DOI: 10.17469/O2112AISV000014

SARA PICCIAU, DOMENICO DE CRISTOFARO, ALESSANDRO VIETTI

Phonological patterns in the predictions of a syllable-based end-to-end ASR system

This paper explores the role of syllables in automatic speech recognition (ASR) systems, focusing on its linguistic implications. Traditionally, ASR has focused on processing speech at the segmental level, but recent research suggests the importance of syllabic processing for robust recognition. We trained a neural ASR model to recognize phonological syllables and conducted a linguistic analysis on its output. Our objective was to observe how various factors, such as syllable token frequency, lexical accent position, syllable type, and parts of speech, influence the neural representation of syllables. To achieve this, we developed a fine-grained linguistic annotation system to overcome the limitations of quantitative metrics like Word Error Rate. By applying Multiple Correspondence Analysis, we identified patterns of association between the neural network's output behavior and linguistic features of speech. Specifically, the study demonstrates that the network compensates for low-frequency syllables through substitution strategies, particularly with absent tokens, which are complex syllables often occurring in proper nouns, common nouns, or numerals. Unstressed high-frequency tokens, such as subordinating conjunctions and determiners, tend toward deletion, while mid-frequency syllables with simple structures (CV) achieve optimal recognition, indicating the network's ability to reflect natural language processing patterns based on token frequency and syllabic complexity. Our findings provide insights into the role of syllables in ASR and contribute to ongoing research in this field.

Keywords: syllable, ASR, pretrained neural models, discrete speech units.

1. *Introduction*

The syllable plays a central role in the spoken word recognition process. In a view that sees the traditional 'segmental' and 'suprasegmental' planes as integrated levels of an interdependent prosodic system, the syllable plays a pivotal role in helping the listener extract lexical and syntactic structure from acoustic information (Beckman, 1996; Hawkins, Smith, 2001). Specifically, the syllable constitutes the linguistic unit in which information useful for speech segmentation, rhythmic patterns, and lexical access is encoded (McQueen, Dilley, 2020).

In automatic speech recognition research, the basic unit of processing has traditionally been the segment. However, the use of the syllable, or phonetic units of similar size, has often been called upon as an alternative signal processing strategy that would apply to a larger time scale than the segment, so as to be more robust in the face of variability across speakers and speech styles (Greenberg 1999; Morgan, Bourlard & Hermansky, 2004; Coro, Massoli, Origlia & Cutugno, 2021). Similarly, in the most

recent research in ASR, the role of the syllable in processing is tested within Transformer-based neural models both through their use as tokens, instead of phonemes and subwords, for recognition of written words, and by trying to trace their role in the internal speech representations of the neural network (Anoop, Ramakrishnan, 2023; Cho, Mohamed, Li, Black & Anumanchipalli, 2024; Vitale, Cutugno, Origlia & Coro, 2024; Shon, Kim, Hsu, Sridhar, Watanabe & Livescu, 2024).

In this paper, we trained a neural model for ASR to process and recognize phonological syllables and assemble them into words. The purpose of this study is to conduct a linguistic analysis on the output of syllabic processing of a neural speech recognition system. A large acoustic model was therefore fine-tuned to map the speech signal onto a phonological transcription that was segmented into syllables and words (see § 2.2).

The primary objective of the linguistic analysis is to observe variation in the final neural representation of syllables based on the effect of syllable token frequency, lexical accent position, syllable type, and parts of speech (to which the syllable belongs). The effects of syllable-based phonetic restructuring of word boundaries, although present and identified in the analysis, will not be the subject of this study.

To identify the different types of "behavior" of the neural network in rescanning syllables and words, a fine-grained linguistic annotation system was developed, a system that was necessary to overcome the opacity of purely quantitative metrics such as Word-Error-Rate or, in our case, Token-Error-Rate. By doing so, it is possible to more precisely "count" prediction types and associate them with linguistic features of speech. This was performed using Multiple Correspondence Analysis, an exploratory multivariate statistic method that revealed patterns of association between the output behavior of the neural network and a set of linguistic factors. Although conducted with different computational methods and on a phonetically annotated dataset, the results of our exploratory analysis are compared with those of Schettino, Vitale & Cutugno (2023).

2. *Methodology*

The workflow for this research project was structured as follows: first, we collected and automatically transcribed the data for the training. Next, we designed a rule-based syllabification algorithm and integrated it in the tokenizer used to fine-tune the ASR model. Following the training, we tested the model and extracted a sample of 300 predicted sentences to conduct a qualitative linguistic analysis. We then categorized discrepancies between predictions and references at both token and word level. Subsequently, we generated a comprehensive database, including detailed information such as token composition and word and token frequency, using a semi-automatic annotation approach. Lastly, we treated deviation events as dependent variables and examined their relation to the features characterizing each token.

