

# Finite-state Multimodal Parsing and Understanding

**Michael Johnston**

AT&T Labs - Research  
Shannon Laboratory, 180 Park Ave  
Florham Park, NJ 07932, USA  
johnston@research.att.com

**Srinivas Bangalore**

AT&T Labs - Research  
Shannon Laboratory, 180 Park Ave  
Florham Park, NJ 07932, USA  
srini@research.att.com

## Abstract

Multimodal interfaces require effective parsing and understanding of utterances whose content is distributed across multiple input modes. Johnston 1998 presents an approach in which strategies for multimodal integration are stated declaratively using a unification-based grammar that is used by a multidimensional chart parser to compose inputs. This approach is highly expressive and supports a broad class of interfaces, but offers only limited potential for mutual compensation among the input modes, is subject to significant concerns in terms of computational complexity, and complicates selection among alternative multimodal interpretations of the input. In this paper, we present an alternative approach in which multimodal parsing and understanding are achieved using a weighted finite-state device which takes speech and gesture streams as inputs and outputs their joint interpretation. This approach is significantly more efficient, enables tight-coupling of multimodal understanding with speech recognition, and provides a general probabilistic framework for multimodal ambiguity resolution.

## 1 Introduction

Multimodal interfaces are systems that allow input and/or output to be conveyed over multiple different channels such as speech, graphics, and gesture. They enable more natural and effective interaction since different kinds of content can be conveyed in the modes to which they are best suited (Oviatt, 1997). Our specific concern here is with multimodal interfaces supporting input by speech, pen, and touch, but the approach we describe has far broader applicability. These interfaces stand to play a critical role in the ongoing migration of interaction from the desktop to wireless portable computing devices (PDAs, next-generation phones) that offer limited screen real estate, and other keyboard-less platforms such as public information kiosks.

To realize their full potential, multimodal interfaces need to support not just input from multiple modes, but synergistic multimodal utterances optimally distributed over the available modes (John-

ston et al., 1997). In order to achieve this, an effective method for integration of content from different modes is needed. Johnston (1998b) shows how techniques from natural language processing (unification-based grammars and chart parsing) can be adapted to support parsing and interpretation of utterances distributed over multiple modes. In that approach, speech and gesture recognition produce  $n$ -best lists of recognition results which are assigned typed feature structure representations (Carpenter, 1992) and passed to a multidimensional chart parser that uses a multimodal unification-based grammar to combine the representations assigned to the input elements. Possible multimodal interpretations are then ranked and the optimal interpretation is passed on for execution. This approach overcomes many of the limitations of previous approaches to multimodal integration such as (Bolt, 1980; Neal and Shapiro, 1991) (See (Johnston et al., 1997)(p. 282)). It supports speech with multiple gestures, visual parsing of unimodal gestures, and its declarative nature facilitates rapid prototyping and iterative development of multimodal systems. Also, the unification-based approach allows for mutual compensation of recognition errors in the individual modalities (Oviatt, 1999).

However, the unification-based approach does not allow for tight-coupling of multimodal parsing with speech and gesture recognition. Compensation effects are dependent on the correct answer appearing in the  $n$ -best list of interpretations assigned to each mode. Multimodal parsing cannot directly influence the progress of speech or gesture recognition. The multidimensional parsing approach is also subject to significant concerns in terms of computational complexity. In the worst case, the multidimensional parsing algorithm (Johnston, 1998b) (p. 626) is exponential with respect to the number of input elements. Also this approach does not provide a natural framework for combining the probabilities of speech and gesture events in order to select among multiple competing multimodal interpretations. Wu et.al. (1999) present a statistical approach for selecting among multiple possible combinations of speech

and gesture. However, it is not clear how the approach will scale to more complex verbal language and combinations of speech with multiple gestures.

In this paper, we propose an alternative approach that addresses these limitations: parsing, understanding, and integration of speech and gesture are performed by a single finite-state device. With certain simplifying assumptions, multidimensional parsing and understanding with multimodal grammars can be achieved using a weighted finite-state automaton (FSA) running on three tapes which represent speech input (words), gesture input (gesture symbols and reference markers), and their combined interpretation. We have implemented our approach in the context of a multimodal messaging application in which users interact with a company directory using synergistic combinations of speech and pen input; a multimodal variant of VPQ (Buntschuh et al., 1998). For example, the user might say `email this person` and `this person` and gesture with the pen on pictures of two people on a user interface display. In addition to the user interface client, the architecture contains speech and gesture recognition components which process incoming streams of speech and electronic ink, and a multimodal language processing component (Figure 1).

