Universal-WER: Enhancing WER with Segmentation and Weighted Substitution for Varied Linguistic Contexts

Samy Ouzerrout
University of Orléans
France
samy.ouzerrout@etu.univ-orleans.fr

Abstract

Word Error Rate (WER) is a crucial metric for evaluating the performance of automatic speech recognition (ASR) systems. However, its traditional calculation, based on Levenshtein distance, does not account for lexical similarity between words and treats each substitution in a binary manner, while also ignoring segmentation errors.

This paper proposes an improvement to WER by introducing a weighted substitution method, based on lexical similarity measures, and incorporating splitting and merging operations to better handle segmentation errors.

Unlike other WER variants, our approach is easily integrable and generalizable to various languages, providing a more nuanced and accurate evaluation of ASR transcriptions, particularly for morphologically complex or low-resource languages.

1 Introduction

Automatic speech recognition (ASR) is now ubiquitous in our daily lives, facilitating translation, video transcription, note-taking, and interactions with voice assistants. While advances in deep learning models have significantly improved ASR system accuracy, challenges remain, particularly for underrepresented and morphologically complex languages. Despite these advancements, evaluating the performance of ASR systems remains essential to ensure their accuracy and reliability.

Word Error Rate (WER) is still the benchmark metric for evaluating transcription quality, but it relies on the Levenshtein distance, which does not account for lexical imprecision or segmentation errors, limiting its relevance in the face of linguistic diversity.

Low-resource languages often exhibit complex morphological structures (Lupyan and Dale, 2010), making them particularly vulnerable to segmentation errors, especially in the case of agglutinative languages.

WER applies a double penalty to these errors, artificially inflating the error rate. Furthermore, these languages are often characterized by high dialectal diversity, leading to inappropriate penalties for variations that are not inherently errors. Lastly, WER treats all lexical substitutions in a binary manner, overlooking minor variations that could be considered acceptable. These limitations highlight the need for a more precise evaluation, better suited to linguistic diversity.

Two main approaches stand out to improve WER calculation: on the one hand, models incorporating weightings based on word meaning, such as the Weighted Word Error Rate (WWER) (Shichiri et al., 2007), and on the other hand, methods like the Phoneme Error Rate (PER) (He and Radfar, 2021), which assess recognition at the phonemic level. Recently, evaluation methods based on language models have also emerged. However, these solutions have limitations, particularly in terms of implementation complexity and generalization to all languages.

Our work proposes an improved version of WER, tailored to the specificities of ASR transcriptions, by introducing weighted substitution based on lexical similarity measures, as well as splitting and merging operations to better handle segmentation errors. This approach aims to ensure adaptability to various languages and different usage contexts.

2 Introducing WER and the Levenshtein Distance

The *Word Error Rate (WER)*, the main metric used to evaluate the performance of ASR systems, cal-

Proceedings of the 9th International Workshop on Computational Linguistics for Uralic Languages, pages 29–35
November 28-29, 2024 ©2024 Association for Computational Linguistics

culates an error rate: he lower the rate (with a minimum of 0), the better the recognition. The maximum rate is unbounded and can exceed 1 (Wikipédia, 2023). The WER formula is given by:

$$WER = \frac{S + D + I}{N}$$

This calculation is based on the *Levenshtein distance*, an algorithm that measures the similarity between two sequences by counting the minimum number of operations required to transform one sequence into another (Levenshtein, 1966). The algorithm recognizes three operations:

- S is the number of **substitutions** (errors where one word is replaced by another),
- D is the number of **deletions** (missing words in the transcription),
- I is the number of **insertions** (extra words added compared to the reference text),
- N is the total number of words in the reference text.

The algorithm works by constructing a matrix where each cell represents the alignment cost (by insertion, deletion, or substitution) of a segment from the input sequence (transcription) with a segment from the target sequence (reference text). The cost calculation is performed iteratively, comparing the elements of the two sequences.

		t	**a**	**c**
	0	1	2	3
c	1	1	2	2
a	2	2	1	2
t	3	2	2	2

Figure 1: example of a matrix for aligning the sequences "cat" and "tac".

