Multiple Dimension Levenshtein Edit Distance Calculations for Evaluating Automatic Speech Recognition Systems During Simultaneous Speech

Jonathan G. Fiscus[1], Jerome Ajot[1], Nicolas Radde[1], and Christophe Laprun[1,2]

[1] National Institute of Standards and Technology (NIST), 100 Bureau Drive, Stop 8940, Gaithersburg, MD 20899, USA

[2] Systems Plus Inc., 1370 Piccard Drive, Suite 270, Rockville, MD 20850, USA

{jfiscus,ajot} @ nist.gov, nicolas @ radde.org, chris.laprun @ jboss.com

Abstract

Since 1987, the National Institute of Standards and Technology has been providing evaluation infrastructure for the Automatic Speech Recognition (ASR), and more recently referred to as the Speech-To-Text (STT), research community. From the first efforts in the Resource Management domain to the present research, the NIST SCoring ToolKit (SCTK) has formed the tool set for system developers to make continued progress in many domains; Wall Street Journal, Conversational Telephone Speech (CTS), Broadcast News (BN), and Meetings (MTG) to name a few. For these domains, the community agreed to declared sections of simultaneous speech as 'not scoreable'. While this had minor impact on most of these domains, the highly interactive nature of Meeting speech rendered a very large fraction of the test material not scoreable. This paper documents a multi-dimensional extension of the Dynamic Programming solution to Levenshtein Edit Distance calculations capable of evaluating STT systems during periods of overlapping, simultaneous speech.

1. Introduction

The Rich Transcription (RT) evaluation series have focused on the building technologies that generate "rich transcriptions". Rich transcriptions are defined as the combined output of Speech-To-Text (STT) systems and metadata detection systems. Past RT evaluations have included technology evaluation tasks for STT systems and metadata technologies like SUs, Disfluencies, Diarization "Who Spoke When", and Diarization "Source Localization" (Fiscus et al., 2004).

A roughly annual series of RT evaluations (Garofolo et al., 2004; Fiscus et al., 2005) since 2000 has focused on developing RT technologies for the Meeting Domain. The Meeting Domain contains a significant amount of overlapping speech, however existing STT evaluation tools were not capable of scoring simultaneous speech. A prototype tool was developed for the 2004 RT evaluation that successfully scored up to three simultaneous speakers. The prototype tool was an adaptation of the SCLITE token alignment engine in the NIST Scoring ToolKit[1] (SCTK) to support the alignment of three or more Directed Acyclic Graphs (DAGs) of word tokens simultaneously. The SCLITE DAG alignment algorithm itself was suggested is an extension to the Dynamic Programming (DP) solution to Levenshtien Edit Distance calculations proposed in Kruskal and Sankoff (1983).

While the PERL implementation proved feasibility, it could only score up to three simultaneous in the 2004 RT test set in a reasonable amount of time. In 2005, a new C++-based NIST SCTK alignment module, ASCLITE, was built and released to the RT community which was capable of scoring up to 5 simultaneous speakers in a reasonable amount of time. By using appropriate constraints during the alignment process, the procedure was used to score 98% of the Rich Transcription Spring 2005 (RT-05S) (Fiscus et al., 2005) distant microphone test sets.

Three main topics will be described in this paper. First, we will discuss our three Stream-Based evaluation models for STT systems. Second, we will discuss the techniques used by the ASCLITE alignment engine to perform the Stream-Based alignments by briefly introducing the DP solution to sequence alignment and describing in more detail the extensions to DAG alignments and multi-DAG alignments. Thirdly, we will provide a brief indication of the results gleaned from the RT-05S evaluation.

2. STT Evaluation with Multi-Stream Models

Simultaneous overlapping speech presents a clear challenge for both STT systems and the STT evaluation protocols. As evidenced in the recent DARPA EARS evaluations, and for that matter recent RT evaluations, system developers have not directly addressed overlapping speech within their systems. Instead they have focused on decoding a single person's speech. The approach has lead to remarkable improvements in performance (Fiscus et al., 2005), but the pervasiveness of overlapping speech indicates the issue must be addressed. Figure 1 shows a plot of testable material in the RT-04S, and RT-05S evaluations as a function the number of active speakers.

