An Annotation Scheme for Complex Disfluencies

Peter A. Heeman[†], Andy McMillin[‡], J. Scott Yaruss[*]

[†] Center for Spoken Language Understanding
OGI School of Science & Engineering
Oregon Health & Science University

[‡] Hearing & Speech Institute
Beaverton, Oregon

[*] Stuttering Center of Western Pennsylvania
Dept. of Communication Sciences and Disorders
University of Pittsburgh

Abstract

In this paper, we present an annotation scheme for disfluencies. Unlike previous schemes, this scheme allows complex disfluencies with multiple backtracking points to be annotated, which are common in stuttered speech. The scheme specifies each disfluency in terms of word-level annotations, thus making the scheme useful for building sophisticated language models of disfluencies. As determining the annotation codes is quite difficult, we have developed a pen and paper procedure in which the annotator lines up the words into rows and columns, from which it is straight-forward for the annotator to determine the annotation tags.

Index Terms:: disfluencies, stuttered speech, annotation scheme

10.21437/Interspeech.2006-55

1. Introduction

In conversational speech, disfluencies are very common [4]. Hence, it is important to build disfluency modeling into speech recognizers. Modeling disfluencies is also important for dealing with the speech of people who stutter, both for determining what they are saying for spoken language applications, but also, as part of future automatic stuttering assessment tools.

To deal with the wide range of disfluencies of both people who do not stutter and of those who do, we need an annotation scheme that can capture the full variety of disfluencies. The scheme also needs to capture what is happening at the word level, so that sophisticated language models can be built. Finally, the scheme should allow all types of disfluencies to be annotated in a consistent and parsimonious way.

In this paper, we present a scheme that covers multi-iteration repetitions, revisions, editing terms, and starters, as well as any complex clustering of these. A problem with the scheme is that it is difficult to determine the word-level annotations. Therefore, we present a pen-and-paper method in which annotators line up the transcribed words into rows and columns. From this depiction, it is easy to determine the appropriate annotation tags. We start with simple repetitions, and then add revisions, editing terms, and starters.

2. Related Research

Repair Structure: There has been extensive research analyzing disfluencies in the speech of non-stutterers. The most common disfluencies studied are revisions and single-iteration repetitions. In a scheme that has been used by a number of researchers [1, 2, 3, 4], disfluencies are decomposed into 4 parts: (1) a *reparandum*, which is the speech that is replaced by other speech; (2) an *interruption point*, which is the time when the reparandum ends; (3) optional *editing terms*, such as "um" and "let's see"; and (4) an *alteration*, which is the replacement for the *reparandum*, as illustrated in Fig. 1. Removing both the reparandum and the editing terms gives the speaker's intended utterance. This approach identifies all of the words involved in a disfluency, and the role that each plays. However, this approach does not address how to annotate clustered disfluencies or multi-iteration repetitions, which are very common in stuttered speech [5, 6, 7].

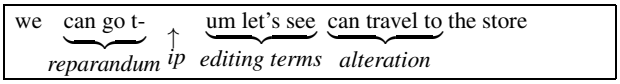

Figure 1: Four parts of repetitions and revisions

Systematic Disfluency Analysis: An instrument proposed for assessing the severity of stuttering is Systematic Disfluency Analysis (SDA; [8]). With this approach, the user annotates each disfluency with a tag for the type of the disfluency. For example, the code **I** indicates an interjection (similar to the editing term of Fig. 1), $\mathbf{Rsy}^3$ indicates a syllable repetition with 3 iterations, and $\mathbf{P}^V$ indicates a prolongation accompanied by "visible tension". For clustered disfluencies, SDA uses a compound annotation, joined together with '+'s. These codes are placed above the transcribed words that are involved. An example of an annotation of a compound disfluency is shown in Fig. 2 (from [8]). A problem with SDA is that it does not capture word-level effects. For the example in Fig. 2, it is not possible to determine automatically that the first "g-", third "g-", and "garage" are involved in the first sound repetition; that the second "g-" and "going" are involved in the second sound repetition; and that "he is going to the" is the first phrase repetition; and that "to the" is the second phrase repetition.

