Prosody, Time Types, and Linguistic Design Factors in Spoken Language System Architectures

Dafydd Gibbon
Fakultät für Linguistik und Literaturwissenschaft
Universität Bielefeld
Postfach 100131, D-4800 Bielefeld 1
gibbon@asl.uni-bielefeld.de

Summary Attempts to extend work on speech and natural language systems to the broader spoken language (SL) domain rapidly meet with bottlenecks due to temporal features of spoken language at different levels, and to the projection problem connected with speaker and language variation in spoken language, from speech style through dialectal to multilingual variation. It is suggested that part of the solution to the bottleneck problems is to consider typical SL problems like the 'Prosodic Paradox', connected with integrating the notion of time into linguistic descriptions. Four notions of time are distinguished, and a novel arrangement of linguistically motivated components for SL system architectures is suggested. Finally, consequences of this framework for the specification of a prosodic parser are discussed.

Zusammenfassung Wenn Ergebnisse von Sprachtechnologie und natürlichsprachlichen Systemen auf die gesprochene Sprache (SL) im weiteren Sinne ausgedehnt werden, trifft man rasch auf diverse Flaschenhalssituationen; diese betreffen die zeitlichen Eigenschaften gesprochener Sprache sowie das Projektionsproblem der Variation, von Sprecher und Sprechstil bis zu dialektalen und multilingualen Varianten. Als Teil einer Lösung für diese Flaschenhalsprobleme werden typische SL-Probleme wie das 'prosodische Paradoxon' und die Integration eines Zeitbegriffs aus linguistischer Sicht untersucht. Vier relevante Zeitbegriffe werden unterschieden, und eine neuartige Anordnung linguistisch motivierter Architekturkomponenten vorgeschlagen. Zum Schluß werden einige Konsequenzen für einen prosodischen Parser besprochen.

1 Spoken language, SL systems, SL system architectures

The description and modelling of spoken language (SL) is a complex endeavour, in which a number of well-known development bottlenecks have to be overcome. These bottlenecks are partly quantitative, relating to the size of systems and, more important, to the size of the resources involved in data and knowledge acquisition logistics, which requires specialised linguistic skills and techniques from phonetics to dialogue analysis. The bottlenecks are partly qualitative, involving theoretical problems of the variability of linguistic units and speaker behaviour, from simple repetitions, through speech style, speech register, dialect, to complex multilingual variation. In many of these areas, results obtained from previous work on written language (WL) and keyboard dialogue, or in text-to-speech and other speech front end (SFE) systems are not directly transferable, as they abstract away from central parameters of SL such as time dependence, prosody, constraints on rapid fluent dialogue, or the projection problem in flexible but restricted multi-code communication.

Treatment of these bottlenecks requires increasing attention to domain-specific linguistic constraints and support for empirical techniques of spoken language data and knowledge acquisition, as well as new software engineering concepts. This range of variability is too great to be captured by statistical means alone, and requires language models of higher complexity. A common criticism of such suggestions is that we do not know whether the linguistic categories are the best categories; this is a feature of all empirically testable - i.e. falsifiable - theories and systems, however. This paper is an attempt to outline central linguistic design features for SL systems. They will need to be relativised in the context of non-linguistic system design factors. However, linguistic design features relate closely to cognitive factors, which also affect other design choices such as incremental and signal-synchronous processing with partial analysis and top-down prediction from multiple knowledge sources; other things being equal, non-hybrid and non-*ad hoc* domain-oriented solutions are to be preferred.

Intuitively, spoken language as understood here is defined as follows: *Spoken Language is the use of restricted forms of verbal dialogue among a restricted group of people with a common core of restricted and (partially) shared cultural and communicative code conventions, for a restricted and (partially) shared task, and within a restricted and (partially) shared domain of discourse.* A spoken language system (SL system) is taken to be a software and hardware package which will support, augment, and substitute for some features of SL which a fluent member of a SL user group might reasonably be expected to have mastered. Solutions to the problem of developing SL systems depend on parametrisation of the above dimensions, and on further reference to political, economic, intellectual, and resource oriented development factors.

