SPEECH CORPUS CREATION CRITERIA

Dr. Ivan Kraljevski

Fraunhofer Institute for Ceramic Technologies and Systems IKTS

Table of contents

1 Introduction

Description: Formulate criteria (recommendations, guidelines) for speech recordings.

- Formulation of criteria for appropriate speech recordings and monitoring of the development of the speech corpus
- Automatic analysis and revision of phonemes/diphones/triphones and generation of further texts for speech recordings

Impact: professional voice recordings in studio quality can be planned and performed to train the speech recognizer in Upper Sorbian.

Prerequisites:

- Text from the target domain, e.g., travel planning, smart home.
- Access to the recording studio to view the technical conditions on-site.

The general procedure in creating a speech corpus consists of the following stages, which are described in more detail in the appropriate sections:

1. Acquiring textual content that matches the intended domain of application.
2. Preparing the data and metadata to stage speech recording sessions.
    a. Text corpus (with or without word classes)
    b. Vocabulary and lexicon creation
    c. Balanced recording prompts
    d. Test or pilot recordings (pre-validation)
3. Speaker recruitment
4. Staging recording sessions (release validations)
5. Finalizing the corpus production (final validation)
6. Data security and storage
7. Usage and distribution (legal aspects)

2 Text Corpus Production

The common problem in developing specific Automatic Speech Recognition (ASR) solutions is the lack of appropriate domain-specific data, effectively rendering reliable statistical language modeling impossible.

To accomplish the task for a given domain, it is necessary to collect information about the frequently used terms and their context in the sentences. This kind of information can be found in text material of the target domain, and it is used to build the speech recognition system's dictionary and the language model.

The system's performance is highly dependent on the quantity and quality of the text material.

2.1 Text Corpus Collection

2.1.1 Domain Definition

The first step for creating a practical speech application is to define the target domain and the application use-cases. A domain represents a topic or set of topics or a situation in which verbal communication takes place.
The complexity of the domain and the vocabulary size influence the design of the lexicon and the language model. The baseline acoustic model could be, when necessary, adapted to the user in the case of the speaker-dependent speech application.

When the domain is defined, the next step is to collect relevant textual data (corpus). The corpus must match the target domain and adequately represent the expected use cases. In some applications, there is scarce and sometimes no textual data available at all to be used for reliable modeling.

2.1.2 Use-cases Definition and Analysis

If there is not enough data for the target domain, the available textual content does not reflect the actual human conversation. Hence, the domain must be described with representative examples.
To find out how people would communicate with a machine using speech, it is necessary to involve as many users as possible in the early stage of development.
At first, an online survey can be conducted with potential users in a preparatory phase, clarify general expectations, and obtain suggestions for the speech application design. That will give the general directions for further textual data collection or generation.
Subsequently, conversation examples must be designed and implemented for usage scenarios that elicit spontaneous utterances from users.

2.1.3 Domain-specific and General-purpose Corpus

The following approaches could be employed separately or combined:

1. **Web scraping.** Use of web crawler tools to collect web pages and recognize different types of content within given criteria (language, domain-specific keywords) and store only the type of content specified by the user, e.g., article titles or authors from a news website, or prices and product descriptions from a commercial website.
    - Pros: fast and automatic collection of large amounts of data and creating a general-purpose corpus suitable for statistical modeling.
    - Cons: hard to restrict to only one domain, noisy content that must be post-processed and normalized.

2. **Prepare use-case examples.** As many as possible human participants provide written examples for a specific domain use. For instance, they could be asked to give several examples in the case where they are requesting an action from an intelligent home appliance, like "set the blinds in the kitchen to half" or "make the light in the kitchen darker" and so on.
    - Pros: controlled and exact specification of sentences.
    - Cons: involving many participants, introducing human bias, considerable effort to provide the required amount of text, not suited for statistical language modeling.

3. **Textual Data Augmentation.** In many domain-specific applications, the texts are not available in their original form. To overcome this problem, textual data augmentation should be employed, where the adequate quantity of text used to train language models is significantly increased.

   One way is to create artificial data based on given examples, while another assumes that additional out-of-domain texts are available, which can be used for language modeling.

   - In the first approach, the textual data can be produced by human experts in the way they believe it could be spoken as input to the domain-specific system. Subsentence phrases and placeholders define the example sentences. The placeholder represents various word classes that can be general concepts (such as time, date, numbers, ZIP codes) or custom-defined (rooms in the house). The word class is usually specified either by grammar or word lists. The grammar restricts the possible word sequences providing validated output (e.g., the sentence "thirty-fifth of May," even if it is correctly recognized, will be not considered as word class "date"). The word classes in the from list contain words belonging to a specific concept (e.g., proper names, cities, states, months, days) where each word is equally possible, or their occurrence could be weighted.
     The problem is that the developers could be easily biased by providing non-diverse sentences by repeating similar phrase structures. Automated procedures for sentence generation from context-free-rule-based templates also could be used for data augmentation. The rules are expanded into alternative phrase instantiations to complete a sentence, and the templates could be designed to cover appropriate statistical frequencies.
     - Pros: Templates definition easier than manually writing examples.
     - Cons: Artificial generated sentences, the syntax might be wrong.

   - The second approach is practical only if word-class language modeling is employed. Namely, many n-grams are relatively domain-independent, and the statistical relevance of the domain-specific data could be improved if they are included in the training.
     For example, the n-gram contexts where numerical data appears, like currency, dates, days, months, years, etc., could be included in the domain-specific texts.
     - Pros: Easier definition than manually writing many examples.
     - Cons: The words in a class can be found in a different context introducing ambiguities (e.g., number as a currency and as year).

   The word classes must be precisely defined in both approaches, and the texts should be tagged with the appropriate class labels. Several well-known algorithms exist for this task, such as the contextual semantic tagger.
   Different context-free grammar rules were designed to generate many "artificial" sentences to perform automated data augmentation.
   Such grammars are powerful enough to describe most of the structure in spoken language and restrictive enough to have efficient parsers. Nevertheless, they are inappropriate to be used for robust ASR since the grammar is almost always incomplete.

4. **"Wizard-of-Oz (WOz)".** The Wizard of Oz experiment is a method that allows a speaker to interact with a speech interface without knowing that a

human rather than a computer is generating the responses. It is applied to elicit natural and spontaneous speech interaction.

WOz experiments could be staged to simulate a user experience, perform tests with user target groups, and test concepts and hypotheses even before a functioning system exist. The benefits for user interface development are twofold: on the one hand, text corpora can be obtained through parallel data recording. On the other hand, user experiments allow conclusions about user behavior, expectations, and user experience design for fault-tolerant, robust behavior.
The verbal interaction is simulated in carefully prepared scenarios where spontaneous speech was elicited from the participants trying to solve a given task. The speakers are free to formulate the conversation with a mockup speech recognizer without explicit prompting, and their speech is recorded. The recordings must be transcribed orthographically, marked for noise, intelligibility, hesitations or pauses, and dialectal pronunciation variants.

