ON WORD ERROR RATE DEFINITIONS AND THEIR EFFICIENT COMPUTATION FOR MULTI-SPEAKER SPEECH RECOGNITION SYSTEMS

Thilo von Neumann[1], *Christoph Boeddeker*[1], *Keisuke Kinoshita*[2], *Marc Delcroix*[2],
Reinhold Haeb-Umbach[1]
[1]Paderborn University [2]NTT corporation, Japan

ABSTRACT

We propose a general framework to compute the word error rate (WER) of ASR systems that process recordings containing multiple speakers at their input and that produce multiple output word sequences (MIMO). Such ASR systems are typically required, e.g., for meeting transcription. We provide an efficient implementation based on a dynamic programming search in a multi-dimensional Levenshtein distance tensor under the constraint that a reference utterance must be matched consistently with one hypothesis output. This also results in an efficient implementation of the ORC WER which previously suffered from exponential complexity. We give an overview of commonly used WER definitions for multi-speaker scenarios and show that they are specializations of the above MIMO WER tuned to particular application scenarios. We conclude with a discussion of the pros and cons of the various WER definitions and a recommendation when to use which.

Index Terms— word error rate, meeting recognition, Levenshtein distance

1. INTRODUCTION

While Automatic Speech Recognition (ASR) systems may be complex, at least their generally accepted performance measure, the Word Error Rate (WER), is seemingly simple. It is given by the Levenshtein distance between the recognized word sequence and the ground truth transcription divided by the number of words in the ground truth transcription. The Levenshtein distance is defined as the minimal number of substitution, insertion and deletion operations required to turn one word string into another. The distance can be efficiently computed by use of dynamic programming, and tools, such as NIST's `sclite` [1], are widely used in the community.

Conventional ASR systems process recordings of a single speaker at the input and output a single transcription, hence we call them SISO. However, today's ASR systems have emerged from SISO systems to systems that process recordings of multiple speakers at their input (potentially overlapping), and produce multiple output word sequences, e.g., for meeting transcription. We call these systems MIMO.

The final goal for a MIMO transcription system is answering the question who spoke what and when, i.e., providing both temporal information and speaker labels together with the transcribed words [2, 3]. The input to a MIMO system can contain speech overlaps, which poses a challenge not only for ASR but also for WER computation, where the conventional WER definitions are not applicable. Many works on ASR, however, focus only on sub-problems and provide varying degrees of detail, e.g., neglecting speaker identification [4–7], temporal information [6, 8] or even utterance order [6]. Evaluating such systems poses an additional challenge as there is no clearcut definition of WER that serves all purposes.

Consequently, a number of WER definitions have emerged that are tailored to specific use cases, such as utterance-wise WER (uWER) [5], Speaker Attributed WER (SA-WER) [9], Concatenated Minimum Permutation WER (cpWER) [10], and Optimal Reference Combination WER (ORC WER) [7]. Which of these WER definitions are applicable heavily depends on the tackled problem and applied network architecture; not all definitions are applicable in all cases. We here propose a generalized model for computing a WER in MIMO situations that poses minimal requirements w.r.t. ancillary information: It does neither require speaker labels nor timing information. It can consequently be used as a scoring tool for arbitrary ASR engines providing arbitrarily detailed information. It treats an utterance as an atomic unit that always has to appear continuously in an ASR system's output and ensures that the order of utterances uttered by each speaker is maintained. We show that our MIMO WER definition is a generalization of many existing WER definitions.

Equally important to a sound definition of WER for MIMO is an efficient algorithm for computing it. Finding the matching between multiple input and output sequences is a combinatorial problem that can become intractable with a naive implementation. In [7, 11], for example, it was stated that the ORC WER could not be computed for a recording containing many utterances. An efficient algorithm for the evaluation of multi-output models was presented in [12], and released under the name `asclite`. It casts the alignment of a system output to multiple reference transcriptions as a multi-dimensional Levenshtein distance calculation that can be computed by dynamic programming. Their interpretation of the multi-dimensional Levenshtein distance, however, allows words to be transcribed on an arbitrary output channel, i.e. it is not counted as an error when an utterance is split over multiple channels (which the MIMO WER and many other WER definitions penalize). This type of error, however, can render a transcription practically unusable. Their tool additionally requires detailed temporal information about word begin and end times, which is not readily available for all of today's ASR approaches, e.g., End-to-End systems [8, 11].

