7. Resources for technical communication systems

Dafydd Gibbon

1. System resource requirements

1.1. Overview of topics

Technical communication systems are defined here as devices or device networks which intervene in the communication channel between speaker and addressee. Technical communication contrasts with face-to-face communication. The devices or device networks may be audio alone or audio-visual (multimodal), and standalone devices (such as computers with software for word processing, lexicon databases, dictation or satellite navigation) or complex systems such as telephone networks and chat or voice-over-internet-protocol (VoIP) communication on the internet (see also Allwood and Elisabeth Ahlsén 2012; Martin and Schultz 2012; Lücking and Pfeiffer 2012; all in this volume).

The topic of resources for technical communication systems is extensive and complex; the present article in the context of a general handbook therefore does not aim at providing detailed recipes for resource specification, compilation and use, but concentrates on generic considerations, and focuses on two specific cases: for text-based communication systems on lexicographic resources, and for speech-based communication systems on speech synthesis systems. References to specialised handbooks and other relevant literature are made at the appropriate places.

There are many general considerations in connection with resource oriented topics in the technical communication system area which are common to a wide range of system development areas. Among these are issues of reusability, interoperability over different platforms, cost-effectiveness, use-case and scenario dependency, as well as data collection paradigms such as the 'crowd-sourcing' of data on the internet from arbitrary or selected internet users, and 'cloud sourcing', the out-sourcing of resources, tools and other systems to internet-based resources, tools and systems.

A major issue is standardisation of categories and formats of resources for information exchange, either as a local set of consistent conventions, or as conformance to *de facto* standards set by influential institutions and companies (such as formats for media and word processor files), or to internationally agreed standards defined by the *International Standards Organisation* (ISO[1]) (see also Trippel 2012 in this volume). A little known standard which is relevant for technical communication systems is, for example, the language code and name standard ISO 639–3, in which information such as the following (for Eng-

lish and its names in other major languages) is recorded: “eng English English anglais inglés 英语 английский Englisch”. In many practical applications, country names have been used instead of language names, which can lead to confusion.

Many generic requirements such as these are currently in a state of rapid fluctuation, and will probably remain so. Consequently they are only referred to but not expounded in detail in the present article.

Similarly, legal issues concerning intellectual property rights, patents and trademark registration play a central role in creating and providing resources; specific issues in speech and multimodal communication concern data protection issues associated with the ease of identification of voices, and the even greater ease of identification of faces.

Ethical issues are also involved, not only in the deployment of systems, but also in the compilation of data resources, the extremes being data collected without consent of the recorded parties, and data recorded with explicit, informed and signed consent.

Issues such as these must be addressed in practice, but can only be mentioned and not handled here because of their variety, complexity and task-dependence, and also because of their often national and culture-specific character.

1.2. Systems and resources

Like any other scientific and engineering enterprise, the development of speech systems, language systems and multimodal systems (referred to here in brief as Human Language Technology systems, HLT systems) is dependent on the availability of adequate empirical, technical and human resources for their development. This three-way distinction between empirical resources (for instance texts, recordings in different media), technical resources (tools for processing empirical resources) and human resources will be maintained in the present contribution wherever necessary.

In the present context, the discussion of resources for the technical transmission components of such systems (e.g. resources for the encoding, transmission and decoding of acoustic signals, hearing aid technology, optical character recognition, font implementation) are largely excluded. The focus is on resources for components which are specifically within the linguistic and phonetic sub-domains of HLT systems.

The terms ‘system’ and ‘resource’ in the context of technical communication are illustrated informally in this section, and then treated selectively but in more detail later. Where more detail is required, specialised literature with comprehensive information is recommended. Some of these technical communication systems can be realised as standalone systems, others are embedded

in larger systems which are not *per se* communication systems. Several contributions to the present volume discuss different kinds of system and resources for these systems. System types and specific resource types are also discussed in most other contributions to the present handbook.

Technical communication systems as understood here include but are not limited to the following, not all of which can be discussed in the present context:

1. Speech input systems: speech-to-text recognition systems, dictation applications, machine and user interface control systems, including prosthetic systems such as voice command systems for motor-impaired users.
2. Speech output systems: geographical information system output, dictation readback components, prosthetic systems such as screen readers for the blind, visualisers for the deaf.
3. Speech dialogue systems: human-machine interaction in tutorial systems, information systems, scheduling and booking systems.
4. Natural language systems: information retrieval (search and parsing) components, database-to-text generation components.
5. Multimodal systems: tutorial systems with avatars, robotic systems with embodied agents, video games, assistive systems with Braille and other tactile output, map-based geographical information systems, systems with other sensors (e.g. airflow, skin resistance, gesture).

Resources for systems such as these include the following:

1. Raw data:
   a. Audio, visual and synchronised audio-visual recordings of interactions in standardised audio and visual formats.
   b. Handwriting, print, keyboard, stylus and finger touch screen input streams in a variety of formats.
2. Annotated data:
   a. Transcriptions of raw data in symbolic notations, produced either manually or automatically.
   b. Annotations of raw data, in which individual segments (tokens) in transcriptions are associated with time-stamps, i.e. temporal pointers (and/or space-stamps, with print) in the raw data.
3. Generalised data:
   a. *Lexicons* (alternative plural: 'lexica') or dictionaries, i.e. inventories of basic language unit (word, idiom) types, for each of which multiple tokens can occur in the data. Each unit type is associated with further kinds of lexical information (typically: phonemic, morphological, syntactic, semantic, pragmatic). A principled distinction between dictionary and lexicon is not made here. A useful informal distinction is often made between *semasiological lexicons* or readers' dictionaries (the most fam-

iliar kind, in which a wordform is known and the meaning is to be found) and *onomasiological lexicons* or writers' dictionaries (in which a known concept, represented by a known word, is known, and a wordform is to be found), for example a *thesaurus* (plural: thesauri) based on a concept hierarchy, or a *terminological dictionary.* Other organisational forms of lexicon are pronunciation dictionaries and multilingual dictionaries, which are not easily classified as semasiological or onomasiological in the usual senses of the terms. Lexicons are dealt with in more detail in Subsection 2.2.

b. *Grammars*, i.e. rule sets which determine the co-occurrence of segment types (not only words but also syllables, speech sounds) with each other, either in sequence (as with words in a sentence) or in parallel (as with sentences and intonation). Some grammatical information about local constraints on word sequences, parts of speech (POS), etc. is typically also encoded in the lexicon. In practical systems, straightforward finite state automata (or regular grammars) formalisms are often used (Beesley & Karttunen 2003); for some theoretical linguistic purposes such formalisms are too restricted.
c. Statistical *language models*, i.e. pairs of segments or segment sequences with probabilities or sets of probabilities, as in *Hidden Markov Models* (HMM) or *Artificial Neural Nets* (ANN). From a linguistic point of view, a language model is a special case of a grammar, either an extremely simple Finite State Automaton or Regular Grammar with probability-weighted transitions (as in statistical diphone and triphone models), or in more sophisticated probabilistic Context Free Grammar, in which nodes in hierarchical structures containing word and sentence constituents have probabilistic weights (for further detail consult Jurafsky & Martin 2000 and Carstensen et al. 2010; see also Martin and Schultz 2012 as well as Paaß 2012 in this volume).

1.3. Intellectual resources

1.3.1. Notations

Intellectual resources are notations, symbolisms, formalisms, interfaces, i.e. means of representing data, facts, figures, models and theories. Some of these resources are standardised in order to facilitate exchange of information, some are introduced *ad hoc* for specific, often temporary purposes, or in order to support competitive development of proprietary systems, and others, e.g. text document formats such as PDF (Adobe) and RTF (Microsoft Corporation) and audio formats such as WAV (Microsoft Corporation) are proprietary notations respectively which have become *de facto* standards (see also Rahtz 2012 in this volume).