- Pros: a collection of relevant in-domain speech and texts.
- Cons: manual speech transcription, a small amount of the collected in-domain data.

2.1.4 Recommendation

For optimal results, one or more of the approaches should be combined. Depending on the application, it is possible to perform a weighted combination of a larger general-purpose corpus with much smaller in-domain text data with specified word classes. Such an approach will yield a model that includes the flexibility of the general language model, where an arbitrary sequence of words could be recognized, with the strict definition of word classes that limit the outcomes to the valid syntax of the class.

2.2 Text Corpus Postprocessing

In most cases, depending on the source, the collected textual data is not in the desired format and needs to be carefully cleaned and normalized.
The first step is to transform the data into a machine-readable format when the texts are in from page scans or images. Then the data must be converted into the same character encoding according to the adopted convention (for instance, UTF-8).
In general, besides the graphemes, the text might also contain digits in various formats, such as numbers, date, time, duration, amounts, and special characters in physical measures and equations. Often many abbreviations should be de-abbreviated into complete phrases.
When derived from scraped website pages, the texts contain boilerplate content, words of different origin written with other grapheme sets than the intended language.
In the case of e-books, the text extraction could lead to wrong line breaks and characters formatting.
Even when the text data is an already processed corpus, the internal format and the metadata may significantly differ among language resource providers (commercial or research organizations).

2.2.1 Text Conversion

The sources of textual data are often in a form that cannot be processed directly, for instance, printed books and documents. Such sources should be converted to machine-readable format employing the following approaches:

1. Manual transcriptions of sources that cannot be automatically transcribed.

2. Conversion of electronic documents (PDF, Word, RTF) into the desired format.
3. Optical-Character-Recognition (OCR) of manuscript scans.

2.2.2 Text Encoding

To perform any processing of already converted textual data, it is necessary to identify their encoding, Byte-Order-Marks (BOM), the End-of-Line (EOL) convention, and convert all the data in the same format.
The text could be encoded in some of the general standards:

- Unicode, UTF-8, UTF-16, UTF-32
- ISO-*, ISO/IEC 8859-1 and ISO/IEC 8859-15
    - Upper Sorbian ISO 8859-2 (Latin-2)
- ASCII (7/8 bits)

Byte-Order-Mark can be either:

- Little Endian (LE), or
- Big Endian (BE).

End-of-line conventions according to the operating system:

- Windows (CR LF)
- Linux (LF)
- Macintosh (CR)

Software tools for character encoding conversion.

- Notepad++ (for smaller textual files)
- iconv (Linux, MSYS)
    - `iconv -f ISO-8859-1 -t UTF-8 in.txt > out.txt`
- Windows Powershell
    - `gc -en string in.txt | Out-File -en utf8 out.txt`

2.2.3 Text Normalization

Normalization is the transformation of the text into a single canonical form by existing or newly defined conventions. Normalizing text guarantees that the textual corpus will be consistent before further processing or analysis. There is no general approach, and the normalization procedure is very defined on the type of the text and the intended application or analysis procedure.
Text normalization is a part of Natural Language Processing (NLP) and generation (NLG) used in speech to text and text to speech conversions. Numbers, dates, acronyms, and abbreviations are non-standard "words" that need to be considered differently depending on the given context.

For instance:

- "€200" would be pronounced as "zweihundert Euro" in German.
- "VI" could be pronounced as "fau- i", or "sechste" depending on word context

In context-independent normalization, for instance, removing non-alphanumeric characters or diacritical marks, regular expressions would suffice.

For example, the sed script:

```
sed -e „s/\s+/ /g“ inputfile
```

would convert sequences of one or more whitespaces into a single space.
Domain knowledge of the language and a pre-existing vocabulary are necessary for more complex text normalization.

2.3 Text Corpus Analysis

The objective of the textual corpus analysis is to transform and derive helpful information of the normalized corpus that can be directly used to develop the language model and the speech application.

For instance, vocabularies serve as a basis for creating lexicons. The vocabulary size can be limited to the top N to reduce the size and the complexity of the lexicon and the language model. N-gram counts are required for language models creation, word frequencies to restrict the size of the vocabulary, entities as elements of word classes.

Some of text analysis and data mining methods that are useful in building speech applications:

- Word frequency represents a list of words and their frequencies, and it could define the top-N most frequent words in a vocabulary.
- Word collocation, which words commonly appear near each other, not necessarily preceding or the following words.
- Concordances are instances and contexts of a given word or set of words, for identifying and analyzing patterns in the corpus.
- N-grams counts of common N-words phrases, unigrams, bigrams, trigrams, etc.
- Parts of speech tagging (POS), detecting and classifying words as nouns, verbs, adjectives, adverbs, and based on its definition and context.
- Named entity recognition (NER) presents the identification of names, places, time, and periods, etc.

2.3.1 Vocabulary Definition

Vocabulary is derived from a normalized text, or it is defined before from existing linguistic resources. If the text is processed that all words have the same capitalization rules (upper case or lower case), each word with the same canonical pronunciation will appear only once in the vocabulary.
Vocabulary can be created by using the stream editor (SED). The example is given for Linux (MSYS), the corpus is streamed to the SED to tokenize into words converted into upper-case, and a unique sorted set is provided.

```
cat train.corpus | sed -e 's/\s*/\n/g' | sed -E
's/^(.*)$/\L\1/' | sort -u > train.vocab
```

The vocabulary is used as an input for lexicon creation by applying grapheme-to-phoneme transformation either by statistical models (Finite-State-Transducers) or by regex rules.

2.3.2 Word Classes

An important aspect of text corpus processing is identifying tokens (words, subwords, abbreviations, numerical expressions) and assigning them to a word class. That allows a significant reduction in the complexity and vocabulary size and improves the generalization of the language models, in both statistical- and context-free-grammars (CFG)-based models.

For instance, a textual corpus that contains all the numbers in the corresponding context is required; it will have an infinite size and will be impossible to acquire. In the case of CFG grammars, there should be definitions for each number, which renders such grammar impossible to create.
Instead, for statistical models, each number is replaced with a word-class label, and in the grammars with special rules that build the numbers using word sub-units.

2.3.2.1 Definition

Word classes are a narrower definition of Part-of-Speech (POS) and, in this case, describe words with the same semantical meaning. For instance, number, date, or time. Word classes can be open or closed. Open word classes do not have limited members or strictly defined rules, such as proper names (Name, Surname), names of products, etc. Contrary, closed classes are strictly defined, and no arbitrary members can be added, for example, names of weekdays, months which are finite lists, or dates and time which follow a strict format.
A more advanced approach uses morphologically inspired word classes (sub-word units), significantly improving language modeling in highly inflected languages (Upper and Lower Sorbian).
The designers of speech applications should be able to define their word classes as unordered sets according to the domain specification. Additionally, to use pre-defined word classes with the possibility to extend their content.
Employing word classes in language modeling also partially solves Out-of-Vocabulary words (OOVs), they can be included in the existing vocabulary and the corresponding class without the need to re-create the language model.