We provide an efficient implementation of the MIMO WER that is also based on the multi-dimensional Levenshtein distance, but includes the constraint that an utterance must be matched consistently and uninterruptedly with one hypothesis channel. It leads to an efficient implementation of the ORC WER with a complexity that is polynomial in the number of utterances instead of exponential, as it was the case in [7].

The contributions of this paper are as follows: (a) We propose a generalized WER for MIMO ASR systems and present an efficient implementation that also leads to efficient implementations of other WER definitions, (b) we discuss its relationship to and pitfalls of existing WER definitions, and (c) we release a software package for MIMO WER computations named `MeetEval`[1].

[1] `https://github.com/fgnt/meeteval`

2. MIMO WER

Recently, transcription systems have emerged that handle complex recordings containing speech of multiple speakers or speech overlaps. Such systems process multi-party speech and produce a hypothesis that can contain multiple channels, thus we here call them MIMO ASR[2]. For such systems, an alignment between multiple references (e.g., one per speaker) and multiple hypothesis channels (not necessarily one per speaker) has to be found before a WER can be computed.

A common way to compute a WER in such a case is utterance-wise evaluation, i.e., utterance-wise WER (uWER) [4, 5, 13]. A recording is here not evaluated in its entirety, but utterances are extracted from the input signal using ground truth utterance begin and end times before the ASR system is applied. For each utterance, the SISO WER is computed and the output channel is selected that minimizes the error. Among others, this has the drawback that any text produced by the ASR system outside the ground truth utterances is ignored, so the performance is constantly over-estimated.

For a more realistic assessment an evaluation is needed of the full hypothesis, also called continuous evaluation [5]. While an ideal MIMO ASR system provides information about speaker identities and timing in addition to the transcription, i.e., answering the question "who said what and when?" [2, 3], many systems focus only on sub-problems and neglect speaker identification [4–7] or temporal alignments [6, 8]. Many of the already proposed WER definitions for continuous evaluation (see Section 3 for discussion) are not applicable in all scenarios.

2.1. A generalized WER definition for multi-speaker ASR

We propose two natural assumptions for a WER model that satisfies all use cases stated before: (i) The order of utterances produced by the same speaker cannot be changed by the ASR system; and (ii) a continuous portion of speech uttered by a single speaker without significant pauses should always appear continuously on one hypothesis channel. We call such a continuous portion of speech an utterance and a violation of assumption (ii) a channel switch. Note that these two assumptions do not state any temporal ordering between utterances of different speakers. We do not assume that a system estimates diarization or speaker information because we argue that the WER should be applicable even if such information is not available.

Given I reference sequences of utterances $\mathcal{R}_i$ and J hypothesis channels $\mathcal{H}_j$ as sequences of words, we find the assignment of reference utterances to hypothesis channels that minimizes the SISO WER over all outputs, respecting the aforementioned assumptions (i) and (ii). We call this error rate the MIMO WER. It does only count transcription errors and no speaker attribution errors (in case speaker information is available).

Finding the MIMO assignment is computationally demanding as the number of valid assignments is exponential in I, J and the number of utterances in the references. The next section discusses how to find such an assignment for the MIMO WER by an extension of the Levenshtein distance.

2.2. Efficient computation of the MIMO WER

The Levenshtein distance [14] between two word sequences $\mathcal{R}$ and $\mathcal{H}$ can be efficiently computed with the Wagner-Fischer algorithm [15], which is a dynamic programming algorithm. With the indices r and h, representing indices into the word sequences $\mathcal{R}$ and $\mathcal{H}$, respectively, a two-dimensional distance matrix L is filled as follows: starting the recursion with $L(0,h) = h$ and $L(r,0) = r$ the entries of the matrix are recursively computed as

$$L(r,h) = \min \begin{cases} L(r-1,h-1) + C_{\text{corr/sub}} \\ closeL(r,h-1) + C_{\text{ins}} \\ L(r-1,h) + C_{\text{del}}, \end{cases} \tag{1}$$