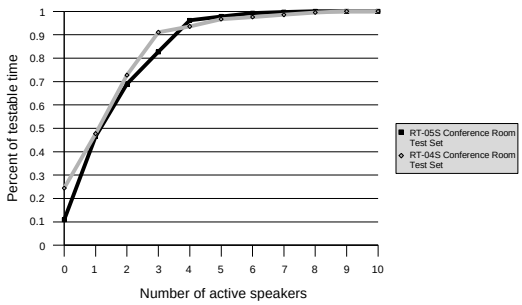

Figure 1: Cumulative percentage of testable time in various Meeting data test sets.

Clearly, ignoring overlapping speech (51% of the RT-05S conference room test set) means a large fraction of the STT challenge is not being evaluated. The remainder of this section describes the requirements NIST imposed on the scoring protocol and then explains the three stream

1 http://www.nist.gov/speech/tools/index.htm

based evaluation models:

- Single system-to-single reference stream alignments
- Single system-to-multiple reference stream alignments
- Multiple system-to-multiple reference stream alignments

2.1 Word Sequence Streams

A natural way to think of transcript is the "Word Sequence Stream" model. The speaker, or any noise source for that matter, emits a stream of annotatable events. Each stream is independent, e.g., speakers talk over one another or doors open and shut independently, and events attributed to a single stream are sequential and non-overlapping. Thus, a DAG nicely represents a stream.

Streams play an important role in our scoring protocol. Each reference speaker is represented as a separate stream of words. The reference transcripts used in NIST STT evaluations have always used this representation. STT system output can also be represented as a stream. Current state-of-the-art STT systems output a series of time marked words as a single stream since these words do not overlap and are not attributed to a specific speaker. However, it is conceivable given the current work in Blind Source Separation (McDounough et al., 2004), researchers are extremely close to building STT systems capable of multi-stream output in the meeting domain.

All three evaluation models pre-segment the entire recording in order to constrain the alignment search space by building small Reference Segment Groups (RSGs). RSGs are built using the time breaks between reference segments as segmentation points to identify independent time regions of words to align. If two or more reference segments overlap in time, as in the case of simultaneous speech, the RSG includes both segments and potentially more segments until an independent unit is found. If there is a time gap between reference segment, an RSG is created without reference segments so that insertion errors not caused by human sources can be included in the performance statistics. System words are assigned to each RSG by determining in which RSG the word's time midpoint lies.

Inside an RSG, each reference speaker's segments comprise a stream for alignment. The system output comprises a single stream within an RSG unless the system has identified speakers, in which case each system speaker is represented by separate stream.

2.2 Requirements for Evaluating Overlapping Speech

We have motivated the need for evaluation overlapping speech. However, since NIST can not require developers to directly address the challenge, the solution for evaluating overlapping speech must accommodate existing technology as well as prepare for the future.

To bridge the gap between existing technology and the desired vision of RT transcripts with speaker attribution, we required our solution to allow any hypothesized word to map to any reference speaker's word so long as the word sequence, in both the system and reference transcript, is strictly maintained. This obviously gives an advantage to systems, but is indeed more demanding on the evaluation protocol. We refer to this capability as "flexible stream alignment"

The protocol must also support the evaluation of systems that combine STT with speaker attribution. It would be simple to evaluate such a system by performing a tiered scoring protocol that first uses the Speaker Diarization (NIST 2005) evaluation protocol to compute system-to-reference speaker mappings followed by the string alignment on the mapped speakers, but convolving speaker detection and STT errors would yield difficult to interpret results. We believe a better solution is to apply the flexible stream alignment capability to multiple system streams. The advantage is that we can evaluate a system with and without taking into account the system's stream IDs.

2.3. STT evaluation models

We define three STT evaluation models below using the stream-based approach to represent transcripts. All models make use of the Dynamic Programming solution to computing Levenshtein Distances as a means to map, or align, system and reference word sequences onto each other. Section 3 describes the well known alignment method (Levenshtein 1966) and NIST's extensions for evaluating overlapping speech.

2.3.1. Single System-to-Single Reference Stream Alignments

For years the STT research community has used DP string alignments to evaluation STT systems. The NIST SCoring ToolKit (SCTK) contains the 'sclite' string alignment module which implements a two dimensional, DAG alignment algorithm. The algorithm, described in section 3.2, is capable of performing the word-to-word mapping as visualized in Figure 2. The straight lines connect words that are mapped together as either correct or substituted words. The arrows that do not link words together represent insertions and deletions. Note that a single, optimal path is found through both DAGs simultaneously even though many optimal paths may exist.

