Computers and Electrical Engineering 93 (2021) 107238

Contents lists available at ScienceDirect

Computers and Electrical Engineering

journal homepage: www.elsevier.com/locate/compeleceng

Psycho-acoustics inspired automatic speech recognition☆

Gianpaolo Coro [a,*], Fabio Valerio Massoli [a], Antonio Origlia [b], Francesco Cutugno [b]

[a] *Istituto di Scienza e Tecnologie dell'Informazione "Alessandro Faedo" – CNR, Pisa, Italy*
[b] *Università degli Studi di Napoli Federico II, Napoli, Italy*

ARTICLE INFO

Keywords:
Automatic speech recognition
Deep learning
Long short term memory
Convolutional neural networks
Factorial hidden Markov models
Hidden Markov models
Speech
Psycho-acoustics
Syllables

ABSTRACT

Understanding the human spoken language recognition process is still a far scientific goal. Nowadays, commercial automatic speech recognisers (ASRs) achieve high performance at recognising clean speech, but their approaches are poorly related to human speech recognition. They commonly process the phonetic structure of speech while neglecting supra-segmental and syllabic tracts integral to human speech recognition. As a result, these ASRs achieve low performance on spontaneous speech and require enormous costs to build up phonetic and pronunciation models and catch the large variability of human speech. This paper presents a novel ASR that addresses these issues and questions conventional ASR approaches. It uses alternative acoustic models and an exhaustive decoding algorithm to process speech at a syllabic temporal scale (100–250 ms) through a multi-temporal approach inspired by psycho-acoustic studies. Performance comparison on the recognition of spoken Italian numbers (from 0 to 1 million) demonstrates that our approach is cost-effective, outperforms standard phonetic models, and reaches state-of-the-art performance.

1. Introduction

Spoken language recognition in human beings is natural, robust, and effective. People recognise each other's words also in situations of high background noise and reverberation using complex multi-channel information processing [1]. However, emulating and understanding these mechanisms goes beyond our current technological capabilities [2,3]. Nevertheless, automatic speech recognition has evolved in the last decades to propose high-performance commercial products to the large public [4,5]. The approaches of modern automatic speech recogniser (ASRs) are poorly related with human speech recognition processes, and building ASRs requires significant economic investments [6]. Indeed, high costs depend mainly on the manual preparation of large corpora of audio samples annotated at multiple levels (usually from sentence to phonetic levels), pronunciation models, and grammars. As a result, the implementation of high-performance ASRs is usually bounded to few and large corporations and their applications are confined to simple sentence transcribers or interactive voice responders. These ASRs usually neglect acoustic indicators such as intonation and supra-segmental tracts that are essential in human spoken dialogues, because these would be expensive in terms of modelling and data preparation. Consequently, modern ASRs have still issues at recognising spontaneous and conversational speech with a high accuracy [7–10].

The classic ASR architecture we took as a reference to build our ASR, is made up of four main processes (Fig. 1): (i) feature extraction, (ii) acoustic unit recognition, (iii) language model, and (iv) decoding. The first process extracts real numbered vectors of acoustic features out of an audio file. The temporal scale of these features is usually strictly sub-segmental and reflects the search

☆ This paper is for regular issues of CAEE. Reviews processed and approved for publication by the co-Editor-in-Chief Huimin Lu.

* Corresponding author.

E-mail addresses: coro@isti.cnr.it (G. Coro), massoli@isti.cnr.it (F.V. Massoli), antonio.origlia@unina.it (A. Origlia), cutugno@unina.it (F. Cutugno).

https://doi.org/10.1016/j.compeleceng.2021.107238
Received 6 November 2020; Received in revised form 18 May 2021; Accepted 26 May 2021