where $C_{\text{corr/sub}}, C_{\text{ins}}$ and C_{del} are the costs of a correct match or substitution, an insertion, and a deletion operation, respectively. After L has been filled, the Levenshtein distance between the two word sequences is given by $L(|\mathcal{R}|, |\mathcal{H}|)$, where $|\cdot|$ denotes the length of the sequence. The costs for substitution or deletion are $C_{\text{corr/sub}} = C_{\text{corr}}$ if $\mathcal{R}(r) = \mathcal{H}(h)$ and $C_{\text{corr/sub}} = C_{\text{sub}}$ otherwise. While for a correct recognition $\mathcal{R}(r) = \mathcal{H}(h)$ the transition cost is zero ($C_{\text{corr}} = 0$), substitutions, insertions and deletions incur costs of typically $(C_{\text{sub}}, C_{\text{ins}}, C_{\text{del}}) = (1,1,1)$, while other values have also been used [12, 16]. The optimal alignment between reference and hypothesis can be found by a backtracking pass through L, starting from $(|\mathcal{R}|, |\mathcal{H}|)$ and moving along the path given by the transitions that achieved the minimum in Eq. (1), until ending in $L(0,0)$.

While a brute force computation of all possible alignments between $\mathcal{R}$ and $\mathcal{H}$ would have a complexity that is exponential in the number of words, this dynamic programming algorithm has a complexity that grows only linearly with the length of the word sequences. Its complexity is given by $\mathcal{O}(3|\mathcal{R}||\mathcal{H}|)$ where the factor of 3 is the number of computations required to obtain one matrix element, see Eq. (1).

2.2.1. Multi-dimensional Levenshtein distance

Moving from the above single reference and single hypothesis case to I references and J hypothesis channels $\mathcal{R}_1, ..., \mathcal{R}_I$ and $\mathcal{H}_1, ..., \mathcal{H}_J$, the computational complexity can again be dramatically reduced from a brute-force search by extending the two-dimensional Levenshtein distance matrix to a multi-dimensional tensor. The indices into L are extended to multi-indices $\mathbf{r} = (r_1, \dots, r_I)$ and $\mathbf{h} = (h_1, \dots, h_J)$, where r_i indexes $\mathcal{R}_i$ and h_j indexes $\mathcal{H}_j$. If we first neglect assumption (ii) of the MIMO WER, the distance tensor is recursively filled by:

$$L(\mathbf{r},\mathbf{h}) = \min_{\substack{i\in\{1,\dots,I\} \\ j\in\{1,\dots,J\}}} \begin{cases} L(\mathbf{r}-\mathbf{e}_i^{(I)}, \mathbf{h}-\mathbf{e}_j^{(J)}) + C_{\text{corr/sub}} \\ L(\mathbf{r}-\mathbf{e}_i^{(I)}, \mathbf{h}) + C_{\text{del}} \\ L(\mathbf{r}, \mathbf{h}-\mathbf{e}_j^{(J)}) + C_{\text{ins}}, \end{cases} \tag{2}$$

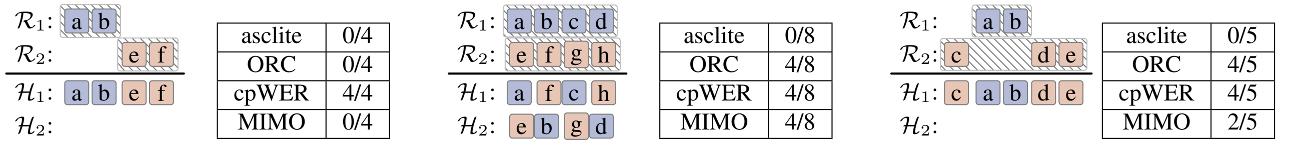
where $\mathbf{e}_i^{(I)}$ is an I-dimensional multi-index containing all zeros, except for a one in the i-th position. Similarly, $\mathbf{e}_j^{(J)}$ is J-dimensional with a one on the j-th component.