The convention which word classes to be used in a speech application must be agreed upon before the corpus processing. The textual corpus must be tagged with the adopted class labels. In the most straightforward approach, the detected words are replaced with class tags.
For instance, different dates formats, like 27.08.2020, 2020/08/27, or 27-08-20, are replaced with <DATE>, time 13:30, or 01:30 PM with <TIME>.
It is essential to define the classes correctly according to the context since words could have different meanings in a different context. Numbers alone could define years and the amount of a currency, postal codes, or credit card numbers.

2.3.2.2 Word Class Parsing

The most challenging part in using word classes is the correct and reliable annotation of word classes in the textual corpora. Manual annotation is the most time-consuming approach, and in some cases, impossible given the amount of data.
Automatic methods rely on the slot-filling approach or shallow semantic parsing with Named-entity recognition (NER). It is grouped into detecting words (names) and their classification into word classes (entities) such as location, person, or temporal expressions.
Linguistic grammar-based techniques can be used as well as statistical models such as machine learning. Expert-created grammar-based systems are usually better for the cost of the effort of creating the grammar. On the other hand, the statistical systems

need a large amount of manually annotated training data. Alternatively, semisupervised approaches can be employed to avoid part of the labeling effort.
Popular Natural Language Processing (NLP) tools used for semantic parsing and named-entity recognition are Natural Language Toolkit (NLTK)[1], scientific and research-oriented toolkit SpaCy[2] , which is more production-oriented, Apache OpenNLP[3] for more advanced text processing services, General Architecture for Text Engineering (GATE[4]), a Java suite of tools.

The main issue with using such tools with the existing language-dependent models it is difficult to introduce new languages, particularly with fewer resources.
In such a case, a simple approach would be to use lookup tables. Lookup tables are lists of words to help extract entities that have a known set of possible values. They should be as specific as possible. For example, to extract country names, the lookup table should contain all countries in the world:

Afghanistan, Albania, …, Zambia, Zimbabwe.

Another approach is to use a custom rule-based matching parser with regular expressions (regex) such as Stanford RegexNER[5]. Here the named entities should have well defined structure, such as time and date, here are some simple examples:

```
<DATE>: 27.08.2020,2020/08/27  /(0[1-9]|[12]\d|3[01]).(0[1-9]|1[0-2]).([12]\d{3})/
<TIME>: 13:30, or 01:30        /^(0?[1-9]|1[0-2]):[0-5][0-9]$/
```

More complex rules are necessary to cover all the correct time and date expressions.
The main advantage of both approaches is that they a language-independent and highly customizable. However, their creation requires expert knowledge (in writing regex rules), restricted to simple applications.
Current State-of-the-Art solutions use deep-learning approaches with large pre-trained transformer models (BERT, Turing-NLG, GPT-3) that can be tuned for a specific use-case.

2.3.3 Context-Free-Grammars

Design of Context-Free-Grammars (CFG) that could be used as standalone or combined with statistical language models is also dependent on the textual corpus analysis and the speech application definition. Any language expression that could be defined in the form of grammar should be replaced with a CFG and appropriate label. In this sense, CFG defines a word-class or a named entity. Many existing speech-enabled products are using grammar-based technology to define word classes.

A CFG describes a specific and restricted domain type of language (e.g., command and control). Grammars allow specifying possible inputs precisely, where, for example, a particular word(s) might be repeated only two or three times. The problem is that the speaker could accidentally skip the words which the grammar requires, and the whole recognition will not provide any results. It is recommended to make grammars more flexible, where instead of strictly defined phrases, a Bag of Words (BOW) allowing arbitrary order is used for the cost of the recognition performance. Grammars usually do

[1] https://www.nltk.org/

[2] https://spacy.io/

[3] https://opennlp.apache.org/

[4] https://gate.ac.uk/

[5] https://nlp.stanford.edu/software/regexner.html

not have probabilities for word sequences like in the statistical language models, but some elements might be weighed.

One method for creating speech-recognition grammars is to generate them automatically from a large corpus of input data. Another one is to design the grammar from a specification of the application manually and then test and modify the grammar by experimenting with end-user speech input.

2.3.3.1 Formal Definition

Human languages are close to CFGs, at least syntactically. CFG can have concepts defined in terms of other grammars (possibly themselves, recursively).
They are context-free because the definition of a concept is the same, regardless of the context in which it occurs, i.e., independent of where it is embedded in another grammar. In textual form, CFGs are formally defined as:

- A finite set of terminal symbols (i.e., words in the vocabulary)
- A finite set of non-terminal symbols (the concepts, such as <date>, <person>, <move-command>, <command-list> etc.
- A special non-terminal symbol (usually S), the start state of the CFG
- A finite set of production rules

Each rule that defines a non-terminal is a possibly non-empty sequence of other symbols, each of which may be a terminal or a non-terminal. There may be multiple such definitions for the same non-terminal.
The language generated by a CFG is the set of all sentences of terminal symbols that can be derived by expanding its special non-terminal symbol S, using the production rules. CFGs can be made probabilistic by attaching a probability to each production rule. These probabilities are used in computing the overall likelihood of a recognition hypothesis (sequence of words) matching the input speech. Whenever a rule is used, the rule probability is applied.

2.3.3.2 Finite-State-Grammars

Finite-state grammars (FSG) are a subset of CFGs, every language accepted by an FSG is also accepted by some CFG; however, not every CFG has an equivalent FSG.

FSGs can generate an infinite set of sentences utilizing a finite number of rules operating upon a finite vocabulary. The sentences are generated according to the choices made, traversing from left (initial state) to the right (final state) through a finite number of internal states.
FSG can represent structures that are modeled by recursive rules only if the recursive node is at the left or right edge of a rule but not when it is on terminal nodes.

2.3.3.3 Writing Grammars

Grammars can be written in many formats such as Bakus-Naur form (BNF), Extended BNF (EBNF), Augmented BNF (ABNF), Java Speech Grammar Format (JSGF), or Extended markup language (XML). In the narrower context of speech recognition defined by the W3C standard "Speech Recognition Grammar Specification (SRGS)" that uses ABNF and XML formats. The ABNF and XML form have the expressive power of a CFG and grammar processor that does not support recursive grammars of a finite state machine or FSG.

