ISCA Archive
http://www.isca-speech.org/archive

ITRW on Disfluency in
Spontaneous Speech (DiSS'01)
Edinburgh, Scotland, UK
August 29-31, 2001

How to Repair Speech Repairs in an End-to-End System

Jörg Spilker, Anton Batliner, Elmar Nöth

University of Erlangen-Nuremberg, Germany
{spilker,batliner,noeth}@informatik.uni-erlangen.de

Abstract

If automatic speech processing wants to deal with spontaneous speech, it has to deal with disfluencies in general and speech repairs in particular as well. The paper describes the processing of speech repairs in the VERBMOBIL system and discusses the special requirements of real-time systems. With respect to this criterion, the VERBMOBIL approach and its results are compared to other work. All these results are based more or less on the evaluation of a stand alone process, not integrated in a speech system. The ultimate goal is, of course, the use and the evaluation of the impact of such a repair process in a real-time, end-to-end system. An evaluation method based on this idea is presented and some preliminary results are given.

1. Introduction

A characteristic feature of spontaneous natural human-human dialogues are disfluencies. The more speech systems are intended to deal with natural dialogues, the more important becomes the problem of handling disfluencies and in particular speech repairs. There is no exact definition of the term "speech repair", but based on the evaluation of the German VERBMOBIL corpus,[1] speech repairs in our sense comprise the following four phenomena:

- in-word repairs
- modification repairs
- pivot constructions
- fresh starts

In an **in-word repair**, the speaker interrupts within a word and corrects a part of it. A typical example is the correction in *Termei-minkalender (app-ain-ointment calendar)*. **Modification repairs** correct part of the whole sentence, but do not change the syntactic construction. In contrast to other studies, we define **repetitions** as a special case of modification repairs, where the corrected part and the correction are identical. An example for a modification repair is the following sentence: *ja ist in Ordnung* Montag *<äh> Sonntag den fünften (yes it's okay* Monday *<uh> Sunday the fifth)*. In a **pivot construction** (anacoluthon), the syntax of a sentence changes from the initial construction to a different one, whereby one part of the sentence belongs to both constructions. One of the few examples we found is: *ich bin vom vierzehnten bis zwanzigsten Mai <äh> <hm> bin ich . . . (I am from the fourteenth to the twentieth of May <uh> <hm> I am . . .)*. The underlined term *vom vierzehnten bis zwanzigsten Mai (from the fourteenth to the twentieth of May)* is the Pivot, which is part of the first –unfinished– syntactic construction *ich bin vom vierzehnten bis zwanzigsten Mai (I am from the fourteenth to the twentieth of May)* and of the second – finished – syntactic construction *vom vierzehnten bis zwanzigsten Mai <äh> <hm> bin ich . . . (from the fourteenth to the twentieth of May <uh> <hm> I am . . .)*. **Fresh starts** do not have a pivot; the construction is aborted and a completely new one is started: *also wenn wir das – das ist der Montag (so if we that – that is the Monday)*. Commonly each repair is segmented in the four parts **reparandum**, **editing term**, **interruption point**, and **reparans**; an example is given in figure 1:

- **reparandum**: the "wrong" part of the utterance
- **interruption point (IP)**: boundary marker at the end of the reparandum
- **editing term**: special phrases, which indicate a repair like "well", "I mean" or filled pauses such as "uhm", "uh" (optional, most of the time missing)
- **reparans**: the correction of the reparandum

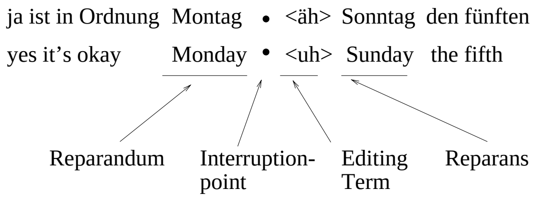
ja ist in Ordnung Montag • <äh> Sonntag den fünften
yes it's okay Monday • <uh> Sunday the fifth
Reparandum
Interruption-point
Editing Term
Reparans

Figure 1: A repair example

2. The VERBMOBIL System

The goal of the VERBMOBIL project (1992–2000) was to build a speech-to-speech translation system supporting the tasks of appointment scheduling, travel planning and help desk. In the German part of the VERBMOBIL corpus, 21% of all turns contain at least one repair. Most of them (82%) are modification repairs. We therefore concentrate on this type of repairs. Modification repairs have a strong correspondence between reparandum and reparans. We could measure this in terms of length of reparandum and reparans (see table 1) and part-of-speech (POS) replacements. For almost all POS-categories, the speakers prefer to modify a word in the reparandum with a word which belongs to the same POS category in the reparans. Thus there is no need for a complete syntactic analysis to detect and correct most modification repairs even if repairs are characterized by violation of syntactic and semantic well-formedness [9]. We implemented a statistical approach as a filter process between the speech recognition engine and the syntactic parser. Starting with the word hypotheses graph (WHG) produced by the

[1] This work is part of the VERBMOBIL project and was funded by the German Federal Ministry for Research and Technology (BMBF) in the framework of the Verbmobil Project under Grant BMBF 01 IV 701 V0. The responsibility for the contents of this study lies with the authors.