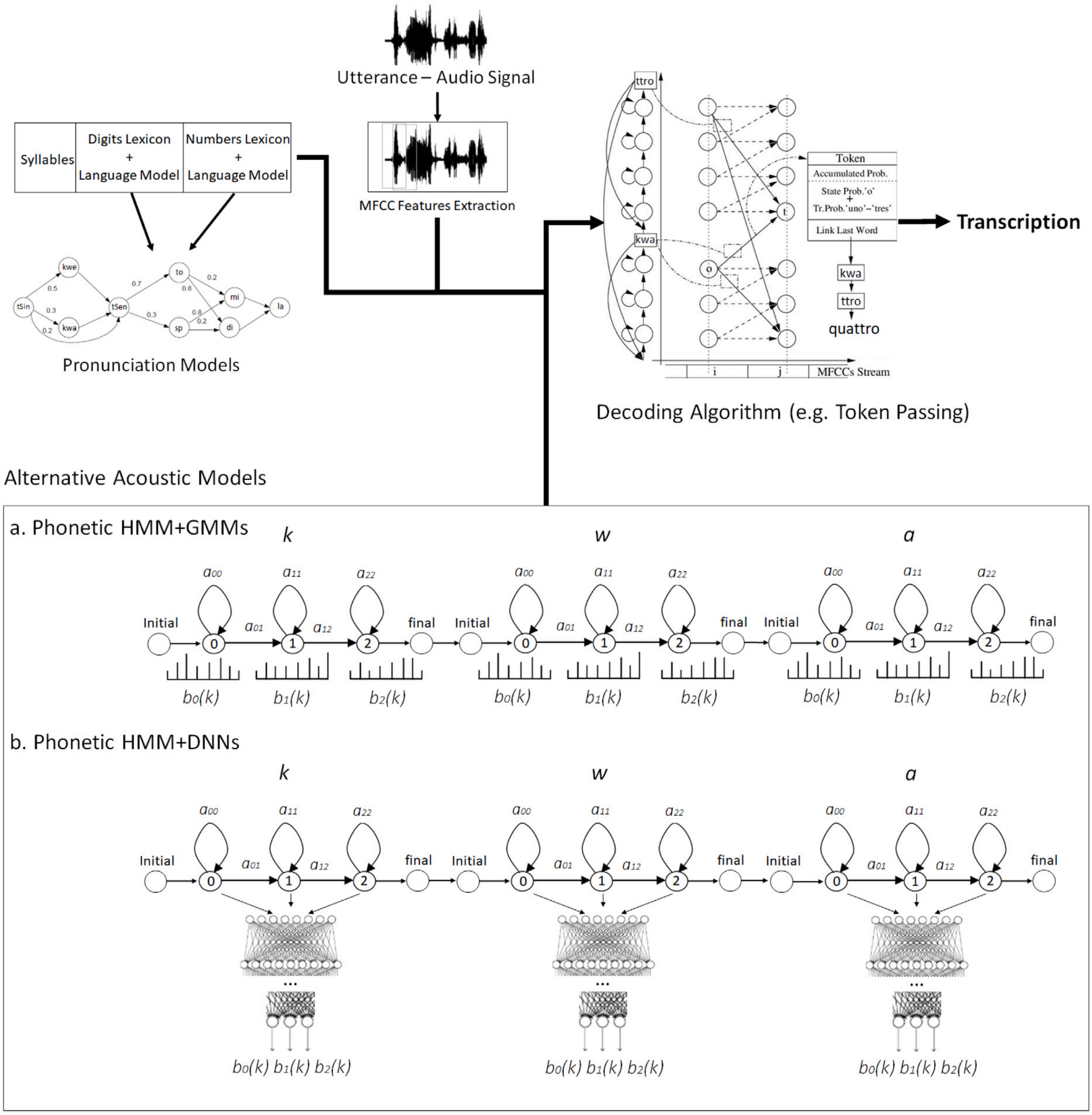

Fig. 1. Diagram of a standard ASR with alternative acoustic models: (a) tri-phonetic HMMs using GMM emission probabilities, (b) tri-phonetic HMMs using a DNN to simulate emission probabilities.
Source: The Token Passing schema is adapted from [11].

for stationary spectral conditions as much as possible. The second process – acoustic unit recognition – extends the temporal scope of the speech chain processing. The acoustic units used in most ASRs are classically related with the acoustic characteristics of the phonotactic distribution of co-articulatory processes. Indeed, most ASRs use contextual phones/tri-phones as acoustic units. This approach embeds coarse assumptions on the internal dynamics of the speech signal and attempts to model a form of contextual prediction of acoustic phenomena related with the nature of connected speech. Consequently, a classic ASR includes a large (~cubic) combination of elementary acoustic phonetic models. A further process estimates the likelihoods of these combined models to a segment of speech signal. State-of-the-art ASRs use between ~2000 and ~10,000 h of speech to train these models [12,13]. However, these acoustic models are not robust enough to manage the reduction processes that are frequent also in clear speech and that are more easily treated with a syllable-based approach [14–16]. The third process – language model – calculates the joint probability of a sequence of words and guides the ASR search among alternative words during the recognition. This model uses a grammar that specifies the permissible structures of the language. Statistical grammars (e.g. N-grams, [17]) are used to model complex languages and are automatically learned from large textual corpora. On the basis of a grammar, the language model assigns a higher probability to more likely word sequences. As a prerequisite, this process requires specifying all possible allowed words (the *lexicon*) and all different ways in which the used units of speech can be combined to build these words (*pronunciation models*). The language model is dependent on the particular lexicon and dialogue context the ASR is meant to manage. State-of-the-art large-vocabulary ASRs include a 10^6 order of magnitude words, and the language model is trained on tens of gigabytes of reference texts [18,19]. The fourth process – decoding – combines acoustic models with the language model to produce the most probable transcription of an input audio file. In most ASRs, acoustic models are implemented as Hidden Markov Models (HMMs, [20] and [21]), and the decoding process is strictly dependent on their state-based nature, where initial and final states establish the acoustic boundaries between consecutive speech units [22]. This type of HMM is a double stochastic model defined by (i) a finite set of states (ii) a transition matrix between consecutive states, (iii) a set of *emission* probability densities for each state to be associated with the vectors of acoustic features at a certain time, and (iv) a set of initial-state probability densities. Alternative models have been proposed to enhance HMM performance, for example Factorial HMMs (FHMMs, [23]) use sets of HMMs all with the same number of states and independent of each other, except for the fact that the emission probability of one state of an HMM depends also on the states of the other HMMs [24]. Most state-of-the-art ASRs use Deep Learning models to (i) classify speech-units, (ii) model HMMs emission densities, and (iii) extract acoustic features [25–30]. In particular, Deep Neural Networks (DNNs) are commonly used to model emission probabilities [31–33] and in some cases are replaced by Recurrent Neural Networks and Long Short-Term Memory models (LSTMs) [34–37]. LSTMs model longer-term dependencies between the elements of the input sequence [38–40] and have demonstrated high performance when used to classify single phonemes and syllables [34–37].

End-to-end ASRs are valid alternative architectures and can reach state-of-the-art performance [41–49]. These systems jointly learn all ASR components in one integrated approach, which reduces training and decoding time. However, they require an amount of training data that is by far higher than what is required by classic ASR architectures [50–53].

