Match ‘em: Multi-Tiered Alignment for Error Analysis in ASR

Phoebe Parsons[1] **Knut Kvale**[2] **Torbjørn Svendsen**[1] **Giampiero Salvi**[1]

[1]Department of Electronic Systems, NTNU, Trondheim, Norway
[2]Telenor Research and Innovation, Oslo, Norway
{phoebe.parsons, torbjorn.svendsen, giampiero.salvi}@ntnu.no, knut.kvale@telenor.com

Abstract

We introduce “Match ‘em”: a new framework for aligning output from automatic speech recognition (ASR) with reference transcriptions. This allows a more detailed analysis of errors produced by end-to-end ASR systems compared to word error rate (WER). Match ‘em performs the alignment on both the word and character level; each relying on information from the other to provide the most meaningful global alignment. At the character level, we define a speech production motivated character similarity metric. At the word level, we rely on character similarities to define word similarity and, additionally, we reconcile compounding (insertion or deletion of spaces). We evaluated Match ‘em on transcripts of three European languages produced by wav2vec2 and Whisper. We show that Match ‘em results in more similar word substitution pairs and that compound reconciling can capture a broad range of spacing errors. We believe Match ‘em to be a valuable tool for ASR error analysis across many languages.

1 Introduction

Metrics like word error rate (WER) provide a simple, automated way of understanding how well an automatic speech recognition (ASR) system is performing. However, this simplicity fails to capture the nuance regarding the severity of transcription errors, both in terms of spellings and semantics. Efforts have been made to improve WER. These include adding new metrics around information lost by mistranscriptions (Morris et al., 2004) and weighting kewyords more heavily in WER (Nanjo and Kawahara, 2005). Attempts to optimize the alignment between transcriptions have utilized articulatory features (Cucchiarini, 1996) as well as semantic distances (Roy, 2021). Additionally, new metrics such as SemDist (Kim et al., 2021) and Aligned Semantic Distance (Rugayan et al., 2022) have been developed to utilize the embedding vector space, instead of aligning the words themselves, to calculate the severity of errors. However, as all these metrics only aim to summarize the quality or utility of an ASR output, they do not provide details on the types or severity of commonly made errors.

Understanding the types of errors that ASR systems make has been of interest for many years. The goals are both understanding how wrong a transcription really is, as well as identifying specific areas for improvement. In Goldwater et al. (2010), the authors create individual word error rate to determine which words are frequently missed and which factors account for misrecognitions. In Vasilescu et al. (2012), the authors compare the ability of humans and automatic transcriptions to disambiguate homophonic or near-homophonic words that are frequently missed by ASR. Words that are frequently missed in conversational speech for Dutch, English, and German are analyzed in Lopez et al. (2022). The authors in Wirth and Peinl (2022); Salimbajevs and Strigins (2015) manually classify ASR errors for both their severity and type to understand how ASR is performing on German and Latvian speech, respectively.

Despite the benefits of metrics and error analysis, there are several factors that can be limiting to these tools. For semantic metrics, knowledge of the language (semantic embeddings, word importance) is crucial. However, access to such resources is not readily available for certain languages. Similarly, analysis of ASR errors is often reliant on manual efforts to label the errors made, thus limited by the amount of human hours

Proceedings of the Joint 25th Nordic Conference on Computational Linguistics and 11th Baltic Conference on Human Language Technologies (NoDaLiDa/Baltic-HLT 2025), pages 440–447
March 3-4, 2025 ©2025 University of Tartu Library