Eq. (2) is equivalent to the derivations in [12] for simple word strings and simplifies to Eq. (1) for $I = J = 1$. Filling L with Eq. (2) requires $\mathcal{O}((IJ+I+J)(\prod_i^I |\mathcal{R}_i|)(\prod_h^J |\mathcal{H}_h|))$ steps, which is exponential in the number of references I and hypothesis channels J. Here, the number of computations to obtain one element of L is given by IJ, the number of substitutions, plus I deletions plus J insertions. This formulation implies the first MIMO assumption that the order of utterances produced by the same speaker must not change, but it does not include the second assumption that an utterance should be consistent on an output.

2.2.2. Proposed: MIMO Levenshtein distance

In order to include utterance splits in the error count, i.e., assumption (ii), we introduce a channel change token <ct> that is inserted between utterances in all reference word sequences $\mathcal{R}$. Updates across

[2] MIMO here does not refer to multiple microphone inputs. This discussion holds for both single- and multi-microphone systems.



asclite	0/4
ORC	0/4
cpWER	4/4
MIMO	0/4

(a) cpWER counts diarization errors while speaker-agnostic WERs do not.

asclite	0/8
ORC	4/8
cpWER	4/8
MIMO	4/8

(b) `asclite` does not penalize channel switches within an utterance.

asclite	0/5
ORC	4/5
cpWER	4/5
MIMO	2/5

(c) ORC WER can over-estimate the WER when annotations are faulty.

Fig. 1: Toy examples to demonstrate differences of WER definitions. Each box a represents a word and a light gray box ▨ represents an utterance. Errors counted by the different WER definitions are given in the tables.

Table 1: Comparison of continuous evaluation WERs.

	speaker-attributed	speaker-agnostic		
	cpWER	asclite	ORC	MIMO
Penalizes				
Speaker Confusion	✓	–	–	–
Channel Switches	✓	–	✓	✓
Temporal Errors	–	✓	(✓)[3]	–
Examples (applicable to)				
CSS + Diar. [3]	✓	(✓)[4]	(✓)[4]	(✓)[4]
Diar. + Sep. + ASR [10]	✓	(✓)[4]	(✓)[4]	(✓)[4]
CSS + ASR ($J < I$) [5]	–	✓	✓	✓
SOT ($J = 1$)				
FIFO [6, 8]	–	–	✓	✓
+ speaker ID [2]	✓	–	✓	✓
non-FIFO [6]	–	–	–	✓
MT-RNN-T [20] SURT [11] ($J < I$)	–	(✓)[5]	✓	✓

references and hypotheses are only allowed at these change tokens. We represent the currently active mapping of reference to hypothesis channel by a pair (i, j) where reference $\mathcal{R}_i$ is matched to hypothesis channel $\mathcal{H}_j$. The constrained update equation to compute the MIMO WER is given by (neglecting the superscript of the multi-index):

$$L_{i,j}(\mathbf{r}, \mathbf{h}) = \begin{cases} \min\limits_{\substack{\tilde{i} \in \{1,\dots,I\} \\ \tilde{j} \in \{1,\dots,J\}}} L_{\tilde{i},\tilde{j}}(\mathbf{r} - \mathbf{e}_{\tilde{i}}, \mathbf{h}); \quad \text{if } \mathcal{R}_i(r_i) = \text{<ct>} \\ \min \begin{cases} L_{i,j}(\mathbf{r} - \mathbf{e}_i, \mathbf{h} - \mathbf{e}_j) + C_{\text{corr/sub}} \\ L_{i,j}(\mathbf{r} - \mathbf{e}_i, \mathbf{h}) + C_{\text{del}} \\ L_{i,j}(\mathbf{r}, \mathbf{h} - \mathbf{e}_j) + C_{\text{ins}}. \end{cases} \end{cases} \tag{3}$$

The lower min operation is equal to the two-dimensional string matching in Eq. (1) along the slice of L determined by i and j which allows leveraging efficient algorithms for Levenshtein distance computation, such as [17, 18], also in the multi-dimensional case. Eq. (3) collapses to Eq. (2) if every second word in every reference $\mathcal{R}_i$ is <ct>. Using Eq. (3) causes L to be relatively sparse in $\mathbf{r}$ which slightly reduces the complexity compared to Eq. (2). When choosing $\mathbf{r}$ as a scalar ($I = 1$), this formulation allows for an efficient computation of the ORC WER (see Section 3.2.2).