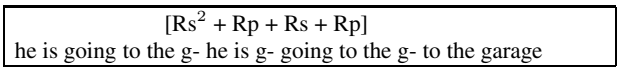

Figure 2: SDA annotation of a clustered disfluency

CHAT: Bernstein Ratner, Rooney and MacWhinney (1996) extended the CHAT annotation scheme, so that disfluencies typical of stuttered speech can be annotated. Their goal of keeping the extension compatible with the existing CHAT codes and CHAT utility programs (for calculating mean utterance length, etc.) resulted in different conventions for each type of repetition. For example, the following utterance with a word repetition, "that's that's that's interesting", is transcribed and annotated as "that's(/2) interesting", where **(/2)** indicates 2 iterations of "that's". The phrase repetition, "I'm going to I'm going to study later", is annotated as "I'm going to I'm+going+to@pr study later". The lack of regularity in coding different types of disfluencies will make it difficult to extract regularities from them. Furthermore, it is not clear how clustered disfluencies can be annotated in this scheme.

Shriberg '94: Shriberg developed an annotation scheme for overlapping disfluencies [10]. Two disfluencies *overlap* if their reparandum and/or alterations have some words in common. Shriberg assumes that overlapping disfluencies can individually be written using the approach of Fig. 1, with reparandum and alteration nested in each other, in a recursive fashion. To annotate a disfluency, she uses square brackets to enclose the reparandum and

alteration of each repair and a period to mark the interruption point. For example, the utterance, "a total of of total of seven hours", is annotated as "a [total [of . of] . total of] seven hours", where "[of . of]" is the inner repair, and "[total of . total of]" is the outer repair. Note that the outer repair uses the alteration of the inner repair, but not the reparandum. A problem with this approach is that some overlapping disfluencies, which Shriberg terms *partially chained structures*, run contrary to her nesting assumption. For these disfluencies, Shriberg forces one disfluency to be the "embedded disfluency" and uses an ad hoc operator to annotate them. The use of this operator will likely not only be confusing to annotators, but will lead to difficulties in developing a language model that captures the true regularities of overlapping disfluencies.

3. Basics of the Annotation Scheme

In previous studies of disfluencies of non-stutterers, we extended the repair-structure approach so that not only revisions and single-iteration repetitions can be annotated, but also overlaps of them [11]. For overlapping disfluencies, we focused on annotating the reparandum, interruption point and editing terms, but not the alteration. Each time the interruption point of a disfluency is encountered (starting from the beginning of the audio signal), the annotator determines how far the speaker is *backtracking* in their speech. This gives the reparandum of the disfluency. The words in the reparandum are not available to be used in the reparandum of subsequent disfluencies, but reparanda of subsequent disfluencies can backtrack to include words that precede the reparanda of previous disfluencies. To illustrate, consider the utterance "a total of of total of seven hours", which has 2 repetition disfluencies. The first disfluency is the repetition "of". Fig. 3 shows the utterance with the first disfluency's reparandum denoted by an arrow showing how far back the speaker backtracked. The second disfluency is the repetition of "total of". As the first instance of "of" is part of the first disfluency's reparandum, it cannot be used in the reparandum of the second. Hence, the reparandum of the second is the first instance of "total" and the second instance of "of", as shown in Fig. 4. This way of annotating reparanda is similar to what is done in DialogueView [12].

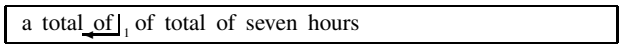

Figure 3: First disfluency marked

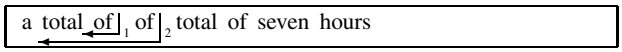

Figure 4: Both disfluencies marked

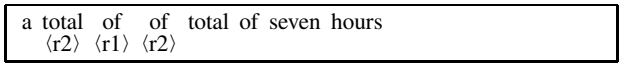

Figure 5: Word-level annotations

The reparandum markings illustrated in Fig. 4 can be turned into word-level annotations: we use the symbol $\langle \mathrm{r}\, i \rangle$ to mark each word of the reparandum of disfluency i. The resulting annotations are shown in Fig. 5. Note that the reparandum of the first disfluency is embedded inside the reparandum of the second disfluency.

Now consider the utterance "a total of total of s- of seven hours". The disfluencies in this example ("total of total of' and "of s- of seven") partially overlap, which causes difficulties for Shriberg's annotation scheme. However, as illustrated in Fig. 6, these disfluencies do not pose a difficulty for our scheme as we do not require the reparandum and alteration of disfluencies to nest inside each other.

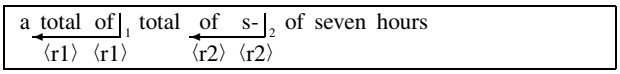

Figure 6: Partial overlap

4. Vertical Alignment Method

For overlapping disfluencies, it can be difficult for annotators to determine which words belong to which disfluency's reparandum. Hence, we developed a *vertical-alignment* method to simplify this task [13], which consists of 4 steps.

Step 1: Format all of the words as a single line of text. Then, after each interruption point (and its optional editing term), start a new line. The result of this step is shown in Fig. 7 for the utterance "a total of of total of seven hours".