2.1 Preparation of the data

2.1.1 Dataset

The data used to fine-tune and test the model has been collected from the Italian dataset of the Common Voice 13.0 corpus (Ardila, Branson, Davis, Henretty, Kohler, Meyer, Morais, Saunders, Tyers & Weber, 2020). It consists of approximately 30 hours of read speech, with each audio file having a maximum length of 7 seconds. The prompts for reading are gathered from Wikipedia and other public domain resources, resulting in the presence of several low frequency words and proper nouns. Furthermore, the dataset is crowd-sourced, meaning that it is robust against inter-speaker and diatopic variation. Aiming to a qualitative analysis centered on phonological aspects, we opted to train and evaluate the model using the phonemic transcriptions of the sentences instead of their original orthographic format. We used the WebMAUS Basic tool provided by the Bavarian Archive for Speech Signal (Schiel, 2015) to transcribe our dataset into X-SAMPA. Before feeding the model with the training data, we ensured the quality of the transcriptions by manually correcting acronyms, ordinal numbers, foreign proper nouns, and other few elements that caused a discrepancy between the original graphemic sentence and its automatic transcription.

2.1.2 Tokenization

While several syllabification algorithms are already available for processing audio signal (Cutugno, Passaro & Petrillo, 2001; Cutugno, Origlia & Schettino, 2018) and text (Iacoponi, Savy, 2011; Bigi, Petroni, 2014), we chose to implement our own tokenization algorithm on textual data. Proceeding this way, we facilitated the integration of the syllabifier in the tokenizer class *Wav2Vec2PhonemeCTCTokenizer* required by the model to perform the training (Transformers, 2023). Moreover, we obtained the output in an optimal format, thus avoiding the need for additional processing of the predictions before conducting the linguistic analysis. Our rule-based syllabification algorithm acts according to the Maximal Onset Principle (Kahn, 1976) and the Sonority Sequencing Principle (Clements, 1990). To streamline data processing and reduce the complexity of managing exceptions, we opted to treat /s/ + plosive clusters and geminates as tautosyllabic onsets, despite acknowledging the lack of phonological grounding in this approach (Marotta, Vanelli, 2021). This computational simplification allowed for a more efficient handling of syllable structures within the model.

(1) aspettare → a.spe.tta.re

In tokenized sentences, blank spaces separate tokens, while pipe symbols separate words. This structure allows us to observe how the model performs word parsing while retaining the information concerning the recognition of individual tokens.

(2) lo studente aspetta → lo | stu dEn te | a spe tta

2.2 Training setup

The initial model used for the experiment is a pretrained Transformer-based model implemented by Microsoft and named WavLM-Large (Chen, Wang, Chen, Wu, Liu, Chen, Li, Kanda, Yoshioka, Xiao, Wu, Zhou, Ren, Qian, Qian, Zeng, Yu & Wei, 2022). This model was trained on 94k hours of English raw audio data. Since its training is based on pure audio signal, its capabilities extend beyond conventional speech recognition tasks, encompassing various other speech-related applications, such as speaker identification. Its proficiency in tasks beyond speech recognition underscores its capacity to extract not only spoken words but also valuable additional knowledge, such as for example speaker-related information. The fine-tuning process for this model included more than just training it with Italian data; it also involved replacing the existing tokenizer. As previously described, the tokenizer relied on a rule-based syllabification algorithm, which encompasses a set of categories, or what can be better defined as syllable-based vocabulary. Following a principle of economy, the vocabulary of tokens consists of the 487 highest frequency syllables and the phonemic inventory of Italian, reaching a total of 517. This approach ensures the transcription of a broad range of possible utterances.

2.3 Linguistic analysis

2.3.1 Categorization of prediction types

The analysis of transcriptions generated by speech recognition systems has been long instrumental in refining the accuracy of these models (Meripo, Nimshi, Konam & Sandeep, 2022; Palmerini, Savy, 2014; Perero-Codosero, Espinoza-Cuadros & Hernández-Gómez, 2022). This process involves examining the transcription to identify areas for improvement within the model. However, transcription error analysis is often conducted on models that transcribe the audio signal into string of characters, or graphemes. As a result, some errors may not be readily traceable to the actual content of the audio. When manually examining the transcriptions generated by our trained ASR system, we noticed discrepancies that could more likely have been attributed to the phonetic organization of connected speech rather than to a deficiency in the training process of the ASR system.

(3) per i brasiliani → pe i | bra zi lja ni

In the context of this study, labeling the deviations found in predictions as errors is misleading. We found it more suitable to categorize them as prediction types instead. In this category the occurrence of unmarked events, that is, when tokens were correctly transcribed by the model, is also included.

We defined prediction types to serve as prediction tags (PT) to categorize the result of the comparison between individual tokens, for both predicted (*Pt*) and reference (*Rt*) units. The classification is based on a system of ordered suffixes that denotes the type of phenomenon at different levels of granularity, as follows:

1. word-level event (if applies);
2. token movement direction (if applies);
3. operation/equality;
4. token or word level.