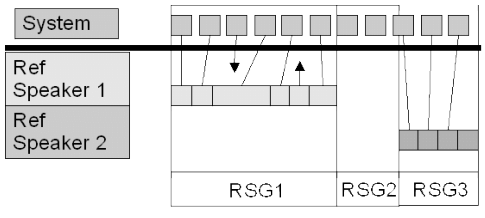

Figure 2: Single System to Single Reference Stream alignment Example.

The system transcript is represented as a single stream, and the reference transcript is represented by another. This evaluation model only works for RSGs with zero or

one reference speaker segments.

2.3.2. Single System-to-Multiple Reference Stream Alignments

As a step towards evaluating speaker attributed STT systems, we have defined the "Single System-to-Multiple Reference Stream Alignments" model. In the model, the system output is still represented as a single stream, but now the reference transcript consists of multiple streams. This evaluation model requires a multi-dimensional search to ensure an optimal alignment solution is found. Section 3.3 describes NIST's multidimensional extension to DP alignments. In the algorithm, the system word stream, and each reference speaker's word streams are considered a dimensions in the alignment.

As Figure 3 shows, "flexible stream alignment" permits system words to be mapped to either reference stream.

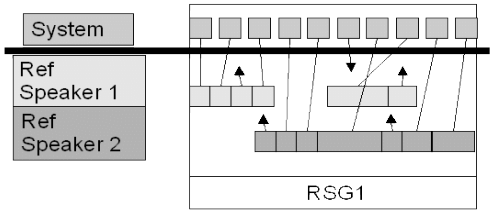

Figure 3: Single System to Multiple Reference Stream alignment example.

While this method can be effectively used to score overlapping speech, attributing insertion errors to reference speakers is undefined during periods of overlap. For instance the 6th system word could be assigned to either reference speaker. ASCLITE uses the following heuristics to assign insertions to reference speakers.

- Insertions with a midpoint occurring in one reference stream are associated with that speaker.
- Insertions with a midpoint occurring in multiple reference streams are distributed equally amongst the streams.

2.3.3. Multiple System-to-Multiple Reference Stream Alignments

To evaluate systems capable of STT with speaker attribution, we have defined the "Multiple System-to-Multiple Reference Stream Alignments" evaluation model. Figure 4 contains a picture of this model where both the system and reference speakers are represented as multiple streams. The same multi-dimensional alignment method described in Section 3.3 is capable of solving this alignment. The only change is now each system-defined speaker is a stream.

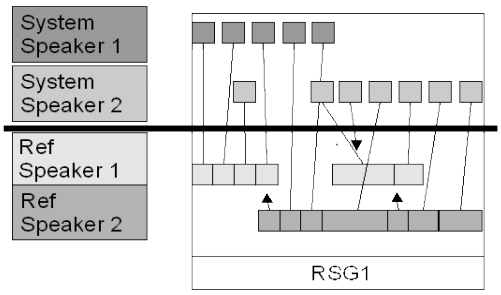

Figure 4: Multiple System-to-Multiple Reference Stream alignment example

Like the single system-to-multiple reference model, insertion errors are assigned using the heuristics define in section 2.3.2.

3. String Alignments

Levenshtein Distance calculations (Levenshtein 1966) are an efficient and reliable method for computing string alignments. The Levenshtein distances are used in spell checking, DNA analysis, and speech recognition. The Levenshtein Distance is a measurement between two strings that represents the smallest number of insertions, deletions, and substitutions required to change the first string into the second.

In speech recognition evaluation, the Levenshtein Distance is used to compare two linear graphs where nodes are words rather than two strings of characters. This difference with the original algorithm is minimal because the process is still the same. These two strings of words are the System and the Reference streams as defined in the "STT Evaluation Models" section.

3.1 Dynamic Programming Solution

Levenshtein distances are commonly computed with a Dynamic Programing Solution (DPS). The solution is a two pass algorithm. The first pass fills the 2 dimensional Levenshtein Distance Matrix (LDM) and the second is a back trace to retrieve the list of operations to transform the first string of words into the second string of words.