Figure 7: Start a new line at backtracking points

Step 2: Vertically align words from different lines that the speaker uses as replacements for one another. Fig. 8 shows the result: the 3 instances of "of" and 2 instances of "total" are aligned.

Figure 8: Vertically align words

Step 3: Determine how far back the speaker backtracked in starting each line. Fig. 9 shows the result: the speaker backtracked over the 3rd column to start the second line, and backtracked over the 2nd and 3rd column to start the third line.

Figure 9: Mark extent of reparanda

Step 4: Determine which words belong to which reparandum of each backtracking: a word belongs to the first backtracking beneath it. The result, shown in Fig. 10, is the same word annotations as Fig. 5. The first instance of "total" belongs to the second backtracking, as the first backtracking does not extend beneath it, while the second one does. The final version of the speech can be read from the bottom word in each column of the vertical alignment, and so is "a total of seven hours".

Figure 10: Annotate words with disfluency indices

4.1. Applying the Vertical Alignment Method

To further illustrate how the vertical alignment method simplifies the annotation process, consider the example from Fig. 6. In the first step, we format the words so that a new line is started after each interruption point: after "of" and "s-". In the second step, we align the two instances of "total" and "of", and the instance of "s-" and "seven", as shown in Fig. 11. In the third step, we draw the arrow for the first backtracking so that it spans the second and third columns as the second line starts at the second column; and draw

the arrow for the second backtracking so that it spans the third and fourth columns. In the fourth step, we mark the first instance of "total" and "of" as belonging to the first backtracking, and the second instance of "of" and only instance of "s-" as belonging to the second. The resulting tags are identical to those in Fig. 6.

Figure 11: Annotating partial overlap

The scheme can be easily applied to multi-iteration repetitions and clusters of such disfluencies in stuttered speech [13]. For multi-iteration repetitions, each repetition is associated with a separate backtracking point, as shown in Fig. 12. Here, the first reparandum is the first instance of "can", and the second reparandum is of the second instance of "can".

Figure 12: Multi-iteration Word Repetition

The stuttering literature distinguishes between sound, word, and phrase repetitions. The utterance "it c- it c- it can have mountains or be flat" has a clustered disfluency consisting of a sound and word repetition, in which the speaker says the word and sound, and then repeats them twice. We annotate this clustered disfluency as having 2 backtrackings (and reparanda), which each contain both a word and sound repetition. Fig. 13 shows the vertical alignment for this example. As the example illustrates, the reparanda of our annotation scheme sometimes combine elements from what are viewed as multiple disfluencies in the stuttering literature.

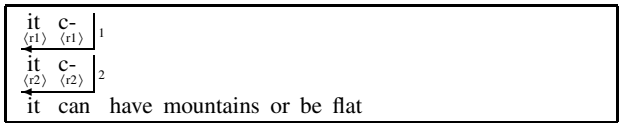

Figure 13: Word & Sound Repetition

5. Additional Disfluencies

Revisions: A revision is where a speaker backtracks, but does not strictly repeat what was just said, but modifies it. In Fig. 14, the speaker replaced "back in the water" with "back into the water". In our annotation scheme, we mark each word of the the reparandum with the code '⟨r*i*⟩', where *i* is the index for this backtracking. In the vertical alignment method, we format the utterance so that a new line is started after the reparandum, and vertically align the words so that words that are replacements for others are in the same columns. Hence, the two instances of "back", "the", and "water" are in the same columns, as are "in" and "into".

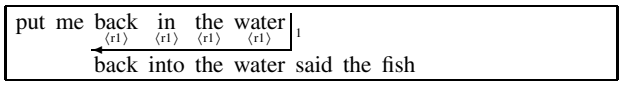

Figure 14: Revision with word replacement

A speaker might insert or omit words in the revision, as shown in Fig. 15. In the example in Fig. 15, the revision includes the inserted word "of". As there is no corresponding word "of" in the original speech, we put a dash in the column above "of". Similarly,

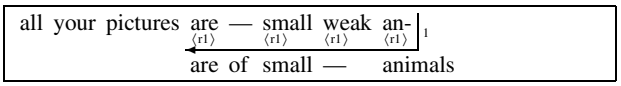

Figure 15: Omitted & inserted words

the revision omitted the word "weak". Hence, we put a dash in the column under the word "weak".