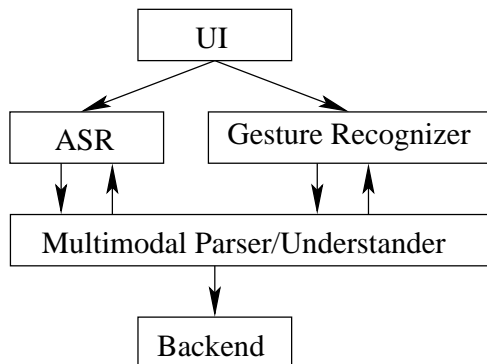


Figure 1: Multimodal architecture

Section 2 provides background on finite-state language processing. In Section 3, we define and exemplify multimodal context-free grammars (MCFGs) and their approximation as multimodal FSAs. We describe our approach to finite-state representation of meaning and explain how the three-tape finite state automaton can be factored out into a number of finite-state transducers. In Section 4, we explain how these transducers can be used to enable tight-coupling of multimodal language processing with speech and gesture recognition.

## 2 Finite-state Language Processing

Finite-state transducers (FST) are finite-state automata (FSA) where each transition consists of an input and an output symbol. The transition is traversed if its input symbol matches the current symbol in the input and generates the output symbol associated with the transition. In other words, an FST can be regarded as a 2-tape FSA with an input tape from which the input symbols are read and an output tape where the output symbols are written.

Finite-state machines have been extensively applied to many aspects of language processing including, speech recognition (Pereira and Riley, 1997; Riccardi et al., 1996), phonology (Kaplan and Kay, 1994), morphology (Koskenniemi, 1984), chunking (Abney, 1991; Joshi and Hopely, 1997; Bangalore, 1997), parsing (Roche, 1999), and machine translation (Bangalore and Riccardi, 2000).

Finite-state models are attractive mechanisms for language processing since they are (a) efficiently learnable from data (b) generally effective for decoding and (c) associated with a calculus for composing machines which allows for straightforward integration of constraints from various levels of language processing. Furthermore, software implementing the finite-state calculus is available for research purposes (Mohri et al., 1998). Another motivation for our choice of finite-state models is that they enable tight integration of language processing with speech and gesture recognition.

## 3 Finite-state Multimodal Grammars

Multimodal integration involves merging semantic content from multiple streams to build a joint interpretation for a multimodal utterance. We use a finite-state device to parse multiple input streams and to combine their content into a single semantic representation. For an interface with  $n$  modes, a finite-state device operating over  $n+1$  tapes is needed. The first  $n$  tapes represent the input streams and  $n+1$  is an output stream representing their composition. In the case of speech and pen input there are three tapes, one for speech, one for pen gesture, and a third for their combined meaning.

As an example, in the messaging application described above, users issue spoken commands such as `email this person` and that `organization` and gesture on the appropriate person and organization on the screen. The structure and interpretation of multimodal commands of this kind can be captured declaratively in a multimodal context-free grammar. We present a fragment capable of handling such commands in Figure 2.

S → V NP ε:ε:]	NP → DET N
CONJ → and:ε:	NP → DET N CONJ NP
V → email:ε:email([	DET → this:ε:ε
V → page:ε:page([	DET → that:ε:ε
N → person:G <sub>p</sub> :person(	ENTRY
N → organization:G <sub>o</sub> :org(	ENTRY
N → department:G <sub>d</sub> :dept(	ENTRY
ENTRY → ε:e <sub>1</sub> :e <sub>1</sub> ε:ε:]	
ENTRY → ε:e <sub>2</sub> :e <sub>2</sub> ε:ε:]	
ENTRY → ε:e <sub>3</sub> :e <sub>3</sub> ε:ε:]	
ENTRY → ...	