Example of a small grammar written in JSGF:

```
#JSGF V1.0;
  grammar hello;
  public <greet> = (good morning | hello) (
  ivan | frank | daniel | jan );
```

2.3.3.4 Grammars in UASR

A grammar file is an ASCII file defining a lexicon and a finite-state language model. Grammar files are organized by lines. The line format is:

```
<PREFIX>: information
```

Blank lines and lines starting with the hash character (#) are ignored. White spaces are significant. Sequences of white spaces – except line breaks – behave like a single space. The following prefixes are supported:

Prefix	Description
LEX:	One lexicon entry
GRM:	One grammar rule

Lexicon Entries

Each lexicon entry maps one orthographic to one phonetic string (i. e., a string of HMM names). The format of a lexicon entry is:

```
LEX: <orthograpic> <phonetic>
```

Orthographic strings are arbitrary. They must, however, not contain white spaces, colons (:) or commas (,). It is recommendable to avoid special characters – except underscores (_) – altogether.
Phonetic strings are sequences of HMM names. They must not contain white spaces. Phonetic strings may contain variants in the form

```
<prefix>(<variant1>|<variant2>[|...])<suffix>
```

A variant can be empty or contain nested variants. For instance, the lexicon entries

```
LEX: sieben zi:(b(@|)n|m)
LEX: bisher bIsh(e:|E:)(r|6)
```

will expand to the pronunciation variants with multiple orthographic strings, and the entries:

```
LEX: sieben zi:b@n
LEX: sieben zi:bn
LEX: sieben zi:m
LEX: bisher bIshe:r
LEX: bisher bIshE:r
LEX: bisher bIshe:6
LEX: bisher bIshE:6
```

are equivalent to the previous examples.

Grammar Rules

A grammar rule has the format:

```
GRM: <left-symbol> <right-symbol>[ <right-symbol> [...]]
```

where <left-symbol> is a named state or a non-terminal transition and <right-symbol> is a named state, a terminal, or a non-terminal transition. Symbol strings must not contain white spaces, colons (:) or commas (,). Though it is recommendable to avoid any special characters, parenthesis (()), brackets ([]), and periods (.) are explicitly allowed.

Here are few examples for grammar rules definitions:

```
## discrete in percent
GRM: <BRIGHTNESS> <LAMP> DAJ:DAJ <PERCENT> PROCENTOW:PROCENTOW
GRM: <BRIGHTNESS> <LAMP> <PERCENT> PROCENTOW:PROCENTOW
GRM: <BRIGHTNESS> <LAMP> <PERCENT> PROCENTOW:PROCENTOW PROŠU:PROŠU
```

Extensive descriptions of the formats are given in the UASR documentation Section tools/REC_PACKDATA.xtp - Writing Grammars (https://rawgit.com/matthias-wolff/UASR/master/manual/index.html).

2.4 Recordings Prompt Selection

2.4.1 Phonological Distribution

Dependent on the task for which the speech corpus is specified (speech recognition or speech synthesis), phonetic units' analysis can be employed to set criteria for the corpus content (not only by the vocabulary).
The content could be determined by phonological units, like phonemes, syllables, morphemes.
In the case of general-purpose speech recognition, it is required that each speaker repeats every possible phoneme in various contexts (triphones). In the case of concatenative speech synthesis systems, that is, every diphone combination in the spoken language.

2.4.2 Selection of Phonetically Rich Prompts

The aim is to select a minimal set of sentences with a maximal possible coverage of phonemic units (phonemes/diphones/triphones).
The prerequisite is the existence of the grapheme and phonemic inventories, pronunciation model, and normalized text corpus or corpora that will serve as the pool from where the recording prompts will be drawn.

2.4.3 Toolkit

The tool is a python-based script, loosely based on the existing tool, with expanded functionality. Besides generating project files and configurations for speech recording sessions, also prepare text resources for statistical language modeling (normalized corpus, vocabulary, lexicon, word counts).

```
Usage: python3 BASgenerator.py HSB.yaml
```

YAML configuration file contain the following parameter keys:

```
# Configuration file

input_txts:
    - "cv.hsb"
    #- "sorbian_institute_monolingual.hsb"
    #- "web_monolingual.hsb"
    #- "witaj_monolingual.hsb"
    - "smartlamp.hsb"
    #- "vocabulary.txt"
    #- "adaptation.corp"
    #- "sl.corp"

phoneme_inventory:  "phonmap_v3.txt"
exceptions_file:    "exceptions_v3.txt"
output_dir:         "corpus"
database:           "db-hsb-asr"

mode:                   "sampa"         # "uasr" or "sampa"
basic_type:             # "phones", "diphones", "triphones" or empty not generating sentences
case:       "uc"            # "uc" - upper case, "lc" - lower case, "or" - original
split:      1               # number of separate datasets
offset:     0              # id starting number
num_sentences: 1000         # maximal number of sentences in all splits
min_duration:  0        # minimal duration in seconds of speech per sentence (0-3s)
no_speakers:   1            # number of possible speakers per split
user_vocab:
        #- "user1.vocab"    # use existing vocabularies to restrict the sentences from the corpus
        #- "user2.vocab"
names:
    - "phones"
    - "diphones"
    - "triphones"
scoring_type: 1 # 1, 2 or 3; choose 2 for negative log-prob weights, 3 for sentence weights

uasr_map:
    'w': 'U v'
    'ts': 't s'
    'tS': 't S'
    'jn': 'j n'
    'dZ': 'd S'
    'dS': 'd S'
    'ng': 'n g'
    'Z': 'S'
    '1': 'Y'
    'e': 'e:'
    'i': 'i:'
    'ij': 'i: j'
    'u': 'u:'
    'o': 'o:'
    'y': 'y:'

digraphs:
    'C_H': 'CH'
    'D_Ź': 'DŹ'
```

The script produces the following folder structure and files:

```
|-corpus            - files required for language modeling
| |-hsb.corp                  - normalized text corpus
| |-hsb.freq                  - word frequencies
| |-hsb.rmvd                  - removed sentences
| |-hsb.vocab                 - vocabulary

|-uasr_configurations – files required for acoustic model training
| |-grammar
| | |-hsb.grm       - word-loop recognition grammar
| |-info
| | |-classes.txt   - observed phoneme classes (subset of phoneme inventory)
| | |-default.cfg   - configuration file template (to be accordingly adjusted)
| | |-default.ipt   - override some functions in dLabPro/UASR
| |-lexicon
| | |-hsb_sampa.lex           -lexicon file with space as phonemes delimiter
| | |-hsb_sampa.ulex          -lexicon file without phonemes delimiter

|-transliterations  - to match audio recordings (speakers, sets, prompts)
| |-HSB-1/RECS
| | |-0001
| | | |-0001HSB_1_000.trl     - target speaker id 0001, set 1, prompt id 000
| | | |-0001HSB_1_001.trl     - target speaker id 0001, set 1, prompt id 001
| | | |-0001HSB_1_002.trl
 ...
| |-HSB-2/RECS
 ...
| | |-0010
| | | |-0010HSB_2_198.lab     - target speaker id 0010, set 2, prompt id 198
| | | |-0010HSB_2_199.lab     - target speaker id 0010, set 2, prompt id 198

|-sentences         - selected sentences for each dataset
| |-sentsel_diphones_3_sampa_1.txt      - plain list of sentences for the prompts
| |-sentsel_diphones_3_sampa_2.txt      - plain list of sentences for the prompts

|-speechrecorder              - BAS speech recorder project files (per dataset)
| |-HSB-1
| | |-HSB-1.prt                         - prompt ids with the orthography
| | |-HSB-1_project.prj                 - project file
| | |-HSB-1_script.xml                  - the script with prompts
| | |-HSB-1_speakers.xml                - the speakers dummy config
```

The "speechrecorder" folder should be moved to the working directory of the BAS Speech Recorder application and opened as an existing project for the recordings.