Well-known open intellectual resources range from the *International Phonetic Alphabet* (IPA[2]; easily accessible on the internet) for representing speech sounds, through tagsets for parts of speech (van Halteren 1999), to notations for predicate logics, attribute logics, and to the hierarchy of formal languages and formal grammars which underlies syntax formalisms as well as parsers and generators for these. Some of these are regulated by the community, some are regulated by specific bodies, for example as HTML (by W3C[3]) and IPA (by the *International Phonetic Association*); on standard notations for speech, see Gibbon et al. (1997).

In discussion of resources, it is convenient to make a distinction between specific and generic data representation notations.

1.3.2. Specific data representation notations

By 'data' is meant observable or in general physical objects and their properties. A data representation notation provides a model for describing data in a systematic way, and may or may not be related to an explicit and coherent formal theory.

Classic cases of data representation notations are phonetic alphabets such as the IPA, which provide an exhaustive and consensually standardised vocabulary (see also Trippel 2012 in this volume) for representing speech sounds and their properties. The IPA is not based on an explicit and coherent theory of speech sound production, transmission or perception, but has been developed pragmatically since the late 19th century in terms of its empirically demonstrated usefulness. Ostensibly, the IPA is based on the physiological constraints on speech sound production, and the consonant chart comes very close to reaching this goal. However, the vowel diagramme is better explained by acoustic theory. Tone and intonation, on the other hand, are represented by icons for percepts. Terms for phonation types such as 'breathy' and 'creaky' are auditory metaphors.

In a somewhat more general sense, the HTML tree graph structured text representation language is domain-specific data because it is designed for text data, not other data types. HTML has a 'semantic interpretation' in terms of actual formats in terms of the CSS (*Cascading Style Sheet*[4]) language, which defines more specific features of the physical appearance of texts. Other data types, such as complex media objects like videos and graphics are not included in the HTML formalism, but HTML includes a pointer ('anchor') concept for linking to these. The pointer concept enables the construction of arbitrary, not necessarily tree-structured texts, i.e. hypertexts. HTML and CSS are formally defined: both express tree structures with sets of attributes attached to the tree nodes, each attribute associated with a numerical or textual value. The semantics of HTML expressions is given a relatively general definition in terms of text

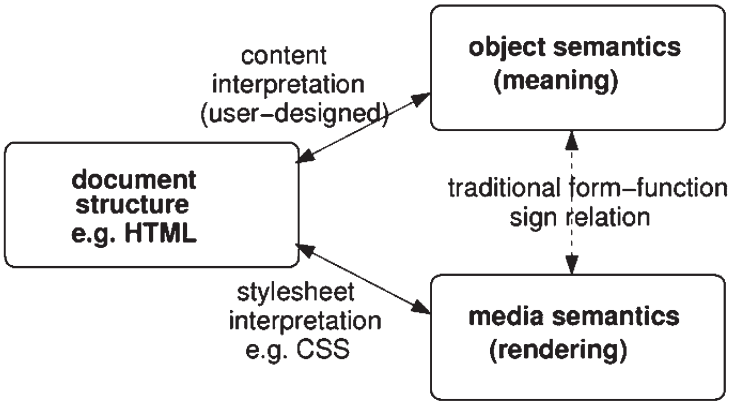

Figure 1. Text Content – Structure – Rendering (CSR) model

rendering, i.e. appearance, but not in terms of conventional object semantics, i.e. 'meaning' in the usual sense of the term, which has a different kind of semantic interpretation function outside the text and remains the concern of the user. CSS expressions provide a more detailed rendering interpretation. The relation between conventional semantic interpretation of meaning and 'semantic' interpretation of rendering properties is illustrated in Figure 1.

At the most detailed text level of the character or letter, the internationally standardised data representation notation is Unicode, which provides a uniform numerical encoding system for a wide range of standardised alphabets for languages, data description notations, and logics. The media semantics of Unicode entities are, as with HTML and CSS, given in terms of glyphs, i.e. renderings of characters as defined by specific font categories. The object semantics of Unicode entities is left to the intuition of the user, as with HTML (Figure 1), and is frequently inconsistent. For example, the IPA 'semantics' of phonetic properties is not coherently expressed: characters are not kept within a well-defined coherent code-page, but, where they are similar to other characters, e.g. Latin characters, they are defined elsewhere in the Unicode character set. There is no layer of abstraction which includes the object semantics of the characters.

1.3.3. Generic data representation notations

The term 'generic' in 'generic data representation notation' indicates that this notion of 'data' is not restricted to observable or physical objects, but may capture any kind of entity. While HTML and its parent formalism SGML, for example, were specifically related to texts and text-like objects when they were introduced, its successor XML is generic (see Stührenberg 2012 in this volume): XML can and is used to represent any kind of object and its properties,

from observable data through texts, archives, and computer programmes (however, very many domain-specific versions of XML have been developed for special purposes, such as VoiceML for speech synthesis objects).

Like HTML, XML expresses tree structures with sets of attributes attached to the tree nodes, each attribute associated with a numerical or textual value. Where other, more complex data structures are required, additional implicit or explicit notational conventions are required. Where an XML document is linked to another entity (such as another XML document), far more complex graph structures can be created, for example. These structures are extrinsic to XML and need other means for monitoring, consistency checking, parsing, etc., than the context-free (or even finite state) parsers which are appropriate for tree analysis.

The following example shows the structure of an automatically generated XML archive document based on an interview (abbreviated with "…"; names, attributes and values modified for publication):

```
<?xml version="1.0" encoding="UTF-8"?>
<accident COST="NEGLIGIBLE" PERSONNUMBER="3">
<title>Walking incident</title>
<report> Yesterday I … </report>
<participant gender="Mr" firstName="An" initialName=""
   lastName="Other" institution="Home News Ltd."
   contact="none" involvement="Victim" role="Witness"/>
<episode-collection>
<episode-item type="ITEM">Yesterday I …</episode-item>
<episode-item type="ITEM">and …</episode-item>
…
</episode-collection>
</accident>
```

An XML 'element' or 'object' has a body consisting of a string of text whose start and end are delimited by tags; the tags are delimited by angle brackets, and the start tag may contain attribute-value pairs representing properties of the element. A second type of element consists of a single tag; an example of this is the first line of the example is a special tag for the element "xml" with metadata about the XML version. The "accident" element which follows occupies the rest of the example, with the start tag on the second line and the end tag on the last line. In the start tag, there are two attribute-value pairs, first the attribute "COST" with the value "NEGLIGIBLE", second the attribute "PERSONNUMBER" with the value "3". Embedded in the body is a series of other elements, "title" (full element), "report" (full element), "participant" (tag only) and "episode-collection" (full element). The element "episode-collection" contains a series of more deeply embedded elements, all with the name "episode-item". The elements "accident", "participant" and "episode-item" contain at-

tribute-value pairs. The depth of embedding is not restricted (in principle, the tree can contain recursion, which would be required, for example, if sentences with subordinate clauses were to be represented). The overall tree graph in this example, of depth three, can be recognised; each node ('element') is associated with a flat tree of attribute-value pairs, of depth two.

The tree graph data structure represented by XML imposes restrictions; not all data types can be comfortably represented by tree structures. Embedded tables, for example, are not simple tree structures, because there are additional constraints on the agreement of the width of rows in the table; theoretically, embedded tables can be represented by indexed context-sensitive languages (a special kind of Type 1 formal language which is more complex than a Type 2 or context-free formal language (cf. Hopcroft et al. 2006). However, in practice this formal property has to be 'faked' in the processing algorithm or by the human designer: the even-branching fan-out needs to be additionally calculated (or provided manually).

Other generic data representation notations are used in standard database technologies, the most prominent type being the linked relation tables used in *Relational Database Management Systems* (RDBMS).