Word-level events are defined as instances where the misplacement of Pt affects canonical word boundaries. There are three possible situations: the merging of two or more words (*mer*); a single word split in two or more parts (*div*); lastly, tokens that are moved within the boundaries of an adjacent word (*mv*). The direction of the shift in the latter case is marked with the suffixes *forw* and *back*. At the core of each PT is the type of operation, namely substitution (*sub*), deletion (*del*) and insertion (*ins*). If no operation is detected, the equality tag (*eq*) is assigned. Additionally, the suffix syl is added when the event involves a token but not an entire word. On the other hand, if the syllable token is also a word, the suffix word is added to the tag. The detailed list of PT is provided in Appendix 1, while an example of annotation is illustrated in example 4 and Tab. 1.

(4) *Rs*: kom prEn de |i | kwa ttro | sti li | ka nO ni tSi |

Ps: kom prEn de | kwa ttro | sti lli ka | a nO ni tSi

Table 1 - *Prediction tags associated to prediction and reference token*

PT	*Pt*	*Rt*
eq_syl	kom	kom
eq_syl	prEn	prEn
eq_syl	de	de
del_syl_word	-	i
eq_syl	kwa	kwa
eq_syl	ttro	ttro
eq_syl	sti	sti
sub_syl	lli	li
mv_back_eq_syl	ka	ka
ins_syl	a	-
eq_syl	nO	nO
eq_syl	ni	ni
eq_syl	tSi	tSi

2.3.2 Creation of the database

The next step of our work aimed to create a detailed database focused on the single *Pt* and *Rt*, with the corresponding PT serving as a pivotal point. With a total of 300 sentences consisting of approximately 5900 tokens, we opted for a semi-automatic

approach to assign PT; manual revision was anyway necessary to ensure the correct tagging in cases of ambiguity.

To facilitate the computation of the comparison between *Pt* and *Rt*, we deleted the blank spaces serving as segment separators in the tokens that were not to be found in the training vocabulary. To obtain tokens without blanks, reference sentences were syllabified with our algorithm, while the predictions required a different approach as some recognized tokens lacked a potential nucleus for syllable construction. To handle these cases, we developed a specific algorithm to directly assemble predicted tokens instead of relying on syllabification. In instances where the algorithm's output was ambiguous, we manually adjusted the information to ensure that the segments were reconducted to the correct tokens, as showed in example 5:

(5) Algorithm output → 'zvva'

After manual adjustments → (*Pw*) 'zv va' ↔ (*Rw*) 'zve va'

We then proceeded with the design of the master algorithm to achieve three main goals:

1. identify correspondence between predicted and reference words (*Pw* and *Rw,* respectively) in the prediction and reference sentences (*Ps* and *Rs,* respectively) despite the mismatches caused by marked prediction types;
2. track the tokens that were misplaced during the recognition and assigned within non-canonical word boundaries;
3. operate the comparison between *Pt* and *Rt* and assign a *PT* label to each, based on detected prediction types.

The algorithm relies on a dynamic index (*current_i*), which retrieves the to-be-compared words *Pw* and *Rw* from *Ps* and *Rs*. This index changes based on the boolean values of flags that are activated when prediction types are detected during the comparison of the previous pair of *Pw* and *Rw*.

Within each loop, *current_i* is initialized, and a pair of *Ps* and *Rs* is split into its *Pw* and *Rw* words.

Table 2 - *Items compared by the algorithm at a sentence, word, and token level*

Ps	['al ko ni', 'ne i', 'swO i li vi', 'sO no', 'sta ti', 'tra do tti']
Rs	['al ku ni', 'de i', 'swO i', 'li bri', 'sO no', 'sta ti', 'tra do tti']
Pw	['al', 'ko', 'ni']
Rw	['al', 'ku', 'ni']
Pt	'al'
Rt	'al--

While the dynamic index is less than the length of the longer sentence among *Ps* and *Rs*, the pair of words to be compared token-wise is retrieved by indexing the sentences with *current_i*. To establish whether there is a correspondence between *Pw* and *Rw* and, consequently, recognize them as matching and comparable elements, we first consider the length of the sentences and then their similarity at a word level.

This is determined in two ways: firstly, by checking in *Pw* the presence of tokens contained in *Rw*. Secondly, by calculating the Levenshtein distance between the concatenated tokens of the list as a single string and comparing it to a threshold that varies in function of the token's length. Two strings are similar if the score is above 0.75 when token's length is equal or shorter than 3 characters or is above 0.50 if it's higher.

The comparison between *Pw* and *Rw* proceeds as follows.