3 Speech Corpus Production

3.1 Detailed Corpus Specification

The detailed specification of the speech corpora must include all the necessary information about the intended features, recording procedures, quality monitoring, and validation.
Because of the substantial costs and effort for collecting a speech corpus, maximizing the number of features in a corpus production is desirable. That should ensure the reusability of the data for future speech-related projects or even different kinds of research (paralinguistics, phonology, sociology).
Some essential parts of the specification are the domain, speaker profiles, speaking style, number of speakers, size of the speech content, number of samples, total duration, distribution, license, personal data protection, microphones, recording equipment, and software, documentation, and logging.

3.2 Speaker Recruitment

3.2.1 Speaker Demographics

The speaker recruitment should be conducted considering the following aspects:

- Speakers' gender, usually 50 % : 50%;
- Speakers' age, custom-defined convention, e.g., younger or older than specific age, or distributed in groups;
- Native speakers;
- Dialectal distribution;
- Social factors;
- Education;

3.2.2 Number of Speakers

The size of the speech corpus is not necessarily correlated with the number of speakers. Depending on the task, the number of speakers could be defined as:

- 1 to 5, basic research or for speech synthesis;
- 5 to 50, research in speech phenomena, evaluation and/or adaptation of existing speech recognition models or systems;
- 50-100, development and training of conventional speech recognition or speaker verification systems.
- x 1000, training with deep learning of end-to-end speech recognition systems.

3.3 Speaking Style

The speech corpus can be recorded from native speakers or, in exceptional cases, non-native speakers with a foreign accent[1].

1. Read speech, elicited from the speakers by displaying prompts (with sheets or on display): books (audiobooks, LibriSpeech[2]), newspapers, formal documents, encyclopedic (The Spoken Wikipedia Corpora[3]), dictation.
2. Answering speech, the prompts are questions that the speaker should answer, with restricted or unrestricted vocabulary.
3. Command and Control Speech, where the speakers are using only an apriori known set of commands.
4. Descriptive, the speaker's answer is elicited by asking to describe a particular object, usually by showing a picture of the object.
5. Non-prompted speech is not entirely spontaneous and is restricted by some rules (like in WoZ experiments using defined protocols or pictograms to perform a task).
6. Spontaneous or conversational speech is determined by a topic or a task, such as dialogs between two or more speakers, narratives, giving a talk (TED-LIUM[4]), commenting on sport, conversation in meetings.

3.4 Recording Protocol

An essential part of the meta-data in any project where the speech corpora are collected is the recording protocol.
That is a machine-readable (CSV, XML, JSON, YAML, etc.) documentation that contains all information about the critical aspect of the recording sessions.
When the speech corpus contains only recordings under the same conditions, only one recording protocol for the complete corpus is required, otherwise, per each recording session.
It is recommended that the recording protocol contain as much information as possible to allow future expansions and enhancements of the corpus or provide enough insights about existing corpora.
The minimum requirement is that the protocol should contain the following: the session identifier, the speaker identifier, the date of the recording session, the environmental and the technical recording conditions.

3.4.1 General Information

3.4.1.1 Session ID

The session ID identifies one recording within the speech corpus. It consists of characters that give broad categorical information about the recording such as, the language, gender of the speaker, type of the recording setup, domain, and a number.
The session ID is often used in file names within the corpus to denote files that belong to the same recording session.

[1] Schiel, Florian, Christoph Draxler, Angela Baumann, Tania Ellbogen, and Alexander Steffen. "The production of speech corpora." (2012).

[2] https://www.openslr.org/12

[3] https://nats.gitlab.io/swc/

[4] https://www.openslr.org/7/

Usually, the recording software automatically creates the session IDs, and manual creation of IDs should be evaded to avoid duplicate or malformed IDs.

3.4.1.2 Speaker ID

A critical aspect of any recording project is to ensure the speakers' anonymity. The real name of the speaker is mapped to an identifier (ID) by using specified conventions (for instance, 3 or 4 alphanumeric characters). The ID will also be used in the speaker profile, signal file headers, and comments. The convention must ensure that the speaker's name will never be disclosed by knowing the ID, and the mappings should never be part of the corpus. If the data protection legislation in the country allows, it is recommended to store the mapping in a safe place. That allows identification of the speaker in the copyrights revocation or recruitment to avoid double speakers for future extensions of the corpus.

3.4.1.3 Date of recording

The recording protocol must contain the exact date and time of recording, which the operator can do automatically or manually.

3.4.2 Environmental conditions

Environmental conditions are described with the following parameters: room acoustics, the sources of noise, and the presence of cross talk of other speakers. The noise or cross talk is the most important and has to be described in a recording protocol.

3.4.2.1 Room acoustics

The room acoustics description should give information about the location of the recording sessions. Indoor (studio or studio-like conditions) or outdoor (field), or even particular room types (studio, office, living room, or even vehicles)

3.4.2.2 Sources of noise

All the sources of noise or background noise found in the recorded signals should be described. For instance, the noise produced by the host computer/sound card, spinning disks, ventilation, noise caused by cellular phones, fluorescent lights, hum (50/60 Hz), office/home/street noise, machines, etc. The protocol should contain information as to whether the background noise is natural or artificial.

3.4.2.3 Crosstalk

The information about the presence of crosstalk where other human voices are part of the background noise (babble, cocktail party) should also be included in the protocol.

3.4.3 Technical conditions

The following information is necessary to describe technical recording conditions: microphones, recording devices, sampling frequency, signal resolution (bits per sample), coding, the speaker's distance to the microphone, prompting, durations, and volume settings.

3.4.3.1 Microphone(s)

The protocol should contain the microphone manufacturer's name, type, and microphone(s), such as a directional, headset, or close-talk microphone. E.g., Ears-free headset Beyerdynamik NEM 192.

3.4.3.2 Recording device

The protocol should contain the manufacturer's name, the type, and equipment(s) used for the recording. E.g., Intel Core 7 workstation with on-board ACI87 sound controller chip, pre-amplifier Beyerdynamik MV 100 set to +20dB.