In this paper, a comparison between ASRs using both conventional and non-conventional approaches is presented. In particular, a novel approach for an ASR is proposed (Fig. 2) that uses several possible alternative acoustic models. Each time the ASR is instantiated, one among four models is used for acoustic unit modelling. Three of these models (FHMM, CNN, LSTM) are inspired by studies on the involvement of psycho-acoustic related features of human speech recognition, i.e. the multi-temporal processing of the speech signal at syllabic and phonetic levels [1,14,54,55]. Our study complies with the idea that although speech recognition in humans and machines is implemented in different ways, they should compute the speech signal in a similar way [56]. Other studies have investigated this similarity at a computational level, to build ASRs that accounted for the high variability of acoustic realisations of lexical representations, speaker independence, and new-word recognition [57]. For example, the Shortlist and SpeM ASRs addressed these properties by pursuing the hypothesis that human speech processing separates pre-lexical (abstract phonological representations before processing) and lexical levels [57,58]. Shortlist-B proposed a further pre-processing of the speech signal to reflect the characteristics of human pre-lexical processing [59]. However, these ASRs still worked at a phonetic-scale (i.e. with phonetic base units) and were mainly conceived for Hidden Markov Models-based acoustic units. Instead, our ASR uses syllabic-scale units and different acoustic model implementations and embeds a new decoding algorithm that is independent of the acoustic model used. The proposed acoustic models address syllable-related dynamics inspired by multi-temporal processing studies. Our results show that these models – especially two involving deep learning models – can use a limited training material to gain performance that is comparable with that of state-of-the-art systems on a non-trivial recognition task. Furthermore, our ASR can outperform standard-approach ASRs built upon the same training material. One drawback is its higher computational complexity, which requires using parallel or distributed processing for operational applications. As benchmark experiments, the recognition of spoken Italian digits (0–9) and numbers ranging from 0 to 1 million (excluded) from telephone-quality recordings were used. Digit recognition was used to compare ASR performance on controlled speech with a simple grammar and a low variability in the utterance of the syllables. Instead, the 0–1 million number experiment was used to test ASR performance with a non-trivial language model and with noisy audio that potentially included features of spontaneous speech (omissions, uncertain speech, false starts, etc.). In these experiments, when our ASR used LSTM syllabic acoustic models through an exhaustive decoding algorithm it had comparable performance with the state-of-the-art Google speech-to-text service [19] although it was trained with just one hour of speech samples. Overall, our experiment is a preliminary approach to open the way for questioning base ASR components and thus to provide a cost- and resource-effective solution to build ASRs.

Our results support the hypothesis that involving psycho-acoustic and supra-segmental information in an ASR, through the modelling of long and short term dynamics, likely increases its performance. This is an important topic impacting many different situations where general-purpose ASRs may not be applicable. First of all, ASRs based on DNNs are challenging for low-resource languages, which may be cut-off from a number of speech interfaces. Domain-specific recognition is also a challenge as it often poses strict constraints to ASRs. For example, pathological speech depends on the effect that a disease may have on human voice and requires strong ASR customisation. Further, domain-specific applications may use words or expressions that are not modelled by general-purpose systems, and sensitive data may not be sent to third parties for transcription. From the point of view of the open source community and of small enterprises, it is important to have the option not to depend on large companies to include speech-to-text capabilities in their applications. In general, a psycho-acoustically motivated solution provides significant adaptation capabilities and flexibility.

Overall, the main research question addressed by this paper is: *Can cognitive and psycho-acoustic theories on the syllable's role in human speech recognition inspire effective syllabic models and ASR architectures?*

This paper is organised as follows: Section 2 describes the assumptions, the material, and the models used in our ASR and alternative baseline ASRs. Section 3 reports the performance comparison between all ASRs on the recognition of syllables, digits, and numbers. Section 4 discusses the results and draws the conclusions.