Linguistic design features for SL system architectures differ along all of these dimensions from design features for WL or keyboard dialogue or SFE systems. One of the main design features concerns the different roles of *time* in SL as opposed to *time* and *space* in WL and SFE systems. A key role in this respect is played by prosody, which has both declarative and procedural functions in the temporal domain: prosodic units have meaning in terms of speaker states and speech styles, and they 'point' indexically in time to focussed constituents in SL, but they also mark states in the processing of SL, such as constituent chunking, iteration, termination at word, phrasal, textual and dialogue levels. The main focus in this paper is on the consequences of prosody and its pervasive multifunctionality as a fundamental design problem for SL system architecture, and the problem will be illustrated with special reference to prosodic morphology as a prototypic SL problem. The following sections deal with the SL domain, the problem of time and the Prosodic Paradox in SL, an integrated approach to linguistic components for SL system architectures, and consequences for prosodic parsing as a specific example of a SL processing problem.

2 The SL Domain

As a canonical reference point, the most elementary form of SL will be defined as follows: *Canonical Spoken Language is dyadic auditory dialogue between users of a homogeneous language variety for a cooperative common task with few and uniquely identified concrete objects.* At the opposite corner in the multidimensional space of spoken language varieties (SL variety space) is the complex communicative problem faced by a stranger in an unknown land with a language he does not speak. The traveller in SL variety space rapidly finds himself in various forms of multi-code situation: the dimensions of speaker variation, dialect and sociolect variation, speech style variation (e.g. formal-informal; humorous-boring), speech register variation (architect-doctor-politician...), multi-modal communication, and the fact that within one variety, there are multi-code elements such as style-shifting, code-switching, the use of foreign words, and citations. There have been many attempts to define empirically the specific 'design features' (Hockett 1958) or 'constitutive factors' (Jakobson 1960) involved in language as a means of communication, as opposed to other systems of human or animal behaviour or to machine operations; the main approaches are semiotic theory, behaviourism, empiricist functionalism (London), rationalist functionalism (Prague) and discourse analysis. The variety space for restricted SL and WL in international contexts is discussed by Gibbon (1981, 1985, 1992). Computational linguistics has so far been less concerned with contextual and communicative issues than with syntax theory or lexicography; a 'Contextual Revolution' may perhaps be around the corner, influenced by AI models of keyboard dialogue and recent developments in situation and discourse semantics, but also by the exigencies of SL modelling.

3 Time Types and the Prosodic Paradox

The problem of time in the SL domain will be discussed in linguistic terms, rather than in terms of contemporary temporal logics or of processing theory. SL differs most obviously from WL in its indexical properties: SL is constituted by behaviour in time, WL is generally taken to be the printed word in two-dimensional space. However, WL is, on closer inspection, also human temporal behaviour mediated by various instruments (e.g. finger movements with a pen or on a keyboard), and at this level it is comparable with SL. A simple analogue (with similar algorithms and analysis steps) to current state of the art single word speech recognition components is Optical Character Recognition (OCR) of well-separated, clearly formed characters from pre-defined font types; a more sophisticated analogue is the recognition of more sloppily written characters and words. Somewhat analogous to the general problem of speech recognition is that of the recognition and understanding of fluent handwriting in pen-based input.

One of the basic precepts of the 'competence' idealisation in linguistics, as with structural linguistics in general, is that linguistic descriptions abstract away from performance factors and are therefore 'timeless' or 'asynchronic'. This view has led to various difficulties, for instance, how to deal with context change in text semantics, or with 'iconic' properties of utterances in which temporal word order reflects temporal event order, or what to do with phonological features such as duration. The difficulty which I would like to focus on here is the Prosodic Paradox (cf. Gibbon 1987):

The Prosodic Paradox

(1) *The defining feature of prosodic categories and structures is the patterning of phonetic features such as pitch and loudness over temporal domains longer than a phonemic segment.*

(2) *Knowledge of prosodic categories and structures belongs to the competence of speakers of a given language.*

From (1) we see immediately that prosodic categories and structures crucially have temporal properties. From (2), however, we may conclude on the general premise of the atemporality of competence descriptions, that prosody has no temporal properties. These characterisations are by no means 'straw men', but are commonly to be found in the literature. A solution to the Prosodic Paradox is proposed here in terms of temporal type distinctions for levels of SL description. Although this may seem, in retrospect, to be an obvious solution, no systematic solution has been proposed previously. Four temporal type levels will be distinguished for this purpose; they co-exist as different perspectives on SL systems and utterances: *Category time*, *Structural time*, *Algorithm time* and *Process time*.