Reparandum	Type	Reparans	#
RR1	IW	RR1	580
DD1	IW		495
MM1	IR	MM1	486
RR1	IR	RR1	411
MM1 MM2	IR	MM1 MM2	111
DD1	IR		108
RR1	IW	II1 RR1	101
MM1 RR1	IR	MM1 RR1	100
MM1 RR1	IW	MM1 RR1	85
MM1	IR	II1 MM1	74
RR1 MM1	IR	RR1 MM1	53
RR1 RR2	IW	RR1 RR2	41
DD1 DD2	IR		35
MM1 MM2 RR1	IW	MM1 MM2 RR1	31
MM1	IR	II1 II2 MM1	27
RR1	IW	II1 II2 RR1	26
RR1 RR2	IR	RR1 RR2	25
MM1 RR1 MM2	IR	MM1 RR1 MM2	25
MM1 MM2 MM3	IR	MM1 MM2 MM3	24
MM1 MM2 RR1	IR	MM1 MM2 RR1	22
DD1	IW	II1	22
			2860

Table 1: Patterns for Modification Repairs (>20 tokens; 3559 patterns in the corpus, ordered by frequency); each word in the reparandum/reparans is annotated as either MM: Match, RR: Replacement, DD: Deletion, or II: Insertion; the integers (1, 2, and 3) relate the words in the reparandum/reparans with each other; IW means interruption with "w"ord fragment, IR interruption without word fragment. Example: *... on* RR1 *Monday* MM1 IR *next* RR1 *Monday* MM1 *...*

word recognizer, a prosodic module detects possible IPs. For each of these IPs, a stochastic model tries to find an appropriate repair by guessing the most probable segmentation. Repair processing is seen as a statistical machine translation problem where the reparandum is a translation of the reparans. For every repair found, a path representing the speaker's intended word sequence is inserted into the lattice. In the last step, a lattice parser selects the best path. The complete architecture is shown in figure 2.

2.1. Detection of Interruption Points

The prosodic module classifies each word boundary in the WHG as a regular or an irregular boundary. Irregular boundaries are seen as hypotheses for IPs. For each word boundary, a vector with 121 prosodic features is determined. Prosodic events like irregular boundaries are characterized by local changes in the acoustic parameters. Tests showed that a context of two words to the left and to the right of the actual word is sufficient for detection. The features are selected to give information about F0, energy, duration, pause, and POS-categories; details are given in [1]. A classification of a subsample of the VERBMOBIL database with neural networks and 559 IPs vs. 51.486 "normal" word boundaries (i.e., a relation of 1:100!) yielded the following results: recall for IPs: 90%, recall for normal word boundaries 64% which means that there are many false alarms. This is a general problem of binary statistical classifiers in cases where the proportion of the two classes is extremely unbalanced.

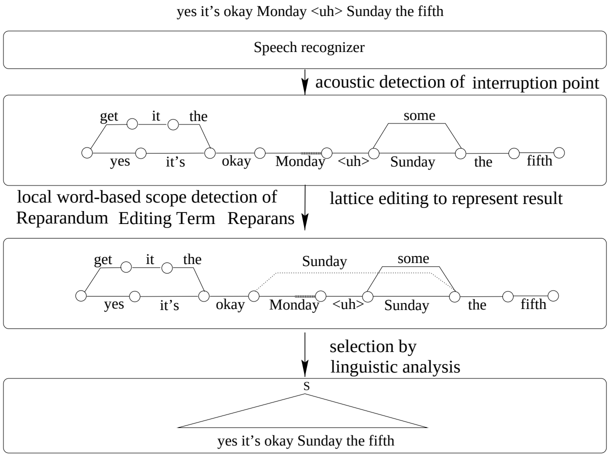
yes it's okay Monday <uh> Sunday the fifth
Speech recognizer
acoustic detection of interruption point
get
it
the
some
yes
it's
okay
Monday
<uh>
Sunday
the
fifth
local word-based scope detection of
Reparandum Editing Term Reparans
lattice editing to represent result
get
it
the
Sunday
some
yes
it's
okay
Monday
<uh>
Sunday
the
fifth
selection by
linguistic analysis
S
yes it's okay Sunday the fifth