1. If length(*Pw*) == length(*Rw*), we compare *Pt* and *Rt* pairwise to check for equality or similarity and we assign the tags *eq_syl* or *sub_syl* respectively.
2. If length(*Pw*) != length(*Rw*), the algorithm searches the correct correspondence between the words by retrieving the adjacent *Pw* and *Rw* indexing *Ps* and *Rs* with *current_i*; then, it attempts to classify the mismatch by identifying one of the following events based on a set of defined conditions exemplified below:
   a. word merging: len(*Pw*) > len(*Rw*), deviance(len(*Pw*)) = len(*Rs*[*current_i*+1]), deviance(*Pw*) in *Rs*[*current_i*+*1*], *Pw* in *Rs*[*current_i*+*1*], *Ps*[*current_i*+1] in *Rs*[*current_i*+2];
   b. token insertion: len(*Pw*) > len(*Rw*), deviance(len(*Pw*)) != len(*Rs*[*current_i*+1]), *Pw* != *Rs*[*current_i*+1], *Ps*[*current_i*+1] != *Rs*[*current_i*+2];
   c. token movement backwards: len(*Pw*) > len(*Rw*), len(*Ps*[*current_i*+1]) < len(*Rs*[*current_i*+1]), *Pw* ~ *Rw*;
   d. token movement forwards: len(*Pw*) > len(*Rw*), len(*Ps*[*current_i*+1]) > len(*Rs*[*current_i*+1]), deviance(len(*Pw*)) = (len(*Rs*[*current_i*+1]) - len(*Ps*[*current_i*+1]));
   e. word division: len(*Pw*) < len(*Rw*), *Ps*[*current_i*+1] ~ *Rw*;
   f. token deletion: len(*Pw*) < len(*Rw*), *Pw* ~ *Rw, Pw* != *Rs*[*current_i*+1].
3. Once the event is detected, the related flag is activated, and the analysis proceeds at a token level with the comparison between each *Pt* and *Rt* in *Pw* and *Rw*.
4. The resulting labels are added to the label list. If none of the conditions set is met or ambiguities arise, the label "manual annotation" is appended to notify the need for manual annotation of the involved sentence.
5. Based on the flag activated in the previous loop, the *Pw* and *Rw* to compare next are defined as follows:
   - no flag activated → *Ps*[*current_i*], *Rs*[*current_i*];
   - word division → *Ps*[*current_i*], *Rs*[*current_i* -1];
   - word merging → *Ps*[*current_i* -1], *Rs*[*current_i*];
   - word deletion → *Ps*[*current_i*-1], *Rs*[*current_i*];
   - word insertion → *Ps*[*current_i*], *Rs*[*current_i*-1].

Due to the high quality of the recognition performed by the model, most of the tags consisted in "eq_syl" and were assigned correctly, meaning that the correspondence between the words and tokens was found smoothly.

A challenging case was represented by assimilations, due to the fact that one of the tokens involved was not deleted or moved within the word but merged in another token. This represents a problem for the similarity function, since one

of the two Rt appears to be missing and is tagged with the "del" PT, despite the fact that it became part of an adjacent Pt during the recognition and, therefore, it has been recognized. In such cases, even though it may appear misleading from a linguistic point of view, the "del" PT it is functional for the algorithm: in fact, to ensure the computational integrity of the final data frame, it remains necessary to assign PT equal to the token count.

(6) *Ps*: non | ra ddZun se | ma i | tSi | pr**e** ttSe ttsjo na li

Rs: non | ra ddZun se | ma i | tSi fre | **e** ttSe ttsjo na li

PT: ['eq_syl'], ['eq_syl', 'eq_syl', 'eq_syl'], ['eq_syl', 'eq_syl'], ['mv_back_eq_syl'], ['aggl_sub_syl', '**aggl_del_syl**', 'aggl_eq_syl', 'aggl_eq_syl', 'aggl_eq_syl', 'aggl_eq_syl']

Given the exploratory nature of our research, we conducted a manual revision and correction of the provisional label list, despite recognizing that this approach is not optimal. During this check, we decided to discard few predictions due to the poor quality of the recognition, that made it impossible to assign PT as exemplified below:

(7) *Ps*: il | pa ti ko la | kom pi kka i | me re tSe

Rs: in | par ti ko la re | lo | im pe JJa va | l | i dE a | del | su o | mo nu men to | fu ne bre |

Lastly, we added detailed information in correspondence of each row and obtained a database with the following structure:

- "filename": ID of the file within the Common Voice dataset;
- "sentence": original sentence transcription in graphemic format as extracted from Common Voice;
- "tokenized_P": predicted sentence tokenized into phonological syllables (model's output format);
- "tokenized_R": predicted sentence tokenized into phonological syllables;
- "deviation": the presence or absence of a marked event in correspondence of the predicted token when compared to the reference token;
- "prediction type": type of recognition (see § 2.3.1 and Appendix 1);
- "token_P": predicted token (*Pt*);
- "token_R": reference token (*Rt*);
- "word": reference word in graphemes;
- "number_of_tokens": number of tokens in the reference sentence;
- "number of words": number of words in the reference sentence;
- "freq_tok_P": relative frequency of the predicted token calculated within the whole dataset used to train the model;
- "tok_type_P": syllabic type of the predicted token;
- "freq_tok_R": relative frequency of the reference token calculated within the whole datasets used to train the model;
- "tok_type_R": syllabic type of the reference token;
- "in_vocab_P": presence or absence of the predicted token in the training vocabulary;

- "in_vocab_R": presence or absence of the reference token in the training vocabulary;
- "stress_R": presence or absence of phonological stress in the reference token, extracted with G2P (Reichel, Kisler, 2014; Goslin, Galluzzi & Romani, 2014; Spinelli, Sulpizio & Burani, 2017);
- "pos_tags": part of speech tag of the reference word, assigned with spacy library (Honnibal, Montani, 2017);
- "freq_word_relative": frequency of reference words calculated on the entire dataset;
- "abs_freq_word_R": absolute frequency of reference words calculated on the entire dataset.