1.4. Operational resources

Operational resources are the tools for manipulating data, and the underlying algorithms on which the implementations are based. Tools are task specific, and therefore depend not only on the modality choice but also on the scenario choice involved. The focus will be on software tools such as parsers and visualisation software. Hardware tools (such as specific computers, audio and video recording devices, specialised input and output devices such as special keyboards, touch screens, braille pads and printers) will not be dealt with. A rough categorisation of operational resources will be discussed below.

2. Resources for text systems

2.1. Informal overview

The main resources for text-based systems, which include information retrieval services of most kinds, are, in general, large collections of texts (for many purposes harvested from the internet), and the search tools for investigating the composition of these texts, whether standalone non-linked texts or hypertexts. Such tools ranging from the 'find and replace' string search function of text editors and word processors to the keyword oriented search of help systems and the keyword plus heuristics (popularity; advertising; 'Did you mean …?') full text

search of internet browsers. These tools, as found for instance in the development and uses of a word processor, utilise resources based on many levels of linguistic, and computational linguistic knowledge.

First, a prominent example from text linguistics is constituted by the 'styles' or format templates which determine both the text structure and appearance of a document require parsing tools which can handle the categories involved. In computational linguistic terms, language units such as characters, words, sentences, paragraphs, documents (articles, books etc.) are assigned appropriate attribute-value structures (feature structures) which need form the basis of text parsers which in turn are used to assign the text rendering or 'appearance semantics' in print. Character codes are assigned implementations in fonts (nowadays typically Unicode) with visual font properties (glyphs and highlighting attributes). The parsing of larger units of language such as words implies the recognition of word boundaries and (for hyphenation), the recognition of the internal structure of words in terms of characters, syllables and morphemes, phonological and morphological analyses. Word prediction resources and spelling correctors require dictionaries. The parsing of sentences for capitalisation, selection and grammar checking requires the recognition of sentence constituents and sentence boundaries. The handling of paragraphs demands a facility for handling their properties such as left, right, top, bottom boundaries (i.e. margins). The most ubiquitous text unit for current word processors is the paragraph: any title, heading, caption, etc. is typically handled as a special type of paragraph, distinguished from other paragraph types by differences in paragraph properties such as top and bottom spacing, left and right indenting, line alignment (left, right, centre, justified), as well as by fonts and their attributes.

Second, an independent formatting layer determined not by language properties but by media properties must be handled. The page structure of a book requires the handling of line-breaks, page-breaks. The page structure of a newspaper requires in addition the non-linear handling of article breaks and continuations. The 'megastructure' of a dictionary requires the handling of cross-references. The constraints on a scientific paper to provide supporting evidence for the content may require a word processing system to provide automatic handling of cross-references for the table of contents, figures (and lists of figures), tables (and lists of tables), footnotes, term indices and bibliographical references. The file architecture of a hypertext network requires the handling of link anchors and targets. At character level, the format layer requires the handling of kerning (character spacing and overlap), ligatures (blends of more than one character) and diacritics (accent marks).

A special case of a text system is the lexicon database, as the basic resource either for a printed dictionary or encyclopaedia, or for a hyperlexicon on the internet or as part of a help system. This example will be dealt with in a separate subsection.

2.2. Example of resource creation for text-based systems – lexicography

Lexicography is the scientific and technological discipline concerned with dictionaries, lexicons, and encyclopaedias. Lexicographic methods are taken partly from the humanities (in particular the language sciences – the 'art of lexicography'), and partly from computational linguistics. Many technical communication systems contain lexicons – or, indeed actually are lexicons – and a lexicon is arguably the most complex linguistic component of a technical communication system (cf. van Eynde et al. 2000).

Theoretical lexicography is concerned with the structure of lexicons and with types of lexical information; the study of types of lexical information is also known as lexicology. Applied lexicography is concerned with the analysis, creation and evaluation of dictionaries, lexicons and encyclopaedias. A dictionary itself may be regarded as a system, most clearly when the dictionary is stored as an electronic database, processed with operational lexicographic resources for distribution on CD-ROM or DVD, or on the internet. In the present discussion, no distinction will be made between dictionary, lexicon and encyclopaedia; for discussion of this distinction reference should be made to the extensive lexicographic literature (see the final section).

The media in which a dictionary is implemented, the architecture of the dictionary and the requirements which are to be met by the architecture and the implementation will be determined by the dictionary use cases and, more specifically, by the dictionary market. A non-exhaustive list of typical use cases might (not including embedded lexicon subsystems) include the alphabetic dictionary (organised by wordforms), the thesaurus and the synonym dictionary (organised by meaning), the idiom dictionary, the bilingual dictionary, the pronunciation dictionary, the rhyming dictionary, the concept-based terminological dictionary.

2.2.1. Lexicon resource structure

In generic terms, any dictionary is a set of *lemmas* (singular: 'lemma'; alternative plural: 'lemmata') organised in a specific well-defined *macrostructure* such as a list or a tree hierarchy, the lemmas each being associated with a well-defined *microstructure* of *data categories*. Additionally, lemmas (alternative plural: 'lemmata') may be interlinked with cross-references and additional explanations; the cross-references constitute the *mesostructure*. The overall structure of the dictionary, together with its published metadata and perhaps also any additional explanatory information is sometimes referred to as the *megastructure*.

Consequently, simplifying the issue, lexicographic resources must first of all contain specifications of the megastructure, macrostructure, microstructure and mesostructure in terms of the desired use cases. For practical applications, each kind of structure requires its own particular combination of empirical,

technical and human resources. The classic case of the semasiological alphabetic dictionary will be taken as an example of such specifications, and the structure types of such dictionaries (which in general apply, *mutatis mutandis*, to other dictionary types) will be outlined as follows.

1. *Macrostructure*. The macrostructure of the alphabetic dictionary is a list of *headwords* or *lemmas* sorted alphabetically, each being the first element in a *lexical entry* or *article* which otherwise contains lexical information about the headword. Macrostructures have certain specific features:
   a. Attention must be paid to the sort order; the traditional ASCII sort order is inadequate in the context of international Unicode conventions, and needs to be specified explicitly in each case. While alphabetic sorting is adequate for many languages with alphabetic orthography, it is evidently less adequate for syllabic scripts and inadequate for logographic scripts. For languages with a very small set of lexicalised prefixes (many African languages), simple alphabetic arrangement is also inadequate.
   b. The alphabetic dictionary is a variety of *semasiological* dictionary, in which the headword represents a wordform and the lexical information concerns the meaning of the wordform. The inverse relation is found in *onomasiological* dictionaries like the thesaurus, where the headword represents a known concept and the lexical information concerns the wordform.
2. *Microstructure.* In the simplest case, the microstructure of a lexicon is an ordered list of types of lexical information. The microstructure of a technical dictionary resource is the most complex part, and also the most difficult to standardise, despite cooperative efforts going back many decades (Atkins et al. 2008; van Eynde et al. 2000). These types of lexical information (also known among computational lexicographers as '*data categories*') concern the following main properties of words:
   a. Word *form* (spelling and hyphenation; pronunciation and prosody, e.g. stress or tone).
   b. Word *structure* (internal: prefixes, suffixes, constituent words in compounds; external: part of speech, grammatical restrictions).
   c. Word *meaning* (descriptive components such as abstractness, animacy, pragmatic components such as style, taboo).
   d. Inter-article *cross-references* (to synonyms, antonyms, examples, sources, etc.).
   e. In a *lexical database*, also *metadata* about the lexicographer, date of processing, comments).
   f. *Hierarchical information*: in more complex cases, the microstructure can be hierarchical, organised as a set of related sub-entries, typically words derived or compounded from the same root.

