Layered Charts for Speech Translation

Jan W. Amtrup*

University of Hamburg, Computer Science Department,
Vogt-Kölln-Str. 30, D-22527 Hamburg, Germany
amtrup@informatik.uni-hamburg.de

Abstract. We are going to present an architecture for natural language processing systems especially designed for spontaneous speech. We introduce the notion of a layered chart for information separation and information hiding. Complex distributed systems can be built using this approach which do not rely on a central data structure or control model. We describe a typed feature formalism with appropriateness suitable for such systems by providing a compact and relocatable storage schema for feature structures. We set out an architecture for spontaneous speech translation that relies on partial parsing to minimize complex operations at an early syntactic stage, but rather carries out utterance integration after building up small scale syntactic objects. We adopt the notion of a variable depth of analysis by providing a module specialized to the translation of idioms and show how an global scoring schema for sub-paths in graphs can be used to block further operation on idioms already accounted for.

1 Introduction

In this paper, we are going to present an architecture for natural language processing systems especially designed to have advantages in the application to spontaneous speech. Properties of spontaneously produced spoken language include the frequent advent of hesitations, repairs and the heavy use of intonation and prosody. Moreover, utterances very often do not comply to standard syntax of written language, especially with regard to congruence or word order. Frequently, incomplete sentences constitute the meaning of a speakers' utterance.

It must be observed that human natural language comprehension and production is inherently incremental in nature. Concentrating on the analysis side of the field, this feature can be shown to be present at various levels. The cohort model of [Marslen-Wilson & Tyler, 1980] and others assigns this property to spoken word recognition. A competitive model of lexical access is assumed. Starting with the first 200 milliseconds of speech input, a set of lexical items is constructed which contains the words compatible with the input. This set is narrowed down while the input continues, with finally only one active word remaining, which is taken to be recognized.

Even on higher linguistic levels, there are examples of the incremental nature of human language processing. [Niv, 1993] introduces a principle to *avoid new subjects* which prefers object readings of noun phrases over subject readings if the NP at hand is not already known in the current discourse. This interaction between syntactic interpretation and discourse implies incremental syntactic processing. The same holds for the preference of certain words in a cohort that are strongly preferred given the context in the current utterance. Those are primed against other, acoustically equally matching lexical items [Zwitserlood, 1989].

* This research was partly funded by the Federal Ministry of Education, Science, Research and Technology (BMBF) in the framework of the VERBMOBIL Project under Grant 01 IV 101 A/O.

To accept incrementality as a fundamental paradigm for natural language processing with computers thus seems to be appropriate. But even if one does not follow this bionic approach, incrementality is a prerequisite for advanced NLP applications, e.g. dialogue systems that might interrupt a user during his or her turn or systems for simultaneous interpretation.

From an engineering point of view, incrementality leads to tremendous difficulties during processing. Consider word graphs as the interface between speech recognition and linguistic processing. A word graph is a directed, acyclic, connected graph whose vertices denote points in time and whose edges represent hypotheses about which word was spoken during the time covered by the edge. Several different hypotheses may be active at once, an acoustic score is used as a measure to determine how well the input signal and the word model within the recognizer match (cf. [Oerder & Ney, 1993] for a description of the suitability of word graphs as interface between speech recognition and linguistic analysis, and [Amtrup et al., 1997] for an evaluation of size measures for word graphs from the language-processing perspective).

If the word graphs are constructed incrementally, they are only left-connected, i.e. they contain paths that cannot be pursued further in the future. The result is that these graphs contain a lot more word hypotheses than those where dead ends were removed during a backward pass. Our tests show that incremental word graphs can be ten times larger, averaging at about 480 edges per second of speech signal.

But pure size is not the only problem to account for. The search space for modules to follow speech recognition is far less structured than before. A syntactic parser for example is no longer able to start analysis with the acoustically best-rated utterance hypothesis, since a global optimum is not known in advance. This is a serious problem, mostly because of performance figures that degrade dramatically when losing this constraint.

What we propose here is to use a distributed system of modules, all operating on a data structure that is uniform throughout the application. In the next section, we will describe the design and properties of layered charts. Section 3 presents a formalism especially suited for distributed layered charts and section 4 sets out the architecture and mode of operation of an interpretation system designed around the data structure proposed here. Finally, in section 5 we contrast our approach with a similar one published earlier.