Figure 2: Architecture for repair processing

2.2. Segmentation

As mentioned before repair segmentation is mainly based on statistical machine translation (SMT) [3]. The SMT approach assumes that a speaker who produces the source sentence S originally wants to produce the target sentence T. Transferring this approach to repair processing, we assume that, if a speaker produces the reparandum (RD), he/she originally wanted to produce the reparans (RS). SMT defines a scoring function for a pair (S, T) which could be adopted for repair processing without further changes:

$$P(RD|RS) = \sum_a P(RD, a|RS)$$

a is the alignment, which describes the link between words in RD and RS; SMT is based on the hypothesis that words of the source sentence are linked to words in the target sentence.[2] If the stronger assumption is made that a word of the source sentence could only be linked to one word in the target sentence, a can be described as a vector $a_1^m = a_1 \dots a_m$ with $a_i \in 0 \dots l$. If the word RD_j is linked to RS_i then $a_j = i$. If it is not connected to any word in RS then $a_j = 0$. m denotes the length of RD and l the length of RS. Without any further assumptions we can infer the following:

$$\begin{aligned} P(RD, a|RS) \quad &= \quad P(m|RS) * \\ & \prod_{j=1}^{m} P(a_j|a_1^{j-1}, RD_1^{j-1}, m, RS) * \\ & \quad P(RD_j|a_1^j, RD_1^{j-1}, m, RS) \quad (1) \end{aligned}$$

The conditional probabilities in equation (1) cannot be estimated reliably from any corpus of realistic size, because there are too many parameters. For example both P in the product depend on the complete reparans RS. Therefore we simplify the probabilities by assuming that m depends only on l, a_j only on j, m, and l, and finally RD_j on RS_{a_j}. So equation (1) becomes

$$\begin{aligned} P(RD, a|RS) &= P(m|l) * \\ & \prod_{j=1}^{m} P(a_j|j, m, l) * P(RD_j|RS_{a_j}) \qquad (2) \end{aligned}$$

[2] A couple of words in the source sentence describes the same concepts as a couple of words in the target sentence.

These probabilities can directly be trained from a manually annotated corpus, where all repairs are labeled with begin, end, IP, and editing term, and for each reparandum, the words are linked to the corresponding words in the reparans. All distributions are smoothed by a simple back-off method [8] to avoid zero probabilities with the exception that the replacement probability $P(RD_j|RS_{a_j})$ is smoothed in a more sophisticated way. It is calculated by a linear interpolation of replacement probabilities for the words, the corresponding POS tags, and the semantic class

$$\begin{aligned} P(RD_j|RS_{a_j}) = \\ &\alpha * P(Word(RD_j)|Word(RS_{a_j})) \\ &+ \beta * P(SemClass(RD_j)|SemClass(RS_{a_j})) \\ &+ \gamma * P(POS(RD_j)|POS(RS_{a_j})) \end{aligned} \tag{3}$$

with $\alpha + \beta + \gamma = 1$.

2.3. Processing word hypothesis graphs

The scoring function is integrated in the system on top of the prosodic annotated WHG from the recognizer. For each path through the WHG that contains an IP hypothesis, all possible segmentations, i.e., all possible (RD, RS) pairs, must be scored. In practice we reduce this set to pairs, where RD and RS are at most four words long, because we found that this restriction holds for 96% of all repairs in the VERBMOBIL corpus. Editing terms are characterized by a closed list of short phrases. Thus if after an IP such a phrase is found, it is skipped to build the (RD, RS) pair. If the score of a pair is above a heuristic threshold, the pair is accepted as a repair and an alternative path is inserted in the WHG. The resulting WHG is finally analyzed by a stochastic parser, which selects according to its model the best scored path and therefore can accept or reject the repair.

3. Constraints in real systems

Before we discuss our results and compare them to other approaches, we want to discuss the evaluation framework: the ultimate goal of our approach was a full and easy integration into a real-life system. Thus we cannot expect perfect strings as input. The state-of-the-art interface of a speech recognizer are WHGs with many co-occurring hypotheses. They represent an enormous search space, so an efficient algorithm is necessary to guarantee a time behavior in almost real time. In addition there exist no prominent repair markers, as hypothesized by some authors, to reduce the search space to the relevant points.