3. *Mesostructure.* The cross-references in the dictionary constitute a more or less explicit network of relationships between words. The following kinds of relation or network structure may be noted:
   a. The main relations are typically between synonyms and antonyms.
   b. A lexical system such as a WordNet (Fellbaum 1998) uses an elaborated and explicit version of this kind of cross-reference structure as its macrostructure.
   c. In addition, implicit cross references are made by the use of category names such as the parts of speech: a term such as 'noun', or a pronunciation transcription, is not explained for each entry, but reference must be made to the introductory sketch grammar in the megastructure of the dictionary.
   d. An interesting formal feature of the mesostructure of lexicons is often *cyclicity* in cross-references, usually unintended. The 'cross-reference depth' of the mesostructure of a dictionary could contain, for example, the following: the word 'thing' is defined in terms of the nearest kind 'object', the word 'object' is defined in terms of the nearest kind 'entity', and the word 'entity' is defined in terms of the nearest kind 'thing'. Without references to external examples, this cyclicity is inevitable.
4. *Megastructure*. The megastructure defines the structure of the overall implementation of the complete dictionary: the actual organisation in a book, a database or on a website. The most straightforward case is the book: the front matter, including the title page, publication metadata page (with date, author and copyright and publisher details), foreword and preface, table of contents; the sketch grammar; the body (lemma-based list of articles); back matter (e.g. publisher's advertising).

The core of an alphabetic dictionary of this kind is the lexical information contained in the microstructure, and the empirical, technical and human resources for acquiring this lexical information form the largest single kind of resource in a lexicography workbench (see Figure 2).

The entries in Figure 2 are from a traditional alphabetically organised semasiological dictionary. The entries have a microstructure which can be formally modelled as a vector or row in a matrix, with the following elements: headword (simultaneously representing orthographical lexical information), pronunciation (in a variety of IPA transcription), an abbreviation (*adj*, *n*) for the *part of speech* (POS, syntactic category), a definition with a modifying relation in the case of the adjective, an identification of the domain ('tech', i.e. technical), and a classical dictionary definition in the case of the nouns. Additionally, structurally (morphologically) related words are given, such as the adverb 'lexically', the agentive noun 'lexicographer', and, in the case lf 'lexis' there is a mesostructural cross-reference, 'compare VOCABULARY'.

lex·i·cal /'leksɪkəl/ *adj tech* of or about words —~ly /kli/ *adv*
lex·i·cog·ra·phy /ˌleksɪ'kɒgrəfi‖-'kɑ:-/ *n* [U] the writing and making of dictionaries —-pher *n*
lex·i·col·o·gy /ˌleksɪ'kɒlədʒi‖-'kɑ:-/ *n* [U] *tech* the study of the meaning and uses of words
lex·i·con /'leksɪkən‖-kɑ:n, -kən/ *n* **1 a** a dictionary **b** a list of words with their meanings **2** *tech* all the words and phrases used in a particular language
lex·is /'leksɪs/ *n* [U] *tech* all the words that belong to a particular subject or language, or that a particular person knows —compare VOCABULARY

Figure 2. Examples of lexicon articles from a traditional dictionary (the Langenscheidt-Longman Dictionary of Contemporary English, 1987)

The classical dictionary definition is also known technically as a *definition by nearest kind and specific differences* (also in Latin: *definitio per genus proximum et differentia specifica*). In the entry 'lexicon', for example, one definition is simply a synonym, but the next has the nearest kind 'list' and the specific differences 'of words' with a further specific difference 'with their meanings'. The technical term for the headword (left-hand side) in a classical dictionary definition is the *definiendum* (Latin for 'to be defined'), and the right-hand side, with the nearest kind and the specific differences, is the *definiens* (Latin for 'defining').

Traditionally, the information for the data categories in the microstructure is acquired from three main resource types: the lexicographer's knowledge of the language; extensive collections of texts in the language; other dictionaries. The ultimate criterion for a practical dictionary will be the lexicographer's knowledge of the language. However, traditional dictionaries are beset with preferences and idiosyncrasies (definitions, spelling variants, alternative plurals) introduced by the lexicographer. Other dictionaries may be a useful source of information, but if used their idiosyncracies will be perpetuated. Basing a dictionary on extensive corpus resources has the advantage of comprehensiveness and facilitates the development of consensual lexical information.

2.2.2. Lexicon creation

The main type of resource for modern dictionaries is the corpus or text collection, which is processed by modern computational and manual lexicographic methods. For a large general purpose alphabetic dictionary, the corpus will contain a selection of word tokens of the order of tens of millions of word tokens (or more), which may well reduce to a set of word types of the order of a hundred

thousand (depending on corpus size and the definition of 'word'), yielding a type-token ratio greater than 1:100. For a spoken language system, the lexicon will in general be much smaller, based on a scenario-specific vocabulary.

Resource processing consists of the following main steps, which requiring appropriate software tools (note that the procedures listed after tokenisation are not necessarily conducted in the order given):

1. Tokenisation. Individual word tokens are identified, including abbreviations, numbers, prices, dates, punctuation, identification of complex layout objects such as tables.
2. POS (*part of speech*) tagging. Each token is provided with a label (or set of labels) constituting a hypothesis about its part of speech; the European EAGLES (*Expert Advisory Groups for Language Engineering Systems*[5]) developed a standard POS tagset for European languages, which has been extended and applied to other languages (these sets are in flux; consult the internet for up-to-date details).
3. Word token and word type list creation. A list of (possibly inflected) word types is extracted from the set of tokens, often also in conjunction with the word token frequencies.
4. Lemmatisation. A list of lemmas is created from the list of word types, involving stemming in the simplest case, and morphological analysis in the general case (cf. Jurafsky & Martin 2000 Carstensen & al. 2010).
5. Concordancing. A context dictionary consisting of a list of items (types, lemmas, tags, etc.) and the contexts in which they occur in the texts. The best known kind of concordance is the KWIC (*KeyWord In Context*), a simple list of words and their left and right context strings.
6. Word sketching (Atkins & Rundell 2008). Extraction of a maximum of (grammatical and other kinds of similarity) information about lemmas based on their distribution in the texts.
7. Dictionary database compilation. Semi-automatic (moderated) entry of information into the lexical database.
8. Manual editing of lexicon articles (definitions, etc.).
9. Production. Selection, organisation and formatting of lexical information for the intended dictionary megastructure.

These procedures apply, with suitable modifications, to the compilation of other types of dictionary, including dictionaries for use in multilingual, speech-based and multimodal communication systems.

The following examples of KWIC concordances illustrate one of the important types of lexicographic resource (characters simplified).

The first example is taken from an interactive concordance on the internet, as a lexicographic resource for the Verbmobil speech-to-speech translation project in the early 1990s, for the phrase 'jede Woche' ('every week'):

```
6 × jede Woche
cd1_m004n_FRB003: rzigsten Woche jeweils einmal jede Woche .
cd2_m018n_ANP014: ja , dann k“onnen wir ja in jede Woche zwei
   Termine legen und dann h
cd2_m021n_BEP005: ht besser , das “ahm entweder jede Woche
   zu machen , also ein eine Woc
cd3_m025n_TIS002: wochenweise , ne , weil jeden jede Woche
   einen Termin , und dann m“uss
cd3_m025n_RAL009: also , ich glaube , jede Woche am gleichen
   Tag , das kriegen
cd3_m027n_MPU015: ag festlegen , sondern uns hm jede Woche
   einen anderen Tag aussuchen .
```

The second example is an extract from an automatically created printed concordance entry for texts from the Nigerian language Ibibio (characters simplified; the word ‘abasi’ means, approximately, ‘lord’, ‘ruler’):

abasi:
- ukpe ikpe ke esop idan ye ukwooro iko ke ufok **abasi**
- mme okwooro iko **abasi**
- ukpono **abasi** eyeyin
- **abasi** ukot
- **abasi** imaan
- **abasi** ison ye akwa abasi ibom

2.2.3. Lexicon resource acquisition

Lexicon resource acquisition is a complex procedure, which may, however, be reduced to a sequence based on levels of abstraction. A useful hierarchy of such stages in lexicon acquisition, some of which apply to the acquisition of other linguistic resources such as grammars, is shown in Figure 3.