(1) *Category time*, T_{cat}: The null case, applicable to the 'competence' or long term storage aspects of a SL system; from this perspective, a category of whatever size may be seen as a point, or as a-temporal. Since a point in time has no temporal parts (there may be other dimensions, of course), the parts of any object at this point have no temporal parts either. Temporally interpretable prosodic features have purely mnemonic status at this abstract level.

(2) *Structural time*, T_{str}: The most abstract level which has an empirically interesting notion of time. Categories are conceived as temporal intervals, and relations over these intervals are defined: precedence, overlap, inclusion (cf. Bird & Klein 1990). Other than these, categories have no temporal properties. These relations are expressed in linguistic grammars and structural descriptions; they are most clearly relevant to facts about linear precedence and autosegmental (prosodic) tier association. In the special case of immediate precedence, a virtual point (represented for instance by a vertex or node in a chart parse of a sentence) separates two intervals; this notion of point is quite different from that in T_{cat}. In an empirical interpretation, intervals have a second order temporal property of duration (based on subjective or statistical generalisations). The duration property is a quasi-constant, dependent on contextual factors. Appropriate formalisations for T_{str} involve event semantics.

(3) *Algorithmic time*, T_{alg}: The location of states of a machine within a temporal coordinate space, and of temporal transition relations within the state space. The points may be interpreted as points in time, and the relations as minimal intervals (differences) between these points. A special case is the clock time represented by a hardware timing diagram and interpreted in terms of the instructions of a specific processor. Algorithmic time defines temporal complexity as linear, logarithmic, polynomial, or non-polynomially hard.

(4) *Process time*, T_{pro}: Measured linear signal time (or subjective judgment time), an independent empirical variable calibrated relative to a clock system such as the physiology of a speaker or hearer, or oscillations in an electronic system. Process time is an independent variable calibrated in terms of points, which are first-order quasi-constants (more or less precise constants) in a temporal coordinate system. A special case may be termed T_{utt}, the timing of utterance tokens and their contexts in terms of either subjective or clock time. 'Real-time' behaviour is when the T_{pro} of a system process is linearly related to an independent process which it models, for instance when a speech analyser functions in time $T_{pro}=nT_{utt}$; in the special case of on-line real-time, $n=1$. Current work in formal declarative phonetics is based on a denotational semantics with T_{utt} as the domain for T_{str}. Appropriate formalisations for T_{pro} involve point-based interval semantics.

Using these definitions, a number of otherwise confusing aspects of prosody features can be sorted out in terms of definitions at these different levels. Competence definitions are at type level (1). Prosodic theories which contain grammatical and lexical descriptions of discourse, text, phrase and word structure, with their associated prosodies (such as intonation, tone) as well as auditory paralinguistic features and visual gestures are at type level (2). Theoretical computational linguistics is concerned, *inter alia*, with type level (3). Experimental phonetics, experimental psycholinguistics, psychoacoustics and speech technology are concerned mainly with type level (4). A typical task for 'on-line' experimental phonetics and experimental psycholinguistics is "find a function which relates T_{utt} to some independently defined T_{pro}, and to T_{str}"; an application to prosodic parsing is given by Gibbon & Braun (1988).

A complicating factor for prosody is that there are three major temporal domains for prosodic timing (cf. Gibbon 1987; also Tillmann & Mansell 1980, Chao 1968), corresponding to the phonemic, word and supra-word levels respectively; that is, T_{utt} and its relations to other type levels is factorised into at least three, only partially synchronised temporal domains:

(1) *Micro-prosodic timing*: < < 250 ms: subsyllabic segments; pitch perturbations, intrinsic pitch

(2) *Core prosodic timing*: ca. 250...1500 ms: syllables within words, and accent peaks and tones

(3) *Macro-prosodic timing*: > > 1500 ms, words in phrases, sentences, texts and discourses; pauses and intonation contours in larger contour hierarchies. Macro-prosodic timing encompasses a scale with further divisions, with so-called 'paragraph intonation' or 'paratones', and discourse intonation at higher levels (cf. Gibbon & Richter 1984).