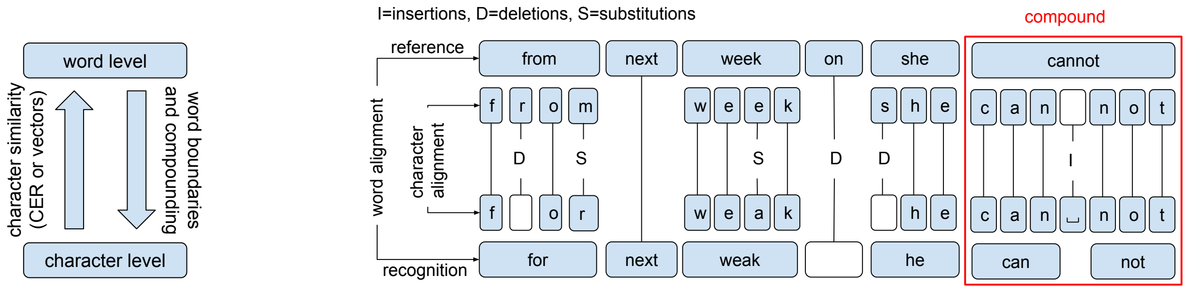

Figure 1: Left: interaction between word and character level in Match'em. Right: An example alignment showing the multi-tiered (word and character) approach as well as compounding.

available to contribute to the task. Lastly, many existing metric and evaluation paradigms are designed with the assumption that words operate as atomic units — an assumption challenged by end-to-end ASR systems where output is generated at the character or sub-word level.

In this paper, we propose a new framework, that we call Match 'em, for aligning ASR generated and reference text that operates both at the word and character level. The goal is to provide a better match between words and characters thus allowing for a detailed analysis of the common mistakes produced by ASR systems. Additionally, this method opens the possibility to use foundation ASR models trained on massive amounts of data to study phenomena related to variability in speech production by analyzing the ASR errors in detail; such phenomena include dialectal variation or pronunciation variation in second language learners or in speakers with speech sound disorder.

The contributions of the paper can be summarized as:

- We introduce a new framework for ASR output and reference alignment that operates on the word and character level. Each level influences the other level with the goal of obtaining an optimal global alignment.
- We introduce a character dissimilarity metric based on speech production to guide the within-word character alignments.
- At the word level, we define a word dissimilarity metric that inherits similarities from the character level. We also implement an algorithm for reconciling compounding (insertion or deletion of spaces)
- We evaluate the method on transcripts of three European languages obtained by two state-of-the-art ASR models (wave2vec2, and Whisper), showing that Match 'em produces more meaningful alignments both in terms of word similarities and character similarities.
- We make all the code available.

2 The Match 'em framework

The standard Levenshtein alignment considers three edit operations (insertion, deletion, and substitution) when transforming the hypothesis text into the reference text (Levenshtein, 1965). The edit costs (that is, the penalty for any of the three edits) are also fixed before alignment occurs. This method is traditionally used to separately compute either word error rate (WER) at the word level or character error rate (CER) at the character level. The Match 'em framework we propose operates both at the word and the character level *simultaneously*. The alignment at each level is influenced by information coming from the other, as illustrated by Figure 1. In the figure, we can see that words that are spelled similarly are aligned, the characters within the words are aligned, and the breaking up of a compound word is accounted for. Details on how each of these components was achieved follows in the subsections below.

2.1 Character- and Word-Level Metrics

The first step in defining the Match 'em algorithm is to define metrics both at the character and word level. At the character level, we introduce a dissimilarity metric based on speech production, similar to the method in (Cucchiarini, 1996). We define a set of vectors of articulatory features for each letter in the target language's alphabet. To accommodate the different parameters by which vowels and consonants are defined, separate vec-

Vowels

value	height	front/back	rounding
0	high	back	false
1	mid	mid	true
2	low	front	

Consonants

value	voice	class	nasal	place	lip rounding
0	false	stop	false	bilabial	false
1	true	affricate	true	labio-dental	true
2		trill		avleolar	
3		fricative		retroflex	
4		approximate		palatal	
5				velar	
6				uvular	
7				glottal	

Table 1: Articulatory features used to define the character-level metric for vowels (left) and consonants (right). For example, the vector [0, 2, 1] would be interpreted to mean a high, front, rounded vowel /y/, whereas the vector [1, 0, 1, 2, 0] would represent a voiced, nasal, alveolar stop /n/. The currently defined vector system does not account for every word sound and would need to be adjusted or expanded as new languages were used.

tor definitions are used for each class. Examples of the vector spaces are provided in Table 1. Each character is then assigned to one or more vectors depending on its typical pronunciation(s). Doing this, the method account for characters that might be commonly realized as two distinct phones (e.g., the Norwegian ”r” is a dialect marker and can be realized as either an alveolar tap or uvular trill (Kvale and Foldvik, 1992)).