Figure 2: Multimodal grammar fragment

The non-terminals in the multimodal grammar are atomic symbols. The multimodal aspects of the grammar become apparent in the terminals. Each terminal contains three components  $W:G:M$  corresponding to the  $n + 1$  tapes, where  $W$  is for the spoken language stream,  $G$  is the gesture stream, and  $M$  is the combined meaning. The epsilon symbol is used to indicate when one of these is empty in a given terminal. The symbols in  $W$  are words from the speech stream. The symbols in  $G$  are of two types. Symbols like  $G_o$  indicate the presence of a particular kind of gesture in the gesture stream, while those like  $e_1$  are used as references to entities referred to by the gesture (See Section 3.1). Simple deictic pointing gestures are assigned semantic types based on the entities they are references to.  $G_p$  represents a gestural reference to a person on the display,  $G_o$  to an organization, and  $G_d$  to a department. Compared with a feature-based multimodal grammar, these types constitute a set of atomic categories which make the relevant distinctions for gesture events predicting speech events and vice versa. For example, if the gesture is  $G_p$  then phrases like *this person* and *him* are preferred speech events and vice versa. These categories also play a role in constraining the semantic representation when the speech is underspecified with respect to semantic type (e.g. *email this one*). These gesture symbols can be organized into a type hierarchy reflecting the ontology of the entities in the application domain. For example, there might be a general type  $G$  with subtypes  $G_o$  and  $G_p$ , where  $G_p$  has subtypes  $G_{pm}$  and  $G_{pf}$  for male and female.

A multimodal CFG (MCFG) can be defined formally as quadruple  $\langle N, T, P, S \rangle$ .  $N$  is the set of nonterminals.  $P$  is the set of productions of the form  $A \rightarrow \alpha$  where  $A \in N$  and  $\alpha \in (N \cup T)^*$ .  $S$  is the start symbol for the grammar.  $T$  is the set of terminals of the form  $(W \cup \varepsilon) : (G \cup \varepsilon) : M^*$  where  $W$  is the vocabulary of speech,  $G$  is the vocabulary of  $\text{gesture} = \text{GestureSymbols} \cup \text{EventSymbols}$ ;

$\text{GestureSymbols} = \{G_p, G_o, G_{pf}, G_{pm}, \dots\}$  and a finite collections of  $\text{EventSymbols} = \{e_1, e_2, \dots, e_n\}$ .  $M$  is the vocabulary to represent meaning and includes event symbols ( $\text{EventSymbols} \subset M$ ).

In general a context-free grammar can be approximated by an FSA (Pereira and Wright 1997, Nederhof 1997). The transition symbols of the approximated FSA are the terminals of the context-free grammar and in the case of multimodal CFG as defined above, these terminals contain three components,  $W$ ,  $G$  and  $M$ . The multimodal CFG fragment in Figure 2 translates into the FSA in Figure 3, a three-tape finite state device capable of composing two input streams into a single output semantic representation stream.

Our approach makes certain simplifying assumptions with respect to temporal constraints. In multi-gesture utterances the primary function of temporal constraints is to force an order on the gestures. If you say *move this here* and make two gestures, the first corresponds to *this* and the second to *here*. Our multimodal grammars encode order but do not impose explicit temporal constraints. However, general temporal constraints between speech and the first gesture can be enforced before the FSA is applied.

### 3.1 Finite-state Meaning Representation

A novel aspect of our approach is that in addition to capturing the structure of language with a finite state device, we also capture meaning. This is very important in multimodal language processing where the central goal is to capture how the multiple modes contribute to the combined interpretation. Our basic approach is to write symbols onto the third tape, which when concatenated together yield the semantic representation for the multimodal utterance. It suits our purposes here to use a simple logical representation with predicates  $\text{pred}(\dots)$  and lists  $[a, b, \dots]$ . Many other kinds of semantic representation could be generated. In the fragment in Figure 2, the word *email* contributes  $\text{email}([$  to the semantics tape, and the list and predicate are closed when the rule  $S \rightarrow V NP \varepsilon:\varepsilon:]$  applies. The word *person* writes  $\text{person}([$  on the semantics tape.