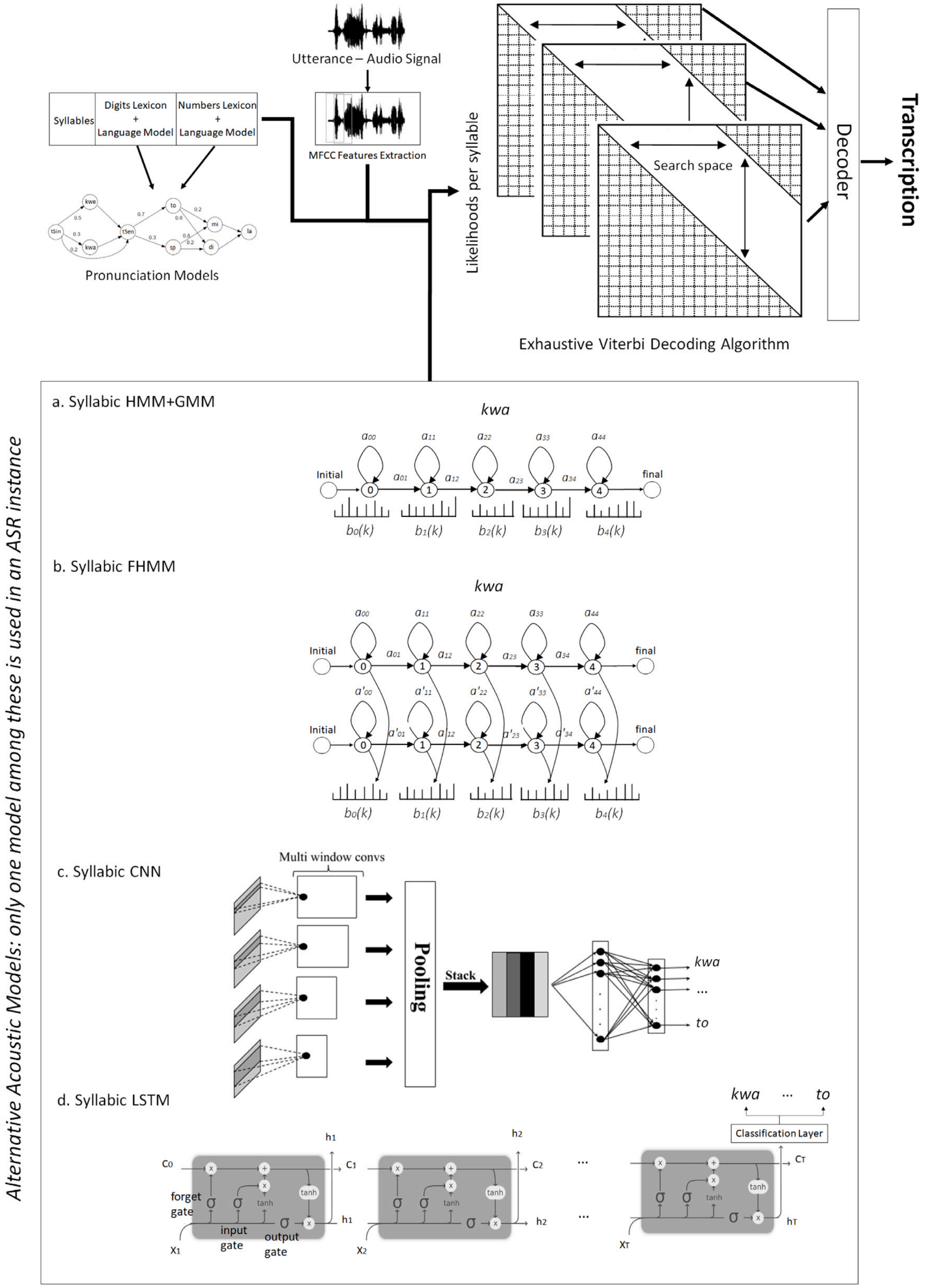

Fig. 2. Diagram of our ASR with acoustic models used alternatively (only one in an ASR instance): (a) syllabic HMMs using GMM emission probabilities, (b) syllabic Factorial HMMs, (c) Convolutional Neural Network using multi-temporal windows, with one output neuron for each syllable, and (d) Long Short Term Memory model, with one output for each syllable.

2. Material and methods

2.1. The base unit of speech

The base unit of speech is the minimal form of acoustic information around which human spoken language recognition is organised [60]. Indeed, the assumption that just one base unit exists is an exemplification of automatic modelling, since linguistic studies have instead indicated that language is organised around a combination of units with different temporal ranges [14]. Generally, human speech recognition uses several units with different time-scales, each containing coherent information at a given linguistic or paralinguistic level, and likely processes these units concurrently [1]. In automatic speech recognition, the base unit of speech is usually modelled as one unit (i) having a high number of manifestations, (ii) spectrally defined, and (iii) allowing the implementation of computationally efficient algorithms. The following subsections describe two base units commonly used in ASRs: Phonemes[1] and syllables.

2.1.1. Phoneme

Spoken language continuum is still commonly represented by a string of phonetic symbols (e.g. the International Phonetic Alphabet). This representation "hides" co-articulatory transitions and partial supra-segmental labelling (mainly word stress) [61], but allows representing an entire language using a large combination of few tens of symbols. The pronunciation of the string of symbols varies from person to person and from word to word. Generally, a phonetic symbol in the IPA alphabet is associated to a set of reference spectral frequencies (fundamental and formant frequencies) in its stationary section, and all transitional and dynamic spectral variations are assumed to be at the head and the tail of this section. These complex dynamics depend on the variable shape of the vocal tract and the possible activity of the vocal cords during the production of the sound. The identification of a phoneme from a portion of the signal spectrum requires catching the exact period in which the vocal tract has a defined structure that produces a stationary signal, and capturing expected transitions towards the following speech sound. ASRs apply an iterative window of ∼10 ms, running on the speech signal to capture both transitions and dynamics. Consequently, the identification of a speech segment as a given unit requires using statistical models that take into account its spectral context and the large variability of the phoneme across phonotactic contexts and speakers. Most ASRs use tri-phones as base units. However, this assumption neglects a large amount of information with higher time range contained in the spoken realisation of syllables and words [62,63].

2.1.2. Syllable

One empirical definition of *phonetic syllable* (or *pseudo-syllable*, [64]), is reported in [65]:

> "[...] a continuous voiced segment of speech organised around one local loudness peak, and possibly preceded and/or followed by voiceless segments".

This definition is application-oriented and is useful in automatic segmentation processes. However, while keeping the term *phonetic syllable*, we adopt a more precise definition by Roach [66, p. 70] that better accounts for co-articulation dynamics:

> "[...] consisting of a centre which has little or no obstruction to airflow and which sounds comparatively loud; before and after that centre [...] there will be greater obstruction to airflow and/or less loud sound".

Thus, a syllable can also be seen as a 100–250 ms segment of signal constructed around a high energy peak (*nucleus*), possibly preceded by an increasing energy slope (*onset*) and followed by a tail of decreasing energy (*coda*).