The distance (dissimilarity) between two characters (either vowel-to-vowel or consonant-to-consonant) is computed as the normalized Euclidean distance between the corresponding vectors. Comparing vowels and consonants in this articulatory space is not meaningful. Instead, the cost is set at 1.0 for most vowel-consonant substitutions, the same cost as a substitution of two completely different characters. With vowel-approximants, the cost is lowered to 0.9 to allow for the gestural and perceptual similarities. This value was chosen through experimentation and visual inspection of the resulting alignments. The cost is also set at 1.0 for any character-punctuation substitution. If multiple definitions character vectors are provided (e.g. in accounting for two realizations of ”r”), the vector with the lowest resulting dissimilarity is used.

As these vectors’ purpose is merely to support a character distance score, not to offer linguist truth, there are known simplifications and omissions in the vector definitions. As an example, di- or trigraphs are not captured in the letter vectors.

In practice, we find that defining these character vectors to be straight-forward for languages with available orthographic to phonetic mappings. Even for languages which the authors were unfamiliar, vector definition was quick.

				Standard	Match ‘em
cats	run	very	quickly	costs	costs
	cat	runs	quick	4	3.286
cat		runs	quick	4	2.555
cat	runs		quick	4	**1.869**
cat	runs	quick		4	2.583

Table 2: Potential alignments for the two phrases *cats run very quickly* and *cat runs quick*. The cumulative costs for each alignment is given for the standard and Match ‘em approaches.

At the word level, the dissimilarity between two words is computed by performing an alignment between the within-word characters of the two words in question (see Figure 1 (right) for an example). This alignment is guided either by the character dissimilarity defined previously, or by the simpler, character-naïve CER. This dissimilarity is then used as the substitution cost when aligning words. Insertion and deletion costs at the word level are left at 1.0.

2.2 Multi-tier Alignment

The Match ‘em alignment makes use of the dissimilarity metrics defined in Section 2.1 to perform multi-tier alignment at the word and character level. Both levels use dynamic programming similarly to the Levenshtein method. However, at the word level, character-based word dissimilarity is used as cost for substitutions. Similarly, at the character level articulatory character dissimi-

	edit costs				
		cats	run	very	quickly
	0	1 ←	1 ←	1 ←	1 ←
cat	1 ↑	1 ↖	1 ↖ ←	1 ↖ ←	1 ↖ ←
runs	1 ↑	1 ↖ ↑	1 ↖	1 ↖ ←	1 ↖ ←
quick	1 ↑	1 ↖ ↑	1 ↖ ↑	1 ↖	1 ↖ ←

	cumulative costs				
		cats	run	very	quickly
	0	1 ←	2 ←	3 ←	4 ←
cat	1 ↑	1 ↖	2 ↖ ←	3 ↖ ←	4 ↖ ←
runs	2 ↑	2 ↖ ↑	2 ↖	3 ↖ ←	4 ↖ ←
quick	3 ↑	3 ↖ ↑	3 ↖ ↑	3 ↖	4 ↖ ←

Table 3: Standard approach: step-by-step edit costs (left) and cumulative costs (right) for aligning the two phrases *cats run very quickly* and *cat runs quick* using the standard approach. Backtrace arrows indicate from which cell the cost is computed.

