A Syntactic Framework for Speech Repairs and Other Disruptions

Mark G. Core and Lenhart K. Schubert
Department of Computer Science
University of Rochester
Rochester, NY 14627
mcore,schubert@cs.rochester.edu

Abstract

This paper presents a grammatical and processing framework for handling the repairs, hesitations, and other interruptions in natural human dialog. The proposed framework has proved adequate for a collection of human-human task-oriented dialogs, both in a full manual examination of the corpus, and in tests with a parser capable of parsing some of that corpus. This parser can also correct a pre-parser speech repair identifier resulting in a 4.8% increase in recall.

1 Motivation

The parsers used in most dialog systems have not evolved much past their origins in handling written text even though they may have to deal with speech repairs, speakers collaborating to form utterances, and speakers interrupting each other. This is especially true of machine translators and meeting analysis programs that deal with human-human dialog. Speech recognizers have started to adapt to spoken dialog (versus read speech). Recent language models (Heeman and Allen, 1997), (Stolcke and Shriberg, 1996), (Siu and Ostendorf, 1996) take into account the fact that word co-occurrences may be disrupted by editing terms[1] and speech repairs (*take the tanker I mean the boxcar*).

These language models detect repairs as they process the input; however, like past work on speech repair detection, they do not specify how speech repairs should be handled by the parser. (Hindle, 1983) and (Bear et al., 1992) performed speech repair identification in their parsers, and removed the corrected material (reparandum) from consideration. (Hindle, 1983) states that repairs are available for semantic analysis but provides no details on the representation to be used.

Clearly repairs should be available for semantic analysis as they play a role in dialog structure. For example, repairs can contain referents that are needed to interpret subsequent text: *have the engine take the oranges to Elmira, um, I mean, take them to Corning.* (Brennan and Williams, 1995) discusses the role of fillers (a type of editing term) in expressing uncertainty and (Schober, 1999) describes how editing terms and speech repairs correlate with planning difficultly. Clearly this is information that should be conveyed to higher-level reasoning processes. An additional advantage to making the parser aware of speech repairs is that it can use its knowledge of grammar and the syntactic structure of the input to correct errors made in pre-parser repair identification.

Like Hindle's work, the parsing architecture presented below uses phrase structure to represent the corrected utterance, but it also forms a phrase structure tree containing the reparandum. Editing terms are considered separate utterances that occur inside other utterances. So for the partial utterance, *take the ban- um the oranges*, three constituents would be produced, one for *um*, another for *take the ban-*, and a third for *take the oranges*.

Another complicating factor of dialog is

[1] Here, we define editing terms as a set of 30-40 words that signal hesitations (*um*) and speech repairs (*I mean*) and give meta-comments on the utterance (*right*).

the presence of more than one speaker. This paper deals with the two speaker case, but the principles presented should apply generally. Sometimes the second speaker needs to be treated independently as in the case of backchannels (*um-hm*) or failed attempts to grab the floor. Other times, the speakers interact to collaboratively form utterances or correct each other. The next step in language modeling will be to decide whether speakers are collaborating or whether a second speaker is interrupting the context with a repair or backchannel. Parsers must be able to form phrase structure trees around interruptions such as backchannels as well as treat interruptions as continuations of the first speaker's input.

This paper presents a parser architecture that works with a speech repair identifying language model to handle speech repairs, editing terms, and two speakers. Section 2 details the allowable forms of collaboration, interruption, and speech repair in our model. Section 3 gives an overview of how this model is implemented in a parser. This topic is explored in more detail in (Core and Schubert, 1998). Section 4 discusses the applicability of the model to a test corpus, and section 5 includes examples of trees output by the parser. Section 6 discusses the results of using the parser to correct the output of a preparser speech repair identifier.

2 What is a Dialog

From a traditional parsing perspective, a text is a series of sentences to be analyzed. An interpretation for a text would be a series of parse trees and logical forms, one for each sentence. An analogous view is often taken of dialog; dialog is a series of "utterances" and a dialog interpretation is a series of parse trees and logical forms, one for each successive utterance. Such a view either disallows editing terms, repairs, interjected acknowledgments and other disruptions, or else breaks semantically complete utterances into fragmentary ones. We analyze dialog in terms of a set of utterances covering all the words of the dialog. As explained below, utterances can be formed by more than one speaker and the words of two utterances may be interleaved.