3. *Data analysis and results*

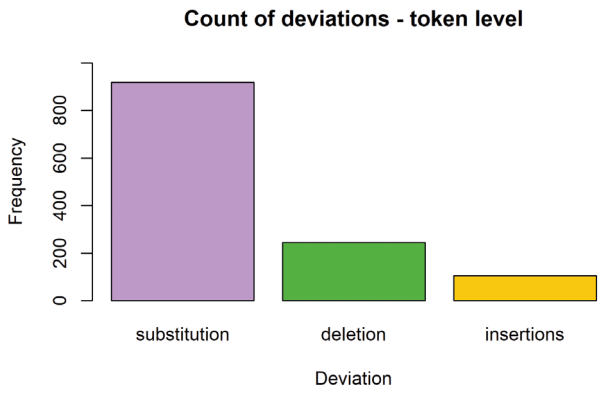
To perform the analysis on our prediction database we first examined the distribution of prediction types. In the second part of the analysis, we considered the patterns of association between prediction types, token frequency, presence in the training vocabulary, and lexical stress using Multiple Correspondence Analysis (MCA).

The quality of the recognition achieved by the syllable-based fine-tuned ASR model is reflected in the number of deviations found in the predictions: only 28 % of the tokens were affected by some marked recognition types, meaning that most tokens were recognized correctly and, therefore, "eq_syl" is the most frequent category (72 %).

The detailed distribution of marked prediction types is displayed in the following figures. Our system of structured labeling allows us to focus separately on the phenomena that are traceable to the token level and on those affecting the sentence structure through word boundary misplacement.

If we consider the high-level recognition labels that mark the type of operation at a token level (Fig. 1), it emerges that substitution is the most frequent operation, followed by deletion and insertion. This indicates that most tokens that were not recognized precisely are still to be found in the transcription hypothesized by the model. On the other hand, the deletion and insertion of tokens (including those which also correspond to entire words, like prepositions, determiners or some auxiliary verb forms) represent a more consistent discrepancy in the recognition. It should be noted that the use of automatically generated phonological transcriptions as a reference causes an increment in the number of substitutions due to the speech variability that characterizes the corpus.

Figure 1 - *Count of deviations at a token level*



Shifting our focus to the labels assigned to prediction types impacting canonical word boundaries, Fig. 2 shows the distribution of the operation/equality tags within the three categories. It can be observed that merging is the most frequent process with 401 tokens belonging to the words involved. Divided words follow with 206 occurrences, while the movement of single tokens to adjacent words is less frequent with a total of 48 instances. Additionally, the movement label applies to single tokens, unlike the other categories that consider all the tokens within the affected words. Examining the distribution of equality, it is evident that the tokens involved in merged and divided words were mostly recognized correctly, while substitution is the second most performed operation. It is interesting to notice that token deletion happens more often in merged words, while the amount of token insertion is proportionally higher in words that undergo division. Regarding tokens affected by movement, the distribution of equal and substituted tokens is nearly identical. Deletions do not apply since moved tokens cannot be missing from the prediction due to their intrinsic nature. As a result, insertions are not relevant either, because inserted tokens were not present in the reference and cannot be considered moved.

Figure 2 - *Count of deviations at a word level*

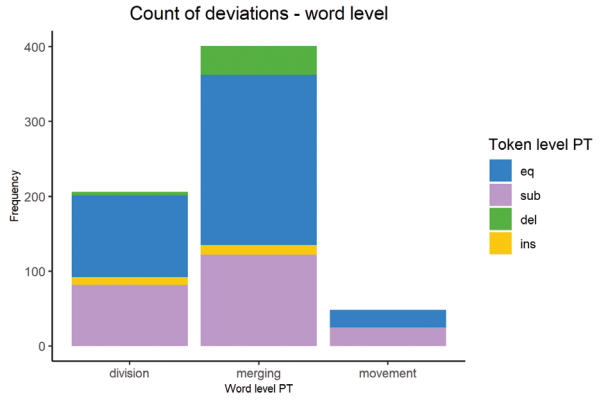
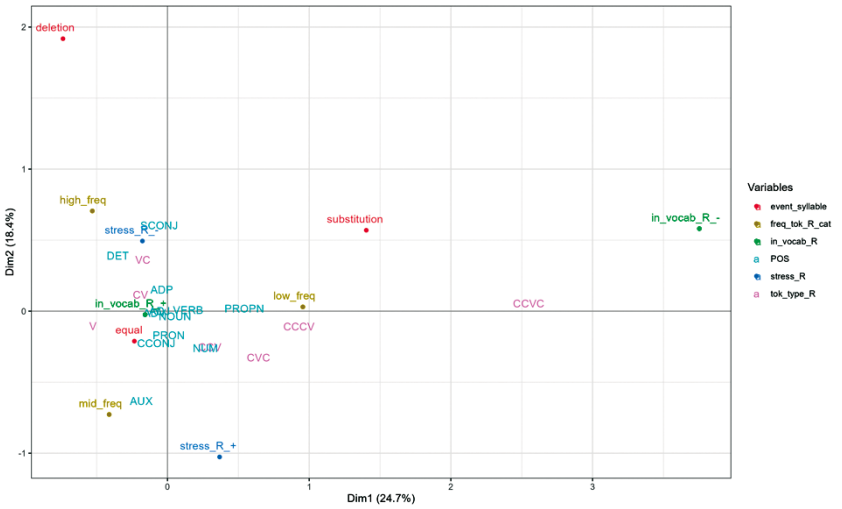