2.1 Substitution Cost Calculation

The Levenshtein distance calculates the shortest path in the matrix by combining the costs of insertion, deletion, and substitution. The costs of insertion and deletion are fixed at 1. Regarding substitution, the algorithm assigns a cost of **0** if the units being compared are identical and a cost of **1** if they differ.

This mechanism, called *binary substitution*, means that the units are considered either entirely identical or different. Each cell of the matrix is defined as the minimum between:

$$D(i,j) = \min \begin{cases} D(i-1,j)+1 & \text{(case of a deletion)} \\ D(i,j-1)+1 & \text{(case of an insertion)} \\ D(i-1,j-1)+ & \\ \text{sub\_cost}(A[i],B[j]) & \text{(case of a substitution)} \end{cases}$$

where the *substitution cost* is defined as:

$$sub\_cost(A[i],B[j]) = \begin{cases} 0 & \text{si } A[i] = B[j] \\ 1 & \text{si } A[i] \neq B[j] \end{cases}$$

Thus, minimum cost to transform one sequence into anotherto another is obtained by following the minimal cost path in this matrix. This mechanism is essential for WER calculation, but it has several limitations due to its application to whole words.

3 Challenges of WER Based on Levenshtein Distance

This cost calculation method is effectively used in the character error rate rate (CER), where the comparison units are individual characters. In this context, each character is compared to another, and the substitution decision is naturally binary: either the units are identical (cost of 0), or they differ (cost of 1).

However, the WER, which compares entire words, has significant limitations, as highlighted by (Shigeki et al., 2023). Due to its binary approach, the WER mainly compares orthographic forms rather than the words themselves, which penalizes minor variations, such as "advisor" and "adviser."

These orthographic variations also include space insertions, as in "doghouse" and "dog house," which are double-counted in WER calculation. This type of situation is treated as a segmentation error.

These limitations affect all languages, but they are particularly pronounced in languages with complex morphology, minority languages, and those with limited resources.

3.1 Weaknesses of Binary Substitution

The binary logic of the Levenshtein distance in substitution cost calculation is problematic in the context of WER, as it treats words as homogeneous entities, without considering their lexical similarity.

For example, the words "hello" and "allo" are phonetically and orthographically closer than "hello" and "sunny." However, Levenshtein distance assigns the same substitution cost (1) to both word pairs, thus failing to distinguish minor errors from major ones.

Traditional WER lacks any mechanism to weight errors based on lexical similarity. As a result, two words differing only by minor variations are treated as if they were significantly divergent.

This approach oversimplifies linguistic errors, significantly limiting WER's ability to accurately assess the performance of ASR systems.

3.2 Segmentation Errors

Levenshtein distance does not account for segmentation errors, such as word splitting or merging, which are common in ASR transcriptions.

For example, if "keyboard" is transcribed as "key board," traditional WER calculation treats this as two distinct errors: a substitution and an insertion. However, this is actually a single splitting error.

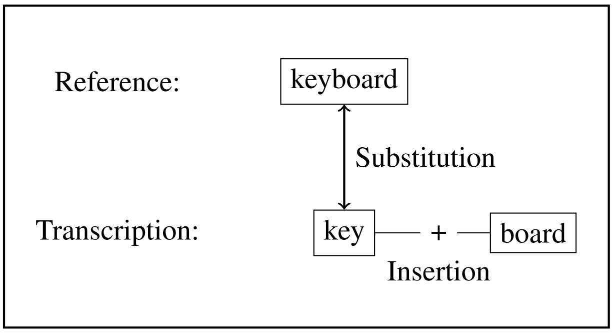

Figure 2: Double counting of segmentation errors.

Similarly, when a compound expression like "ice cream" is transcribed as a single word "icecream," this constitutes a merging error.

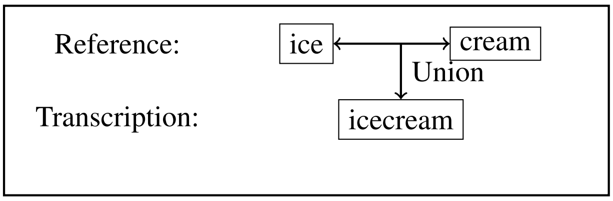

Figure 3: Merging error in transcription

The omission of segmentation errors leads to an inaccurate evaluation of transcriptions, overlooking aspects specific to speech recognition. Moreover, these segmentation issues are often considered less severe than insertions and deletions.