Specifications of recordings:

- Sampling frequency (8, 16, 22.5, 44.1 kHz)
- Sample type and width (16 or 32 bit)
- Channel number (mono, stereo)

3.4.3.3 Microphone position and distance

If the placement and distance of microphones change during the recordings or from session to session, they should be logged in the recording protocol.
It is good to identify the microphones before a recording, e.g., tapping on each microphone in a pre-defined sequence. That is especially true if the microphones can be moved or their cables are detached.

3.4.3.4 Other parameters

Depending on the needs, more specific coded recording parameters that could influence the recording process can be included or be defined:

- Recording supervisor
- Recorded domain(s)
- Instructions to the speaker(s)
- Session duration
- Prompting type (paper, face-to-face, screen, voice)
- Acoustics, reverberation, Signal-to-Noise (S/N) ratio, etc.
- Presence of supervisor
- Moderator
- Wizard-of-OZ
- Type of speech
- Comments

3.4.4 Examples or Recording Protocols

HSB Corpus

Table modified of similar from [1]	
Session ID	+ SES + 4-digit number set automatically
Speaker ID	+ 4-digit number set automatically
Date of recording	21.-25.09.2020
Environmental conditions + Room Acoustics:	Quiet office environment, near studio quality
Sources of Noise/Background Noise:	There is only one computer, the recording laptop - no other noise sources. Some interference from cabling (long XLR)
Cross Talk:	No other noise sources
Technical recording conditions	
Microphones	Shure SM58 Sennheiser
Recording device(s)	Notebook with power adapter Zoom H6 audio interface EXH-6 combo capsule (gain set on 9, no attenuation) USB-cable for Zoom H6
Technical specifications of recorded signals	Sampling rate: 44.1 KHz Bits per sample: 16 Length per prompt: 7.9523 sec
Placement and distance to the microphone(s)	Distance monitor – speaker: 0.8 - 1 m Distance speaker – microphone: 20 - 25 cm Left – Sennheiser (gain 9, no attenuation) Right – Shure (gain 9, no attenuation)
Name or ID of the recording	HSB-1, HSB-2, HSB-3
Details about the recorded domain(s)	General domain (Common Voice) and SmartLamp
Details about instruction to the speaker(s)	To read the prompts precisely as displayed
Duration of the recording session	1 hour
Prompts type (paper, face-to-face, screen, voice)	Screen
Emotional speech (yes/no)	No
Acoustics (reverberation, S/N ratio, etc.)	Pop shield(s) in between (or plastic bags)
Supervisor present	Yes
Interpreter present	Yes, the supervisor
WOZ:	N.A.
Type of speech	Read
Comments	Microphone standers x 2 Long XLR cables x 2 Long HDMI cable USB audio interface (for backup) Acer Monitor with power adapter ICY Box external HDD, cables, power adapter Extension cord with three sockets Extension cord with five sockets Headphones for audio check

3.5 Recording Software

The software that will be used for collection speech should provide automated and manual speech recording, platform independence, clear and understandable user interface, easy project and recording devices configuration, multimedia prompts, different ways of advancing in the recording session, multiple displays for the operator and the speaker, possibility to repeat specific prompts.

3.5.1 BAS SpeechRecorder

The most suitable tool for this task is the BAS SpeechRecorder[1] developed at the Bavarian Archive for Speech Signals (BAS)[2].

3.5.2 Signal File Formats

There are many standardized signal file formats. In most cases, a signal file format consists of a header containing information on the signal, sample rate, type and width, machine format, number of channels, etc.), and a container of digitized signal samples. The formats of the speech signal that are commonly used in State-of-the-Art speech recognition systems are the:

- RIFF WAVE (audio/wav, audio/x-wav, audio/.wave) starts with a binary file header followed by a sequence of data chunks.
- RAW, a container of the digitized signal without header, implicit or explicit specifications (e.g., by extensions: .dea, .al .la : ALAW, 8 bits; .deu, .ul .lu : ULAW, 8 bits)
- NIST SPHERE[3] (audio/x-nist), readable header in plain text (7 bit US ASCII) with the binary signal data.

Other formats are seldom directly used, and the audio channel is converted into RIFF WAVE:

- MP3 (audio/mpeg)
- AIFF (audio/x-aiff)
- MP4 (video/mp4)
- AVI (video/avi)
- QuickTime (video/quicktime)

3.5.3 Sampling Frequency

The sampling rate should be at least twice the maximum frequency in the digitized signal. In speech signals, the highest frequency is just below 8kHz; therefore, the most used sampling rate is a minimum 16kHz minimum. In telephone recording, the bandwidth is technically reduced to 300Hz - 3300Hz, and a sampling rate of 8kHz is used. Most audio devices process only the sampling rates 44.1kHz, 22.05 kHz, and 11.025 kHz (audio CD standard), and they should be chosen according to the given criteria (Nyquist law or Shannon theorem). It is recommended to use standard sample rates of 8kHz (telephone recordings), 16kHz, and 22.05kHz (on-site or field recordings).

[1] https://www.bas.uni-muenchen.de/Bas/software/speechrecorder

[2] http://www.bas.uni-muenchen.de/

[3] https://www.bas.uni-muenchen.de/forschung/Bas/BasFormatsdeu.html#NIST

3.5.4 Sample Type and Width

The definition of the format of a single sampling value is the sample type. The width of a sample defines the bits required to represent a value on a storage medium:

- PCM (linear), usually 16 bits, either with signed integer values (-32768 to +32767) or unsigned (0 to 65535).
- ALAW (World) or ULAW (US), 8 bits, where the bit order may be reversed. Decompressed ALAW roughly correspond to 13 bits linear PCM and decompressed ULAW to 14 bits PCM.

Formats that use more than one byte per sample, the machine format must be defined:

- Big Endian, most significant byte first (10), (Motorola)
- Small Endian, least significant byte first (01), (Intel)

3.5.5 Project Configuration

The selected prompts are included in the BAS SpeechRecorder (BAS-SR) configuration files, for each dataset split, in the corresponding project folders.
The BAS-SR folder ("speechrecorder" folder) should be copied to the workstation to be used for speech recordings.
The recorded speech is stored in the RECS subfolder under the speakers' name folders.

3.6 Speech Signal Quality

To ensure the best possible quality of the speech recordings, the equipment must be set in such a way that avoids any distortions and additional sources of noise.

3.6.1 Signal Amplitude

The amplitude of the recorded speech signal should always be in the ranges of the maximal and minimal amplitude values (gain control).

3.6.2 Clipping

If the amplifier delivers an analog signal with amplitude out of the range of the AD converter, then clipping will occur. That must be avoided in any case. However, even with the optimal setting of the audio equipment gain, the speaker can speak louder than anticipated, and the amplitude will be clipped on the minimum-maximum ranges. Signals with clipping should be excluded from the final release of the corpus.