Syllables are probably the units around which human speech production developed [67]. Several studies have highlighted the importance of syllables in human speech perception, because syllables can be perceived in a spoken word also when they are not actually uttered (*mirage* effect) [68–70]. However, in "Categories" Aristotle already observed the vague nature of syllables, i.e. although each syllable is heard as separated from the other, generally they do not have defined boundaries. Indeed, disfluencies and reduction processes, observed also in clear but connected speech, can cause segment cancellation and indeterminacy of clear syllabic boundaries [71]. However, the rhythmical structure is always preserved, which means that more prominent units are usually less reduced and thus guarantee the preservation of speech-chain intelligibility [72]. From a psycho-acoustic point of view, a syllable contains much more information than the sequence of sounds constituting it [73,74]. Different brain activation patterns have been observed in human subjects hearing sequences of syllables or single syllables alternatively [75,76]. These experiments have also highlighted specific activation patterns in different brain areas, corresponding to multiple temporal scales of phonetic, syllabic, and supra-syllabic lengths.

One drawback of using syllable as a base unit in ASR, is that it is difficult to numerically describe all syllabic-scale ($\sim 100--250\,\mathrm{ms}$) speech properties that psycho-acoustic studies have indicated as related with human speech recognition robustness to speaker differences and adverse environmental conditions [14,77]. Also, syllable boundaries lack a consistent psycho-acoustic and linguistic definition that makes acoustic model specification non-uniquely defined [18,78]. Indeed, most syllabic-scale features are related with prosody, energy contour, and slow modulations (around 4 Hz). Several studies have demonstrated that incorporating

[1] For brevity, in our model descriptions we will improperly use the term "phoneme" to both indicate classes of speech sounds (phonemes) and their realisations (phones).

Table 1
Overall set of 42 pseudo-syllables involved in our experiment, plus two silence annotations: “sil” indicates a long silence (⩾ 200 ms), whereas “sp” indicates a shorter pause.

di	dje	do	due	dze	kwa	kwan	kwe	kwin	la	lle
mi	nno	no	o	ran	ro	se	sei	sil	sp	ssan
sse	ta	ti	to	tre	tren	tSa	tSen	tSi	tSin	tSo
ttan	tte	tto	ttor	ttro	tu	u	un	van	ve	ven

this information in syllabic acoustic models can increase the performance of an ASR [79–82]. However, these studies have also highlighted that it is not convenient to re-use standard ASR algorithms and assumptions when using syllabic base units [73,83,84]. Generally, most syllabic ASRs either represent syllables as sequences of phonetic acoustic models or build one acoustic model per syllable (or demi-syllable) while using phonetic features extracted from ~10 ms signal windows.

From a speech-processing operational point of view, pseudo-syllables bring more advantages than tri-phones. The automatic segmentation of a speech signal into pseudo-syllabic segments is facilitated by the correlation of these units with the modulations of sonority movements. In every group of sounds, there are as many syllables as clear relative peaks of sonority [85]. Furthermore, speech-intensity change is correlated with tonal speech perception [86] and is typically maximum between pseudo-syllables’ onsets and nuclei. These properties allow detecting tonal units automatically [87–89]. Moreover, the acoustic correlates of pseudo-syllables can be used to model pitch movements and produce effective pitch contour stylisation, especially over signal segments with complex spectral content [90]. These characteristics allow to build automatic pseudo-syllable classification and segmentation algorithms based on sound-intensity and spectral entropy analysis, overcoming common issues related with sonorant consonants with high intensity (e.g. nasals sounds) [91]. Moreover, automatic emotion detection and tracking models can be more efficient using pseudo-syllabic-scale analyses, instead of phonetic-scale analyses, by harnessing nuclei’s spectral richness and extracting information on speech rate and style [92].

Overall, pseudo-syllables are (i) widely used in psycho-acoustic studies as the base unit of analysis, (ii) correlated with observable neural activity, (iii) phonetically describable through specific intensity and spectral patterns, (iv) automatically detectable by computationally efficient algorithms, and (v) based on clear phonetic templates. Thus, using pseudo-syllables in ASR allows including human-related patterns and automatically identifying the signal segments that should be extracted and annotated to train the acoustic models and define word pronunciation models. The Italian part of the corpus used in this paper (Section 2.2) was also annotated at the pseudo-syllable level to foster experiments that could explore these operational advantages. For all these reasons, in this paper acoustic models are based on pseudo-syllables, although the term *syllabic model* is used for simplicity.

2.2. Lexicon, language model and acoustic features

The experiment reported in this paper uses a controlled lexicon to test the performance and the properties of different acoustic models and decoding algorithms. This lexicon was selected to require short preparation, analysis, and development times, and also to produce a non-trivial language model that included sufficiently varied speech and some characteristics of spontaneous and large-vocabulary contexts. Based on these requirements and following the suggestions of other works [73,83], the range of numbers between 0 and 999,999 – hereinafter indicated as [0,1M) – was selected as a benchmark vocabulary. The Speecon Italian corpus [93,94] includes sentences in this numeric range, recorded at 16 kHz from 400 different speakers with telephonic quality (with 25 dB ± 3 dB signal-to-noise ratio). Moreover, Speecon includes annotations for numbers and digits (i.e. from 0 to 9) at the phonetic, syllabic, and sentence levels. A total of 42 syllables (Table 1) and 19 phonemes – plus two silence models – and ~220 syllabic combinations were sufficient to build up a language model for the lexicon of numbers in [0,1M). Although the number of phonemes involved is not far from that of the whole Italian language (~32), 42 syllables are just a subset of the thousands of syllables of Italian. However, even with these syllables, spoken long numbers present characteristics of spontaneous speech, e.g. omissions, false starts, dialect inflexions, and uncertain speech.