From the point of view of storage in a lexicographic system, all the objects represented by the inside boxes in Figure 3 are data of different kinds, for which different data structures are required. However, from a linguistic point of view it is convenient to distinguish between *corpus data* and *lexicon data*, as in Figure 3.

The primary corpus of *raw data* consists of the formatted text material to be analysed (and in the case of speech, recordings). The raw data may include bilingual information from parallel or comparable corpora if a bilingual or translation dictionary is being compiled.

The secondary corpus of *processed data* consists of the character streams aligned with the raw data, which are segmented into tokens of the required linguistic units (such as characters and character sequences, affixes, words) in a tokenisation step. In a speech corpus the units may be phonemes and syllables, or prosodic units such as accents and tones; in a video corpus the units may be

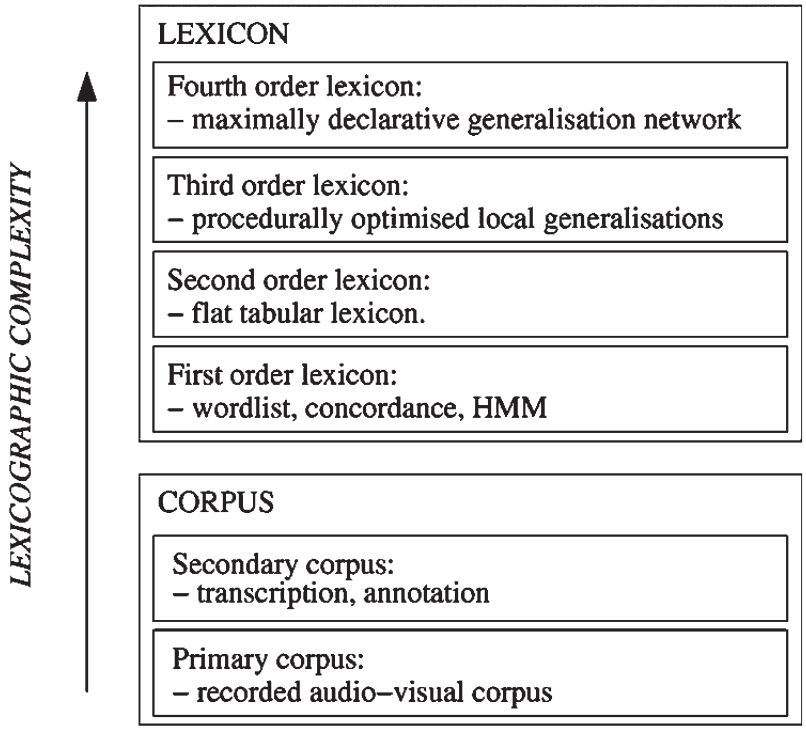

Figure 3. Levels of lexicon data types

gesture configurations, visemes (facial movements, particularly of the lips, associated with phonemes).

The lexicon data and structures are of many kinds depending on the required lexicon use cases. However, the types shown in the figure have some generic validity for all types.

The *first order lexicon* is the concordance as outlined above, for example a list of word-context pairs. Concordancing is a standard procedure in all lexicon construction.

The *second order lexicon* is rarely formulated explicitly, but represents an intermediate stage between concordances and standard lexicon databases: wordform tokens from the corpus are reduced to wordform types and listed separately if, and only if, they have no distinguishing lexical properties of form or meaning. If they have even one distinguishing lexical property, they are listed separately. In this respect the second order lexicon represents an intermediate stage between the concordance, with no abstraction over sets of entries, and the third order lexicon, with use case specific abstractions.

The *third order lexicon*, the most common type, is lemma based, and brings polysemous and perhaps homonymic items together, provided that they have

the same lemma. It is at this stage that distinctions between different use cases become apparent, since the organisation at this level is based on procedural convenience, for instance for semasiological lexicons (with wordform lemmas and semantic lexical information) versus onomasiological lexicons such as thesauri (plural of thesaurus), with concept-based lemmas and form based lexical information. The semasiological-onomasiological distinction is rather simplistic: it is not easy to assign a WordNet, a translation dictionary or a pronunciation dictionary, for example, to the one or the other, though both are clearly third order lexicons.

The *fourth order lexicon* generalises over use cases and models lexical information according to declarative (logical) rather than procedural (use case based) criteria. The fourth order lexicon is mainly of theoretical interest and is less well known than the others, and distinguishes a maximum of linguistic generalisations about pronunciation, grammar and meaning from a minimum of exceptions, based on information extracted from the lexical information available at the lower order levels. The fourth order lexicon uses formalisms such as inheritance graphs (representing implication hierarchies, taxonomies), but has nevertheless been used in practical communication systems (Gibbon & Lüngen 2000). In principle, the fourth order lexicon provides an ideally compact form of storage for lexical information which is intended for re-using in widely different use cases; the generally hierarchical (or even more complex) form is well-suited to the contemporary XML-based data structures used in resource storage.

3. Resources for speech and multimodal systems

3.1. Informal overview

From the perspective of hardware resources, the fields of speech systems and multimodal systems are highly complex, and involve many more components which are dependent on specialised hardware than text based systems. For this reason they cannot be covered comprehensively or in detail in a general handbook article. Fortunately, there exist a number of relatively comprehensive and widely used handbooks on speech resources and, to some extent, on resources for multimodal systems, which should be consulted (see the final section). The present article is restricted to generic considerations and to the specific example of speech synthesis systems.

A general rule is that speech resources are orders of magnitude larger, more complex (and more expensive) to make than text-based resources. A further empirical rule is that multimodal resources are orders of magnitude larger, more complex (and more expensive) than speech resources. Speech systems are often

embedded in multimodal systems. Speech and visual communication resources also share properties and techniques at a generic level: both are concerned with the processing of physical symbols and the mapping of segments of these on to symbolic strings and other patterns. This can also apply to text, but only when text is scanned from analogue sources such as handwriting on paper and represents a kind of visual information. *Optical character recognition* (OCR) algorithms used for analysing scanned text share properties with those needed for speech and vision decoding.

The more abstract and language-related levels of speech and multimodal resources are practically identical to those involved in text processing. Therefore it is appropriate to concentrate more on the form-oriented aspects of pronunciation. It is not possible within the confines of the present context to account for the visual aspect of multimodal resources, including gestural communication (signing by deaf communicators, and conversational gesture). In a previous handbook in the field (Gibbon et al. 1997) a distinction was made between three phases of data acquisition for corpus building and processing, each of which requires rather different operational resources in the form of human procedures or software and hardware tools: the *pre-recording*, *post-recording*, and *post-recording* phases. For further details consult also Gibbon et al. (2000).

1. *Phase I: pre-recording phase*. The pre-recording phase is concerned with use case specific scenario and experiment design based on requirements specifications for later processing in the post-recording phase. These requirements and design issues determine the materials (equipment, prompts, texts, scenario layout, participants etc.) and procedures to be used in the recording phase. Many contextual details need to be taken into consideration: for instance, in a noisy application environment the 'Lombard effect' (change of voice characteristics) is to be found, therefore recording and testing under studio conditions may be inappropriate. This preparatory phase is arguably the most complex phase, and the specialised literature should be consulted: if resource design is not right, the implementation will not be right: 'garbage in, garbage out'.
2. *Phase 2: recording phase*. During the recording phase, scenario or experiment-specific recordings are made as raw data (see Figure 3). For specialised purposes, software (or otherwise) controlled randomised or structured prompts (e.g. for systematic testing and experimentation purposes), specialised environments (e.g. sound-proofed rooms; noisy car or airplane settings; telephone; 'Wizard of Oz' or 'man-behind-the-curtain' simulated human-computer communication) may be needed. For more general purposes, less formal environments with across-the-table dialogue involving face-to-face or hidden communicators may be suitable. Here, too, the specialised handbooks should be consulted.