	edit costs				
		cats	run	very	quickly
	0	1 ←	1 ←	1 ←	1 ←
cat	1 ↑	1/4 ↖	1 ←	1 ←	1 ←
runs	1 ↑	1 ↑	1/3 ↖	1 ←	1 ←
quick	1 ↑	1 ↑	1 ↑	1 ↖	2/7 ↖

	cumulative costs				
		cats	run	very	quickly
	0	1 ←	2 ←	3 ←	4 ←
cat	1 ↑	**0.25** ↖	1.25 ←	2.25 ←	3.25 ←
runs	2 ↑	1.25 ↑	**0.583** ↖	**1.583** ←	2.583 ←
quick	3 ↑	2.25 ↑	1.583 ↑	1.583 ↖	**1.869** ↖

Table 4: Match 'em approach: step-by-step edit costs (left) and cumulative costs (right) for aligning the two phrases *cats run very quickly* and *cat runs quick* using the Match 'em approach. Backtrace arrows indicate from which cell the cost is computed.

larities are used to align characters within words.

As a demonstration of the benefit of Match 'em, let us consider the examples provided in Table 2. Here we have four different potential alignments between the reference text *cats run very quickly* and the hypothesis text *cat runs quick*. All words between the reference text and the hypothesis are different ("cats" and "cat" are, for example, made different by the addition of the "s"). This means that with the similarity naïve standard approach used in WER with edit costs fixed at 1.0, any and all alignments are equally valid and the resulting alignment will be chosen at random. The local edit costs and the cumulative costs for the standard alignment can be found in Table 3.

Unlike with the standard alignment, Match 'em discounts the costs of substituting similar words. Thus, although "cats" and "cat" are different words the cost for substituting them is only 1/4 (the CER between them). Thus, as shown in Table 4 (left), the costs for substitutions of the words "cats", "run", and "quickly" are less than one and when incorporated into the full costs (Table 4 (right)) an obvious best path is presented: one which results in the third alignment in Table 2.

2.3 Compounding

After the preliminary word-level alignment described in Section 2.2, Match 'em accounts for errors around compound words or, equivalently, it accounts for errors created by adding or deleting one or more space characters. With the standard Levenshtein alignment, the breaking up or creation of a compound word inflicts two edits: a deletion or insertion, as well as a substitution. For example, in Figure 1 (right), there would be a substitution between the words "cannot" (reference) and "can" (ASR) as well as an insertion of the word "not" (ASR). However, the difference really is the insertion or deletion of a space (a character). As exemplified by the figure, Match 'em allows to classify this as a single word substitution at the word level, and as a single character insertion (the space) at the character level. It accomplishes this by iteratively checking the neighbouring words to every edit (substitutions, insertions or deletions). For every iteration, the neighbouring word is attached to the current word if the operation results in a lower character level cost. In the example, "not" is attached to "can" because this results in a reduction of word dissimilarity from 3 ("cannot" vs "can") to 1 ("cannot" vs "can not"). This process is repeated as long as the cost decreases, allowing for compounds of several words.

3 Experiments

To evaluate the impact of this new alignment method, audio in three different European languages was transcribed using two state-of-the-art ASR model architectures. The three languages (Norwegian, Italian, and English) were chosen for a variety of reasons. Firstly, Match 'em requires languages with alphabets for which character articulatory vectors can be defined—thus excluding languages that use syllabaries or logographies, such as Japanese or Chinese, respectively. Also, these three languages cover multiple language families (Germanic and Romance), orthographic depths (Norwegian and Italian spellings being largely phonetically written as opposed to English being irregular (Seymour et al., 2003)), and dialectal variations (both Norwegian and Italian contain a large amount of dialectal variation compared to English (Kinder and Savini, 2004; Skjekkeland, 1997)). Additionally, Norwegian Bokmål allows for multiple legal spellings of words (e.g., *vet* and *veit* both being legal spelling for the present tense of *å vite* ("to know"). Lastly, Norwegian utilizes compounding of words (again with common, but perhaps less legal, variations to spellings) to a higher degree than English or Italian, which gives us an opportunity to test how Match 'em performs on this aspect.