3.3 Morphologically complex languages

Morphological richness poses major challenges for ASR systems, which struggle to handle word inflections (prefixes, suffixes, etc.), thereby increasing the number of lexical forms and creating rare or unseen structures in training data (Morris, 2021).

In highly inflected languages, even small lexical variations can have a disproportionate impact on ASR performance. A simple error in a suffix or internal inflection can significantly increase WER, despite an otherwise accurate transcription. WER, by treating each word as a whole unit, does not account for this morphological variability.

Agglutinative languages, such as Finnish and Estonian, present particular challenges for speech recognition due to their morphological complexities. Words are formed by concatenating roots with numerous affixes, resulting in long lexical units and generating multiple word forms. This presents several difficulties for ASR systems:

- **Vocabulary explosion** : For Finnish, a lexicon of 400,000 words can still lead to a high rate of out-of-vocabulary words (Kurimo et al., 2006). This complicates the accurate transcription of these unknown words, and WER, not accounting for this complexity, severely penalizes variations that might be considered minor in the context of these languages.

- **Segmentation errors**:A poorly trained ASR system might split these elements into multiple words or merge them incorrectly, leading to multiple errors in the Word Error Rate (WER) calculation.

3.4 Minority and Low-Resource Languages

Often characterized by complex morphology, minority languages are subject to the same constraints mentioned earlier.

In evolutionary linguistics, (Lupyan and Dale, 2010) showed that languages spoken by large cosmopolitan communities, with many non-native speakers, tend to simplify their morphology over time. In contrast, minority languages, spoken in smaller communities, generally retain complex morphological structures. Native speakers of these languages share an intuitive understanding of these complex rules, allowing the language to preserve these features.

Similarly, (Lindenfelser, 2020) explains that languages with fewer non-native speakers or those that have not been significantly influenced as a second language (L2) tend to retain or even develop complex morphological systems, such as elaborate inflection systems for nouns or verbs.

Although low-resource languages are often minority languages, some are also widely spoken. This lack of data imposes various constraints on ASR systems:

- **Lack of diverse data**:Limited and insufficiently diverse corpora affect the ability of ASR models to correctly recognize lexical and linguistic variations.
- **Transcription errors related to data quality**: Errors often stem from poor quality or lack of standardization in transcriptions, rather than an intrinsic weakness of the system.
- **Inability to handle dialectal variations**: The same word or phrase may be pronounced differently depending on the region, dialect, or speaker. Models trained on a standard form (or specific dialect) often fail to recognize variants from other regions.
- **Difficulty in handling accents**: The phonetic diversity is often vast but under-documented, complicating ASR models' ability to accurately process these regional variations or accents.
- **Low phonological standardization**: The lack of formal rules for pronunciation and segmentation makes it difficult for ASR models to manage words effectively.

These training limitations lead to multiple errors, disproportionately increasing the WER, even when the divergences do not reflect actual inaccuracies.

4 Proposed New Method for WER Calculation

4.1 Lexical Measures for Substitution Cost

To overcome the limitations of binary substitution in WER calculation, we introduce the use of lexical similarity measures such as the Jaccard index, CER (character error rate), or cosine similarity. These measures calculate a continuous dissimilarity cost between 0 and 1, reflecting the actual difference between words.

$$sub\_cost(A[i], B[j]) = similarity(A[i], B[j])$$

Table 1 presents the error rates (in percentage) for different word pairs. the higher the value, the more dissimilar the words are. The last column indicates the algorithmic complexity of each method.

It is important to note that CER can exceed 100% when the transcription is significantly longer than the reference word, due to its calculation based on Levenshtein distance, which penalizes excessively long transcription sequences.

While these measures can be combined to offer a holistic evaluation, this increases the complexity of the process and, therefore, the execution time.

4.2 Split and Merge Operations

We introduce two new operations for WER calculation: splitting and merging. These operations aim to correct common segmentation errors in transcriptions produced by ASR systems. A word may be incorrectly split into two segments or, conversely, merged into one.

For example, when the word "input" is transcribed as "in put," a single merging operation would suffice to correct this error, rather than treating it as two distinct errors. By incorporating these operations into WER calculation, our approach improves the accuracy of this metric by accounting for word segmentation errors in ASR transcriptions.