3. *Phase 3: post-recording phase*. The post-recording phase is essentially the implementation phase for the specifications and design developed in the pre-recording phase. The following generic procedure applies to most speech and multimodal data.
   a. In processing speech and multimodal primary data transcriptions (assignment of strings of labels to recordings, without time-stamps) and annotations (assignment of labels to segments of recordings, with time-stamps). Parallel annotations may be assigned to the same data (the layers of parallel annotation streams are known as 'tiers' – rhyming with 'fears', not with 'flyers'). The annotation procedure may be manual (required for initial bootstrapping), semi-automatic (e.g. automatic but post-edited) or automatic (using statistical annotation software with training component). Information for archiving and further processing is extracted from the annotations.
   b. The next steps are generally the extraction of a list of word tokens from the annotations and the creation of a machine-readable pronunciation dictionary using standardised orthographic and phonetic (more usually: phonemic) coding conventions. Although Unicode is generally used for text-based systems, it is very much oriented towards output for printing, rather than convenient input or processing. In the speech technology context, standardised custom alphabets are generally used, the most common of which, in multilingual resources, is still the SAMPA (sometimes 'SAM-PA'), i.e. SAM Phonetic Alphabet, developed in the European Commission funded SAM ('Speech Assessment Methodology') project in the 1980s and extended to cover all languages in the early 1990s.
   c. Several efficient free speech annotation software tools are available, such as Praat[6] (probably the best known), Transcriber[7], WaveSurfer[8] (in view of the rapid development in the field, the internet should be consulted for further information). There is much less agreement about 'alphabets' for annotating video signals, though there are a number of software packages for video annotation, the most widely used being Anvil[9], Elan[10] and EXMARaLDA[11] (see Dipper et al. 2004; Rohlfing et al. 2006).
   d. Further post-recording analysis, e.g. creation of lexicons, word models, grammars etc., is closely related to the analogous levels in text-based system resource development. Finally, evaluation of resource type and quality is an essential part of current best practice in resource creation and deployment.

Beyond these generic aspects of the pre-recording, recording and post-recording phases are very many technical details: specific algorithms for speech stream transformation and visualisation (including waveform, spectrum, pitch

track), for speech stream segmentation and in multimodal contexts also for video scene line and object detection (see additional references in the final section of this article).

3.2. Example: resources for speech synthesis systems

3.2.1. Resource types

Speech synthesis systems are generally embedded components of systems with more complex functionality. Their specifications therefore depend on the use cases for their technical environment, and on the technical environment itself. Typical uses for embedded speech synthesis systems are public address systems (e.g. railway stations, airports), geographical information systems (e.g. vehicle satellite navigation systems), information systems for non-literate users, dictation software (for readback), screen readers for the blind, speech-to-speech translation software, multimodal systems including robotic systems. In categorising the types of embedding into more complex systems, a two-way distinction is conventionally made between *Text-To-Speech* (TTS) synthesis, where the input is text and the output is speech, and *Concept-To-Speech* (CTS) synthesis, where the input is a conceptual representation (commonly a database) (see also Martin and Schultz 2012 in this volume). The prevalent type is TTS; in practice, CTS systems may also involve text as an interface.

A TTS system requires resources for developing the following subcomponents:

1. Text parser: the text is pre-processed in order to extract implicit information:
   a. The spelling and ultimately the pronunciation of special text components such as abbreviations and numbers must be extracted.
   b. A pronunciation lexicon, usually with additional pronunciation rules, is required.
   c. A parser is needed for disambiguating the structure by picking the correct word readings from the lexicon and delimiting the phrasing of sentences.
   d. A grapheme-to-phoneme (phonetisation) component is used to derive a transcription of the speech sounds for input to the speech processing component.
   e. A prosody module is needed for deriving intonation and accentuation patterns for input to the speech processing component.
2. Signal processing component: conversion from an interface with parsed and phonetised text with added prosodic information into a synthetic speech signal.

The text parser is a special case of the kind of parser which is used in text processing in general, enhanced with phonetisation and prosodic modelling information, and will not be discussed further here.

For the signal processing component there are several different speech synthesis paradigms, including the following main types, for which paradigm specific resources are required:

1. Pre-recorded 'canned' speech. Canned speech is typically used in straightforward information service environments such as satellite navigation systems for vehicles, and for railway station announcements. Systems such as these use a restricted set of utterance templates which permit substitution of station names and times, but also permit a combinatorially large set of new utterances to be synthesised. Canned speech is in principle very comprehensible and very natural, provided that the template units are carefully designed and produced, with close attention paid to the correct prosody (intonation and accentuation), and to appropriate transitions between canned speech units.
2. Unit concatenation speech synthesis. Small units, such as phonemes, diphones, demi-syllables and sometimes larger units, are concatenated to form words and sentences. There are three main approaches, each of which requires different kinds of resource:
   a. Diphone synthesis is one of the first kinds of concatenative speech synthesis, and is still used. In diphone synthesis, pre-recorded speech samples containing all the diphones in the sound system of the language are used, which are concatenated in order to reproduce the patterns of the input syllable and word sequences. A diphone is essentially a pair of phonemes (speech sounds; see below).
   b. Unit selection synthesis, a popular variety of speech synthesis, and in general more natural than diphone synthesis, is based on selecting continuous units from a large recorded corpus. The corpus is designed to contain all the phonemes, generally all the diphones, and perhaps all the triphones (sequences of three phonemes). Units are concatenated after calculating the best possible fit (cost, weight).
   c. *Hidden Markov Model* (HMM) synthesis, a recent development based on stochastic modelling of unit sequences, trained on a suitable corpus.
3. Formant speech synthesis. Formant synthesis is one of the earliest kinds of speech synthesis, and is based on the spectral structure of speech sounds. An acoustic signal is reconstructed from empirical information about vowels, consonants, and the pitch, intensity and duration patterning of the intended synthetic speech signal. In principle, this approach is the most flexible and parametrisable in terms of linguistic and phonetic properties, but is more difficult to use in practical systems than concatenative techniques.

3.2.2. Resource creation

In order to illustrate the resource creation process for a speech system, a traditional diphone concatenation approach is described. For unit concatenation and HMM synthesis, which are more complex and currently under rapid development, the technical literature and internet sources should be consulted (see also the last section for references). However, many of the resources needed for diphone synthesis are also required in similar form for the more complex speech synthesis techniques.