3.1 Datasets

For both Italian and English data, we used the VoxPopuli corpus (Wang et al., 2021), which consists of recordings from the European Parliament. As parliamentary recordings, the speech style is largely spontaneous with a good distribution of speakers. In the Italian corpus, we removed a number of utterances where there was a significant mis-alignment between the audio and human-generated transcriptions.

As Norway is not part of the European Parliment, the NB Tale dataset (National Library of Norway, 2015) was used instead of VoxPopuli. NB Tale is publicly available through the Norwegian National Library's Language Bank and contains a good variety of speakers. In our experiments, we only used the subsection of NB Tale consisting of spontaneous speech recordings produced by native speakers, to better align with the speech style for Italian and English. All of the speech in NB Tale is human-transcribed using the Bokmål written standard.

3.2 Models

To generate transcriptions for our alignment analysis, we employed two end-to-end model architectures, wav2vec 2.0 (Baevski et al., 2020) and Whisper (Radford et al., 2022). The transcriptions are either generated as characters for wav2vec 2.0, or as byte pair encodings (effectively word-level and character-level output (Radford et al., 2019)) for Whisper. These flexible outputs allow for potentially novel spellings and therefore constitute a good test bed for Match 'em. Finally, both wav2vec 2.0 and Whisper have reported impressive accuracies, making them ideal candidates to generate reasonable transcriptions to evaluate.

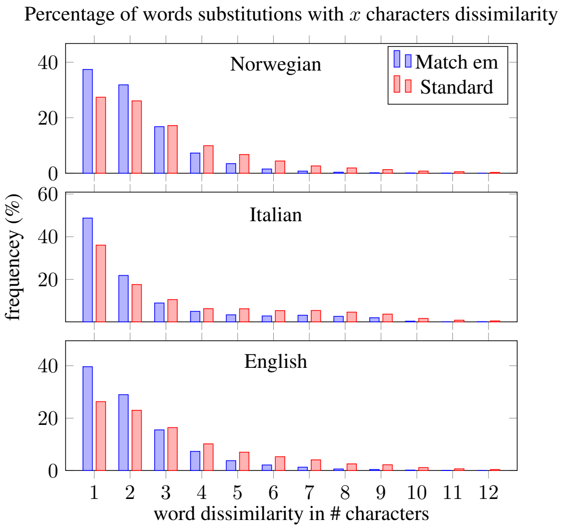

Figure 2: Percentage of word substitutions as a function of word dissimilarity in number of characters. Results are accrued over both wav2vec 2.0 and Whisper model outputs.

For the wav2vec 2.0 architecture, we used different models depending on the language. For Italian and English, we used the VoxPopuli multilingual model (Wang et al., 2021) without a language model (LM). This model contains approximately 300 million parameters. However, the VoxPoluli model does not contain Norwegian. Thus, for Norwegian, we used the 300 million parameter wav2vec 2.0 model created by the Norwegian National Library AI Lab (De La Rosa et al., 2023) run with a LM. For the Whisper architecture, we used the same multi-lingual model (large-v2) for all languages. This model, unlike the wav2vec 2.0 counterparts, was trained to perform multiple tasks, including ASR in English and other languages, any-to-English translation, and non-speech detection. The Whisper model contains 1550 million parameters and was trained on 680,000 hours of loosely-supervised Internet audio (117,000 of those hours being in languages other than English). This model was run with a LM.

3.3 Implementation

We implemented the Match ‘em framework as a Python package[1]. This implementation has been designed with a high degree of flexibility, allowing many features to be specified as runtime parameters. These include selecting which alignment to use (Levenshtein vs Match ‘em), whether compounding should be reconciled, and what kind of character dissimilarity to use (binary or vector based). The articulatory vectors, described in Section 2.1, are included in the Match ‘em repository. The vectors are defined in JSON format and can be easily expanded or edited for other letter-based orthographies.