The omitted and inserted words captured in the vertical alignment method require additional codes in our annotation scheme. We use the code '⟨a*x*⟩' to *anchor* two words that should be in the same column, where a unique *x* is used for the words in each column that need to be anchored. For the disfluency in Fig. 15, the two instances of "small" are given a common anchor, and the instance of "an-" and "animals" are also given a common anchor. All the other column alignments can be inferred.

Editing Terms: Editing terms, as shown in Fig. 1, commonly occur after the interruption point of revisions and single-iteration repetitions. In our scheme, they are given a code of '⟨i⟩', as they are referred to as *interjections* in the stuttering literature. If they immediately follow a reparandum, they are associated with the reparandum's backtracking. In the vertical alignment method, the editing terms are formatted on the same line as its associated reparandum, and displayed in bold, as illustrated in Fig. 16. The editing term "I mean" is associated with the reparandum "a raindrop" marked with index 1. In this example, there are two more backtrackings, both with reparandum "a" and neither with an editing term.

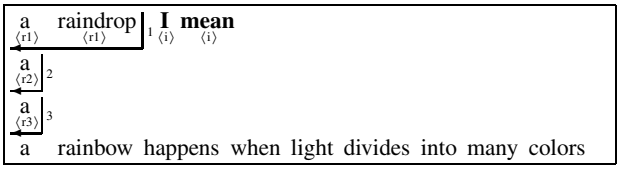

Figure 16: Editing term associated with a reparandum

Editing terms can also occur on their own in what are called covert repairs [1] or abridged repairs [4]. In this case, the editing term does not have an associated reparandum. In the vertical alignment method, we still view it as causing a backtracking: after the editing term, a new line is started with the subsequent words immediately lined up with the beginning of the editing term. Fig. 17 gives an example. The first backtracking has the editing term "um", but with no reparandum. Hence, the next line is formatted so that it starts directly underneath the editing term. Note that this example has a second backtracking, with reparandum "she was only fooling", with index 2. Note that no correspondence is assumed between the editing term "um" and the two instances of "only" that also appear in the same column.

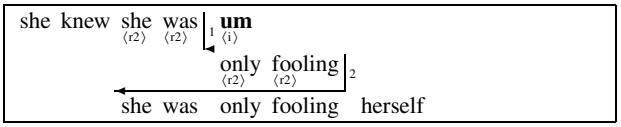

Figure 17: Editing term separate from reparandum

Starters: People who stutter sometimes use *starter* words. Starters are similar to editing terms in that they are not part of what the speaker is trying to communicate. The difference is that starters are used to help (re)start phonation. Hence, starters typically lead, without pause, into the following word. A typical starter is "and".

Just as with editing terms, starters might occur by themselves, or be associated with a reparandum. When it occurs by itself, it is annotated as '⟨s*i*⟩' where *i* is a unique disfluency index. In the

vertical alignment method, such a starter is viewed as causing a backtracking. The starter is formatted in italics on a new line, so that the end of the starter is lined up with the end of the preceding line, as shown in Fig. 18. The reason the starter is on the following line, rather than the preceding line as with editing terms, is to capture that the starter is being used to (re)start phonation.

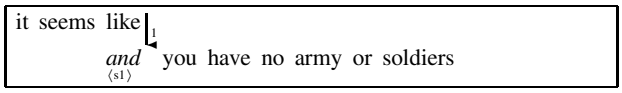

Figure 18: Starter without an associated reparandum

When a starter occurs immediately before the alteration of a reparandum, it is associated with that reparandum's backtracking. In this case the starter is annotated with '$\langle si \rangle$', where i is the same index as is given to the words in the reparandum. In the vertical alignment method, the starter is formatted so that it immediately precedes the alteration, as shown in Fig. 19.

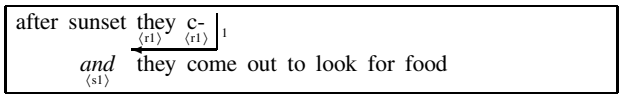

Figure 19: Starter with an associated reparandum

Complex patterns of backtracking involving starters can occur, as shown in Fig. 20. In particular, this example shows two starters occurring right after each other. Because starters are attempts to (re)start phonation, an immediate repetition of a starter represents an unsuccessful attempt to (re)start phonation; hence the starters are viewed as separate backtrackings, each with its own index.