A significant problem we face in adding meaning into the finite-state framework is how to represent all of the different possible specific values that can be contributed by a gesture. For deictic references a unique identifier is needed for each object in the interface that the user can gesture on. For example, if the interface shows lists of people, there needs to be a unique identifier for each person. As part of the composition process this identifier needs

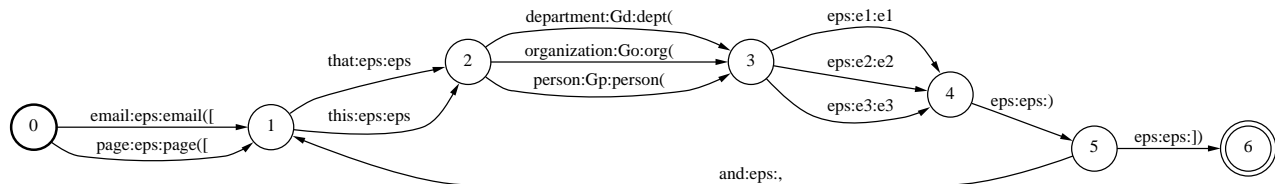


Figure 3: Multimodal three-tape FSA

to be copied from the gesture stream into the semantic representation. In the unification-based approach to multimodal integration, this is achieved by feature sharing (Johnston, 1998b). In the finite-state approach, we would need to incorporate all of the different possible IDs into the FSA. For a person with id *objid345* you need an arc  $\varepsilon:objid345:objid345$  to transfer that piece of information from the gesture tape to the meaning tape. All of the arcs for different IDs would have to be repeated everywhere in the network where this transfer of information is needed. Furthermore, these arcs would have to be updated as the underlying database was changed or updated. Matters are even worse for more complex pen-based data such as drawing lines and areas in an interactive map application (Cohen et al., 1998). In this case, the coordinate set from the gesture needs to be incorporated into the semantic representation. It might not be practical to incorporate the vast number of different possible coordinate sequences into an FSA.

Our solution to this problem is to store these specific values associated with incoming gestures in a finite set of buffers labeled  $e_1, e_2, e_3, \dots$  and in place of the specific content write in the name of the appropriate buffer on the gesture tape. Instead of having the specific values in the FSA, we have the transitions  $\varepsilon:e_1:e_1, \varepsilon:e_2:e_2, \varepsilon:e_3:e_3, \dots$  in each location where content needs to be transferred from the gesture tape to the meaning tape (See Figure 3). These are generated from the *ENTRY* productions in the multimodal CFG in Figure 2. The gesture interpretation module empties the buffers and starts back at  $e_1$  after each multimodal command, and so we are limited to a finite set of gesture events in a single utterance. Returning to the example *email this person and that organization*, assume the user gestures on entities *objid367* and *objid893*. These will be stored in buffers  $e_1$  and  $e_2$ . Figure 4 shows the speech and gesture streams and the resulting combined meaning.

The elements on the meaning tape are concatenated and the buffer references are replaced to yield

S:	email	this person	and	that organization
G:		$G_p e_1$		$G_o e_2$
M:	email([	person( $e_1$	,	org( $e_2$ ) ])

Figure 4: Messaging domain example

*email([person(objid367), org(objid893)])*. As more recursive semantic phenomena such as possessives and other complex noun phrases are added to the grammar the resulting machines become larger. However, the computational consequences of this can be lessened by lazy evaluation techniques (Mohri, 1997) and we believe that this finite-state approach to constructing semantic representations is viable for a broad range of sophisticated language interface tasks. We have implemented a sizeable multimodal CFG for VPQ (See Section 1): 417 rules and a lexicon of 2388 words.

### 3.2 Multimodal Finite-state Transducers

While a three-tape finite-state automaton is feasible in principle (Rosenberg, 1964), currently available tools for finite-state language processing (Mohri et al., 1998) only support finite-state transducers (FSTs) (two tapes). Furthermore, speech recognizers typically do not support the use of a three-tape FSA as a language model. In order to implement our approach, we convert the three-tape FSA (Figure 3) into an FST, by decomposing the transition symbols into an input component ( $G \times W$ ) and output component  $M$ , thus resulting in a function,  $\mathcal{T}:(G \times W) \rightarrow M$ . This corresponds to a transducer in which gesture symbols and words are on the input tape and the meaning is on the output tape (Figure 6). The domain of this function  $\mathcal{T}$  can be further carried to result in a transducer that maps  $\mathcal{R}:G \rightarrow W$  (Figure 7). This transducer captures the constraints that gesture places on the speech stream and we use it as a language model for constraining the speech recognizer based on the recognized gesture string. In the following section, we explain how  $\mathcal{T}$  and  $\mathcal{R}$  are used in conjunction with the speech recognition engine and gesture recognizer and interpreter to parse and inter-

pret multimodal input.