The following resources will be required for a diphone synthesiser:

1. Text processing component:
   a. Text parser as outlined above, which will tokenise words, decode abbreviations, and establish phrasing, and focus points and intervals.
   b. Phonetiser or grapheme-to-phoneme converter, to produce phonemic/phonetic representations of words.
   c. Prosodic model, to utilise the phrasing and focus information to associate pitch, duration and intensity patterning to the word sequence.
2. Signal processing component:
   a. Pre-recording phase:
      i. List of phonemes (and perhaps also major allophones) of the language concerned. The size of the phoneme set varies considerably, depending on the typological properties of the language, from around 15 to several dozen.
      ii. List of diphones based on the list of phonemes. For a phoneme set P the size of the set of diphones is therefore maximally $|P|^2$, the square of the number of phonemes. This set includes both diphones which occur within words and diphones which occur across word boundaries, as well as a pause unit. Since the number of phonemes in the language determines the size of the diphone set, evidently languages vary greatly in the sizes of the diphone sets.
      iii. Prompts containing examples of each diphone. A traditional method of compiling a suitable set of prompts is to find a set of words or longer expressions containing the desired units, and to put these into a standard frame such as “Say X again.” However, a more efficient method of compiling prompts is to create a ‘phonetically rich’ corpus, i.e. to start with a large text corpus and extract the minimum number of sentences which together contain all the diphones in the entire text; this can be automatised, for example by means of a common scripting language such as Unix/Linux shell, Perl, Python or Ruby. If any diphones are missing, further sentences need to be added.

b. Recording phase:
   i. For commercial systems, a professional speaker is usually recorded, in a professional studio. In prototype development, less stringent standards are imposed. Nevertheless, certain technology-dependent conditions need to be met: good voice quality (no 'creaky' or 'breathy' voice), good control of pitch and volume, pronunciation and intonation patterns which are appropriate for the task concerned. For special applications such as video games, in which emotional and aggressive speech varieties are often encountered, special but still highly controlled conditions apply.
   ii. The recording equipment and environment need to be carefully controlled in order to avoid unwanted noise, echo, delayed acoustic feedback to the speaker (cf. the approximately 20 ms syllable-length delay which leads to the 'Lee Effect' of involuntary speech inhibition).
c. Post-recording phase:
   i. Recordings are archived with well-defined file-name conventions, in a suitable data management environment, associated with standardised metadata such as the data proposed by the *Open Language Archive Community* (OLAC[12]; consult the internet for further details).
   ii. Recordings which do not correspond to the requirements are rejected (and re-recorded if diphones are then missing). Audio normalisation (e.g. of intensity) is performed where required, in order to achieve uniform recording properties.
   iii. Annotation of recordings with transcriptions and time-stamps, is carried out using specialised software. The basic kind of annotation, which provides a benchmark for automatic annotation, is manual annotation by phonetically trained skilled annotators. An essential part of the annotation process is the evaluation of the annotations, both by objective means (e.g. monitoring the use of correct annotation labels) and by experimental intersubjective means (e.g. comparing the annotations of independent annotators, which are never 100 % identical but should approach 90 % similarity in use of labels and in time-stamp accuracy as far as possible). An example of annotation will be discussed below. Semi-automatic annotation either uses interactive software for labelling, monitored by a skilled human annotator, or post-editing of automatically generated annotations, Fully automatic annotation, which can be necessary if very large recorded corpora are used, may start with the orthographic prompt text, phonetise the text automatically with a grapheme-to-phoneme converter, and assign time-stamps relating to the recorded signal using statistically trained forced alignment software; post-annotation editing is often

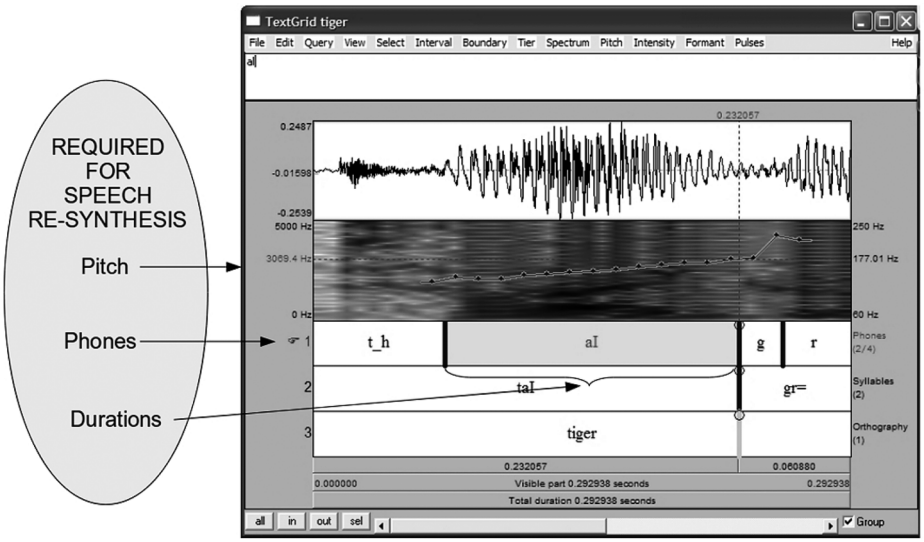

Figure 4. Annotation of speech signal for speech synthesis with the Praat software workbench

necessary, however, unless tests of samples have shown that inaccuracies are not relevant in practice.

iv. Diphone database creation. Diphones are extracted from the annotated recordings and processed in order to create a diphone database or 'diphone voice' for use in the runtime speech synthesis system. Processing can involve pitch extraction and normalisation, and the normalisation of the intensity (volume) and duration of the diphones in order to facilitate the runtime synthesis procedure. Extensive evaluation of the diphone database during the development process by a skilled phonetically trained evaluator is required in order to establish that the diphones it contains can be used without distortion in many different word and sentence contexts.

v. System evaluation. The evaluation of the diphone voice in the runtime system uses extensive and varied perception tests with subjects who represent potential users, based on criteria of comprehensibility and naturalness.

One of the core resources of any speech technology system is the annotated speech recording. In Figure 4 the main features of a typical annotation using the Praat software workbench (Boersma 2001) is shown.

Figure 4 shows the speech signal in three visualisations: as waveform (top), as spectrogramme (second from top), and as pitch track (pitch trace, fundamental frequency trace), superimposed on the spectrogramme. Below the signal vis-

ualisations are three tiers for three annotation streams: phones (phoneme tokens in context), syllables and words. For a diphone synthesiser, the phone tier is the crucial source of information.

For further processing, the stored annotation file is used. The annotation visualised in Figure 4 is stored internally in the Praat system as follows (the 'Phones' tier only):

```
File type = "ooTextFile"
Object class = "TextGrid"

xmin = 0
xmax = 0.2929375
tiers? <exists>
size = 1
item []:
  item [1]:
    class = "IntervalTier"
    name = "Phones"
    xmin = 0
    xmax = 0.2929375
    intervals: size = 4
intervals [1]:
      xmin = 0
      xmax = 0.0718
      text = "t_h"
    intervals [2]:
      xmin = 0.0718
      xmax = 0.2323
      text = "aI"
    intervals [3]:
      xmin = 0.2323
      xmax = 0.25579
      text = "g"
intervals [4]:
      xmin = 0.2557
      xmax = 0.2929
      text = "r `="
```

The characteristics of the annotation can be seen clearly: each annotation label is represented by a numbered segment ('interval') with three properties, two time stamps and a text label: xmin = 0.2323 (the start of the interval), xmax = 0.25579 (the end of the interval) and text = "g" (the transcription label).

Diphones are constructed from pairs of phones, with the beginning of the diphone starting in the middle, or the most 'stable' part of the first phone and continuing to the middle, or the most 'stable' part of the second phone. The definition of 'stable' varies from phoneme to phoneme, and is generally most easy to

identify in vowels and other continuous sounds, and least easy to identify in obstruent consonants.

The more modern techniques of unit selection synthesis and HMM synthesis, as well as others which have not been mentioned, require somewhat different procedures. For example, whereas in standard diphone synthesis the selection of suitable diphones from the recorded corpus is done prior to runtime and stored in a database, in unit selection synthesis the recorded corpus is, essentially, the database, though pre-runtime optimisations and calculation of properties of units in the database are performed. This has consequences for other components of the speech synthesiser and resources for these. In a diphone speech synthesiser such as the well-known MBROLA diphone voice handling system (Dutoit 1997) the pitch and duration values will in general be calculated on the basis of rules of grammar and a prosody description, and included in a well-defined interface between the text parsing and the actual voice handling part. For a unit selection system, the corpus itself in general needs to be annotated with prosodic features, either in a rule-based fashion or derived statistically from the corpus itself.