2 Layered Charts

The common grounds for all kind of information about an utterance to be processed is the temporal extension each hypothesis belongs to. Regardless of linguistic level, what is constructed by the various modules of the system are descriptions about a certain interval of the utterance.

To account for information that is added by processing components, one could use a uniform representation with a monotonic growth of descriptions during derivation[2]. But this carries the burden of unnecessarily processing data portions in modules that actually do not care about them — at least they have to be actively ignored. What one might want instead is the right amount of information present in each module while simultaneously maintaining a consistent Structure of the union of all data. This schema converges to an information hiding model which presents just as much information as is needed by each individual component.

Using a layered chart, the word graph given by a speech recognition module is the ground layer of representation. Each subsequent module may add information of its own. These additional hypotheses take the form of edges within the chart, just like the input words do, but

[2] One could even try to establish a uniform processing schema on all levels of descriptions as for example done by [Weisweber, 1992].

with a different status. Consider fig. 1 as an example word graph. Given a translation system as application, the word edges are processed by a syntactic parser which builds constituent hypotheses from them. Those constituents that satisfy certain target criteria (like being of utterance status) are transferred to semantics or transfer where translations are constructed.

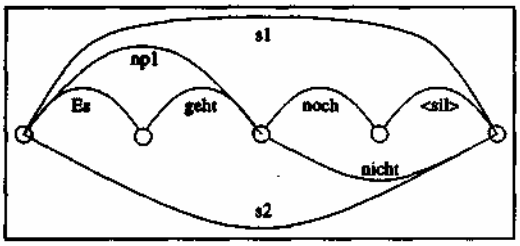

Figure 1. A toy word graph

A first benefit of operating with chart-like structures on all levels of representation is the avoidance of redundant work. Partial derivations can be reused several times, especially with highly compact word graphs: The edges for Es geht in fig. 1 are built only once, regardless of how many continuations exist starting from there.

But layered charts offer more than this advantage: The union of all edges in every component constitutes a directed, acyclic graph where edges with identical span describe identical portions of the word graph (albeit they may describe alternative partial paths through them). All components share the temporal structure of the input utterance at hand. While not strictly necessary for conventional, pipelined system architectures, this property is strongly advantageous for incremental architectures with a more complicated topology. Feedback loops can be easily established. If one component detects the inability to further operate on a partial analysis delivered from another component, it may relate this information towards the originating module and thus enable it to modify its control structure. This is made more easy given the identity relation over edges. Furthermore, work done by different modules on parallel paths within the application can be combined.

Given the micro-structure of the word graph, it is also possible to ensure a consistent modeling of the probability of a derivation. Every hypothesis is assigned a score, starting with purely acoustic scores and language model scores, possibly extending to represent the probability of a syntactic derivation using stochastic grammars and further for more complex analyses. Using this scoring model and assuming a beam search operation in all modules of the system, individual hypotheses can even be rejected by assigning a penalty score in a component which decides an analysis not worth further examination.

An operational view to layered charts allows for the construction of a distributed system based on the separation of different information spaces. Thus, we are able to establish a decentralized control schema within a parallel system. Components work mostly autonomous on their respective part of the information present. If needed, communication takes place using a message-passing paradigm in order to establish an overall system performance in a cooperative way [Amtrup, 1997].

3 A Formalism for Layered Charts

The primary application we investigated using the approach of layered charts is interpreting of spontaneous speech in the domain of appointment scheduling. We follow the symbolic paradigm, resting on feature structures to encode linguistic information. Since we were implementing a distributed system, we aimed at a formalism which is capable of encoding feature structures as contiguous memory blocks. The primary advantage of this approach is the easy distribution of feature structures, which entails no overhead for encoding and decoding during communication. What is sent is effectively a stream of bytes, a direct copy of the memory image of the FS. All components share a common name space, all references within a FS are local to that FS.

The main theoretical drawback of this representation mechanism is the impossibility to share common substructures across several FSs which would result in lower complexity of unification; a detailed analysis of the effects of this decision with regard to overall system performance has not been done yet[3].