To incorporate these operations within the Levenshtein algorithm, we add the following conditions:

```
// Separation
if (j > 1 and (transcript[j] == reference[i-1] +
      reference[i])) then
    d[i, j] := min(d[i, j], d[i-2, j-1] +
        seg_Cost)

// Union
if (i > 1 and (reference[i] == transcript[j-1] +
      transcript[j])) then
    d[i, j] := min(d[i, j], d[i-1, j-2] +
        seg_Cost)
```

Splitting and merging errors, being less severe than insertions and deletions,can be given a reduced cost. Moreover, these errors can be treated as adding or removing a character from a word. The cost can thus be calculated using the CER, which is simplified in this configuration:

$$seg\_cost = \begin{cases} 1/len(reference) & \text{(cost based on CER)} \\ 1 & \text{(standard cost)} \\ 0.5 & \text{(reduced cost)} \end{cases}$$

Method	hello allo	kitten sitting	intention execution	diner dinner	O(n)
Cosine similarity	36.7%	38.3%	38.5%	5.1%	O(n+m)
Fuzzy Wuzzy	33.0%	38.0%	43.9%	8.9%	O(n*m)
Jaro	21.7%	25.4%	36.3%	5.5%	O(n*m)
Sorensen Dice	42.9%	63.6%	60.0%	11.1%	O(n+m)
CER	50.0%	42.86%	55.55%	16.66%	O(n*m)
LCS similarity	40.0%	42.86%	44.44%	16.66%	O(n*m)
Jaccard LCS	50.0%	55.55%	61.54%	16.66%	O(n*m)

Table 1: Comparison of similarity measures with complexity.

4.3 Experimental Analysis

In this study, we carried out transcriptions of Finno-Ugric languages (Finnish, Meadow Mari, and Hill Mari), as well as Dutch and Afrikaans, using the MMS model. The performance of the transcriptions was evaluated using WER, CER, and our UWER version.

Language	WER	CER	UWER
Finnish	0.691	0.136	0.161
Meadow Mari	0.636	0.151	0.242
Hill Mari	0.922	0.313	0.471
Afrikaans	0.384	0.106	0.141
Dutch	0.477	0.104	0.134

For UWER calculation, the segmentation cost (seg_cost) was adjusted according to the CER. Tests with costs set to 1 and 0.5 showed minimal differences, as illustrated below:

cost = cer	cost = 1	cost = 0.5
0.226	0.22837	0.22734
0.24172	0.2473	0.2441

To better visualize the impact of these error rate differences, here are some examples of reference sentences and their transcriptions, with the two measures compared (table 2).

Reference	Transcription	WER	UWER
ja minä huokasin kevennyksestä	ja mina huokasin kevenyksest	0.50	0.10
kaisa syötteli porsasta	kaisa syoteli porsasta	0.33	0.08
oletpa tosiaan lapsellinen	olet pa tosian lapselinen	1.33	0.12
ik ben daar heel blij mee	ik ben dar hel blij me	0.50	0.12
de beatles waren van liverpool	da bitels uaren fan liverpul	1.00	0.36
naaktslakken hebben geen slakkenhuis	naktslaken heben gen slakenhuis	1.00	0.14

Table 2: WER and UWER Comparison

5 Discussion

5.1 Improvement

- **Acoustic versus linguistic errors**: WER does not distinguish between errors caused by acoustic factors (noise, pronunciation) and those of a linguistic nature, assigning them equal weight in the score calculation.
- **Equal penalty for all types of errors**: Although we introduced a dynamic penalty for substitutions, it remains fixed for insertions and deletions. a penalty proportional to the length of inserted or deleted words could, among other things, help mitigate the impact of noise.
- **Combined errors**: Our experimental analyses show that when segmentation and lexical errors are combined, even our metric no longer accurately reflects the transcription quality. For exemple:

Reference	Transcription	WER	UWER
tervetuloa	tervet tuloa	2.00	1.34
slaapwel	slap wel	2.00	1.43

To address this, segmentation operations should be replaced by *substitution_separation* and *substitution_union*, applied without the requirement for equality.