Each rank in the LDM represents a word from one string and each file from the other string. Transitioning from file to file, or rank to rank, represents a step in the transformation where one word is consumed. The cells contain a single number which is the minimum edit cost to arrive at the cell.

3.1.1 Pass 1

The LDM is filled by looking backward from the current search location to find the lowest transition cost into the current cell using the following formula which is called the cost model.

$$d_{i,j} = min \begin{cases} C_{Del} + \min_{y \in pred(j)} d_{i,y} \\ C_{Ins} + \min_{x \in pred(i)} d_{x,j} \\ C_{Subs}(i,j) + \min_{x \in pred(i)} \min_{y \in pred(j)} d_{x,y} \\ C_{Cor}(i,j) + \min_{x \in pred(i)} \min_{y \in pred(j)} d_{x,y} \end{cases} \quad (1)$$

Where:

- $d_{i,j}$ is the minimum distance to cell *(i,j)*
- C_{Ins}, C_{Del}, C_{subs} and C_{Cor} are the predefined costs of Insertion, Deletion, and Substitution
- *pred(i)* are the predecessor coordinates of the word with the coordinate *i.*

In the 2-D topology, the distance is computed by taking the minimum of the predecessors distance plus the transition cost as represented by the arrows in the Figure 5. In the sentence "The Dog", "The" is the predecessor of "Dog" and in the sentence "A Cat", "A" is the predecessor of "Cat".

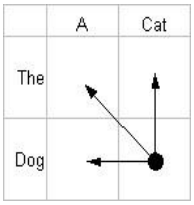

Figure 5: Linear Graph Predecessor Structure

The transformation steps are used to determine which words are correct, substituted, inserted or deleted. The diagonal transition hypothesizes the alignment ('Dog', 'Cat'), the horizontal transition hypothesizes the alignment (*, 'Cat'), and the vertical transition hypothesizes the alignment ('Dog', *). Alignments involving the words can be either correct or substitutions, while the alignments with '*' are either insertions or deletions depending on which dimension is the reference.

The transition costs are defined a priori. For STT scoring, we use the values 0, 3, 3, 4 for C_{cor}, C_{Ins}, C_{Del}, and C_{subs} respectively. The values were chosen to prefer a substitution over adjacent insertion/deletion pairs. Thus, the measured error rate is minimized.

3.1.2 Pass II

The second pass (look back) occurs after all cells of the LDM have been computed. The look back is the process to find one of the minimal paths from the last cell of the LDM to the first cell. For every step of the look back, the next cell of the path is selected using the predecessors model (Figure 5) and the transition costs and values in the predecessor LDM cells. While ties often exist between alternative steps backward, any of the alternatives will yield the same minimal cost alignment.

This algorithm is a *O(m*n)* where *m* and *n* are the length of the strings to align.

The following example is the alignment of the two sentences: "O Brother Where Art Thou" and "Where Are You Now". Figure 6 shows the Levenshtein Distance Matrix and Figure 7 the final alignment, where D is an Deletion, I is an Insertion, C is a Correct word, and S is a Substitution.

		Where	Are	You	Now
	0	3	6	9	12
O	3	4	7	10	13
Brother	6	7	8	11	14
Where	9	6	9	12	15
Art	12	9	10	13	16
Thou	15	12	13	14	17

Figure 6: 2-D Levenshtein matrix

```
O Brother Where Art Thou  -
-   -     Where Are You  Now
D   D       C   S   S    I
```

Figure 7: Alignment of the two strings

3.2 Directed A-cyclic Graph topology extension

To extend the classical topology of a linear strings of words, Kruskal and Sankoff (1966) introduced the notion of aligning Directed A-cyclic Graphs with the DPS. The proposed DAG structure must have a single start node {S} and a single end node {E}. Unlike the linear graphs, every internal node can have multiple previous nodes and/or multiple next nodes.

This structure and topology permits alternative transcriptions to be represented in the word sequence. This is particularly useful for representing unclear speech. Figure 8 is an example of an ambiguous sentence that could be either "This is the cat" or "This is a cat". The words "the" and "a" are both allowable in the sentence.