The formalism we developed models typed feature structures with appropriateness using an array representation where each element consists of a tag field which denotes the type of the node in the feature graph and a data field that holds the content. This kind of modeling bears much resemblance to approaches like [Carpenter & Qu, 1995, Wintner & Francez, 1995] with the main difference being that we do not attach a procedural interpretation as an abstract machine to the FS. The design of FSs ensures hat, once they are created, no reorganization in memory is necessary, neither for further unification, nor for transportation to a different module.

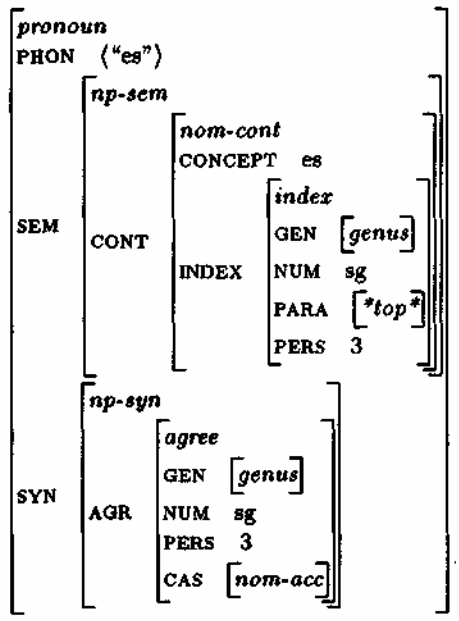

Figure 2. A feature structure for the pronoun "es"

Table 1. Feature structure as array

Node No.	Node Type	Node Value	Desc.
1	TYPE	pronoun	Topmost type of feature structure
2	FEAT	5	Pointer to feature phon
3	FEAT	24	Pointer to feature sem
4	FEAT	11	Pointer to feature syn
5	LIST	0	Feature phon is a list
6	FIRST	8	Pointer to list head
7	REST	10	Pointer to list tail
8	STRING	3	String of length 3
9	1702035456	0	"es"
10	NIL	0	Nil
11	TYPE	np-syn	The syntax part is nominal
12	FEAT	16	Pointer to feature agr
13	FEAT	23	Pointer to feature cas
14	EMPTY	0	Feature dtrs is empty
15	EMPTY	0	Feature subcat is empty
16	TYPE	agree	Type agree
17	FEAT	22	Pointer to feature gen
18	FEAT	21	Pointer to feature num
19	FEAT	20	Pointer to feature per
20	ATOM	3	3rd person
21	ATOM	sg	Singular
22	TYPE	genus	Under-specified gender
23	FEAT	nom-acc	Under-specified case
24	TYPE	np-sem	The semantics part is nominal
25	FEAT	26	Pointer to feature cont
26	TYPE	nom-cont	Semantic contents
27	FEAT	31	Pointer to feature concept
28	EMPTY	0	Feature def is empty
29	FEAT	32	Pointer to feature index
30	EMPTY	0	Feature mods is empty
31	ATOM	es	Main concept
32	TYPE	index	The index
33	FEAT	40	Pointer to feature gen
34	FEAT	39	Pointer to feature num
35	FEAT	37	Pointer to feature para
36	FEAT	38	Pointer to feature pers
37	TYPE	*top*	Under-specified parameter*
38	ATOM	3	3rd person
39	ATOM	sg	Singular
40	TYPE	genus	Under-specified gender

The inventory of descriptions in FSs includes atoms, strings, lists (which are directly encoded, not using a the type hierarchy) and complex types. The description of complex types reserves space for pointers to all features appropriate for a given type (cf. [Wintner & Francez, 1995]), so complex type descriptions can be reused without the need of additional copying. We model well-typed feature structures (cf. [Carpenter, 1992, p. 95]), which means that feature value pointers may be blank. Figure 2 shows the lexical entry for the German pronoun **es** (it), table 1 the resulting array representation in internal form.

[3] A theoretical investigation would, of course, result in a worse complexity class; but this account would neglect time necessary for transport of messages, encoding, decoding, etc.. In this case, an empirical study would be more appropriate.

Currently, we are not modeling disjunction or negation, although the representation is prepared for that. Instead, we added the possibility to compute function values of arbitrary C++ functions and use them as values of a feature. The functions are not called automatically during unification, but the user has to explicitly carry out function evaluation to replace the expressions with the values they render. At the moment we are using functions mainly for the concatenation of lists — e.g. to compute the PHON-feature of complex constituents.