We define an utterance here as a sentence, phrasal answer (to a question), editing term, or acknowledgment. Editing terms and changes of speaker are treated specially. Speakers are allowed to interrupt themselves to utter an editing term. These editing terms are regarded as separate utterances. At changes of speaker, the new speaker may: 1) add to what the first speaker has said, 2) start a new utterance, or 3) continue an utterance that was left hanging at the last change of speaker (e.g., because of an acknowledgment). Note that a speaker may try to interrupt another speaker and succeed in uttering a few words but then give up if the other speaker does not stop talking. These cases are classified as incomplete utterances and are included in the interpretation of the dialog.

Except in utterances containing speech repairs, each word can only belong to one utterance. Speech repairs are intra-utterance corrections made by either speaker. The reparandum is the material corrected by the repair. We form two interpretations of an utterance with a speech repair. One interpretation includes all of the utterance up to the reparandum end but stops at that point; this is what the speaker started to say, and will likely be an incomplete utterance. The second interpretation is the corrected utterance and skips the reparandum. In the example, *you should take the boxcar I mean the tanker to Corning*; the reparandum is *the boxcar*. Based on our previous rules the editing term *I mean* is treated as a separate utterance. The two interpretations produced by the speech repair are the utterance, *you should take the tanker to Corning*, and the incomplete utterance, *you should take the boxcar*.

3 Dialog Parsing

The modifications required to a parser to implement this definition of dialog are relatively straightforward. At changes of speaker, copies are made of all phrase hypotheses (arcs in a chart parser, for example) ending at the previous change of speaker. These copies are extended to the current change of speaker. We will use the term contribution (contr) here to refer to an uninterrupted sequence of words by one speaker (the words between speaker changes). In the example below, consider change of speaker (cos) 2. Copies of all phrase hypotheses ending at change of speaker 1 are extended to end at change of speaker 2. In this way, speaker A can form a phrase from contr-1 and contr-3 skipping speaker B's interruption, or contr-1, contr-2, and contr-3 can all form one constituent. At change of speaker 3, all phrase hypotheses ending at change of speaker 2 are extended to end at change of speaker 3 except those hypotheses that were extended from the previous change of speaker. Thus, an utterance cannot be formed from only contr-1 and contr-4. This mechanism implements the rules for speaker changes given in section 2: at each change of speaker, the new speaker can either build on the last contribution, build on their last contribution, or start a new utterance.

```
A: contr-1          contr-3
B:          contr-2           contr-4
cos        1        2        3
```

These rules assume that changes of speaker are well defined points of time, meaning that words of two speakers do not overlap. In the experiments of this paper, a corpus was used where word endings were time-stamped (word beginnings are unavailable). These times were used to impose an ordering; if one word ends before another it is counted as being before the other word. Clearly, this could be inaccurate given that words may overlap. Moreover, speakers may be slow to interrupt or may anticipate the first speaker and interrupt early. However, this approximation works fairly well as discussed in section 4.

Other parts of the implementation are accomplished through metarules. The term metarule is used because these rules act not on words but grammar rules. Consider the *editing term metarule*. When an editing term is seen[2], the metarule extends copies of all phrase hypotheses ending at the editing term over that term to allow utterances to be formed around it. This metarule (and our other metarules) can be viewed declaratively as specifying allowable patterns of phrase breakage and interleaving (Core and Schubert, 1998). This notion is different from the traditional linguistic conception of metarules as rules for generating new PSRs from given PSRs.[3] Procedurally, we can think of metarules as creating new (discontinuous) pathways for the parser's traversal of the input, and this view is readily implementable.

The repair metarule, when given the hypothetical start and end of a reparandum (say from a language model such as (Heeman and Allen, 1997)), extends copies of phrase hypotheses over the reparandum allowing the corrected utterance to be formed. In case the source of the reparandum information gave a false alarm, the alternative of not skipping the reparandum is still available.

For each utterance in the input, the parser needs to find an interpretation that starts at the first word of the input and ends at the last word.[4] This interpretation may have been produced by one or more applications of the repair metarule allowing the interpretation to exclude one or more reparanda. For each reparandum skipped, the parser needs to find an interpretation of what the user started to say. In some cases, what the user started to say is a complete constituent: *take*

[2] The parser's lexicon has a list of 35 editing terms that activate the editing term metarule.