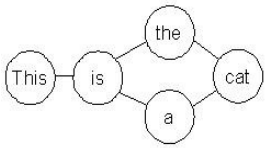

Figure 8: A Directed A-cyclic Graph

To accommodate the new topology, the DPS is been extended in two ways. First, the rank and files represent the topologically sorted list of nodes in the graph. This makes the LDM cell computations simpler because predecessor LDM cells can easily be calculated before the current cell's computation. The second change to DPS is to extend the *pred(i)* function in formula (1) to list all possible transitions into the node as defined by the DAGs.

The following example presents the predecessor computation (Figure 11) for entrances into the final cell of the LDM when aligning the two DAGs in Figures 9 and 10.

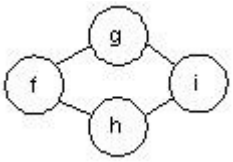

Figure 9: DAG 1

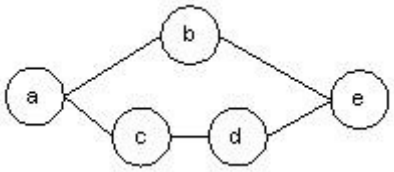

Figure 10: DAG 2

	a	b	c	d	e
f					
g					
h					
i					

Figure 11: Predecessors for the two DAGs

In DAG 1, the predecessors of "i" are "g" and "h". In DAG 2, the predecessors of "e" are "b" and "d". Thus to compute the distance of the cell (e, i), the LDM cells for (e,h) and (e,g) are considered for a Deletion, cells (d,i) and (b,i) are considered for a Insertion, and cells (d,h), (b,h), (d,g), and (b,g) are considered for the Substitution of Correct (depending on if e and i are different or the same).

3.3 Multi-dimensional extension

Two dimensional alignments are not able to align regions of overlapping speech. The technique that we have successfully used to align overlapping speech is to extend the DPS to three or more dimensions. Like the extensions for DAGs alignments, the algorithm was changed in two ways. The first change was to extend distance matrix to multiple dimensions. Coordinates, of course, are represented as vectors.

The second change is to the cost function. The cost function requires drastic changes compared to the DAG extensions because the concepts of correct and substitutions do not map to the higher dimensional alignments. Consider the hypothesized alignment of ("the", "the", "one", *, *) as a multi-dimensional extension of the representation in section 3.1.1. Correctness is replaced with the notion of most frequently occurring identical words. Since the word "the" occurs twice, we assume it should be the least penalized. Any word in the hypothesized alignment that is not one the most frequently occurring words is penalized as a substitution, "one" in the example. The two "*"s are considered InDels (short for insertions or deletions).

The resulting cost formula is (2) and (3).

$$LDM(\vec{A}) = \min_{\vec{B} \in pred(\vec{A})} \{Trans(\vec{B}, \vec{A}) + LDM(\vec{B})\} \quad (2)$$

$$Trans(\vec{B}, \vec{A}) = C_{InDel} * InDels + C_{Subs} * Subs \quad (3)$$

where:

- $\vec{A}$ is a vector indicating the cell in the LDM
- $pred(\vec{A})$ returns the set of predecessor LDM cells for the vector $\vec{A}$
- C_{InDel} is the a priori cost of and InDel
- C_{Sub} is the a priori cost if a substitution
- *InDels* is the number of InDels
- *Subs* is the number if substitutions

For the two dimensional case, the formula simplifies to the same formula as formula (1).

3.1.1 Computational Considerations

The multi-dimensional algorithm is very computationally demanding and memory intensive. The computational complexity is $O(m^n)$ where m is the average segment length and n is the number of dimensions. The LDM matrix has m^n elements and the *pred()* function visits 2^n-1 LDM cells. Fortunately, the computational requirements can be reduced by applying application constraints for STT evaluations.

In both multi stream alignment models, we have introduced following requirements of the alignments:

1. One word from the system can only be aligned with one and only one reference word.
2. A system word and one or more of the reference words can not be simultaneously inserted or deleted.

By applying these requirements, the 2^n-1 visits of the *pred()* functions can be reduced to formula 4.

$$C_s^1 + C_r^1 + C_s^1 \times C_r^1 = s + r + s \times r \quad (4)$$

Where:

s is the number of system streams
r is the number of reference streams
and $s + r = n$ the number of dimensions

From Table 1 we can see the potential savings.