The development procedure, and the creation of appropriate resources and enabling technologies to facilitate development even of a relatively straightforward traditional diphone based system is evidently highly complex, and the specialist handbooks and their bibliographies should be consulted for further information (see the final section).

4. Recommendations for further reference

In the text reference was made to the need to consult specialist literature for further information, background knowledge and development recipes. For up-to-date information, judicious consultation of the internet is advised, particularly of contributions to major conferences in the fields concerned. However, the following short list of publications will serve as a starting point.

1. Text resources: Abeillé (2003), van Halteren (1999) (corpus processing); Fellbaum (1998), Atkins & Rundell (2008), van Eynde & Gibbon (2000) (Lexicography); Beesley & Karttunen (2003) (finite state modelling);
2. Speech resources: Coleman (2005) (overview); Dutoit (1997) (speech synthesis); Gibbon et al. (1997), Gibbon et al. (2000) (spoken language resources and standards); Wahlster (2000), Wahlster (2006) (speech-to-speech translation; speech in mobile devices).
3. General: Carstensen et al. 2010 (computational linguistics and speech technology, in German); Goldfarb & Prescod (XML technologies); Jurafsky & Martin (2000) (speech and language technologies); Hopcroft et al. (2006) (formal languages); Lobin (2010) (text technology).

Since the field of resources for technical communication systems is developing and expanding rapidly, it is advisable also to consult the proceedings of the most important conferences in the field. The central conference series for resources in both speech and text processing areas is the *Language Resources and Evaluation Conference* (LREC[13]) series, whose events take place every two years; for speech and multimodal communication alone, the Interspeech conference series is the major event.

Notes

1 http://www.iso.org/iso/home.html
2 http://www.langsci.ucl.ac.uk/ipa/
3 http://www.w3.org/
4 http://www.w3.org/Style/CSS/
5 http://www.ilc.cnr.it/EAGLES/home.html
6 http://www.fon.hum.uva.nl/praat/
7 http://xml.coverpages.org/transcriber.html
8 http://www.speech.kth.se/wavesurfer/
9 http://www.anvil-software.de/
10 http://www.lat-mpi.eu/tools/elan/
11 http://www.exmaralda.org/
12 http://www.language-archives.org/
13 http://www.lrec-conf.org/

5. References

Abeillé, A. (ed.)
2003 *Treebanks: Building and Using Parsed Corpora*. Dordrecht: Kluwer Academic Publishers.

Allwood, Jens and Elisabeth Ahlsén
2012 Multimodal communication. In: Alexander Mehler, Laurent Romary and Dafydd Gibbon (eds.), *Handbook of Technical Communication*, pp. Berlin/ Boston: De Gruyter.

Andry, François, Norman M. Fraser, Scott, Simon Thornton, Nick J. Youd
1992 Making DATR work for speech: lexicon compilation in SUNDIAL. *Computational Linguistics* 18:3, 245–267.

Atkins, B. T. S., Rundell, Michael
2008 *The Oxford Guide to Practical Lexicography.* Oxford: OUP.

Beesley, Kenneth R., Lauri Karttunen
2003 *Finite State Morphology.* Stanford: CSLI Publications.

Boersma, Paul
2001 Praat, a system for doing phonetics by computer. *Glot International* 5:9/10, 341–345.

Carstensen, Kai-Uwe, Christian Ebert, Cornelia Ebert, Susanne J. Jekat, Ralf Klabunde, Hagen Langer
2010 *Computerlinguistik und Sprachtechnologie*. Heidelberg: Spektrum Akademischer Verlag.

Coleman, John
2005 *Introducing Speech and Language Processing*. Cambridge: Cambridge University Press.

Dipper, Stefanie, Michael Götze and Manfred Stede
2004 Simple Annotation Tools for Complex Annotation Tasks: an Evaluation. In: Proceedings of the LREC Workshop on XML-based Richly Annotated Corpora, 54–62.

Dutoit, Thierry
1997 *An Introduction to Text-To-Speech Synthesis*. Dordrecht: Kluwer Academic Publishers.

Eynde, Frank van, Dafydd Gibbon
2000 *Lexicon Development for speech and language processing*. Dordrecht: Kluwer Academic Publishers.

Fellbaum, Christiane (ed.)
1998 *WordNet. An Electronic Lexical Database*. Cambridge, MA: MIT Press.

Gibbon, Dafydd, Roger Moore, Richard Winski
1997 *Handbook of Standards and Resources for Spoken Language Systems*. Berlin: Mouton de Gruyter.

Gibbon, Dafydd, Harald Lüngen
2000 Speech Lexica and Consistent Multilingual Vocabularies. In: Wahlster (2000), pp. 296–307.

Gibbon, Dafydd, Inge Mertins, Roger Moore
2000 *Handbook of Multimodal and Spoken Dialogue Systems: Resources, Terminology and Product Evaluation*. New York: Kluwer Academic Publishers.

Goldfarb, Charles F., Paul Prescod
2003 *XML Handbook*. Upper Saddle River, NJ: Prentice Hall.

Hopcroft, John E., Rajeev Motwani, Jeffrey D. Ullman
2006 *Introduction to Automata Theory, Languages, and Computation* (3rd ed.). Addison-Wesley.

Jurafsky, Daniel & James H. Martin
2000 *Speech and Language Processing. An Introduction to Natural Language Processing and Speech Recognition*. Upper Saddle River, NJ: Prentice Hall.

Halteren, Hans van (ed.)
1999 *Syntactic Wordclass Tagging*. Dordrecht: Kluwer Academic Publishers.

Lobin, Henning
2010 *Computerlinguistik und Texttechnologie*. Paderborn: Fink.

Lücking, Andy and Thies Pfeiffer
2012 Framing multi-modal technical communication. In: Alexander Mehler, Laurent Romary and Dafydd Gibbon (eds.), *Handbook of Technical Communication*, pp. Berlin/Boston: De Gruyter.

Martin, Jean-Claude and Tanja Schultz
2012 Multimodal and speech technology. In: Alexander Mehler, Laurent Romary and Dafydd Gibbon (eds.), *Handbook of Technical Communication*, pp. Berlin/Boston: De Gruyter.

Paaß, Gerhard
2012 Document classification, information retrieval, text and web mining. In: Alexander Mehler, Laurent Romary and Dafydd Gibbon (eds.), *Handbook of Technical Communication*, pp. Berlin/Boston: De Gruyter.

Rahtz, Sebastian
2012 Representation of documents in technical communication. In: Alexander Mehler, Laurent Romary and Dafydd Gibbon (eds.), *Handbook of Technical Communication*, pp. Berlin/Boston: De Gruyter.

Rohlfing, Katharina, Daniel Loehr, Susan Duncan, Amanda Brown, Amy Franklin, Irene Kimbara, Jan-Torsten Milde, Fey Parrill, Travis Rose, Thomas Schmidt, Han Sloetjes, Alexandra Thies and Sandra Wellinghoff
2006 Comparison of multimodal annotation tools. *Gesprächsforschung* 7: 99–123.

Stührenberg, Maik
2012 Foundations of markup languages. In: Alexander Mehler, Laurent Romary and Dafydd Gibbon (eds.), *Handbook of Technical Communication*, pp. Berlin/Boston: De Gruyter.

Trippel, Thorsten
2012 Controlled language structures in technical communication. In: Alexander Mehler, Laurent Romary and Dafydd Gibbon (eds.), *Handbook of Technical Communication*, pp. Berlin/Boston: De Gruyter.

Wahlster, Wolfgang (ed.)
2000 *Verbmobil: Foundations of Speech-to-Speech Translation*. Berlin, Heidelberg, New York: Springer.

Wahlster, Wolfgang
2006 *Smartkom: Foundations of Multimodal Dialogue Systems*. Berlin: Springer.