[3] For instance, a traditional way to accommodate editing terms might be via a metarule, X -> Y Z ==> X -> Y editing-term Z, where X varies over categories and Y and Z vary over sequences of categories. However, this would produce phrases containing editing terms as constituents, whereas in our approach editing terms are separate utterances.

[4] In cases of overlapping utterances, it will take multiple interpretations (one for each utterance) to extend across the input.

the oranges I mean take the bananas. Otherwise, the parser needs to look for an incomplete interpretation ending at the reparandum end. Typically, there will be many such interpretations; the parser searches for the longest interpretations and then ranks them based on their category: UTT > S > VP > PP, and so on. The incomplete interpretation may not extend all the way to the start of the utterance in which case the process of searching for incomplete interpretations is repeated. Of course the search process is restricted by the first incomplete constituent. If, for example, an incomplete PP is found then any additional incomplete constituent would have to expect a PP.

Figure 1 shows an example of this process on utterance 62 from TRAINS dialog d92a-1.2 (Heeman and Allen, 1995). Assuming perfect speech repair identification, the repair metarule will be fired from position 0 to position 5 meaning the parser needs to find an interpretation starting at position 5 and ending at the last position in the input. This interpretation (the corrected utterance) is shown under the words in figure 1. The parser then needs to find an interpretation of what the speaker started to say. There are no complete constituents ending at position 5. The parser instead finds the incomplete constituent `ADVBL -> adv • ADVBL`. Our implementation is a chart parser and accordingly incomplete constituents are represented as arcs. This arc only covers the word *through* so another arc needs to be found. The arc `S -> S • ADVBL` expects an `ADVBL` and covers the rest of the input, completing the interpretation of what the user started to say (as shown on the top of figure 1). The editing terms are treated as separate utterances via the editing term metarule.

4 Verification of the Framework

To test this framework, data was examined from 31 TRAINS 93 dialogs (Heeman and Allen, 1995), a series of human-human problem solving dialogs in a railway transportation domain.[5] There were 3441 utterances,[6] 19189 words, 259 examples of overlapping utterances, and 495 speech repairs.

The framework presented above covered all the overlapping utterances and speech repairs with three exceptions. Ordering the words of two speakers strictly by word ending points neglects the fact that speakers may be slow to interrupt or may anticipate the original speaker and interrupt early. The latter was a problem in utterances 80 and 81 of dialog d92a-1.2 as shown below. The numbers in the last row represent times of word endings; for example, *so* ends at 255.5 seconds into the dialog. Speaker *s* uttered the complement of *u*'s sentence before *u* had spoken the verb.

80 u:	so	the		total	is
81 s:			five		
	255.5	255.56	255.83	256	256.61

However, it is important to examine the context following:

82 s: that is right
s: okay
83 u: five
84 s: so total is five

The overlapping speech was confusing enough to the speakers that they felt they needed to reiterate utterances 80 and 81 in the next utterances. The same is true of the other two such examples in the corpus. It may be the case that a more sophisticated model of interruption will not be necessary if speakers cannot follow completions that lag or precede the correct interruption area.

5 The Dialog Parser Implementation

In addition to manually checking the adequacy of the framework on the cited TRAINS data, we tested a parser imple-

[5] Specifically, the dialogs were d92-1 through d92a-5.2 and d93-10.1 through d93-14.1

[6] This figure does not count editing term utterances nor utterances started in the middle of another speaker's utterance.