Computing the MIMO or ORC WER is NP-hard for arbitrary I and J. This can be proven by showing that the string MERGE problem, which is NP-complete [19], can be solved by checking if the MIMO WER for specifically arranged inputs is zero. The MIMO assignment must be at least as hard as MERGE, thus, NP-hard.

3. DISCUSSION

Table 1 gives a short overview of the presented WER definition and shows example use-cases where each one is applicable. Fig. 1 displays small scoring examples to highlight their differences.

3.1. Speaker attributed WERs

It is common to compute a Speaker Attributed WER (SA-WER) [9, 10] when an ASR system provides speaker labels, e.g., for Speaker Attributed ASR (SA-ASR) [2]. SA-WER judges both, transcription and speaker attribution errors. The SA-WER definition from [9] can be computed when the estimated speaker labels represent the true speaker identity, i.e., the mapping between estimated speaker labels and reference labels is known. In recent publications, the cpWER [10] is preferred over the SA-WER. It is the overall SISO WER across all channels for the permutation of reference speakers $|\mathcal{R}_i|$ to hypothesis speakers $|\mathcal{H}_j|$ with the minimal error.

The cpWER is a special case of the MIMO model with the constraint that the mapping between reference speakers $|\mathcal{R}_i|$ and hypotheses $|\mathcal{H}_j|$ is bijective, i.e., a permutation in $\mathcal{R}_i$ and $J = I$. Under-estimation (i.e., when $J < I$) is handled by adding empty dummy channels until $J = I$, and over-estimation is handled implicitly by having hypothesis channels that are not matched with any reference. The cpWER is an upper bound on the MIMO WER since it reduces the number of assignments. It can be computed in polynomial time with the Hungarian algorithm [21–23]. An example for the cpWER judging speaker attribution errors is shown in Fig. 1a.

3.2. Speaker agnostic WERs

If speaker information is not available, e.g., for Continuous Speech Separation (CSS) systems [5], a speaker-agnostic WER has to be computed. A speaker-agnostic WER can also be useful when speaker information is available as the difference between a speaker-attributed and a speaker-agnostic WER indicates how many errors can be attributed to mistakes regarding diarization.

3.2.1. asclite

A widely used tool for computing a speaker-agnostic WER is the `asclite` tool [12], as used, e.g., in [5, 6]. It uses a multi-dimensional Levenshtein distance algorithm (see Eq. (2)) to match multiple references to multiple hypothesis channels. It reduces the search space significantly by use of timestamps for words in the references and hypothesis . It has, as such, the tightest bounds on estimated timestamps among the discussed WER definitions and is the only WER that penalizes time annotation errors. This adds the constraint that `asclite` can only be used when the ASR system produces such timing information, which is not available for all methods [8]. It can, nevertheless, have an exploding complexity in certain cases even when temporal alignment information is available [24]. The idea of `asclite` is similar to our MIMO WER when

[3] ORC WER only penalizes temporal errors when the order of utterances is changed by them.

[4] Applicable in theory, but temporal or memory complexity may explode for large numbers of output channels.

[5] This architecture may or may not provide temporal information.

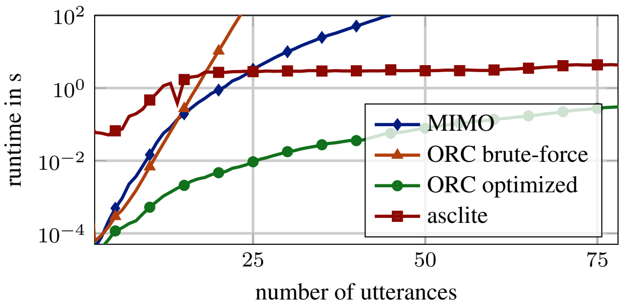


Fig. 2: Runtime comparison of different WERs for a CSS scenario with $I = 4$ speakers and $J = 2$ output channels.

assumption (ii) is dropped, i.e., utterance splits are not counted as an error. It thus produces overoptimistic error rates. An extreme example of this is visualized in Fig. 1b, where words of an utterance are spread over multiple channels which renders the hypothesis practically unusable, while `asclite` computes a WER of zero. All other presented WER definitions respect MIMO assumption (ii).