4 Speech Translation with Layered Charts

After having presented the general concepts of a layered chart and described a formalism suitable for such a distributed system, we continue with a description of an actual application built using these tools. The architecture of our system is set out in fig. 3. A distributed layered chart is the central data structure, which is used and augmented by various modules, depicted with dashed lines. The subset of possible communication links which is needed in this individual case is made up of directional arrows between components. As we focus on speech translation, a HMM-based recognizer is the first component to become active during the users' input. It incrementally builds up a left-connected word graph and delivers the word hypotheses as soon as the are computed.

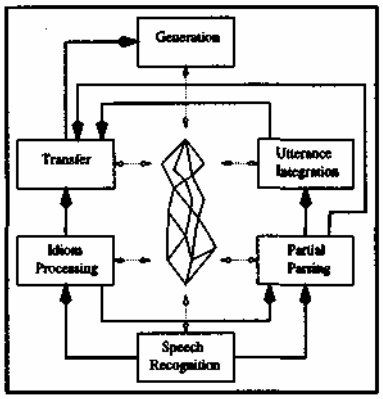

Figure 3. The architecture of the translation system

Table 2. Results of standard and partial parsing

Parm	Meas	T	NIG	IG	PP
Agenda tasks	#	2063	16238	86922	8591
Chart nodes	#	8	50	484	484
Chart edges	#	654	2418	4982	909
Success edges	#	4	4	371	160
Unifications	#	831	3713	13848	1143
Analysis time	s	9.37	94.94	113.64	8.56

The nature of spontaneous speech — especially the wide range of syntactic phenomena not covered in full detail by a standard grammar for written language — suggests to use a partial parsing approach. That way, it is possible to extract small-sized constructions like noun phrases or adverbial phrases and to rely on a later stage to integrate these into verbal complexes or whole utterances. In order to explore the processing difficulties that arise when processing incremental word graphs for spontaneous speech, we carried out some preliminary experiments.

We built a bottom up, left corner parser that works incrementally from left to right. It uses a beam search at each node of the graph and is supported by a bigram language model. We wrote linguistic descriptions for one of the dialogues of our corpus. The type hierarchy used contains 345 type definitions, with an emphasis on syntactic information, but a less elaborate semantic contribution. The grammar had 80 rules, partly to describe date- and time-related expressions. 413 lexical entries constituted the lexicon.

We then evaluated the parser when operating on transcripts (T), non-incremental word graphs (NIG) and incremental word graphs (IG). Tab. 2 shows the results of this evaluation. Additionally, we show results for partial parsing (PP). These numbers were gathered by removing all rules from the grammar covering large-scale syntactic constructs and leaving in rules for NPs, ADVPs and such.

The results show that parsing graphs instead of transcripts takes about ten times longer. But one notices a drastic speedup when doing only a partial parse. In this case, parsing is in fact faster than for transcripts. A partial explanation for this effect is the reduced size of the grammar (65 rules instead of 80). A more crucial point, however, is the reduced complexity and applicability of certain rules. For full parsing in the strict incremental paradigm, one has to construct complement complexes from left to right before encountering a verb form that may subcategorize for them — at least for languages with a possible verb-last position like German. This is a real burden if there are many potential candidates (like NPs) in a graph.

The obvious solution to this problem — to wait for verbs before building up complement structure on the left — can not be easily integrated into the parser due to its strictly left-to-right incremental mode of operation. One would have to create different strategies for "normal" rules that are not allowed to trace back and for rules handling subcategorization that are. A cleaner way of dealing with complement selection and verbal subcategorization together with phenomena of other provenience (relatively free word order, abandoning of utterance parts not relevant in the domain etc.) seems to be the approach of partial parsing: First to build up small structures purely on syntactic grounds and to later integrate them using both syntactic and semantic knowledge.

This combination of partial parser and integrator is shown in fig. 3 on the right hand side. After having analyzed the structure of an utterance and computed the possible semantic contents, the results of this process are given to a transfer stage.