Fig. 3 shows the results of Multiple Correspondence Analysis (MCA), a multivariate statistical technique performed using *FactoMinerR* R package. As mentioned previously, in the analysis we observed the emergence of patterns of association between prediction types (event_syllable), token relative frequency (freq_tok_R_cat), token presence in the training vocabulary (in_vocab_R) and lexical stress (stress_R). In order to facilitate the analysis, the relative frequency of reference tokens was discretized in three categories using quantiles. Part of speech tag (POS) and syllable type (tok_type_R) were not included in the computation of the low-dimensional space but are projected onto it a posteriori, as supplementary variables, to contribute to the linguistic interpretation of the analysis. Since our focus is on reference tokens and their attributes, insertion (which is the least frequent operation) is not relevant in this context. The most complex syllable types (e.g. CCVCC or CCCVC as in 'edward' and 'mainstream') were also excluded from the analysis given their low frequency (less than 10 observations) and since, in most cases, they correspond to proper nouns in English.

MCA is a dimensionality reduction technique for categorical variables based on variance maximization, as in Principal Component Analysis (PCA). The dimensions on which observations are projected are those that represent the greatest variance in the original data (see Fig. 3: Dim1 = 24.7 %, Dim2 = 18.4 %), hence their meaning is derived from the actual distribution of variable values on the new plane. On the left side of the plot, we can observe the distribution of the tokens learned by the model during the training phase. It is interesting to notice in the top section how unstressed high-frequency tokens (greater than 2.23 %), consisting mostly in subordinating conjunctions and determiners, are related to deletion. In the bottom-left section we find the mid-frequency items (range: 0.5 % - 2.23 %). These tokens are characterized by a simple syllabic structure (CV) and tend to be recognized correctly. The tokens that were absent in the training vocabulary are on the right side of the MCA chart. Since these tokens are characterized by a more complex syllabic structure, they are naturally less frequent in the dataset. To compensate for this lack of information, the model adopts substitution as a strategy. The tokens are part (or, in some cases, represent) mostly proper and common nouns, as well as numerals.

Figure 3 - *Multiple Correspondence Analysis (MCA)*



4. *Discussion and future work*

Using a neural network to recognize syllables seems to show how the phonetic structure of the speech signal changes. In our analysis, the phonetic elements that undergo the deletion process are elements that usually disappear in connected and spontaneous speech, which was not expected since the training and test data consisted of read speech. The reconstructed words mainly involve deletion phenomena and known regularities in connected speech, such as element frequencies and certain parts of speech (POS) (Bybee, 2006). An innovative aspect of our study is the role of frequency, which shows up in three ways. For low-frequency syllables, the network compensates for their absence by substituting them. At medium frequency, the network works best, and it optimizes its recognition processing using high frequency syllables, as shown in other studies on spontaneous speech (Schettino, Vitale & Cutugno, 2023). This reveals the network ability to adapt based on how often syllables occur, demonstrating its potential to mimic natural language patterns.

To support linguistic analysis, we have developed a new annotation schema that could lead to a clearer metric than the traditional Word Error Rate (WER) from a linguistic standpoint. This annotation schema can be further developed into a metric. However, the algorithm currently lacks accuracy in some cases. Therefore, we aim to improve it by generalizing it to account for all instances of predictions.

One major limitation of our study is that we did not perform a phonetic analysis of the syllable directly from the acoustic signal, so its actual phonetic structure as well as the resyllabification process were not observed on the phonetic level. Additionally, the number of syllables used in our vocabulary (tokenizer) was limited, thus restricting the generalizability of the results.

Future research must therefore expand primarily in two areas: (1) the analysis of the role of token frequency, as information available to the neural model, in relation to the variables observed here, through the construction of balanced and representative vocabulary; (2) providing a phonetic representation of the syllable grounded in acoustic information that would allow the neural model's behavior to be evaluated not as a "deviation" from phonological expectations, but as a faithful processing of the acoustic characteristics of the speech signal.

In conclusion, we believe that our study shows, albeit in a preliminary way, how complex computational tools such as modern neural networks can be used by linguists as models to simulate and test linguistically relevant hypotheses.