System Streams	Reference Streams	2^n-1	Reduced computation	
			Number	Percentage
1	1	3	3	0%
1	5	63	11	83%
2	2	15	8	47%
5	5	1023	35	97%

Table 1: Look back reductions as a result of STT alignment application contraints

4. Application to RT-05S MDM STT Systems

The Single System-to-Mutliple Reference Stream

Alignment Model was used for the Rich Transcription Spring 2005 (RT-05S) Meeting Recognition STT evaluation (Fiscus et al., 2005) for the Multiple Distant Microphone (MDM) condition. Table 2 contains the results for ICSI/SRI's (Stolcke et al., 2005) primary system on the MDM audio input condition.

	Number of Active Reference Speakers				
	0+1	2	3	4	5
WER	30.2%	38.9%	42.7%	49.5%	50.4%

Table 2: ICSI/SRI RT-05S Conference Room Primary MDM STT System results split by the number of active speakers per segment

By combining all the alignment extensions and using the computation reduction techniques, we were able to score all regions with up to 5 overlapping reference speakers which, from Figure 1, constituted 98% of the test set. The scorer took approximately two hours of computation on a MAC G5 with a duel core 2.3Mhz processor system.

5. Conclusion

In this paper we have discussed the need for evaluating STT systems during simultaneous overlapping speech and three stream-based STT evaluation models to accomplish the evaluation. The three models are:

- Single system-to-single reference stream alignments
- Single system-to-multiple reference stream alignments
- Multiple system-to-multiple reference stream alignments

The models allow system words to be flexibly aligned to any reference speaker. This technique allows existing STT technologies that do not identify who said which word to be evaluated as well as future systems that are capable of indicating the source of each word.

The models rely on the use of a generalization to the DP solution to Levenshtein Edit Distance calculations to three or more dimensions. The computational complexity is managed by reducing the search space to permit legitimate word mappings where a system word can only map to a reference word and vice versa.

We successfully scored up to five simultaneous speakers using in the RT-05S Conference Room test set which comprised 98% additional test material.

6. References

Kruskal, J. B., & Sankoff, D., (1983). An overview of sequence comparison, Time warps, string edits, and macromolecules: The theory and practice of sequence comparison, ed. D. Sankoff, and J.B. Kruskal. (Reading, MA: Addison-Wesley) 382, ISBN 0-201-07809-0.

Levenshtein, V. I. (1966). Binary codes capable of correcting deletions, insertions and reversals, Soviet Physics-Doklady 10, 707-710.

Fiscus, J., Radde, N., Garofolo, J., Le, A., Ajot, J., & Laprun, A., (2005). The Rich Transcription 2005 Spring Meeting Recognition Evaluation: NIST MLMI Meeting Recognition Workshop, Edinburgh. To appear in Springer Lecture Notes in Computer Science Series, Volume 3869, S. Renals and S. Bengio, editors.

Garofolo, J., Laprun, C., & Fiscus, J., 2004, "RichTranscription 2004 Spring Meeting Recognition Evaluation", (2004). in Proc. ICASSP-2004 Meeting Recognition Workshop; Montreal, Canada.

McDonough, J., Raub, D., Wolfel, M., & Waibel, A., (2004) Towards Adaptive Hidden Markov Model Beamformers, submitted to IEEE Transactions on Speech and Audio Processing

NIST, (2005). Rich Transcription Spring 2005 Meeting Recognition Evaluation Plan, http://www.nist.gov/speech/tests/rt/rt2005/spring/

NIST, (2004). Rich Transcription Spring 2004 Meeting Recognition Evaluation Plan, http://www.nist.gov/speech/tests/rt/rt2004/spring/

Stolcke, A., Anguera, X., Boakye, K., Cetin, O., Grezl,, F. Janin, A., Mandal, A., Peskin, B., Wooters, C. & Zheng, J., (2005). Further Progress in Meeting Recognition: The ICSI-SRI Spring 2005 Speech-to-Text Evaluation System. Proc. NIST MLMI Meeting Recognition Workshop, Edinburgh. To appear in Springer Lecture Notes in Computer Science Series, Volume 3869, S. Renals and S. Bengio, editors.

Fiscus, J., Garofolo, J., Le, A., Martin, A., Pallett, D., Przybocki, M., & Sanders, G., (2004). "Results of the Fall 2004 STT and MDE Evaluation", in the Proc. of the Fall 2004 Rich Transcription Workshop.