In our experiment, the Speecon recordings were divided into 80%–20% training and test sets within a cross-validation process and did not include the same speakers. The Speecon syllabic-level annotations allowed to extract ~65 min of speech to train acoustic models and ~13 min to test their performance. Word-level annotations allowed to prepare recordings to test ASRs’ performance on numbers (~140 min) and digits (~55 min). The language model for [0,1M) was built using the CMUCLMTK toolkit v7 [95] as a statistical model trained with syllabic mono-grams, bi-grams, and tri-grams, and had a non-trivial perplexity of 9.8. As a training set for the language model, the linguistic syllabic subdivisions of all numbers in [0,1M) were used, plus their syllabic transcriptions in Speecon. These transcriptions report the syllables actually uttered in long numbers and thus simulate spoken sentence alterations due to continuous speech, which in turn allows building a more realistic syllabic language model. Finally, back-off probabilities were used to account for non-observed syllabic concatenations.

As acoustic features, 13 Mel-frequency cepstral coefficients (MFCCs, [96]) were used, with delta and double-delta features, for a total number of 39 features extracted from ~10 ms windows sliding over the signal with 50% overlap (5 ms). MFCCs are standard features used in ASR [8]. They are extracted out of the application of a filter bank based on the mel scale, which simulates the response of the human auditory system to speech frequencies. Although other types of mel scale-based features could be used [97–99], MFCCs were the set of features that all ASRs involved in our experiments could use. Thus, MFCCs allowed to measure performance differences that depended on the architectures and acoustic models rather than on the signal representation.

Furthermore, MFCCs typically allow to use a lower number of spectral features (typically 13 per window) than alternative methods (e.g. Mel-filter bank energies, which typically require 40 features per window) [100], which was beneficial to avoid overfitting issues with our limited training set.

Although our proposed acoustic models were syllabic (i.e. our ASR used syllables as base units), they were committed to extracting syllabic information from sequences of phonetic-scale features. Indeed, alternative features using syllabic-scale windows directly (100–250 ms) do not have the same consistency and robustness as MFCCs for speech recognition [77,81,101].

2.3. Speech decoding

The decoding process used by our reference ASRs (e.g. Token Passing, [22]) relies on the alignment between sequences of HMM states and the audio signal and makes use of the initial and final states to estimate phonetic and syllabic boundaries. However, some of the syllabic acoustic models proposed in this paper are not made up of sequences of states, thus decoders conceived to work with HMMs could not be used. For this reason, a new decoding algorithm was used that was independent of the nature of the incorporated syllabic acoustic model used. The algorithm described in [102] (hereinafter named *exhaustive Viterbi*) fitted our scopes because it uses syllabic acoustic models as black-boxes. This algorithm optimises the alignment of each acoustic model to the signal because it calculates the likelihoods of the models to all possible sub-sequences of the acoustic features extracted from the audio signal, i.e. it tests all possible alignments of the models to the signal. This approach increases the performance also of standard syllabic HMMs with respect to other decoding algorithms and has also been used in a commercial ASR [103]. In particular, the algorithm calculates the conditional probability distribution $P(W|X)$ of a sequence of n syllables $W = w_1 w_2 .. w_n$ given a sequence of T features $X = x_1 x_2 .. x_n$ extracted from the audio signal, where T is the length of the audio signal. During the calculation, the algorithm combines a syllable-based language model with syllabic acoustic models to find the optimal sequence of syllables W^* associated to X. The output of the algorithm is thus the sequence of syllables that is most probably associated with the audio signal. Through the pronunciation models it is possible to associate lexicon words to the optimal sequence of syllables and produce the orthographic transcription of the audio.

Formally, the algorithm produces the following optimal solution (the demonstration is reported in [102]):

$$P(W^*|X) = \underset{m \in Syl}{\operatorname{argmax}} \{ f(m,T) \cdot E(m) \}$$

where Syl is the complete set of N syllables included in the language model, $E(m)$ is the probability of model m to be an ending syllable, and $f(m,t)$ is the solution to the sub-problem of unit alignment in the time interval [1,t], defined as

$$f(m,t) = max \left\{ \begin{matrix} P(X_1^t|m) \cdot \pi(m) \\ \underset{1 \leqslant t^* < t, n \in Syl}{max} \left\{ f(n,t^*) \cdot P(m|n)^\gamma \cdot P(X_{t^*+1}^t|m) \right\} \end{matrix} \right\}$$

where γ is the language model's weight. Starting from time T, a backtracking process follows the definition of $P(W^*|X)$ to find the best alignment between the models and the signal. In particular, according to the definition of $f(m,t)$, the algorithm efficiently tests all possible alignments of all models to all segments of the audio signal, and thus optimises the models' recognition accuracy. However, one drawback is that it requires a pre-calculation of all models' likelihoods to all subsets of observations, i.e. $P(X_{t_i}^{t_j}|m)$ $\forall m \in Syl, 0 \leqslant t_i \leqslant T, 0 \leqslant t_j \leqslant T$. In particular, the algorithm first computes the V matrix:

$$V = \begin{pmatrix} P(X_1^1|m) & P(X_1^2|m) & ... & P(X_1^{T-1}|m) & P(X_1^T|m) \\ 0 & P(X_2^2|m) & ... & P(X_2^{T-1}|m) & P(X_2^T|m) \\ ... & ... & ... & ... & ... \\ 0 & 0 & ... & 0 & P(X_T^T|m) \end{pmatrix}$$