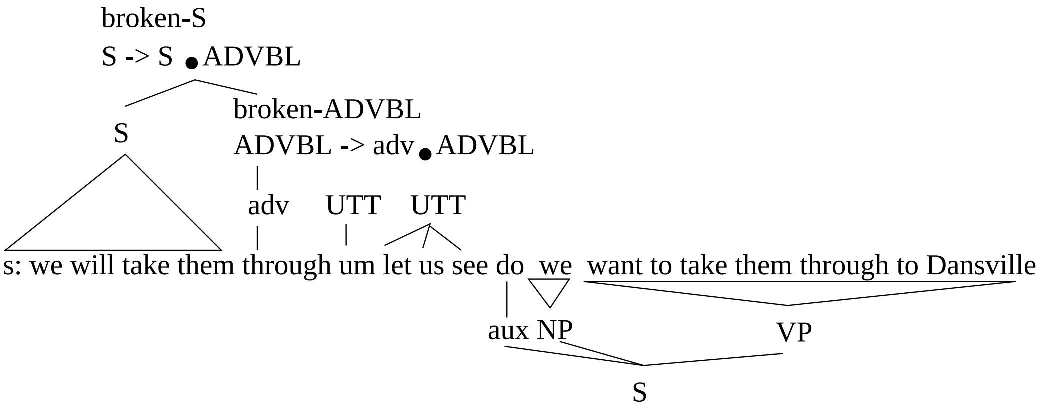

Figure 1: Utterance 62 of d92a-1.2

mented as discussed in section 3 on the same data. The parser was a modified version of the one in the TRIPS dialog system (Ferguson and Allen, 1998). Users of this system participate in a simulated evacuation scenario where people must be transported along various routes to safety. Interactions of users with TRIPS were not investigated in detail because they contain few speech repairs and virtually no interruptions.[7] But, the domains of TRIPS and TRAINS are similar enough to allow us run TRAINS examples on the TRIPS parser.

One problem, though, is the grammatical coverage of the language used in the TRAINS domain. TRIPS users keep their utterances fairly simple (partly because of speech recognition problems) while humans talking to each other in the TRAINS domain felt no such restrictions. Based on a 100-utterance test set drawn randomly from the TRAINS data, parsing accuracy is 62%[8] However, 37 of these utterances are one word long (*okay*, *yeah*, etc.) and 5 utterances were question answers (*two hours*, *in Elmira*); thus on interesting utterances, accuracy is 34.5%. Assuming perfect speech repair detection, only 125 of the 495 corrected speech repairs parsed.[9]

Of the 259 overlapping utterances, 153 were simple backchannels consisting only of editing terms (*okay*, *yeah*) spoken by a second speaker in the middle of the first speaker's utterance. If the parser's grammar handles the first speaker's utterance these can be parsed, as the second speaker's interruption can be skipped. The experiments focused on the 106 overlapping utterances that were more complicated. In only 24 of these cases did the parser's grammar cover both of the overlapping utterances. One of these examples, utterances utt39 and 40 from d92a-3.2 (see below), involves three independently formed utterances that overlap. We have omitted the beginning of s's utterance, *so that would be five a.m.* for space reasons. Figure 2 shows the syntactic structure of s's utterance (a relative clause) under the words of the utterance. u's two utterances are shown above the words of figure 2. The purpose of this figure is to show how interpretations can be formed around interruptions by another speaker and how these interruptions themselves form interpretations. The specific syntactic

[7]The low speech recognition accuracy encourages users to produce short, carefully spoken utterances leading to few speech repairs. Moreover, the system does not speak until the user releases the speech input button, and once it responds will not stop talking even if the user interrupts the response. This virtually eliminates interruptions.

[8]The TRIPS parser does not always return a unique utterance interpretation. The parser was counted as being correct if one of the interpretations it returned was correct. The usual cause of failure was the parser finding no interpretation. Only 3 failures were due to the parser returning only incorrect interpretations.

[9]In 19 cases, the parser returned interpretation(s) but they were incorrect but not included in the above figure.

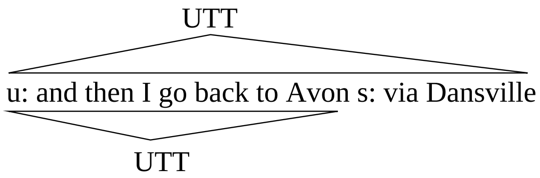

Figure 3: Utterances 132 and 133 from d92a-5.2

structure of the utterances is not shown. Typically, triangles are used to represent a parse tree without showing its internal structure. Here, polygonal structures must be used due to the interleaved nature of the utterances.

s: when it would get to bath
u: okay how about to dansville

Figure 3 is an example of a collaboratively built utterance, utterances 132 and 133 from d92a-5.2, as shown below. u's interpretation of the utterance (shown below the words in figure 3) does not include s's contribution because until utterance 134 (where u utters *right*) u has not accepted this continuation.

u: and then I go back to avon
s: via dansville

6 Rescoring a Pre-parser Speech Repair Identifier

One of the advantages of providing speech repair information to the parser is that the parser can then use its knowledge of grammar and the syntactic structure of the input to correct speech repair identification errors. As a preliminary test of this assumption, we used an older version of Heeman's language model (the current version is described in (Heeman and Allen, 1997)) and connected it to the current dialog parser. Because the parser's grammar only covers 35% of input sentences, corrections were only made based on global grammaticality.