3.6.3 Background Noise

Another extreme is when the signal gain is relatively low, and the background noise becomes dominant (low Signal-to-Noise Ratio). The noise will be amplified if the signal is post-processed by normalization, retaining the same SNR levels.

3.6.4 Room Acoustics

If the recordings are not conducted in a professional studio (or silent room), where the reflections are brought to a minimum, it is most likely that the room acoustics will play a significant role in the speech signal quality.
Most of the sound waves produced by a speaker do not reach the recording microphone directly. They oscillate through the room from the speaker's mouth with the

speed of sound, hit walls, the ceiling, and other surfaces, and are reflected by them, therefore, arriving at the microphone membrane indirectly.
Strong or prolonged reverberation can occur when a comparatively long time passes before the sound level, after the actual sound event, has subsided. That has a profound effect on the quality of the speech recordings, and the effect should be minimized as much as possible because it will strongly influence the acoustic modeling.
Dereverberation is a complex procedure usually applied in the postprocessing stage. To compensate the room acoustics, it is recommended to use microphone arrays with advanced speech processing algorithms such as:

- Voice Activity Detection
- Direction of Arrival
- Beamforming
- Noise Suppression
- De-reverberation
- Acoustic Echo Cancellation

Ideally, the same audio input device is used in the speech recognition system.
If such microphone arrays are not available, to improve the room acoustics, the sound reflections should be suppressed and diffused:

- Large bare wall surfaces should be covered, e.g., with open bookshelves and panels of fabric.
- At least one of the opposing walls should be covered with sound-absorbing material (foam).
- Carpets, thick fabric curtains, upholstered or furniture with slopes and slants help reflect and diffuse the sound waves.

The position of the microphones in the room is also essential. Wall areas or corners should be avoided since, in these zones, the increased sound pressure levels and humming often occur due to sound reflections.
The distance between the microphone and the walls should be at least one meter. In rectangular rooms, the modes can form, and the low-frequency sound waves either amplify each other or cancel each other out.

3.7 Recording Session Quality

To ensure the quality of the corpus collection and avoid any system problems and deviations from the specification of the recording process, regular quality checks should be performed, as monitoring (recording process) and validation (recorded corpus).

3.7.1 Monitoring

In a monitored speech recording, the speaker is asked to follow strict rules to produce spoken content, such as prompts for the read-speech corpora or producing speech of a specific emotional state in the case of a collection corpus of emotional speech.
A supervisor is always present during the recording session to ensure that the speaker correctly reads and pronounces the prompts. Such an approach will reduce the effort or altogether avoid later annotation. Annotating and transcribing speech usually requires 20-30 times the recording duration. Therefore, controlling and modifying the technical and phonetical characteristics during speech recordings is of high importance. The issues occurring during the recording session can be signaled to the operator and the speaker that a particular error occurred.

3.7.2 Validation

Some aspects of the quality of the recordings can be assessed only after the recording session, such as the signal-to-noise ratio of the types of the present noises (stationary or non-stationary noise, reverberations). There is pre-validation, release validations, and final validation of corpora.

- **Pre-validation.** The pre-validation should be performed after some of the speakers have been recorded, the data post-processed, and annotated. For instance, the quality must be verified after pilot or test recordings or after few speakers are recorded.
  The recording process should be paused until the results of the pre-validation are available and corrections (if any) to the processes are made.
- **Release validation.** At defined milestones, release validations are performed. To improve the consistency within the separate releases, the results of a current validation should be considered before collecting speech data for the next release. Therefore, the validation times should be considered in the overall time budget of the corpus collection project.
- **Final validation**. After the speech corpus production is declared to be completed, the final validation is performed. After the final validation (updates), the available budget and the overall schedule should allow some funding and time for corrections after the final validation (updates). It is improbable that a final validation will indicate that there are no errors at all.

Validation timings are defined by the size of the corpus, the time scale, and the intended usage. For instance, only the pre-validation and final validation are required for a smaller corpus that takes less than three months to be produced. In comparison, a larger corpus would need more frequent validation, e.g., individual releases.
Every point in the specifications of the speech corpus should be subject to validation. The project contract or the corpus specifications must describe what will be validated and regarded as an error. The following aspects of the speech corpus are validated [1]:

- Documentation: consistency, completeness, structure
- Metadata: completeness, parsability, contents (samples)
- Signal data: completeness, technical quality, acoustical quality (samples), contents (samples)
- Annotation data: completeness, parsability, contents (samples)
- Media: readability (on different computer platforms)
- Dictionary: completeness, quality (samples)

The validation must be documented and included in the final corpus documentation, and they can be qualitative and quantitative validation. A quantitative result could be that more than 5% of the recordings are clipped. Such validations (formal) are carried out automatically.
Qualitative validations require manual work and are often carried out only on a randomly selected recording from the corpus. What is treated as an error should be clearly defined along with the allowed tolerances.
For example, if the specification defined a 50/50 gender distribution of the speakers, the tolerance should be set as percentage +/- 5.

Quantitative (Formal) Validation Procedure:

- Check for N recording items per speaker.
- Check for empty signal files.
- Check for signals files with clippings; they must always be less than 5%.

- Check for S/N; it must be more than 15 dB.
- Check for correct terminology for all data files according to specs.
- Check for one annotation file per signal file.
- Check if annotation files are parsable.
- Check for complete and parsable speaker profile per speaker.
- Check for complete and parsable recording protocol per recording.
- Check for 50/50% gender distribution +/- 5%.
- Check for age distribution in two groups, 18 - 32 and 32 - 64; both groups have 50% +/- 5%.
- Check for parsability and completeness of dictionary.

Qualitative Validation Procedure:

- Check documentation for completeness and consistency.
- Check 5% randomly selected annotation files by independent manual transliteration and cross-check results; 3%-word errors (including insertions and deletions) allowed.
- Check 10% randomly selected entries from the dictionary for correct pronunciation; 2% phonemic errors (including insertions and deletions) are allowed.

There is no strictly defined procedure for validation; which parameters of the speech corpus should be validated and when depends on the circumstances.

3.7.3 Controlling

Regular periodic tests should be included in the checklists of the operator and the supervisor, ensuring that the recording setup, technical specifications, and monitoring and validations are consistent and according to the corpus specifications.
The controls can be automated to some extent, but the operator/supervisor or an external observer should do random checks. An external observer could be a person designated by the client or a partner institution.

4 Corpus Data Management

4.1 Speaker's Data

The speakers' data should be kept separately from the speech corpus metadata. The speaker ID mapping shall never be distributed with the corpus. It has to be securely archived and allowing later speaker identification for the following reasons:
In later corpus expansion to avoid double recordings from the same speaker
In data copyright revocation, it is necessary to identify the speaker's recordings that have to be removed (or deleted from the corpus). That does not apply to the corpus derivatives, such as acoustic models, statistics where the speaker's data cannot be removed without destroying them.