3.2.2. ORC WER

The ORC WER has been proposed [7] as a speaker-agnostic WER that does not judge temporal alignment but respects MIMO assumption (ii). It is a special case of our MIMO model that merges all I references into one reference $\mathcal{R}^{(\mathrm{ORC})}$, sorted by utterance begin times, and then computes the MIMO WER using only $\mathcal{R}^{(\mathrm{ORC})}$. This is equal the constraint that the global order of utterances must not change. It can thus not be used when the system is allowed to change the ordering of the utterances, as, e.g., a non-FIFO SOT [6] does. Such a re-ordering can also happen implicitly, e.g., when the given timestamps are faulty or ambiguous. Fig. 1c shows an example of such a corner case where two utterances are wrongly labeled as a single utterance and the ORC WER counts all words as substitutions while the ASR system transcribed everything correctly. Such a constellation is, however, often only a corner-case that does not appear frequently in common evaluation scenarios.

The ORC WER is a lower bound on the cpWER since it does not consider speaker errors, and becomes equal to cpWER if no speaker errors are present. It is at the same time an upper bound on the MIMO WER and becomes equal to the MIMO WER when the utterance ordering is unambiguous and the number of errors is small.

The naive implementation in [7] is computationally expensive with a complexity that is exponential in the number of utterances, which caused the authors of [7] to drop examples exceeding 23 utterances for the WER evaluation. The complexity can be reduced to being polynomial in the number of utterances by using Eq. (3) with $I = 1$. It thus allows computing the ORC WER for longer recordings (see Section 4).

4. BENCHMARK

Fig. 2 show the runtime of the different WERs algorithms over the number of utterances to be scored. The benchmark was run on a single core of an AMD Milan 7763 CPU with 2.45GHz in the Noctua2 compute cluster of the Paderborn Center for Parallel Computing (PC^2). We simulated a common CSS scenario with four reference speakers and two system output streams [5]. The cpWER is excluded from the plot because it perform no complex alignment and thus has a much lower complexity compared to the other WERs. The measurements for `asclite` contain a slight offset due to file parsing overhead since no Python interface is available. Note that `asclite` appears very efficient in this benchmark because it uses temporal annotations to narrow down the search space. We here evaluate the MIMO WER in its most general case, i.e., without using temporal information. The MIMO WER can be tuned to use the same information as asclite with a similar complexity and thus runtime.

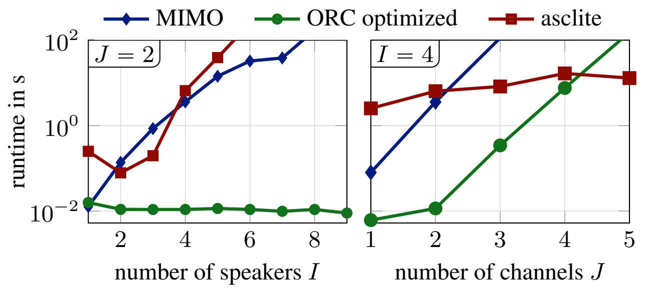


Fig. 3: Runtime comparison for different numbers of speakers I and output channels J. The number of utterances is 25 in both plots.

From Fig. 2 we can see that the MIMO and ORC WERs indeed have a complexity that is polynomial in the number of utterances (concavity of the curve in log scale). The complexity of MIMO WER is greater than that of ORC WER because ORC WER removes the depenency on the number of speakers I. The brute-force implementation of the ORC WER explodes for about 20 utterances, as already reported by [7]. `asclite` shows a behavior that is close to linear in this scenario.

Fig. 3 shows the dependency on the number of speakers and the number of channels, respectively. The complexity of the MIMO WER and `asclite` is exponential in the number of speakers I while the ORC WER has a constant complexity w.r.t. the number of speakers since the references are merged. Being independent of the number of speakers I is important for evaluating CSS systems where the number of speakers I can be arbitrarily large but the number of output channels J is small. The complexity of `asclite` explodes in more complex scenarios, such as generated by the `mms_msg` tool [25]. Here, the only known applicable tool remains our proposed efficient ORC WER implementation.