Linguistic categories are mapped at each level on to two representation types: a PHON mapping via structural and algorithmic time to phonetic utterance token representations, and a SEM mapping through logical form and algorithmic time to the indexical situation of utterance. The two mappings are theoretically entirely analogous.

A major source of confusion in SL processing concerns the status of structural descriptions in SL parsing: traditionally, structural descriptions are free from information about location in temporal coordinates (T_{cat}). However, a more realistic view of parse results is in terms of T_{str} as temporal relations in a system of virtual point coordinates; these have (partial) indexical interpretations in terms of T_{utt} points. Thus, the chart vertices or nodes in a SL chart parser are not simply uniquely indexed as T_{str} at type level (2), but also by quasi-constants derived from the temporal coordinate system of T_{utt}. Indexing by a module-independent variable is necessary for inter-module synchronisation of uniquely identifiable indexical information about the token under analysis. Thus, T_{str} and T_{utt} can be thought of as 'type time' and 'token time', T_{typ} and T_{tok}, respectively. In current speech recognition work, the problem of mapping T_{utt} on to T_{str} is known as 'lattice parsing', in which a stream of word or phoneme hypotheses as input to a system module is organised into a temporally indexed precedence/overlap lattice of competing word or phoneme hypotheses. Where a chart parser is concerned, the problem of relating points in T_{utt} to the virtual points in T_{str} is known as the *lattice-chart mapping problem* (cf. Chien & al. 1990; Section 5 below).

These distinctions, which are clearly of more than theoretical interest, are generally blurred in traditional WL and keyboard dialogue processing. The identity of WL as object language and WL as theory language, as well as the greater familiarity of most people with conscious analysis of written text as opposed to spoken discourse, generally leads parsing theoreticians and practitioners to omit or abstract away from the fact that at runtime, a WL parser *analyses an inscription token* (or a sequence of finger movements over a keyboard) by *assigning it to a type*. In a WL parser, the keyboard and disk hardware and drivers are unobtrusive and efficient A/D-converters which remove all awareness of the millimetres and milliseconds of finger movements or sensitive magnetic media from the mind of the NLP or CL worker, and thus in general insulate him intellectually from the core problems of SL processing. Traditional linguistic training insists that the declarative component of linguistics is about competence, and that linguists describe types, not tokens; description of utterance types with no empirical description of utterance tokens is an evident impossibility, however, whether by a linguist or by a parser.

4 Domain oriented linguistic architecture for SL systems

Component candidates for SL systems may be expected to have standard module properties: modules can be developed independently of each other; modules have an immediate interpretation in terms of a task or domain model, may change internal details without affecting other modules, have specific internal data structures for domain representations and functions which relate these structures both internally and to inputs from other modules. Using the linguistic design criteria discussed in the preceding section, a number of candidates for components of SL systems, and for relations between these components, may be identified; they differ in some ways both from traditional linguistic views and from current notions of system architecture. However, they constitute a strong set of domain-specific constraints on possible module interactions within a SL system architecture. On linguistic and psycholinguistic grounds, there are reasons to suppose that complex language models with top-down and bottom-up interaction are required right down to the sub-word level. The basic SL components defined by this approach are shown in Figure 1. The components constitute design features for the linguistic part of a generic SL architecture.