4.2 File Lists Generators

Usually, the corpus produced in a controlled recording session already contains the lists of recordings. However, in acoustic modeling involving machine learning (HMM/GMM, DNN/GMM, or just DNN training), the corpus should be split into at least the training and the test dataset. The rule of thumb is that the set should be disjunct and that a

recording never occurs in both sets. The same applies to the speakers. The training and the test set should never simultaneously contain recordings from the same speaker. When the evaluation is performed on the phoneme level in terms of phoneme sequences correctness and accuracy, then the test set is not necessary to be from the same domain. In the evaluation case, in terms of word-error-rate (WER) or sentence-error-rate (SER), both sets should be of the same application domain.
The file lists in most speech recognition frameworks are in the format of plain text files where each line contains a full or relative (from the working directory) path to the signal file.

Suggested file lists:

- Master – contains all the recordings in the corpus, serves as a starting point for a generation of other file-lists
- Training – usually a selection of the 70-80% of the master file-list
- Test – the other 20-30% of the master file-list where no signals and speakers appeared in the training
- Development – the test set can be further split into development datasets, used for optimizations of the training parameters; here, the same applies, that all datasets (defined by file lists) should be disjunct.
- Debug – a much smaller dataset used for debugging the configuration files and the training and testing process. It could be part of any of the mentioned datasets.

The UASR framework has a script for manipulation with file lists:
https://rawgit.com/matthias-wolff/UASR/master/manual/automatic/tools/FLST.xtp.html
That allows excluding entries from file lists, merging file lists, filtering and/or randomly splitting file lists, and sorting file lists.
Alternatively, the file list can be managed using Linux command tools, Table editors (Excell or Google Sheets), or database tools.

4.3 Corpus Data Storage

The loss of data has multifold consequences; commonly, the speaker is not available to participate soon again and introduces additional costs for the adopted financial budget.

The recorded data is stored locally at the workstation, and after each recording session, it should be copied to external storage such as a hard disk drive or network-attached storage (NAS).
The transfer time should be minimized, and the recordings transferred fast before the beginning of the next session. A backup should be done as soon as possible and whenever a significant amount of data is available.
The proper amount of storage space of a single recording session should be estimated and ensure enough space for at least two or three additional recording sessions on the recording device.

For a quick and reliable backup, external storage media such as a hard disk drive or USB storage drive are recommended; additionally, CD/DVD-Rs or high-capacity storage tapes can be used as well.

The devices should be disconnected from the equipment when not in use and preferably encrypted and protected with a password. No access for unauthorized personnel to the equipment should be possible.

5 Legal Aspects

5.1 Corpus Copyright Holder and User

Once the speech corpus has been finished and checked out the possibilities of distributing it to others, the following legal relationship to consider is that of the copyright holder(s) and the user(s) of the resource. The copyright holders of speech resources want to protect themselves against unauthorized copying. However, there is no technical way to protect a speech corpus against unauthorized copying; it would hinder data usage. Also, it is almost impossible to prosecute a user for copying speech data: in most cases, they will not even be in the same country, and the evidence of misuse will be impossible to ensure.
The usual approach is to:

- The users sign a license agreement, clearly stating the rights of usage and forbid data copying or re-distribution, or
- Include the usage conditions in the corpus documentation and say that the user automatically accepts the conditions using the speech data.

In general, users of a speech corpus are, in most cases, companies or research institutions, and it is not likely they will explicitly commit fraud by re-distributing the speech corpus.

5.2 Data Protection

Protection of personal data is another critical issue: the speech data and the metadata about the speakers.

- **Speech data** are usually not subject to particular concern about data protection once the speaker has waived his/her rights to the recordings. But it is critical in the case for a biometric database, a particular case of a speech corpus designed to develop and/or evaluate voice authentication systems. It might be that the speakers' data are abused to break into security systems based on the new technology. Although this is rather unlikely, it is recommended to take extra care to ensure that the mapping between speaker data and the speaker ID is inaccessible for everybody, including former staff of the speech corpus production project.
- **Metadata** about the speakers that are personal data like home address, telephone numbers, email, etc., are constantly protected and subject to special laws in most countries. A legal advisor should be contacted about properly storing and protecting these kinds of data.

6 References

1. Schiel, Florian, Christoph Draxler, Angela Baumann, Tania Ellbogen, and Alexander Steffen. "The production of speech corpora." (2012).
2. Kraljevski I., Fischer M., Gjoreski A., Hirschfeld, D. Development of a natural language speech dialogue system for an AR-based, adaptive mobility agent, 2018
3. Ines Wendler, Andreas Jatho, Ivan Kraljevski,Martin Wenzel, Nutzerzentrierter Entwurf von Multimodalen Bedienkonzepten, Conference: ESSV 2017, 28th Conference on Electronic Speech Signal Processing, Saarland University, Saarbrücken, March 15–17, 2017
4. Chungurski S, Kraljevski I, Mihajlov D, Arsenovski S. Concatenative speech synthesizers and speech corpus for Macedonian language. InITI 2008-30th International Conference on Information Technology Interfaces 2008 Jun 23 (pp. 669-674). IEEE.
5. Gebremedhin YB, Duckhorn F, Hoffmann R, Kraljevski I. A new approach to develop a syllable-based, continuous Amharic speech recognizer. InEurocon 2013 2013 Jul 1 (pp. 1684-1689). IEEE.
6. Kraljevski I, Strecha G, Wolff M, Jokisch O, Chungurski S, Hoffmann R., "Cross-language acoustic modeling for Macedonian speech technology applications", In International Conference on ICT Innovations 2012 Sep 12 (pp. 35-45). Springer, Berlin, Heidelberg.
7. Kraljevski I, Hirschfeld D. Language Model Adaptation for Transcription of Banking Protocols. Studientexte zur Sprachkommunikation: Elektronische Sprachsignalverarbeitung 2015. 2015:81-8.

7 Recording Sessions Checklist

1. Before recordings (daily):

1. Prepare equipment and setup.
2. Check if all documents are present.

2. Recording session (60 min):

Preparation phase (max 10 min):

1. Participant gives personal data.
2. The participant signs the consent.
3. Short introduction by the moderator about the lights system and prompts.
4. Operator changes the software setup (HSB1, HSB2, or HSB3).

Recording phase (35 - 40 min):

5. Prompt to be re-recorded only in case of mistakes (moderator).
6. The participant reads the prompts in the given order.
7. The participant can stop the recording session at any time due to fatigue.

3. After recordings (daily):

3. Backup on HDD (whenever possible during the sessions)
4. Shutting down the notebook
5. Switching off the audio equipment
6. Unplug extension cords
7. Lock the room