The effectiveness of the language module without the parser on the testing corpus is shown in table 1.[10] The testing corpus consisted of TRAINS dialogs containing 541 repairs, 3797 utterances, and 20,069 words.[11] For each turn in the input, the language model output the n-best predictions it made (up to 100) regarding speech repairs, part of speech tags, and boundary tones.

The parser starts by trying the language model's first choice. If this results in an interpretation covering the input, that choice is selected as the correct answer. Otherwise the process is repeated with the model's next choice. If all the choices are exhausted and no interpretations are found, then the first choice is selected as correct. This approach is similar to an experiment in (Bear et al., 1992) except that Bear et al. were more interested in reducing false alarms. Thus, if a sentence parsed without the repair then it was ruled a false alarm. Here the goal is to increase recall by trying lower probability alternatives when no parse can be found.

The results of such an approach on the test corpus are listed in table 2. Recall increases by 4.8% (13 cases out of 541 repairs) showing promise in the technique of rescoring the output of a pre-parser speech repair identifier. With a more comprehensive grammar, a strong disambiguation system, and the current version of Heeman's language model, the results should get better. The drop in precision is a worthwhile tradeoff as the parser is never forced to accept posited repairs but is merely given the option of pursuing alternatives that include them.

Adding actual speech repair identification (rather than assuming perfect identification) gives us an idea of the performance improvement (in terms of parsing) that speech repair handling brings us. Of the 284 repairs correctly guessed in the augmented model, 79 parsed.[12] Out of 3797 utterances, this means that 2.1% of the time the parser would have failed without speech repair informa-

[10] Note, current versions of this language model perform significantly better.

[11] Specifically the dialogs used were d92-1 through d92a-5.2; d93-10.1 through d93-10.4; and d93-11.1 through d93-14.2. The language model was never simultaneously trained and tested on the same data.

[12] In 11 cases, the parser returned interpretation(s) but they were incorrect and not included in the above figure.

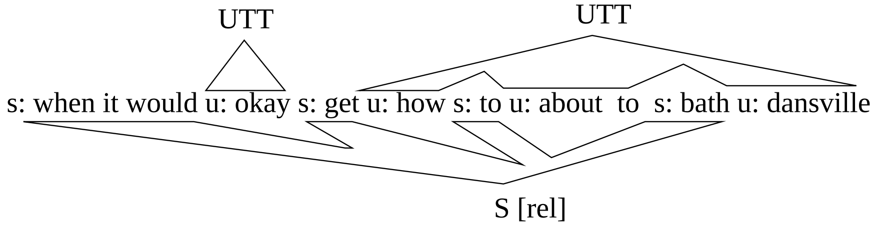

Figure 2: Utterances 39 and 40 of d92a-3.2

repairs correctly guessed	271
false alarms	215
missed	270
recall	50.09%
precision	55.76%

Table 1: Heeman's Speech Repair Results

repairs correctly guessed	284
false alarms	371
missed	257
recall	52.50%
precision	43.36%

Table 2: Augmented Speech Repair Results

tion. Although failures due to the grammar's coverage are much more frequent (38% of the time), as the parser is made more robust, these 79 successes due to speech repair identification will become more significant. Further evaluation is necessary to test this model with an actual speech recognizer rather than transcribed utterances.

7 Conclusions

Traditionally, dialog has been treated as a series of single speaker utterances, with no systematic allowance for speech repairs and editing terms. Such a treatment cannot adequately deal with dialogs involving more than one human (as appear in machine translation or meeting analysis), and will not allow single user dialog systems to progress to more natural interactions. The simple set of rules given here allows speakers to collaborate to form utterances and prevents an interruption such as a backchannel response from disrupting the syntax of another speaker's utterance. Speech repairs are captured by parallel phrase structure trees, and editing terms are represented as separate utterances occurring inside other utterances.