4 Results

4.1 Word substitution similarity

Figure 2 considers word substitution pairs (excluding word insertion or deletion) from both the wav2vec 2.0 and Whisper text. In order to assess the quality of the alignment, we evaluate how many characters are different between the words in a substitution pair. In the figure, we can see that Match ‘em increases the frequency of word-pairs with a small orthographic distance. For example, consider all the pairs with only one character difference. For Match ‘em these account for 37.4% (Norwegian), 48.74% (Italian), and 36.64% (English) of all the substitution errors. These percentages are approximately ten percentage points higher than the corresponding values for the standard alignment (Norwegian: 27.39%, Italian: 36.07%, English: 26.31%).

As Match ‘em better aligns similar words, we can use it to analyze the types of character errors occurring within words. This is fundamentally different than analyzing character errors from standard CER alignment because it allows us to focus on errors in specific parts of the words that carry specific meaning. CER, as is typically computed, ignores word boundaries. Thus, while it may provide insight into which characters are frequently missed, it looses any information that might indicate what role those letters played. The value of character-aware error analysis can be illustrated by (Parsons et al., 2023)

As an example, we investigated word substitutions where only the final character changed. From there, we observed the most common character changes for each language. For Norwegian, these characters were “e” and “r”; while for English, it was the character “s”. For both of these languages, insertion or deletion of these characters will change the quantity of a noun or the tense of a verb. For Italian, the vast majority of words ends in a vowel, where the final vowel marks both gender and quantity of a word. Due to the frequency and similar semantic load, we considered all final vowels in Italian in our analysis. The percentage of all word substitutions containing just this final letter change are presented in Table 5. Through this we see that not only does Match ‘em align more instances of final letter change but that a sizeable amount of all substitution errors are just the final letter change. Such a final letter change might alter a word’s meaning slightly, but will rarely destroy the meaning of an entire sentence. Consequently, depending on the task at hand, those errors may be given higher or lower weight in ASR development.

Dataset	wav2vec 2.0		Whisper	
	Standard	Match ‘em	Standard	Match ‘em
English	3.92	4.67	8.43	10.22
Italian	11.48	13.52	12.04	14.63
Norwegian	5.66	7.62	5.17	6.69

Table 5: The percent of word substitutions produced by Match ‘em alignment where only the final character changed. The most common errors were considered (Norwegian: “e” or “r”, Italian: all vowels, English: “s”).

4.2 Compounds

As described in Section 2.3, Match ‘em also attempts to recognize and rectify compounding errors. Although the majority of compounds include the concatenation of two words, in the Norwegian data we see that Match ‘em is able to account for cases where more than two words are combined, such as “to tusen og tolv” and “totusenogtolv”. Both these written forms are valid in Norwegian and have the same meaning (*two thousand and twelve*). In English, many of the compound pairs are contractions (e.g., “it is” vs. “it’s”, “we are” vs. “we’re”) where the difference is not only the space but also the substitution of character(s) for an apostrophe.

As this method works on the surface level of words, without any context of word meaning(s),

[1] https://github.com/scribe-project/match-em

there is the potential that the compound word pairs while being similar in characters are actually semantically distinct. The most common pair found in our Norwegian data (“og så” - “også”) demonstrates this well because the two variants can be translated to English as *and so* and *also*, respectively. Most contractions that are seen in both English and Italian carry the same semantic content. As an exception, some Italian contractions should be considered as mispellings (like “un Europa” instead of the correct “un’Europa” or “una Europa”). Regardless, as the meanings would still be interpretable by a human, the reduction in penalty for the compounding mistake is well justified. Given the success of the compounding analysis, we believe that more highly synthetic languages, such as Finnish, may be good candidates for Match ‘em analysis in future work.