#### 4 Applying Multimodal Transducers

There are number of different ways in which multimodal finite-state transducers can be integrated with speech and gesture recognition. The best approach to take depends on the properties of the particular interface to be supported. The approach we outline here involves recognizing gesture first then using the observed gestures to modify the language model for speech recognition. This is a good choice if there is limited ambiguity in gesture recognition, for example, if the majority of gestures are unambiguous deictic pointing gestures.

The first step is for the gesture recognition and interpretation module to process incoming pen gestures and construct a finite state machine *Gesture* corresponding to the range of gesture interpretations. In our example case (Figure 4) the gesture input is unambiguous and the *Gesture* finite state machine will be as in Figure 5. If the gestural input involves gesture recognition or is otherwise ambiguous it is represented as a lattice indicating all of the possible recognitions and interpretations of the gesture stream. This allows speech to compensate for gesture errors and mutual compensation.

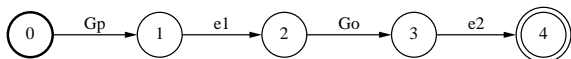


Figure 5: *Gesture* finite-state machine

This *Gesture* finite state machine is then composed with the transducer  $\mathcal{R}$  which represents the relationship between speech and gesture (Figure 7). The result of this composition is a transducer *GestLang* (Figure 8). This transducer represents the relationship between this particular stream of gestures and all of the possible word sequences that could co-occur with those gestures. In order to use this information to guide the speech recognizer, we then take a projection on the output tape (speech) of *GestLang* to yield a finite-state machine which is used as a language model for speech recognition (Figure 9). Using this model enables the gestural information to directly influence the speech recognizer's search. Speech recognition yields a lattice of possible word sequences. In our example case it yields the word sequence *email this person* and that organization (Figure 10). We now need to reintegrate the gesture information that we removed in the projection step before recognition. This is achieved by composing *GestLang* (Figure 8) with the result lattice from speech recognition (Figure 10), yielding transducer *GestSpeechFST* (Figure 11). This transducer contains

the information both from the speech stream and from the gesture stream. The next step is to generate the combined meaning representation. To achieve this *GestSpeechFST* ( $G : W$ ) is converted into an FSM *GestSpeechFSM* by combining output and input on one tape ( $G \times W$ ) (Figure 12). *GestSpeechFSM* is then composed with  $\mathcal{T}$  (Figure 6), which relates speech and gesture to meaning, yielding the result transducer *Result* (Figure 13). The meaning is read from the output tape yielding *email*([*person*( $e_1$ ), *org*( $e_2$ )]). We have implemented this approach and applied it in a multimodal interface to VPQ on a wireless PDA. In preliminary speech recognition experiments, our approach yielded an average of 23% relative sentence-level error reduction on a corpus of 1000 utterances (Johnston and Bangalore, 2000).

#### 5 Conclusion

We have presented here a novel approach to multimodal language processing in which spoken language and gesture are parsed and integrated by a single weighted finite-state device. This device provides language models for speech and gesture recognition and composes content from speech and gesture into a single semantic representation. Our approach is novel not just in addressing multimodal language but also in the encoding of semantics as well as syntax in a finite-state device.

Compared to previous approaches (Johnston et al., 1997; Johnston, 1998a; Wu et al., 1999) which compose elements from  $n$ -best lists of recognition results, our approach provides an unprecedented potential for mutual compensation among the input modes. It enables gestural input to dynamically alter the language model used for speech recognition. Furthermore, our approach avoids the computational complexity of multidimensional multimodal parsing and our system of weighted finite-state transducers provides a well understood probabilistic framework for combining the probability distributions associated with speech and gesture input and selecting among multiple competing multimodal interpretations. Since the finite-state approach is more lightweight in computational needs, it can more readily be deployed on a broader range of platforms.