Since the parser has knowledge of grammar and the syntactic structure of the input, it can boost speech repair identification performance. In the experiments of this paper, the parser was able to increase the recall of a pre-parser speech identifier by 4.8%. Another advantage of giving speech repair information to the parser is that the parser can then include reparanda in its output and a truer picture of dialog structure can be formed. This can be crucial if a pronoun antecedent is present in the reparandum as in *have the engine take the oranges to Elmira, um, I mean, take them to Corning.* In addition, this information can help a dialog system detect uncertainty and planning difficultly in speakers.

The framework presented here is sufficient to describe the 3441 human-human utterances comprising the chosen set of TRAINS dialogs. More corpus investigation is necessary before we can claim the framework provides broad coverage of human-human dialog. Another necessary test of the framework is extension to dialogs involving more than two speakers.

Long term goals include further investigation into the TRAINS corpus and attempting full dialog analysis rather than experimenting with small groups of overlapping utterances. Another long term goal is to weigh the current framework against a purely robust parsing approach (Rosè and Levin, 1998), (Lavie, 1995) that treats out of vocabulary/grammar phenomena in the same way as editing terms and speech repairs. Robust parsing is critical to a parser such as the one described here which has a coverage of only 62% on fluent utterances. In our corpus, the speech repair to utterance ratio is 14%. Thus, problems due to the coverage of the grammar are more than twice as likely as speech repairs. However, speech repairs occur with enough frequency to warrant separate attention. Unlike grammar failures, repairs are generally signaled not only by ungrammaticality, but also by pauses, editing terms, parallelism, etc.; thus an approach specific to speech repairs should perform better than just using a robust parsing algorithm to deal with them.

Acknowledgments

This work was supported in part by National Science Foundation grants IRI-9503312 and 5-28789. Thanks to James Allen, Peter Heeman, and Amon Seagull for their help and comments on this work.

References

J. Bear, J. Dowding, and E. Shriberg. 1992. Integrating multiple knowledge sources for detection and correction of repairs in human-computer dialog. In *Proc. of the 30th annual meeting of the Association for Computational Linguistics (ACL-92)*, pages 56–63.

S. E. Brennan and M. Williams. 1995. The feeling of another's knowing: Prosody and filled pauses as cues to listeners about the metacognitive states of speakers. *Journal of Memory and Language*, 34:383–398.

M. Core and L. Schubert. 1998. Implementing parser metarules that handle speech repairs and other disruptions. In D. Cook, editor, *Proc. of the 11th International FLAIRS Conference*, Sanibel Island, FL, May.

G. Ferguson and J. F. Allen. 1998. TRIPS: An intelligent integrated problem-solving assistant. In *Proc. of the National Conference on Artificial Intelligence (AAAI-98)*, pages 26–30, Madison, WI, July.

P. Heeman and J. Allen. 1995. the TRAINS 93 dialogues. TRAINS Technical Note 94-2, Department of Computer Science, University of Rochester, Rochester, NY 14627-0226.

Peter A. Heeman and James F. Allen. 1997. Intonational boundaries, speech repairs, and discourse markers: Modeling spoken dialog. In *Proc. of the 35th Annual Meeting of the Association for Computational Linguistics*, pages 254–261, Madrid, July.

D. Hindle. 1983. Deterministic parsing of syntactic non-fluencies. In *Proc. of the 21st annual meeting of the Association for Computational Linguistics (ACL-83)*, pages 123–128.

A. Lavie. 1995. *GLR*: A Robust Grammar Focused Parser for Spontaneously Spoken Language*. Ph.D. thesis, School of Computer Science, Carnegie Mellon University, Pittsburgh, PA.

C. P. Rosè and L. S. Levin. 1998. An interactive domain independent approach to robust dialogue interpretation. In *Proc. of the 36th Annual Meeting of the Association for Computational Linguistics*, Montreal, Quebec, Canada.

M. Schober. 1999. Speech disfluencies in spoken language systems: A dialog-centered approach. In *NSF Human Computer Interaction Grantees' Workshop (HCIGW 99)*, Orlando, FL.

M.-h. Siu and M. Ostendorf. 1996. Modeling disfluencies in conversational speech. In *Proceedings of the 4rd International Conference on Spoken Language Processing (ICSLP-96)*, pages 386–389.

Andreas Stolcke and Elizabeth Shriberg. 1996. Statistical language modeling for speech disfluencies. In *Proceedings of the International Conference on Audio, Speech and Signal Processing (ICASSP)*, May.