and then the backtracking procedure rapidly reconstructs the optimal solution. The overall algorithm's complexity is $O(T^2 N^2 C_l)$, i.e. it is quadratic in T and N, and also depends on the complexity of all likelihood calculations C_l by the acoustic models. The complexity of the algorithm can be reduced by introducing constraints on the minimum and maximum $t_j - t_i$ difference, and through a beam search strategy in the $f(m,t)$ calculations that filters out all likelihoods falling under a certain threshold. This strategy strongly reduces the number of non-zero elements in the V matrix and thus reduces computational time. In our experiment, forcing 100 ms $\leq t_j - t_i \leq$ 250 ms and relative likelihood $\geqslant$ 0.5 made our ASR return results in short time without losing performance. Further, since each element of the V matrix is independent of the other, the matrix calculation can be parallelised to (linearly) reduce decoding time (Section 4). The main difference between the exhaustive Viterbi algorithm and Token Passing is that the former treats acoustic models as black boxes. Indeed, exhaustive Viterbi is independent of the acoustic model implementation used and only requires likelihood calculations from these models. Instead, Token Passing is strongly based on the assumption that acoustic models are made up of sequences of states and that the transition from one model to another can occur only from a final state to an initial state. This assumption strongly reduces the computational complexity of the decoding strategy. In particular, Token Passing defines the *minimum alignment cost*, between the vectors X_1^t and a sequence of model states ending in state j, as $s_j(t) = min_{i \in all\ states}\{s_i(t-1) + p_{ij}\} + d_{ij}$. With p_{ij} being a transition cost that is given either by the model state-transition matrix or by the language model (when j is an initial state and $\{i\}$ are the final states of other models). The optimal sequence of states is the one having the minimum cost $S = min_j\{s_j(T)\}$. Using a bi-gram language model, the computational complexity of the unit-to-signal alignment is between $O(TNlog(N)C_l)$ and $O(TN^2C_l)$, where N is the number of connected units (e.g. syllables) and C_l is the complexity of the likelihood calculation of the state-based model used. Instead, the models managed by exhaustive Viterbi can be non-state-based, which is the main reason for its higher computational complexity but also for its higher flexibility.

2.4. Hidden Markov models

Hidden Markov Models (HMMs) are the most used choice for acoustic modelling (Fig. 1-a). Given a sequence of acoustic features X, they estimate the conditional probability distribution $P(X|S)$ of X given a sequence of states $S = s_1, s_2, \dots, s_T$. Based on this definition, the Viterbi algorithm [104] efficiently estimates the likelihood of an HMM to X as the conditional probability $P(X|S^*)$ of the sequence of states S^* that maximises $P(X|S)$ (i.e. the one most likely associated to X). An HMM that models a phoneme is trained (e.g. through the Baum–Welch algorithm) on many examples of acoustic features for that phoneme, in order to model the inter-speaker and inter-word variability of that phoneme. Likewise, a syllabic HMM is trained on the acoustic features of a syllable (Fig. 2-a), i.e. concatenations of phonetic-scale features. Often, ASRs use concatenations of phonetic HMMs to build up di-phone or tri-phone models that are re-trained to better assess inter-model transitions [12]. Modern ASRs need thousands of hours of annotated material to train phonetic acoustic models for large-vocabulary applications. For the experiment reported in this paper, HMM implementations from JAHMM [105], KALDI [31], and CMUSphinx [106] were used to implement syllabic and tri-phonetic HMMs.

For decades, ASRs have used HMMs with Gaussian mixtures (GMMs) to model emission densities [18]. However, current state-of-the-art ASRs use DNNs to model emission densities (Fig. 1-b) as $softmax(o_i(X_t))$, where $o_i(X_t)$ is the value of the activation function in the output layer of the node corresponding to state i [107]. This type of ASR is currently used in many domains and reaches state-of-the-art performance [52,108–114]. The used DNNs are typically multi-layer perceptrons with many layers (~7), with the training phase initialised by a pre-training algorithm. KALDI provides two main implementations of HMM-DNNs, one using Restricted Boltzmann Machines for pre-training and Stochastic Gradient Descent for training (HMM-DNN-nnet1), and the other one using Natural Gradient for Stochastic Gradient Descent and Parameter Averaging (HMM-DNN-nnet2).

2.5. Factorial hidden Markov models

An FHMM is made up of a set of HMMs, all with the same number of states, and inter-dependent emission probabilities usually modelled as multi-variate Gaussians (Fig. 2-b) [24]. FHMMs are particularly suited for speech processing, in particular to model concurrent and overlapping dynamics that are generated by multiple and loosely-coupled processes, as those present in a speech signal [23,115,116]. Multi-temporal ASRs have used this property to model the syllabic and phonetic structures contained in ~200 ms speech segments. In particular, the transition probability distributions of syllabic FHMM acoustic models with 2 parallel HMMs have highlighted the presence of two inter-linked syllabic-scale and phonetic-scale dynamics [80]. These are likely responsible for the higher performance of FHMMs with respect to HMMs in syllable modelling. FHMMs have been used in ASRs with the aim to include results from psycho-acoustic studies on overlapping speech dynamics [24,117,118]. For the experiment reported in this paper, FHMMs were implemented in Java by porting and optimising the original Matlab implementation by Ghahramani [119].

2.6. Deep learning models

Deep Learning (DL) models leverage a multi-layered structure to extract information from raw input data. DNNs are conventional DL models where each layer of the network is assumed to produce an internal representation of the input (*feature map*), with deeper layers producing higher levels of information abstraction. The increasing computational power of graphic processing units (GPUs) has allowed introducing DL models in a vast number of domains, e.g. from computer vision [120–122] to natural language processing [123,124]. In our experimental campaign, two different DL models were implemented – with PyTorch [125] – as acoustic syllabic models (without embedding them in an HMM): a Convolutional Neural Network (CNN, [126]) and a Long Short-Term Memory model (LSTM, [127]).