In ongoing research, we are collecting a corpus of multimodal data in order to formally evaluate the effectiveness of our approach and to train weights for the multimodal finite-state transducers. While we have concentrated here on understanding, in principle the same device could be applied to multimodal

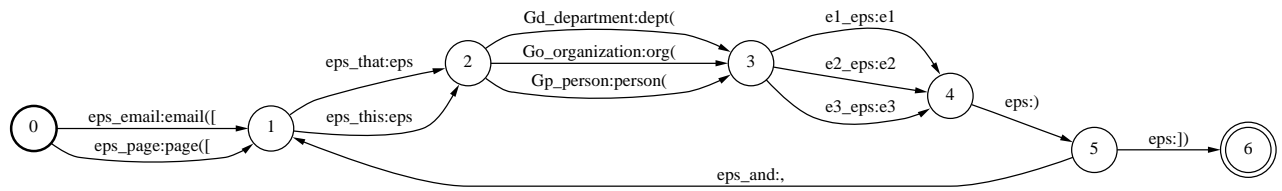


Figure 6: Transducer relating gesture and speech to meaning ( $T:(G \times W) \rightarrow M$ )

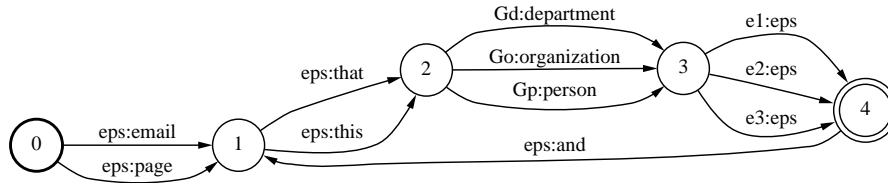


Figure 7: Transducer relating gesture and speech ( $\mathcal{R}:G \rightarrow W$ )

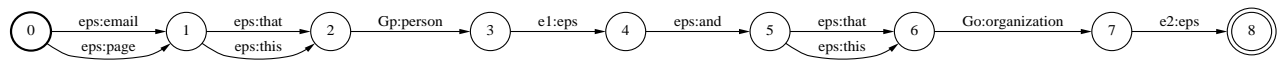


Figure 8: GestLang Transducer

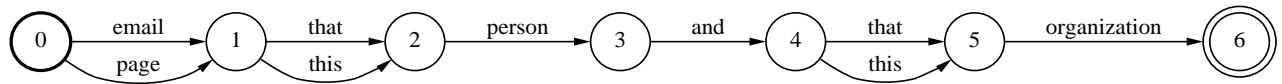


Figure 9: Projection of Output tape of GestLang Transducer

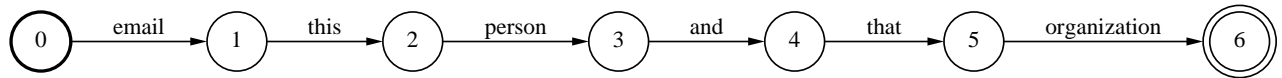


Figure 10: Result from speech recognizer

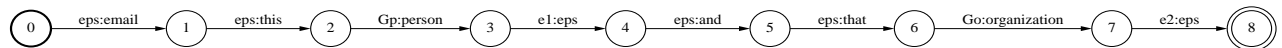


Figure 11: GestureSpeechFST

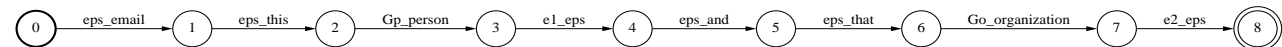


Figure 12: GestureSpeech FSM

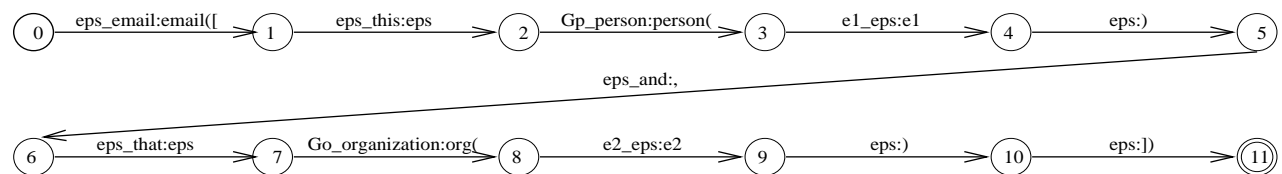


Figure 13: Result Transducer

generation which we are currently investigating. We are also exploring techniques to extend compilation from feature structures grammars to FSTs (Johnson, 1998) to multimodal unification-based grammars.