References

Anoop, C.S., Ramakrishnan, A.G. (2023). Suitability of syllable-based modeling units for end-to-end speech recognition in Sanskrit and other Indian languages. In *Expert Systems with Applications,* 220, 119722. https://doi.org/10.1016/j.eswa.2023.119722

Ardila, R., Branson, M., Davis, K., Henretty, M., Kohler, M., Meyer, J., Morais, R., Saunders, L., Tyers, F.M. & Weber, G. (2020). *Common Voice: A Massively-Multilingual Speech Corpus*. arXiv. https://doi.org/10.48550/arXiv.1912.06670

Beckman, M.E. (1996). The Parsing of Prosody. In *Language and Cognitive Processes*, 11(1–2), 17–68. https://doi.org/10.1080/016909696387213

Bigi, B., Petrone, C. (2014). A generic tool for the automatic syllabification of Italian. In *Proceedings of the First Italian Conference on Computational Linguistics CLiC-it 2014.* Pisa, Pisa University Press, 73-77.

Bybee, J. (2006). From Usage to Grammar: The Mind's Response to Repetition. In *Language*, 82, 711-733. 10.1353/lan.2006.0186

Chen, S., Wang, C., Chen, Z., Wu, Y., Liu, S., Chen, Z., Li, J., Kanda, N., Yoshioka, T., Xiao, X., Wu, J., Zhou, L., Ren, S., Qian, Y., Qian, Y., Zeng, M., Yu, X. & Wei, F. (2022). WavLM: Large-Scale Self-Supervised Pre-Training for Full Stack Speech Processing. In *IEEE Journal of Selected Topics in Signal Processing*, 16, 1-14. 10.1109/JSTSP.2022.3188113

Cho, C.J., Mohamed, A., Li, S.-W., Black, A.W. & Anumanchipalli, G.K. (2024). SD-HuBERT: Sentence-Level Self-Distillation Induces Syllabic Organization in HuBERT. *arXiv*. Retrieved from http://arxiv.org/abs/2310.10803

Clements, G.N. (1990). The role of the sonority cycle in core syllabification. In Kingston, J. & Beckman, M.E. (Eds.), *Papers in Laboratory Phonology: Volume 1: Between the Grammar and Physics of Speech.* 1. Cambridge University Press, 283-333. https://doi.org/10.1017/CBO9780511627736.017

Coro, G., Massoli, F.V., Origlia, A. & Cutugno, F. (2021). Psycho-acoustics inspired automatic speech recognition. In *Computers & Electrical Engineering*, 93, 107238. https://doi.org/10.1016/j.compeleceng.2021.107238

Cutugno, F., Origlia, A. & Schettino, V. (2018). Syllable structure, automatic syllabification and reduction phenomena. In Cangemi, F., Clayards, M., Niebuhr, O.,

Schuppler, B. & Zellers, M. (Eds.), *Rethinking Reduction: Interdisciplinary Perspectives on Conditions, Mechanisms, and Domains for Phonetic Variation*. Berlin/Boston: De Gruyter Mouton, 205-242. https://doi.org/10.1515/9783110524178-007

Cutugno, F., Passaro, G. & Petrillo, M. (2001). Sillabificazione fonologica e sillabificazione fonetica. In Albano Leoni, F., Sornicola, R., Stenta Krosbakken, E. & Stromboli, C. (Eds.), *Dati empirici e teorie linguistiche, Atti del XXXIII, Congresso della Società di Linguistica Italiana*. Roma: Bulzoni, 205-232.

Goslin J., Galluzzi C. & Romani, C. (2024). PhonItalia: a phonological lexicon for Italian. *Behav Res Methods*, 46(3), 872-86. doi: 10.3758/s13428-013-0400-8. PMID: 24092524.

Greenberg, S. (1999). Speaking in shorthand – A syllable-centric perspective for understanding pronunciation variation. In *Speech Communication*, 29, 159-176.

Hawkins, S., Smith, R. (2001). Polysp: A polysystemic, phonetically-rich approach to speech understanding. In *Italian Journal of Linguistics*, 13, 99-189.

Honnibal, M., Montani, I. (2017). spaCy 2: Natural language understanding with Bloom embeddings, convolutional neural networks and incremental parsing. [Python Package] Version 3.8.1. https://spacy.io/

Iacoponi, L., Savy, R. (2011). Sylli: Automatic Phonological Syllabification for Italian. [Computer Program] Version 0.9.8. https://sylli.sourceforge.net/

Kahn, D. (1976). Syllable-based generalizations in English phonology. Ph.D. dissertation, Massachusetts Institute of Technology. https://dspace.mit.edu/handle/1721.1/16397

Marotta, G., Vanelli, L. (2021). *Fonologia e prosodia dell'italiano*. Roma: Carocci Editore.