We can conclude that, while the ORC WER has a few corner cases, it is usually well suited for evaluation with a reasonable execution time. Systems that are incompatible to ORC WER (e.g., non-FIFO SOT) can utilize the MIMO framework and open-source implementation for an efficient WER computation.

5. CONCLUSION

We proposed MIMO WER, a generalized WER definition to assess modern ASR systems that transcribe multiple speakers' utterances to multiple output channels. We embedded it in a comprehensive discussion of existing WERs showing that the MIMO WER can cater a wide range of applications, and that other WER definitions are specialization for particular use cases. An efficient implementation based on a multi-dimensional Levenshtein distance definition is derived for MIMO and ORC, and a channel change token is introduced to penalize the split of utterances over different ASR output channels. The software is provided as an open source tool `MeetEval`[1].

6. ACKNOWLEDGEMENT

Christoph Boeddeker was funded by Deutsche Forschungsgemeinschaft (DFG), project no. 448568305. Computational resources were provided by the Paderborn Center for Parallel Computing.

7. REFERENCES

[1] NIST, "The NIST Scoring Toolkit (SCTK)," 2021.

[2] Naoyuki Kanda, Yashesh Gaur, Xiaofei Wang, Zhong Meng, Zhuo Chen, Tianyan Zhou, and Takuya Yoshioka, "Joint Speaker Counting, Speech Recognition, and Speaker Identification for Overlapped Speech of any Number of Speakers," in *Interspeech 2020*. Oct. 2020, pp. 36–40, ISCA.

[3] D. Raj, P. Denisov, Z. Chen, H. Erdogan, Z. Huang, M. He, S. Watanabe, J. Du, T. Yoshioka, Y. Luo, N. Kanda, J. Li, S. Wisdom, and J. R. Hershey, "Integration of Speech Separation, Diarization, and Recognition for Multi-Speaker Meetings: System Description, Comparison, and Analysis," in *2021 IEEE Spoken Language Technology Workshop (SLT)*, Jan. 2021, pp. 897–904.

[4] Takuya Yoshioka, Hakan Erdogan, Zhuo Chen, Xiong Xiao, and Fil Alleva, "Recognizing Overlapped Speech in Meetings: A Multichannel Separation Approach Using Neural Networks," in *Interspeech 2018*. Sept. 2018, pp. 3038–3042, ISCA.

[5] Z. Chen, T. Yoshioka, L. Lu, T. Zhou, Z. Meng, Y. Luo, J. Wu, X. Xiao, and J. Li, "Continuous Speech Separation: Dataset and Analysis," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, May 2020, pp. 7284–7288.

[6] Naoyuki Kanda, Yashesh Gaur, Xiaofei Wang, Zhong Meng, and Takuya Yoshioka, "Serialized Output Training for End-to-End Overlapped Speech Recognition," in *Interspeech 2020*. Oct. 2020, pp. 2797–2801, ISCA.

[7] Ilya Sklyar, Anna Piunova, Xianrui Zheng, and Yulan Liu, "Multi-Turn RNN-T for Streaming Recognition of Multi-Party Speech," May 2022, pp. 8402–8406.

[8] Naoyuki Kanda, Jian Wu, Yu Wu, Xiong Xiao, Zhong Meng, Xiaofei Wang, Yashesh Gaur, Zhuo Chen, Jinyu Li, and Takuya Yoshioka, "Streaming Multi-Talker ASR with Token-Level Serialized Output Training," Sept. 2022, pp. 3774–3778.

[9] NIST, "The 2009 (RT-09) Rich Transcription Meeting Recognition Evaluation Plan," 2009.

[10] Shinji Watanabe, Michael Mandel, Jon Barker, Emmanuel Vincent, Ashish Arora, Xuankai Chang, Sanjeev Khudanpur, Vimal Manohar, Daniel Povey, Desh Raj, David Snyder, Aswin Shanmugam Subramanian, Jan Trmal, Bar Ben Yair, Christoph Boeddeker, Zhaoheng Ni, Shota Horiguchi, Naoyuki Kanda, Takuya Yoshioka, and Neville Ryant, "CHiME-6 Challenge: Tackling multispeaker speech recognition for unsegmented recordings," in *6th International Workshop on Speech Processing in Everyday Environments (CHiME 2020)*, 2020.