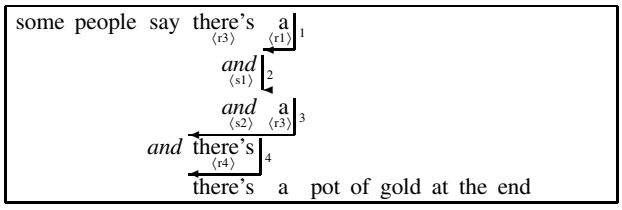

Figure 20: Starter occuring at the backtracking

6. Aligning the transcription with the story

Speech samples for diagnosing stuttering are sometimes collected by having the person read a story. In these cases, it is useful to relate the transcribed words to the words of the story that the speaker was supposed to read, in order to capture reading errors. To accomplish this, we use the vertical alignment method, in which the last row is the story text, as illustrated in Fig. 21. From the alignment, we see that the speaker had two reading mistakes, which were not corrected (and hence are not annotated as a backtracking). The speaker omitted the word "in" and inserted the word "the". We capture the alignment between the transcribed words and the story words through the use of anchors. Each word of the story is given a unique anchor in advance. Any time the speaker deviates from the story text, we anchor the nearby words in the transcription.

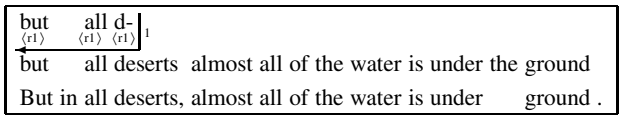

Figure 21: Transcribed words aligned to the story text

7. Conclusion

In this paper, we presented a scheme for annotating disfluencies, including complex ones that are common in stuttered speech. The annotation scheme covers multi-iteration repetitions, revisions, starters, and editing terms. It also allows the transcribed words to be aligned with the story for read-speech tasks. The scheme relates each disfluency to the individual words and sounds that the speaker says, thus allowing precise modeling of disfluencies by an automatic speech recognizer.

For complex disfluencies, determining the annotation tags can be very difficult. Hence, we developed the vertical alignment method, in which the annotator first graphically (with pen and paper) aligns the words into rows and columns. From this depiction, the annotator then determines the annotation tags. We have also developed a tool that takes the transcript and the word-level annotation tags and produces the vertical alignment, which is useful both for visualizing the disfluencies and for double-checking the annotation tags. In future work, we plan to simplify the annotation process: the annotator will use a graphical computer tool to align the words into rows and columns, and then the computer tool will automatically assign the annotation tags.

8. References

[1] W. Levelt, "Monitoring and self-repair in speech," *Cognition*, vol. 14, pp. 41–104, 1983.

[2] D. Hindle, "Deterministic parsing of syntactic non-fluencies," in *Proceedings of the 21st Annual Meeting of the Association for Computational Linguistics*, 1983.

[3] C. Nakatani and J. Hirschberg, "A corpus-based study of repair cues in spontaneous speech," *Journal of the Acoustical Society of America*, vol. 95, no. 3, pp. 1603–1616, 1994.

[4] P. Heeman and J. Allen, "Speech repairs, intonational phrases and discourse markers: Modeling speakers' utterances in spoken dialog," *Computational Linguistics*, vol. 25, no. 4, pp. 527–572, 1999.

[5] C. Hubbard and E. Yairi, "Clustering of disfluencies in the speech of stuttering and nonstuttering preschool children," *J. of Speech and Hearing Research*, vol. 31, 1988.

[6] L. LaSalle and E. Conture, "Disfluency clusters of children who stutter: Relation of stutterings to self-repairs," *J. of Speech and Hearing Research*, vol. 38, pp. 965–977, 1995.

[7] K. Logan and L. LaSalle, "Grammatical characteristics of children's conversational utterances that contain disfluency clusters," *Journal of Speech, Language, and Hearing Research*, vol. 42, pp. 80–91, Feb. 1999.

[8] H. Gregory, J. Campbell, C. Gregory, and D. Hill, *Stuttering Therapy: Rationale and Procedures*. Pearson Allyn & Bacon, 2003.

[9] N. Bernstein Ratner, B. Rooney, and B. MacWhinney, "Analysis of stuttering using CHILDES and CLAN," *Clinical Linguistics and Phonetics*, pp. 169–187, 1996.

[10] E. Shriberg, "Preliminaries to a theory of speech disfluencies," UC Berkeley," Doctoral dissertation, 1994.

[11] P. Heeman, "Speech repairs, intonational boundaries and discourse markers: Modeling speakers' utterances in spoken dialog," U. of Rochester, Doctoral dissertation, 1997.

[12] F. Yang, P. Heeman, K. Hollingshead, and S. Strayer, "DialogueView: Annotating and viewing dialogue with multiple levels of abstraction," *Natural Language Engineering*, to appear, 2006.

[13] P. Heeman, J. Yaruss, and A. McMillin, "Towards a detailed annotation scheme for clustered disfluencies," in *Workshop on the Characteristics and Assessment of Stuttered Speech*, London England, June 2005.