Another problem in real-life systems are word fragments. They play an important role in repair processing but speech recognizers are not able to mark them. With respect to these constraints, three different evaluations are carried out. The first two are stand-alone evaluations measuring the performance of the pure repair process. The third one shows the impact of the repair process on the complete VERBMOBIL system and will be described in the next section.

The first row of table 2 shows the results with the assumption that we have a perfect recognizer that produces no word errors and marks every word fragment. The test set were 441 turns, a subsample of the 559 prosodically classified turns; 118 turns were used for the training of α, β, and γ. Processing time was restricted in two ways. First there is a dynamic deadline, the real-time factor. It is set to five times the length of the turns. Second there is an absolute deadline of 10 seconds. The reference machine was a SPARC Ultra 300MHZ. The "detection" column shows the results for the repair identification task. The "correct seg." column presents the same numbers for the correct segmentation. A segmentation is defined as "correct" if reparandum and editing term are identified. In some cases within complex repairs (repairs within repairs), reparandum and editing term are not identified correctly but, if these segments are removed from the input, the resulting string is the intended word sequence. An example is:

Annotated:	... wann paßt es	| *di*	| *Ihn*	| Ihnen denn ...
Annotated:	... when does it suit	| *y*	| *yo*	| you ...
		RD1	RD2	RS
Recognized:	... wann paßt es	| *di Ihn*		| Ihnen denn ...
Recognized:	... when does it suit	| *y yo*		| you ...
		RD		RS

Therefore we call these results "generalized segmentation". The second row in table 2 presents the same evaluation based on an almost perfect recognizer which is not able to mark fragments.

The decrease of the recall rate from test 1 to test 2 emphasizes the importance of word fragments. Fragments are not only a prominent detection feature but they are also easier to correct. In many cases the correction is simply a deletion of the fragmented word. The decrease of the precision rate is not really a decrease in quality. It comes from the worser ratio of non-repair turns to repair-turns in test 2, where we only left out repairs with word fragments.

A direct comparison to similar work is rather difficult due to very different corpora, evaluation conditions, and goals. Not all approaches deal with the complete repair process but concentrate on either detection or correction. [10] report a recall of 83.4% and a precision of 93.9% in detecting the IP of a repair, but do not discuss the problem of finding the correct segmentation in detail. In addition their results are obtained on a corpus where every utterance contains at least one repair. [12] introduce hidden events to model the IPs of different classes of repairs in the speech recognition process. This reduces their recognition errors by about 0.9% absolute, but nothing is said about recall and precision of IP detection. Likewise they make no suggestion about getting the correct segmentation. An early and comprehensive attempt is described in [2]. They use a pattern matcher to trigger possible repairs and verify these hypotheses with a parser. The simple pattern matching algorithms achieves a recall of 76.1% and a precision of 61.8% for repair detection. 57% of the detected repairs are successfully corrected (43.6% Rec./48.1% Prec.). A second evaluation based on a different test set (26 repairs) includes a verification of the hypothesized repairs by a parser [5]. If the parser finds an unacceptable utterance, the hypothesized repairs are successively parsed until a parseable utterance is selected. In this case a detection recall of 42.3% and a precision of 84.6% is obtained. For correction the values are 30.8% recall and 61.5% precision. They comment that this procedure is not very efficient in a real-time speech system. [7] suggests a parsing approach using a deterministic parser. He assumes a perfect repair detector, so there can be no comparison as for the detection and correction algorithms. An algorithm which is inherently capable of lattice processing is proposed by [6]. They redefine the word recognition problem as identifying the best sequence of words, corresponding POS tags and special repair tags. They report a recall rate of 81% and a precision of 83% for detection, and 78%/80% for correction. The test setup was almost the same as that for test 1 (cf. Table 2). Unfortunately, nothing is said about the processing time achieved with their module. [4] build a parser on

	Detection		Correct seg.		Generalized seg.	
	Recall	Precision	Recall	Precision	Recall	Precision
Test 1	70%	86%	59%	84%	61%	84%
Test 2	48%	77%	47 %	76%	48%	76%

Table 2: Results for repair processing

top of this module in a similar way to [2]. They observe a slight improvement of about 2% in recall but a drop of about 50% in precision.