We propose a chart-based approach for transfer, too. Most of the transfer systems currently available operate in a fashion more suitable for language than for spontaneous speech. They normally consider a sentence as the primary unit of translation, for which a complete syntactic/semantic description is available before transfer operates. This linguistic object is then traversed top-down; most often breadth-first search is used on each layer of the structure.

In order to be applicable for speech applications, we argued elsewhere [Amtrup, 1995] that even the transfer module has to obey incrementality to fit into a speech translation system; with slight modifications on standard chart-parsing algorithms it is possible to apply most results gained from that field to transfer.

The transfer portion of the chart now becomes a separate level of the distributed layered chart in our proposal. There are different sources of information for the transfer module. All analyses generated by the partial parser are delivered here. This enables the generation of syntactically oriented translations of noun phrases or prepositional phrases etc. Next, after utterance integration being done, larger structures with certain semantic information (like case-role assignment) are transmitted from utterance integration to transfer. Supplied with this additional data, transfer results for complete paths through the word graph may be constructed. These are communicated to the generation module which produces a surface form for the translation utterance and hands it to the user.

It has been argued since long that translation systems would benefit if they abandoned the requirement of a single level of description for linguistic data. Evidence from human strategies of interpretation [Künzli & Moser-Mercer, 1995] suggests that some parts of source language utterances are treated automatically without being deeply analyzed by the interpreter. Analogously, a notion of a variable depth of analysis has been proposed, a mechanism that demands

the shallowest analysis possible [Weisweber & Hauenschild, 1990]. We take this approach partly into account by providing a separate module for idioms processing. This module receives the complete word graph and searches incrementally for sub-paths in the graph matching predefined idioms (in our domain mostly greetings and utterance parts necessary to continue dialogue flow). These subparts receive a translation within the idioms processor. The resulting structures are directly given to the transfer module which incorporates them into the current translation, if appropriate. Simultaneously, the sub-paths accounted for are blocked in other modules to prevent an unnecessary deep analysis on them. The partial parser is delivered a rough syntactic structure of the idiom at hand and assigns a penalty score to the word hypotheses in order to prevent them from being considered in further processing steps. Thus, idiom processing is a shortcut for a more detailed analysis done in partial parsing.

5 Related Work

There are several approaches to integral data structures for complex NLP systems consisting of a set of heterogeneous components with certain individual needs for data representation. The most widely used of them is the blackboard [Engelmore & Morgan, 1988], which provides an integral view at all data produced by each module. The main drawback, however, lies in the fact that a centralized control is maintained, which constrains the use of multiprocessor systems due to restricted memory bandwidth. Using distributed blackboards [Jagannathan et al., 1989, part II] might be a first step towards a direction applicable for our purposes. Of particular interest regarding the structure described in this paper is an approach named "Whiteboard" [Boitet & Seligman, 1994] which claims to apply incrementality and parallelism to NLP systems. Essentially, a central data structure is maintained in which all data pertaining to the current computation is stored. So called *managers* retrieve information from there and modify the data to be suitable for the purposes of the components of the system; thus, data formats may differ from the whiteboard and components. Lattices, i.e. directed, acyclic graphs, are chosen as representation formalism on the whiteboard, which is nearly identical to the principle of layered charts used here. The crucial difference between [Boitet & Seligman, 1994] and the approach discussed in this paper regards control and applicability: We favor a distributed control schema with a high degree of autonomous operation of individual components to avoid problems with a centralized data and control structure. We conceptually base computation on a common temporal structure but do not introduce a single central component storing each data item. Furthermore, while whiteboards can only be used to simulate incremental behavior of components, incrementality is one of the main design issues motivating our approach. We use a common formalism to store data on all linguistic levels above speech recognition.

6 Conclusion

We have presented layered charts, a distributed data structure motivated by the nature of spontaneous speech and the assumption that incrementality is a *sine qua non* for certain applications. Layered charts extend the notion of charts in that they are able to store different alternate hypotheses and information of different origin. They strongly support distributed systems without a central control paradigm and thus enable inter-modular parallelism. We described a typed feature formalism with appropriateness which fits into the environment by storing feature structures in a compact, position-independent manner, so that they can be transmitted without linearization overhead. Finally, we introduced an application for interpreting spontaneous

speech incrementally using distributed layered charts. We utilize a partial parsing approach to reduce complexity in syntax and to preserve strict left-to-right incrementality. Utterances are integrated with a slightly looser schema by e.g. allowing complement saturation to the left of a verb. A chart-based transfer approach with a variable depth of analysis is pursued. Idioms are treated without a syntactic mechanism; once found, the sub-paths on which they appear will not be considered again.