Linguistic category type T_{cat}	Semantico-pragmatic interpretation	Phonologico-prosodic interpretation	Inter-pretation procedures T_{alg}	Indexical utterance context $T_{pro}/T_{tok}/T_{utt}$
	T_{str}/T_{typ}			
				Situation
	Discourse referents, utterers		SEM	> Utterance token, signal
Discourse dialogue convent.				
		Discourse intonation, paraling.	PHON	>
	Argument, narrative, ...		SEM	>
Text, 'para.' structure				
		Speech act and text intonation	——PHON——	——>
	Thematic, Predicate interpret.		SEM	>
Sentence, Phrase syntax				
		Sentence intonation accent	PHON	>
	Lexical semantics		SEM	>
Word syntax, morphology				
		Morphophonology, word prosody	PHON	>

Figure 1: Linguistic design components for generic SL architectures

Linguistic design components do not in themselves define modules for a specific SL architecture because modularisation decisions may need also to be based on other factors. The design components can be defined horizontally, i.e. by assigning to each module its own time domain; with few exceptions, vertical grouping does not make so much sense, though there are clearly borderline cases, which may motivate module overlapping, such as clitics, functional units and grammatical words between morphology and syntax, or complex sentences between phrase structure and texts, or the semantics of compound words between word semantics and sentence semantics. Figure 1 represents the following basic linguistic design requirements:

(1) *Definition of categories* (elementary categories, rules of composition, lexicalised categories) *and systems of categories* (in terms of types and sorts), i.e. grammar and lexicon, defining T_{cat} (left column).

(2) *Definitions of the mapping of categories via their forms* (phonological representations) *and meanings* (logical form, semantic representations) defining T_{str}/T_{type} (columns 2, 3).

(3) *Indexical temporal properties* defining $T_{pro}/T_{tok}/T_{utt}$ (right column).

(4) *Processor definitions* for linking information from other modules within the overall architecture concept (cf. Section 5), defining T_{alg} for parsing, generation, or other types of knowledge access (column 4).

The force of this suggestion for explicit models of prosody is that each module not only has its own semantic (and indexical) interpretation, but also that it has its own phonological (and phonetic) interpretations: traditional phonology and word prosody are then phonological interpretation in the word module, while prosody in general is phonological interpretation in the other modules: Phrase, Text, Discourse.

Empirical linguistic constraints on modules such as these reduce the directed communication paths within a modular SL system from the maximum of n^2-n (assuming modules do not communicate in the same sense with themselves) to a smaller number, but greater than the n-1 of traditional serial non-incremental processing. The next section deals with some consequences of this framework for the parser function for the inverse word prosody mapping.

5 A prosodic morphology parser module: design criteria

A basic division of morphology will be assumed in terms of T_{cat} (roughly: immediate dominance) and T_{str} (roughly: quasi-linear precedence; cf. Section 3, and for linguistic motivation, see Gibbon 1990, Bleiching 1991). The word prosodic parser has the task of extracting word constituents in T_{str} from the precedence, overlap and inclusion information provided by the syllable parser, and presenting these as input to the categorial morphotactic parser proper. In doing this, it takes into account distinctive word-level tonal properties (such as the occurrence of accent), meaningful prosodic units (such as accent forms, boundary tones), as well as distinctive phoneme and archiphoneme-like segments, and 'long components' due to morphophonotactic constraints and overlapping assimilations. Below their domain, parsers 'see' only T_{cat}; temporal parts of terminal symbols are irrelevant.

The work to be done by the word prosodic parser is considerable, not only because of ambiguities but also because the search space is enormous: one of its tasks is to identify not only existing lexicalised items, but to identify, as far as possible, well-formed non-lexicalised items and to distinguish these from ill-formed input. Consequently, top-down constraints from multiple knowledge sources, as well as cognitively and phonetically plausible procedural design considerations are required. Word prosody and morph parsing illustrates well the specific requirements for SL processor modules in general. An appropriate parsing procedure needs to cope with the following problems:

(1) Incremental left-right processing based on time-flow and memory characteristics.
(2) Inter-module top-down constraint of predictions to minimise the search space and maximise the deterministic element in SL processing.
(3) Uncertain beginnings and ends of input due to unpredictable noise and performance properties of speakers (parallelled by attention-wandering etc. in hearers).
(4) Ill-formed input, also due to unpredictable noise and performance properties.
(5) Input structures which are more complex than single strings in being graph or lattice structured, and differing from conventional parser inputs in at least four ways:
- competing input hypotheses (ambiguous input; this is partly a construct of current models, and may be minimised by using underspecified input)
- underspecified input for reasons of noise or ambiguity
- non-segmental multi-channel (multi-tier) prosodic event input
- inputs with complex and varying temporal structure, tagged with indices locating them within the temporal coordinates of T_{utt}.