[11] Desh Raj, Liang Lu, Zhuo Chen, Yashesh Gaur, and Jinyu Li, "Continuous Streaming Multi-Talker ASR with Dual-Path Transducers," in *ICASSP 2022 - 2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, May 2022, pp. 7317–7321, ISSN: 2379-190X.

[12] Jonathan G Fiscus, Jerome Ajot, Nicolas Radde, and Christophe Laprun, "Multiple Dimension Levenshtein Edit Distance Calculations for Evaluating Automatic Speech Recognition Systems During Simultaneous Speech," in *Proceedings of the Fifth International Conference on Language Resources and Evaluation (LREC'06)*, Genoa, Italy, 2006, p. 6, European Language Resources Association (ELRA).

[13] Thilo von Neumann, Keisuke Kinoshita, Christoph Boeddeker, Marc Delcroix, and Reinhold Haeb-Umbach, "Graph-PIT: Generalized permutation invariant training for continuous separation of arbitrary numbers of speakers," in *Interspeech*. 2021, ISCA.

[14] Vladimir Iosifovich Levenshtein, "Binary codes capable of correcting deletions, insertions, and reversals," in *Doklady Akademii Nauk*. 1965, vol. 163, pp. 845–848, Russian Academy of Sciences.

[15] Robert A. Wagner and Michael J. Fischer, "The String-to-String Correction Problem," *Journal of the ACM*, vol. 21, no. 1, pp. 168–173, Jan. 1974.

[16] Daniel Povey, Arnab Ghoshal, Gilles Boulianne, Lukas Burget, Ondrej Glembek, Nagendra Goel, Mirko Hannemann, Petr Motlicek, Yanmin Qian, Petr Schwarz, Jan Silovsky, Georg Stemmer, and Karel Vesely, "The Kaldi Speech Recognition Toolkit," 2011.

[17] Gene Myers, "A fast bit-vector algorithm for approximate string matching based on dynamic programming," *Journal of the ACM*, vol. 46, no. 3, pp. 395–415, 1999.

[18] William J. Masek and Michael S. Paterson, "A faster algorithm computing string edit distances," *Journal of Computer and System Sciences*, vol. 20, no. 1, pp. 18–31, Feb. 1980.

[19] Anthony Mansfield, "On the computational complexity of a merge recognition problem," *Discrete Applied Mathematics*, vol. 5, no. 1, pp. 119–122, Jan. 1983.

[20] Ilya Sklyar, Anna Piunova, and Yulan Liu, "Streaming Multi-Speaker ASR with RNN-T," in *ICASSP 2021 - 2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, June 2021, pp. 6903–6907, ISSN: 2379-190X.

[21] Harold W. Kuhn, "The Hungarian Method for the Assignment Problem," *Naval Research Logistics Quarterly*, , no. 2, pp. 83–97, 1955.

[22] J. Munkres, "Algorithms for the assignment and transportation problems," *J. Society for Industrial and Applied Mathematics*, vol. 5, no. 1, pp. 32–38, 1957.

[23] Thilo von Neumann, Christoph Boeddeker, Keisuke Kinoshita, Marc Delcroix, and Reinhold Haeb-Umbach, "Speeding Up Permutation Invariant Training for Source Separation," in *Speech Communication; 14th ITG-Symposium*, Sept. 2020.

[24] Naoyuki Kanda, Jian Wu, Xiaofei Wang, Zhuo Chen, Jinyu Li, and Takuya Yoshioka, "VarArray Meets t-SOT: Advancing the State of the Art of Streaming Distant Conversational Speech Recognition," 2022, Publisher: arXiv Version Number: 2.

[25] Tobias Cord-Landwehr, Thilo von Neumann, Christoph Boeddeker, and Reinhold Haeb-Umbach, "MMS-MSG: A multi-purpose multi-speaker mixture signal generator," in *International workshop on acoustic signal enhancement (IWAENC)*. 2022, IEEE.