Analyzing the difference in compounding errors between wav2vec 2.0 and Whisper gives some insights for the potentially different behaviour of these two models. For English, the top 10 most frequent compounding errors are nearly the same for both models and contain typical contractions (e.g. “it is” vs “it’s”). The numbers of errors are also comparable. For Italian, the Whisper model has a much lower number of compound errors compared to wav2vec 2.0 (see also Section 4.3). For Norwegian, the two models make a comparable number of compound errors, the most common of which is “og så” versus “også”. However, after “og så” and “også”, frequency of specific compound errors is different between the two models. Further analysis of these phenomena may give insights into the workings of these two architectures.

4.3 Standard versus Match ‘em WER

The goal of Match ‘em is to produce a better word alignment for detailed error analysis. It is, however, interesting to study how Match ‘em modifies the WER. If we exclude the compounding reconciliation, the better alignment does not change the total number of errors (insertions, deletions and substitutions), although it may change their relative distribution. Changes in WER are, therefore, an exclusive result of compounding reconciliation, where we keep a single substitution and reduce the number of insertions and deletions. Table 6 demonstrates this by showing the WERs computed with Levenshtein (standard) and Match ‘em alignment for the three test languages and two model architectures. As expected, by resolving compounding errors, Match ‘em results in a lower WER. The reduction is greater for wav2vec 2.0 which, as noted in Section 4.2, produces a higher number of compounding errors than Whisper. As mentioned in Section 2.3, however, it is not clear if lower is truly better here.

Dataset	wav2vec 2.0		Whisper	
	Standard	Match ‘em	Standard	Match ‘em
Norwegian	22.07	21.06	21.50	20.81
Italian	20.55	18.87	13.54	13.28
English	19.87	17.92	14.80	14.49

Table 6: The WER for each language, model, and alignment method.

5 Conclusions

We propose the new Match ‘em framework for creating better alignment between reference and ASR-generated transcriptions both at the word and character level. We show that Match ‘em allows for a deeper understanding of ASR performance compared to WER, by supporting detailed analysis of common errors. By using word dissimilarity metrics and by reconciling compound errors, Match ‘em alignment results in word substitution pairs that are more similar compared to standard Levenshtein alignment. We show that analysis of these substitution pairs can yield insights into the potential semantic impacts of these errors. Our claims are verified across three European languages (English, Italian and Norwegian) and two state-of-the-art ASR architectures (wav2vec 2.0 and Whisper). We believe the Match ‘em framework to be a useful tool for other ASR researchers for gaining insights into their own models’ performances and, more generally, for speech researchers to gain linguistic insights by analyzing ASR errors on large annotated speech corpora.

Acknowledgments

This work has been done as part of the SCRIBE project as funded by the Norwegian Research Council, project number: 322964.

References

Alexei Baevski, Yuhao Zhou, Abdelrahman Mohamed, and Michael Auli. 2020. wav2vec 2.0: A framework for self-supervised learning of speech representations. In *Advances in Neural Information Pro-*

cessing Systems, volume 33, pages 12449–12460. Curran Associates, Inc.

Catia Cucchiarini. 1996. Assessing transcription agreement: Methodological aspects. *Clinical Linguistics & Phonetics*, 10:131–155.

Javier De La Rosa, Rolv-Arild Braaten, Per Kummervold, and Freddy Wetjen. 2023. Boosting Norwegian automatic speech recognition. In *Proceedings of the 24th Nordic Conference on Computational Linguistics (NoDaLiDa)*, pages 555–564, Tórshavn, Faroe Islands. University of Tartu Library.

Sharon Goldwater, Dan Jurafsky, and Christopher D Manning. 2010. Which words are hard to recognize? prosodic, lexical, and disfluency factors that increase speech recognition error rates. *Speech Communication*, 52(3):181–200.

Suyoun Kim, Abhinav Arora, Duc Le, Ching-Feng Yeh, Christian Fuegen, Ozlem Kalinli, and Michael L. Seltzer. 2021. Semantic Distance: A New Metric for ASR Performance Analysis Towards Spoken Language Understanding. In *Proc. Interspeech 2021*, pages 1977–1981.