(6) Output structures which are more complex than sets or lattices of conventional letter-string words, being lattices of (possibly underspecified) feature structures as keys for lexical access; they have the following additional properties:
- outputs are tagged with indices locating them within the temporal coordinates of T_{utt}
- ambiguous outputs are underspecified (in preference to disjunctions)
- outputs distinguish between lexicalised and inferred (compositionally constructed, potential) lexical items

(7) Calculation of analysis confidence weightings on the basis of
- stochastic information on input unit distributions in time
- measurement values and accuracy
- lexicalisation information
- top-down constraints, both module-internal and module-external
- activation spreading principles, in connectionistic implementations

With this number of empirically determined parametrisations to be considered in the parser configuration, it is intutively obvious that considerable experimentation within an elaborate empirical development environment is required before a final specification can be given; a parametrised parser tool prototype has been implemented for many of these features. Only a subset of the features has been considered so far in speech technological systems (cf. Görz 1988, Haton & al. 1991, Sagerer 1990).

Traditional top-down parser modules have a basic iterative algorithm structure with PREDICT-SCAN-COMPLETE steps (a bottom-up parser may be considered to represent the null hypothesis for PREDICT, effectively an unconstrained 'predict anything' strategy). Although they are generally treated as sequential steps, and a minimal sequentiality clearly must be retained, there is much potential overlap in activity between the steps, allowing some parallelisation in principle. In cognitive terms, these three steps may represent local functions corresponding to global access, selection, and integration functions in speech recognition as suggested by Marslen-Wilson (1987).

Any active chart parser will fulfil point (2), and the PREDICT step will incorporate PREDICT information for T_{cat} and T_{str} in the word time domain from higher level contexts, in order to reduce the local search space. Earley's algorithm with its combination of top-down (active) and left-right incremental processing will fulfil point (1). In Earley's original algorithm, only the acceptor algorithm is incremental; the parse extractor presupposes complete analysis. However, there is an elementary modification of the COMPLETE step for incremental parsing (i.e. insertion of the completed structure of an item into the COMPLETED list of the dominating item).

Point (3) can be solved by PREDICTing a new start for the parse domain in question (here: word constituents, morphs and prosodies), at each cycle of the algorithm, and by emitting any syntactically or semantically relevant COMPLETED sub-constituent on completion for processing incrementally in semantics and at the higher levels, rather than waiting for maximal analysis of the word domain. Further generalisations of the PREDICT algorithm can be considered. Point (4) is more difficult, but can in principle be reduced to point (2), with an additional problem of 'bridging the gap' by top-down interpolation, constraint relaxation (cf. Langer 1990) or by analysing noise types.

Point (5) is the hard SCAN task of skipping over the 'signal-symbol barrier': this involves mapping intervals defined in terms of T_{utt} on to virtual interval borders (represented by chart vertices or nodes) in T_{str} (cf. Carson-Berndsen 1992; cf. also Chien & al. 1990 for a more *ad hoc* solution), and defining higher level units in terms of precedence and overlap relations (cf. Carson-Berndsen's event-based approach, 1991). Point (6) is a difficult, but fairly conventional speech analysis problem (cf. Görz 1988).

Procedural linguistic design features for a prosodic morphology or word prosody parser are thus much more complex in task and domain oriented functional terms than the design features for traditional WL parser algorithms, whatever the formal complexity of their core algorithm (which defines T_{alg}) may be. These prosodic processor design features can, on the basis of current evidence, also be applied to the higher levels, though different prosodic and semantic properties at higher levels determine significant differences in information types and core algorithms.

6 Conclusion

A close examination of the linguistic design features underlying SL system architectures showed a wide range of parameters which require different settings in SL systems than in WL or SFE systems. Within this framework, a set of detailed design criteria were formulated for a prosodic parser; the parser was initially specified for the time domain of word prosody; in principle, these criteria also apply for prosodic parsers over the longer time domains. An experimental version of the parser has been implemented within the linguistic word modelling section of the ASL-Nord project, and is currently undergoing tests.