References

[Amtrup, 1995] Amtrup, Jan W. 1995. Chart-based Incremental Transfer in Machine Translation. In *Proceedings of the sixth International Conference on Theoretical and Methodological Issues in Machine Translation, TMI '95,* pages 188-195, Leuven, Belgium.

[Amtrup, 1997] Amtrup, Jan W. 1997. ICE: A Communication Environment for Natural Language Processing. In *Proceedings of the International Conference on Parallel and Distributed Processing Techniques and Applications (PDPTA97),* Las Vegas, NV.

[Amtrup et al., 1997] Amtrup, Jan W., Henrik Heine, and Uwe Jost. 1997. What's in a Word Graph — Evaluation and Enhancement of Word Lattices. In *Proceedings of the 5th European Conference on Speech Communication and Technology, Eurospeech 1997,* Rhodes, Greece.

[Boitet & Seligman, 1994] Boitet, Christian and Mark Seligman. 1994. The "Whiteboard" Architecture: A Way to Integrate Heterogeneous Components of NLP systems. In *COLING-94-: The 15th International Conference on Computational Linguistics,* Kyoto, Japan.

[Carpenter, 1992] Carpenter, Bob. 1992. *The Logic of Typed Feature Structures.* Tracts in Theoretical Computer Science. Cambridge: Cambridge University Press.

[Carpenter & Qu, 1995] Carpenter, Bob and Yan Qu. 1995. An Abstract Machine for Attribute-Value Logics. In *Proceedings of the fourth International Workshop on Parsing Technologies (IWPT'95),* pages 59-70, Prague.

[Engelmore & Morgan, 1988] Engelmore, Robert and Tony Morgan. 1988. *Blackboard Systems.* Reading, MA: Addison-Wesley Publishing Company.

[Jagannathan et al., 1989] Jagannathan, V., R. Dodhiawala, and L. Baum (eds.). 1989. *Blackboard Architectures and Applications.* Boston, Ma.: Academic Press.

[Künzli & Moser-Mercer, 1995] Künzli, Alexander and Barbara Moser-Mercer. 1995. Human Strategies for Translation and Interpretation. In *K195-Activities: Workshops, Posters, Demos,* pages 304-306, Bielefeld.

[Marslen-Wilson & Tyler, 1980] Marslen-Wilson, William D. and Lorraine K. Tyler. 1980. The Temporal Structure of Spoken Language Understanding. *Cognition,* 8:1-71.

[Niv, 1993] Niv, Michael. 1993. *A Computational Model of Syntactic Processing: Ambiguity Resolution from Interpretation.* Ph.D. thesis, Univ. of Pennsylvania.

[Oerder & Ney, 1993] Oerder, Martin and Hermann Ney. 1993. Word Graphs: An Efficient Interface Between Continuous-Speech Recognition and Language Understanding. In *Proceedings of the 1994 IEEE International Conference on Acoustics, Speech & Signal Processing, ICASSP,* Minneapolis, MN.

[Weisweber, 1992] Weisweber, Wilhelm. 1992. Term-Rewriting as a Basis for a Uniform Architecture in Machine Translation. In *COLING-92: The 15th International Conference on Computational Linguistics,* pages 777-783, Nantes, France.

[Weisweber & Hauenschild, 1990] Weisweber, Wilhelm and Christa Hauenschild. 1990. A Model of Multi-Level Transfer for Machine Translation and its Partial Realization. KIT-Report 77, Technical University of Berlin.

[Wintner & Francez, 1995] Wintner, Shuly and Nissim Francez. 1995. Abstract Machine for Typed Feature Structures. In *Proceedings of the 5th Workshop on Natural Language Understanding and Logic Programming,* Lisbon, Spain.

[Zwitserlood, 1989] Zwitserlood, P. 1989. The Locus of Effects of Sentential-Semantic Context in Spoken-Word Processing. *Cognition,* 32:25-64.