J. J. Kinder and Vincenzo M. Savini. 2004. *Using Italian: a guide to contemporary usage*, chapter 1. Cambridge University Press.

Knut Kvale and Arne Kjell Foldvik. 1992. The multifarious r-sound. In *Proc. International Conference on Spoken Language Processing (ICSLP-92)*, pages 1259–1262.

Vladimir I. Levenshtein. 1965. Binary codes capable of correcting deletions, insertions, and reversals. *Soviet physics. Doklady*, 10:707–710.

Alianda Lopez, Andreas Liesenfeld, and Mark Dingemanse. 2022. Evaluation of automatic speech recognition for conversational speech in dutch, english and german: What goes missing? In *Proceedings of the 18th Conference on Natural Language Processing (KONVENS 2022)*, pages 135–143.

Andrew Cameron Morris, Viktoria Maier, and Phil Green. 2004. From WER and RIL to MER and WIL: improved evaluation measures for connected speech recognition. In *Proc. Interspeech 2004*, pages 2765–2768.

Hiroaki Nanjo and Tatsuya Kawahara. 2005. A new ASR evaluation measure and minimum Bayes-risk decoding for open-domain speech understanding. In *Proceedings. (ICASSP '05). IEEE International Conference on Acoustics, Speech, and Signal Processing, 2005.*, volume 1, pages I/1053–I/1056 Vol. 1.

National Library of Norway. 2015. NB Tale – Speech Database for Norwegian.

Phoebe Parsons, Knut Kvale, Torbjørn Svendsen, and Giampiero Salvi. 2023. A character-based analysis of impacts of dialects on end-to-end Norwegian ASR. In *Proceedings of the 24th Nordic Conference on Computational Linguistics (NoDaLiDa)*, pages 467–476, Tórshavn, Faroe Islands. University of Tartu Library.

Alec Radford, Jong Wook Kim, Tao Xu, Greg Brockman, Christine McLeavey, and Ilya Sutskever. 2022. Robust Speech Recognition via Large-Scale Weak Supervision. *arXiv*.

Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language Models are Unsupervised Multitask Learners. *arXiv*.

Somnath Roy. 2021. Semantic-wer: A unified metric for the evaluation of ASR transcript for end usability. *CoRR*, abs/2106.02016.

Janine Rugayan, Torbjørn Svendsen, and Giampiero Salvi. 2022. Semantically Meaningful Metrics for Norwegian ASR Systems. In *Proc. Interspeech 2022*, pages 2283–2287.

Askars Salimbajevs and Jevgenijs Strigins. 2015. Error analysis and improving speech recognition for Latvian language. In *Proceedings of the International Conference Recent Advances in Natural Language Processing*, pages 563–569, Hissar, Bulgaria. INCOMA Ltd. Shoumen, BULGARIA.

Philip Seymour, Mikko Aro, and Jane Erskine. 2003. Foundation literacy acquisition in European orthographies. *British Journal of Psychology*, 94:143–174.

Martin Skjekkeland. 1997. *Dei norske dialektane: tradisjonelle særdrag i jamføring med skriftmåla*. Høyskoleforl.

Ioana Vasilescu, Martine Adda-Decker, and Lori Lamel. 2012. Cross-lingual studies of asr errors: paradigms for perceptual evaluations. In *LREC*, pages 3511–3518. Citeseer.

Changhan Wang, Morgane Riviere, Ann Lee, Anne Wu, Chaitanya Talnikar, Daniel Haziza, Mary Williamson, Juan Pino, and Emmanuel Dupoux. 2021. VoxPopuli: A Large-Scale Multilingual Speech Corpus for Representation Learning, Semi-Supervised Learning and Interpretation. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 993–1003, Online. Association for Computational Linguistics.

Johannes Wirth and Rene Peinl. 2022. Automatic speech recognition in german: A detailed error analysis. In *2022 IEEE International Conference on Omni-layer Intelligent Systems (COINS)*, pages 1–8.