4. End-to-End Evaluation

Within real-life systems, we cannot measure the performance of the repair process in terms of recall and precision. Errors in word recognition or parsing influence the performance. If for example a word in the reparandum or reparans is misrecognized the strong correspondence between reparandum and reparans is obscured. The worst case would be that the recognition error leads to a correct sentence that the repair process should not correct. For the recall value this event is counted as a miss, but from the point of view of repair processing the behavior is totally correct. We therefore measure the impact of repair processing by the changes we found in the results after parsing. The VERBMOBIL system was tested on 276 turn with active and inactive repair processing. The turns contain 90 repairs. In 64 WHGs a repair is hypothesized, but only 12 times the parsing output is changed. A manual inspection of these changes shows that 6 repairs are correct. This means that it was a real repair or a recognition error, that could not be told apart from a repair. Two hypotheses are definitely false alarms, in two cases the hypothesis is correct, but the parser cannot analyze the corrected version. For the rest of the hypotheses, the word recognition was not good enough to decide, whether they were correct or not.

As expected there is a big difference between an idealised environment and the real-life system. But not only word fragments cause problems. Word errors[3] and parsing problems inhibit that repair processing has a greater impact on the complete system.

5. Summary and Conclusion

The term “speech repairs” denotes different phenomena, which have to be handled by different methods. In VERBMOBIL , we concentrate on the most frequent type of repairs, i.e., modification repairs. We found a strong correspondence between reparandum and reparans in syntactic and semantic features, which are utilized in a stochastic approach to repair detection and correction. The promising results on word strings could not be verified in the VERBMOBIL system. One major problem in real-life systems are word fragments, which cannot be marked by state-of-the-art word recognizers. In addition recognition errors and incomplete syntactic analyses reduce the impact of the repair process on the complete system. This first attempt of applying repair processing to a speech-to-speech system shows, that besides a necessary and possible improvement of the repair process itself, the system performance must be enhanced to benefit from such a process.

Modelling the repair segmentation as a stochastic machine translation process offers a great variety of improvements. Our approach models the replacement probability quite simple with very rough assumptions. Och et al. in [11] show and compare more sophisticated approaches, which can be applied to repair processing as well.

6. References

[1] Anton Batliner, Jan Buckow, Heinrich Niemann, Elmar Nöth, and Volker Warnke. The Prosody Module. *Verbmobil: Foundations of Speech-to-Speech Translation.* Springer, Berlin, 2000, pages 106-121.

[2] J. Bear, J. Dowding, and E. Shriberg. Integrating multiple knowledge sources for detection and correction of repairs in human computer dialogs. In *Proc. ACL*, pages 56–63, University of Delaware, Newark, Delaware, 1992.

[3] P. F. Brown, J. Cocke, S. A. Della Pietra, V. J. Della Pietra, F. Jelinek, J. D. Lafferty, R. L. Mercer, and P. S. Roossin. A statistical approach to machine translation. *Computational Linguistics*, 16(2):79–85, June 1990.

[4] M. G. Core and K. Schubert. Speech repairs: A parsing perspective. In *Satellite meeting ICPHS 99*, pages 47–50, 1999.

[5] John Dowding, Jean Mark Gawron, Doug Appelt, John Bear, Lynn Cherny, Robert Moore, and Douglas Moran. Gemini: a Natural Language System for Spoken-Langugae Understanding. In *Proc. ACL*, pages 54–61, 1993.

[6] Peter A. Heeman and James F. Allen. Speech repairs, intonational phrases, and discourse markers: Modelling speakers’ utterances in spoken dialogue. *Computational Linguistics*, 25(4):527–571, December 1999.

[7] D. Hindle. Deterministic parsing of syntactic nonfluencies. In *Proc. ACL*, pages 123–128, MIT, Cambridge, Massachusetts, 1983.

[8] S. M. Katz. Estimation of probabilities from sparse data for the language model component of a speech recognizer. *Transaction on Acoustics, Speech and Signal Processing*, ASSP-35:400–401, March 1987.

[9] W. Levelt. Monitoring and self-repair in speech. *Cognition*, 14:41–104, 1983.

[10] C. Nakatani and J. Hirschberg. A speech-first model for repair detection and correction. In *Proc. ACL*, pages 46–53, Ohio State University, Columbus, Ohio, 1993.

[11] Franz Josef Och and Hermann Ney. A comparison of alignment models for statistical machine translation. In *COLING 00: The 18th Int. Conf. on Computational Linguistics*, pages 1086–1090, Saarbrücken, Deutschland, 2000.

[12] A. Stolcke, E. Shriberg, D. Hakkani-Tur, and G. Tur. Modeling the prosody of hidden events for improved word recognition. In *EUROSPEECH ’99*, volume 1, pages 307–310, Budapest, 1999.

[3] The word error rate for the 276 turns is 24%.