files were checked for their format using the command 'sox -V' and then parsing the output produced by sox. All signal files are valid WAV sound files (RIFF) with the following properties in

[2] see 3.4.

accordance with the documentation as well to the specification: 2 channels, 22050 Hz sampling rate, 16 bit width, signed (linear). All signal files contain a signal of more than 5 sec length. About 4% of the sound files contain saturated samples (clippings); some of these were inspected manually to ensure that the clipping were caused by noise, clicks etc. but not by the speech signal itself. In the inspected files this was never the case. Sox did not report any technically corrupt files.
Status signal files: ok

- Speaker distribution
  10 female and 10 male speakers as stated in the meta data were selected randomly and their speech signal checked for their respective gender. No deviations from the documented gender were found.
  Status speaker distribution: ok

- Documentation, completeness, consistency with corpus
  Apart from the file TRANSCRP_EN.PDF which describes the SpeechDat annotation the documentation of the corpus consists of plain text files only. All documentation (and meta data) files are readable on Macintosh, Linux and Windows. The main documentation is contained in the file REPORT.txt. The following checks have been performed:

  - Contact for requests regarding the corpus: ok
  - Number and type of media: ok
  - Content of each medium: acceptable
    "The corpus contains 47 complete sessions..." - The corpus contains 95 complete sessions. What is meant here is probably: "The corpus contains 47 double sessions recorded in the two recording rooms."
  - Copyright statement and intellectual property rights (IPR): ok
  - Layout of media: file system type and directory structure: ok
  - File nomenclature: explanation of codes used: ok
    "The channel assignment for the microphones is stored in the file TABLE/SESSION.TBL." – A constant channel assignment would be preferable; also it is generally better to separate different signals in individual files and mark them in the file name.
  - Formats of signal and annotation files: ok
  - Coding: PCM linear ok

- Compression: n.a.
- Sampling rate: 22050Hz ok
- Valid bits per sample: 16 ok
- Used bytes per sample 2 ok
- Multiplexed signals: standard RIFF ok
- Clearly stated purpose of the recordings: ok
- Speech type(s): read from screen ok
- Instruction to speakers: acceptable
  A full copy of the instructions is not provided (verbal instruction), but the recording situation makes quite clear how the speakers were instructed.
- Specification of the individual text items: ok
- Specification for the prompt sheet design: n.a.
- Example prompt sheet: n.a.
- Speaker recruitment strategies: not given
- Number of speakers: ok
- Distribution of speakers over sex, age, dialect regions: acceptable
  Only age, mother tongue and gender is given in the speaker profile. Due to the nature of the corpus and the fact that the specifications do not require any additional information, this is acceptable.
- Description/definition of dialect regions: not given
- Recording platform: Macintosh ok
- Position and type of microphone(s): ok
- Company name and type id: ok
- Electret, dynamic, condenser: not given
  Has to be derived from technical sheets of microphones, which are not provided in the documentation.
- Directional properties: see before
- Mounting: ok
- Position of speaker(s) (distance to microphone): ok
- Bandwidth: half of sampling rate ok
- Number of channels and channel separation: ok
- Acoustical environment: ok

  - Unambiguous spelling standard used in annotations: not given Since the prompt texts were provided by the client, the spelling is probably taken as is.
  - Labeling symbols: ok
  - List of non-standard spellings (dialectal variation, names etc.): not given
  - Distinction of homographs which are no homophones: not given
  - Character set used in annotations: plain text ISO 8859-1 ok
  - Annotation manual, guidelines, instructions: ok
  - Description of quality assurance procedures: not given
  - Selection of annotators: not given
  - Training of annotators: not given
  - Annotation tools used: WWWTranscribe ok
  - Lexicon format: ok
  - Lexicon text-to-phoneme procedure: not given
  - Lexicon explanation or reference to the phoneme set: SAM-PA ok
  - Lexicon phonological or higher order phenomena accounted for in the phonemic transcriptions: n.a.
  - Statistical Information: not given
  - Indication of how many files were double-checked by the producer together with percentage of detected errors: not given

  All documentation files are readable on WinX, Linux and Macintosh. Status documentation: acceptable

- Annotation files (transcripts)

  - All annotation files have been check for proper SAM syntax: ok
  - 10% percent randomly selected annotation files were inspected manually against the signal using WWWTranscribe. Less than 1% text errors have been found and less than 2% of noise marker errors (listing in annex A).

  Status annotation: ok

- Lexicon

  - Formal check
    The two SAM-PA lexica have been checked for their format, used SAM-PA symbols and coverage of transcripts. No missing items or errors were found.
  - Content
    15% of randomly selected lexical entries were checked manually against SAM-PA rules. Less than 2% percent phoneme deviation found.

  Status lexicon: ok

- Readability on different platforms
  The two DVDs and the CD containing the documentation were successfully mounted on Macintosh, Linux and WinX.
  Status Readability on different platforms: ok

Validation Tools

Sox was used to check the format of the signal files as well as for clippings.

WWWTranscribe[3] was used to manually check the transcripts and the lexicon.

Other Observations

None.

Comments

The documentation lacks some details, which should be provided by the producer:

- how speakers have been recruited
- which reference was taken for the English and French spelling
- according to which method the pronunciations in the lexica were created
- the selection and training of the transcribers
- quality assurance procedures
- type of microphones
- description of speaker instruction

Result

The corpus WebCommand is in a usable status.

[3] Contact Dr. Chr. Draxler, draxler@bas.uni-muenchen.de, for more information regarding WWWTranscribe.