The cost would then be:

$$\begin{cases} \text{seg\_cost} + \text{similarity}(\ \text{ref}[i-1] + \text{ref}[i],\ \text{hyp}[j]) \\ \text{(for separation)} \\ \text{seg\_cost} + \text{similarity}(\ \text{ref}[i],\ \text{hyp}[j-1] + \text{hyp}[j]) \\ \text{(for union)} \end{cases}$$

5.2 Comparison with Other Methods

The Phoneme Error Rate (PER) and Weighted Word Error Rate (WWER) are variants of WER that attempt to address some of its limitations.

PER (Shichiri et al., 2007) focuses on errors at the phoneme level, offering finer granularity than WER. However, it requires phonetic transliteration of both the transcription and the reference text, making generalization more difficult.

WWER (He and Radfar, 2021), on the other hand, assigns different weights to deletion, insertion, and substitution errors, optimized using dictionaries to weigh words based on their importance. However, this approach relies on the creation of specific linguistic resources and does not sufficiently discriminate substitution costs, limiting its effectiveness.

Apple's "Humanizing WER" method (Apple, 2024) and the work of Hughes (Hughes, 2023) use advanced language models to improve the evaluation of speech recognition systems. HWER weights errors according to their context, offering an evaluation closer to human perception. Despite their potential, these approaches have limitations: complexity of implementation, lack of standardization, potential subjective biases, and difficulty in applying to low-resource languages due to their reliance on language models.

6 Conclusion and Future Directions

This study has highlighted the limitations of WER, especially its inability to account for lexical nuances and segmentation errors, making it unsuitable for morphologically complex or low-resource languages.

We proposed an improved version of WER, which introduces weighted substitution based on lexical similarity, as well as splitting and merging operations. Experimental results show that UWER improves evaluation accuracy across several languages.

Our approach aims to ensure WER's adaptability to the vast linguistic diversity while providing a simple-to-implement solution a simple-to-implement solution, fully interchangeable with WER, without requiring changes to current practices.

By increasing the precision of this metric, we provide a more rigorous evaluation tool capable of revealing the true performance of models, especially for morphologically complex and low-resource languages.

Furthermore, this approach can also be leveraged as a loss function to optimize ASR model training. Although WER is not differentiable, adaptations such as differentiable approximation, reinforcement learning, or optimization via Minimum Bayes Risk (MBR) can be considered to overcome this limitation.

References

Apple. 2024. Humanizing wer. https://machinelearning.apple.com/research/humanizing-wer.

Bradley He and Martin Radfar. 2021. The performance evaluation of attention-based neural asr under mixed speech input. In *Proceedings of ICASSP 2021*, Stony Brook University, NY, USA.

John Hughes. 2023. The future of word error rate. *Speechmatics*.

Mikko Kurimo, Antti Puurula, Ebru Arisoy, Vesa Siivola, Teemu Hirsimaki, Janne Pylkkonen, Tanel Alumae, and Murat Saraclar. 2006. Unlimited vocabulary speech recognition for agglutinative languages. In *Proceedings of the Human Language Technology Conference of the North American Chapter of the ACL*, pages 487–494, New York, USA. Association for Computational Linguistics.

Vladimir Levenshtein. 1966. Binary codes capable of correcting deletions, insertions, and reversals. *Soviet Physics - Doklady*, 10:707–710.

Siegwalt Lindenfelser. 2020. Asymmetrical complexity in languages due to l2 effects: Unserdeutsch and beyond. *Languages*, 5:57.

Gary Lupyan and Rick Dale. 2010. Language structure is partly determined by social structure. *PLOS ONE*, 5(1):e8559.

Ethan Morris. 2021. Automatic speech recognition for low-resource and morphologically complex languages. Master's thesis, Rochester Institute of Technology.

Takashi Shichiri, Hiroaki Nanjo, and Takehiko Yoshimi. 2007. Automatic estimation of word significance oriented for speech-based information retrieval. In *Proceedings of ACL 2007*, pages 204–209, Otsu, Japan.

Karita Shigeki, Sproat Richard, and Ishikawa Haruko. 2023. Lenient evaluation of japanese speech recognition: Modeling naturally occurring spelling inconsistency. *arXiv preprint arXiv:2306.04530*.

Wikipédia. 2023. Word error rate.