It may be suggested on the basis of this study that a detailed examination of functional linguistic criteria is not simply of marginal theoretical interest for the development of multi-variety, including multilingual SL systems, but that it is a prerequisite as a source of combinatorial constraints on the complex language models required for future SL systems. Further, the open empirical problems in the SL context create considerable problems of data and knowledge acquisition logistics, and require sophisticated tools for linguistic and phonetic knowledge acquisition, and extensive cooperation throughout the computational and linguistic community at the international level. Finally, over and above task specific demands, the criterion of the generalisability of results suggests that it also makes economic and scientific sense to take into account a wide range of resources and at the multilingual level, including general and theoretical issues in linguistic typology. These in turn will, like the Time Type issues discussed in this paper, provide information about further constraints on the variability of spoken language.

7 References

Bird, S., E. Klein (1990). Phonological events. *Journal of Linguistics* 26:33-56.

Bleiching, D. (1991). *Default-Hierarchien in der deutschen Wortprosodie*. ASL-TR-19-91/UBI.

Carson-Berndsen, J. (1991). *Ereignisstrukturen für phonologisches Parsen*. ASL-TR-9-91/UBI.

Carson-Berndsen, J. (1992). *An Event-based Phonotactics for German*. ASL-TR-29-92/UBI.

Chao, Y-R. (1968). *Language and Symbolic Systems*. Cambridge, CUP.

Chien, L-F., K.J. Chen, L-S. Lee (1990). An augmented chart data structure with efficient word lattice parsing scheme in speech recognition applications. *Proceedings, COLING 90, Helsinki*, Vol. 2: 60-65. Helsinki.

Gibbon, D. (1981). Idiomaticity and functional variation. A case study of international amateur radio talk. In *Language and Society* 10, 1981:21-42.

Gibbon, D. (1985). Context and variation in two-way radio discourse. In C.Ferguson, ed. *Discourse Processes* 8:395-419.

Gibbon, D. (1987). Prosodic Parsing. Plenary Lecture, Intonation Symposium (2 August 1987), 11th International Congress of Phonetic Sciences, 1-7 August 1987, Tallin, Estonia.

Gibbon, D. (1990). Prosodic Association by Template Inheritance. In: Gazdar, G. & W. Daelemans, eds., *Proc. Int. Workshop on Inheritance in Natural Language Processing*. Tilburg.

Gibbon, D. (1992). Language and software, or: Fritzl's Quest. In: Floyd, C. & H. Züllighoven, eds. *Software Development and Reality Construction*. Berlin, Springer.

Gibbon, D., G. Braun (1988). The PSI/PHI model of prosodic parsing. In *Proceedings, COLING 88, Budapest*.

Gibbon, D., H. Richter, eds. (1984). *Intonation, Accent and Rhythm. Studies in Discourse Phonology*. Berlin, de Gruyter.

Görz, G. (1988). *Strukturanalyse natürlicher Sprache*. Bonn, Addison-Wesley.

Haton, J.-P., J.-M. Pierrel, G. Perennou, J. Caelen, J.L. Gauvain (1991). *Reconnaissance automatique de la parole*. Paris, Dunod Informatique.

Hockett, C.F. (1958). *A Course in Modern Linguistics*. Toronto, Macmillan.

Jakobson, R. (1960). Linguistics and Poetics: Closing Statement. In T. Sebeok, ed., *Style in Language*. Cambridge, Mass., MIT Press, 350-377.

Langer, H. (1990). Syntactic normalisation of spontaneous speech. *Proceedings, COLING 90, Helsinki*, Vol. 3: 180-184.

Magerman, D.M., M.P. Marcus (1991). Pearl: a probabilistic chart parser. *5th EACL Conference*, Berlin. 15-20.

Marslen-Wilson, W.D. (1987). Functional parallelism in spoken word recognition. *Cognition* 25.

Sagerer, G. (1990). *Automatisches Verstehen gesprochener Sprache*. Mannheim, B.I. Wissenschafts-Verlag.

Tillmann, H-G., P. Mansell (1980). *Phonetik: Lautsprachliche Zeichen, Sprachsignale und lautsprachlicher Kommunikationsprozeß*. Stuttgart, Klett-Cotta.