Mcqueen, J.M., Dilley, L. (2020). Prosody and Spoken-Word Recognition. In Gussenhoven, C., Chen, A. (Eds.), *The Oxford Handbook of Language Prosody*. Oxford University Press, 508-521. https://doi.org/10.1093/oxfordhb/9780198832232.013.33

Meripo, N., Konam, S. (2022). ASR Error Detection via Audio-Transcript entailment. 10.48550/arXiv.2207.10849

Morgan, N., Bourlard, H. & Hermansky, H. (2004). Automatic Speech Recognition: An Auditory Perspective. In Greenberg, S., Ainsworth, W.A., Popper, A.N., Fay, R.R. (Eds.), *Speech processing in the auditory system*. New York: Springer, 309-338.

Palmerini, M., Savy, R. (2014). Gli errori di un sistema di riconoscimento automatico del parlato: analisi linguistica e primi risultati di una ricerca interdisciplinare. In *Proceedings of the First Italian Conference on Computational Linguistics CLiC-it 2014 and of the Fourth International Workshop EVALITA 2014*. Pisa: Pisa University Press, 281-285.

Perero-Codosero, J.M., Espinoza-Cuadros, F.M. & Hernández-Gómez, L.A. (2022). A Comparison of Hybrid and End-to-End ASR Systems for the IberSpeech-RTVE 2020 Speech-to-Text Transcription Challenge. In *Applied Sciences*, 12*(*2), 903. https://doi.org/10.3390/app12020903

Reichel, U.D., Kisler, T. (2014). Language-independent grapheme-phoneme conversion and word stress assignment as a web service. In Hoffmann, R. (Eds.), *Elektronische Sprachverarbeitung. Studientexte zur Sprachkommunikation*, 71, TUDpress, Dresden, 42-49.

Sasindran, Z., Yelchuri, H. & Prabhakar, T.V. (2024) SeMaScore: A new evaluation metric for automatic speech recognition tasks, arXiv.org. Available at: https://arxiv.org/abs/2401.07506

Schettino, L., Vitale, V.N. & Cutugno, F. (2023). Syllabic Reduction in Italian Connected Speech: Towards the Integration of Linguistic and Computational Approaches. In *Proceedings of the 20th International Congress of Phonetic Sciences*, 2039-2043. Prague: Guarant International.

Schiel, F. (2015). A Statistical Model for Predicting Pronunciation. In *Proceedings of the ICPhS 2015*, Glasgow, UK.

Shon, S., Kim, K., Hsu, Y.-T., Sridhar, P., Watanabe, S., & Livescu, K. (2024). DiscreteSLU: A Large Language Model with Self-Supervised Discrete Speech Units for Spoken Language Understanding. ArXiv. doi: 10.48550/arXiv.2406.09345

Spinelli, G., Sulpizio, S. & Burani, C. (2017). Q2Stress: A database for multiple cues to stress assignment in Italian. In *Behavior Research Methods*, 49(6), 2113-2126. doi: 10.3758/s13428-016-0845-7. PMID: 28039679.

Transformers. (2023). Tokenization Wav2Vec2 Phoneme (v4.35.2) [Computer software]. Hugging Face. https://github.com/huggingface/transformers/blob/v4.35.2/src/transformers/models/wav2vec2_phoneme/tokenization_wav2vec2_phoneme.py#L94

Vitale, V.N., Cutugno, F., Origlia, A. & Coro, G. (2024). Exploring emergent syllables in end-to-end automatic speech recognizers through model explainability technique. In *Neural Computing and Applications*, 36(12), 6875-6901. https://doi.org/10.1007/s00521-024-09435-1

Appendix

Appendix 1 - *Prediction tags*

Label	*Prediction*	*Reference*
eq_syl	do po | al ku ni |	do po | al ku ni |
sub_syl	mO do | **ve** tSo |	mO do | de tSo |
ins_syl	i | lo ro | a bi ta **tta** |	i | lo ro | a bi t a t |
del_syl	kom ple ta men te | sO - |	kom ple ta men te | so **lo** |
sub_syl_word	**kon** | E | di ven ta to |	non | E | di ven ta to |
ins_syl_word	te | **i** |	ti |
del_syl_word	so pra ttu tto | - | ma ssa ka tSe ts |	so pra ttu tto | **in** | ma ssa tS u se tts |
mv_eq_forw_syl	o ri dZi | **ni** mi ti ke |	o ri dZi ni | mi ti ke |
mv_sub_forw_syl	E stre | **ro** u ma no |	E sse re | u ma no |
mv_eq_back_syl	da **ve** | tra te |	da | ve tra te |
mv_sub_back_syl	tu tta vi a **no** |	tu tta vi a | non |
div_eq_syl	**a** | pu ddZa | da	a ppo ddZa ta |
div_sub_syl	a | pu ddZa | **da**	a ppo ddZa ta |
div_ins_syl	fra | zi **i** |	fra zi |
mer_eq_syl	**kwa ttro** po sti |	kwa ttro | po sti |
mer_sub_syl	**sE** | la u re a to |	si | E | la u re a to |
mer_ins_syl	pu kwe stE **ro** no | kO lle	kwe s t E r mo | ko lle |
mer_del_syl	fi nO - tto |	fi no | ad | O tto |