## References

- Steven Abney. 1991. Parsing by chunks. In Robert Berwick, Steven Abney, and Carol Tenny, editors, *Principle-based parsing*. Kluwer Academic Publishers.
- Srinivas Bangalore and Giuseppe Riccardi. 2000. Stochastic finite-state models for spoken language machine translation. In *Proceedings of the Workshop on Embedded Machine Translation Systems*.
- Srinivas Bangalore. 1997. *Complexity of Lexical Descriptions and its Relevance to Partial Parsing*. Ph.D. thesis, University of Pennsylvania, Philadelphia, PA, August.
- Robert A. Bolt. 1980. "put-that-there": voice and gesture at the graphics interface. *Computer Graphics*, 14(3):262–270.
- Bruce Buntschuh, C. Kamm, G. DiFabrizio, A. Abella, M. Mohri, S. Narayanan, I. Zeljkovic, R.D. Sharp, J. Wright, S. Marcus, J. Shaffer, R. Duncan, and J.G. Wilpon. 1998. Vpq: A spoken language interface to large scale directory information. In *Proceedings of ICSLP*, Sydney, Australia.
- Robert Carpenter. 1992. *The logic of typed feature structures*. Cambridge University Press, England.
- Philip R. Cohen, M. Johnston, D. McGee, S. L. Oviatt, J. Pittman, I. Smith, L. Chen, and J. Clow. 1998. Multimodal interaction for distributed interactive simulation. In M. Maybury and W. Wahlster, editors, *Readings in Intelligent Interfaces*. Morgan Kaufmann Publishers.
- Mark Johnson. 1998. Finite-state approximation of constraint-based grammars using left-corner grammar transforms. In *Proceedings of COLING-ACL*, pages 619–623, Montreal, Canada.
- Michael Johnston and Srinivas Bangalore. 2000. Tight-coupling of multimodal language processing with speech recognition. Technical report, AT&T Labs – Research.
- Michael Johnston, P.R. Cohen, D. McGee, S.L. Oviatt, J.A. Pittman, and I. Smith. 1997. Unification-based multimodal integration. In *Proceedings of the 35th ACL*, pages 281–288, Madrid, Spain.
- Michael Johnston. 1998a. Multimodal language processing. In *Proceedings of ICSLP*, Sydney, Australia.
- Michael Johnston. 1998b. Unification-based multimodal parsing. In *Proceedings of COLING-ACL*, pages 624–630, Montreal, Canada.
- Aravind Joshi and Philip Hopely. 1997. A parser from antiquity. *Natural Language Engineering*, 2(4).
- Ronald M. Kaplan and M. Kay. 1994. Regular models of phonological rule systems. *Computational Linguistics*, 20(3):331–378.
- K. K. Koskenniemi. 1984. *Two-level morphology: a general computation model for word-form recognition and production*. Ph.D. thesis, University of Helsinki.
- Mehryar Mohri, Fernando C. N. Pereira, and Michael Riley. 1998. *A rational design for a weighted finite-state transducer library*. Number 1436 in Lecture notes in computer science. Springer, Berlin ; New York.
- Mehryar Mohri. 1997. Finite-state transducers in language and speech processing. *Computational Linguistics*, 23(2):269–312.
- J. G. Neal and S. C. Shapiro. 1991. Intelligent multimedia interface technology. In J. W. Sullivan and S. W. Tyler, editors, *Intelligent User Interfaces*, pages 45–68. ACM Press, Addison Wesley, New York.
- Sharon L. Oviatt. 1997. Multimodal interactive maps: Designing for human performance. In *Human-Computer Interaction*, pages 93–129.
- Sharon L. Oviatt. 1999. Mutual disambiguation of recognition errors in a multimodal architecture. In *CHI '99*, pages 576–583. ACM Press, New York.
- Fernando C.N. Pereira and Michael D. Riley. 1997. Speech recognition by composition of weighted finite automata. In E. Roche and Schabes Y., editors, *Finite State Devices for Natural Language Processing*, pages 431–456. MIT Press, Cambridge, Massachusetts.
- Giuseppe Riccardi, R. Pieraccini, and E. Bocchieri. 1996. Stochastic Automata for Language Modeling. *Computer Speech and Language*, 10(4):265–293.
- Emmanuel Roche. 1999. Finite state transducers: parsing free and frozen sentences. In András Kornai, editor, *Extended Finite State Models of Language*. Cambridge University Press.
- A.L. Rosenberg. 1964. On n-tape finite state acceptors. *FOCS*, pages 76–81.
- Lizhong Wu, Sharon L. Oviatt, and Philip R. Cohen. 1999. Multimodal integration – a statistical view. *IEEE Transactions on Multimedia*, 1(4):334–341, December.