2.6.1. Convolutional neural network

DNNs (which include CNNs) process signal segments in a "static" way, i.e. like they were images [128,129]. Normally, they do not model time as an internal parameter and this limitation negatively affects their performance in automatic speech recognition with respect to other time-explicit models. In order to account for this issue, a CNN was built to model pseudo-syllables using a multi-temporal analysis within a convolutional stage (Fig. 2-c). This model uses the following operations: Each unit of the convolutional layer is computed by means of multiplications between the input data and a matrix (*kernel*), whose optimal values and size were assessed during the training phase. The kernel size corresponds to the size of the input that is convolved with the kernel (*receptive field*) so that each convolution only looks at a small portion of the input. Through the use of small receptive fields, convolutional layers are generally able to extract and combine *local* information from the input data, i.e. information contained in segments (of speech signal, in our case) with a predefined length. Our CNN used four 1D convolutional layers – each with a different kernel size – that corresponded to different filters and windows on the syllabic signal. The size of each window represents the time scale processed by each convolutional operation. During the convolution, a window stride of one sample maximises the capture of local relations through the signal. In summary, our CNN analyses a syllabic speech signal at multiple time scales through a multi-window processing. After the convolutional step, vector pooling and stacking operations are followed by a flattening operation that projected all the resulting feature vectors (feature maps) on a new 1D vector, whose optimal length was estimated during the training phase. This vector is input to a fully-connected (FC) neural network layer, whose optimal size was estimated during the training phase. A rectified linear unit (ReLU) activation function ($max(0,x)$) is applied to each node of this layer to reduce the vanishing gradient problem [38,130] and favour generalisation capability [131,132]. In order to reduce the risk of data overfitting due to the small amount of training data available, the dropout technique [133] was used on the FC layer. Dropout statistically excludes some nodes of the FC layer from one training session and re-introduces these nodes with their original weights after the non-dropped nodes connections have been trained. At each training step, a new set of nodes is selected to be dropped. Finally, during the inference phase, each node's output is multiplied by a dropout probability to account for their possibly missed training steps. Overall, this procedure simulates an ensemble of a high number of different models whose output is eventually averaged at inference time. The last stage of our CNN is a classification layer, i.e. another FC layer with 44 neurons, one for each unit to recognise. This layer allows classifying the acoustic features of a syllabic signal as one among the syllabic units reported in Table 1. Indeed, a softmax function applied to the layer's outputs makes the CNN overall simulate a posterior probability density $P(W|X)$ of each syllable W given the input vector X [134]. In turn, this reduces the complexity of the decoding algorithm, because the probabilities of all syllables for a signal segment are calculated just after one propagation of the input through the network. The described architecture came after testing a large number of alternative architectures, including chained windows and deeper networks. It was the architecture using the lowest number of parameters and gaining the highest performance on the tasks reported in this paper.

2.7. *Long short-term memory model*

LSTMs are naturally suited to process observation sequences and time series [127], because they consist of one computational unit that is iteratively used to process the observations of an input time series (processing *steps*). The unit uses *gating* mechanisms that process a temporal flow of data while controlling the retain and the release of memorised information. Since an LSTM is suited to simulate a posterior probability density $P(W|X)$ of all acoustic units W given the input features series X, it cannot directly replace an HMM (which calculates likelihood). Furthermore, the processing steps cannot be treated as the sequence of states of an acoustic HMM and thus cannot be used in classic speech decoding processes.

In this paper, an LSTM model was implemented to classify pseudo-syllabic acoustic units directly and was later combined with a speech decoding algorithm, which was able to harness its multi-temporal processing of the speech signal. Our LSTM model's unit processes one vector of the input time series (i.e. one window of acoustic features) at a time. It uses a standard unit characterised by one *forget* gate, one *input* gate, and one *output* gate (Fig. 2-d), all implemented as single-layered neural networks. Within the unit, the cell state c_t is a Real-valued vector that roughly stores the "long-term" memory of the model, whereas the hidden state h_t is the output vector of the LSTM unit that manages "short-term" memory. At each processing step, the LSTM unit receives the current input vector of acoustic features, and the cell and hidden states of the previous unit. The unit outputs a new cell state and a new hidden state. All gates receive the current unit input vector and the previous hidden state as input. As a first operation, the cell state of the previous processing step is multiplied by the output of the *forget* gate, i.e. a neural network with sigmoid activation function with range [0,1], where 0 represents a complete blockading (forget) of an input element and 1 a complete pass (remember). Another process point-wise multiplies the output of a sigmoid-activated neural network (input gate) by the output (*proposed cell state*) of a tanh-activated neural network. The output of this process is summed to the output of the forget gate in order to establish which part of the information retained by the forget gate should be updated. This result is the unit's cell state that is passed to the next LSTM processing step. The hidden state is calculated by first passing the cell state to a tanh function (to re-scale its values in [−1,1]) and then multiplying this result with the output of another sigmoid-activated neural network (output gate). Overall, this final step roughly decides what portion of the cell state is produced as the output of the LSTM unit. As a final step, our LSTM-based syllabic acoustic model uses the last processing step's output as an input to a classification layer, whose input size is equal to the hidden state size. Similarly to the CNN model, a softmax function is applied to the output of this classification layer in order to simulate the posterior probability density $P(W|X)$ of each syllable W given the input vector X, and to reduce the decoding algorithm complexity.

2.8. *Baseline speech recognisers*

Several instances of our ASR were produced to assess its performance depending on the acoustic model used. Each instance used one among the four supported models. It is worth noting that some of these models are more suited for recognising entire syllables, but their performance could be positively or negatively affected by the combination with the decoding algorithm (Section 3.3). In particular, syllabic HMMs were enabled in our ASR architecture to measure the performance enhancement that our decoding algorithm would bring to a classic model. Similarly, FHHMs were used to evaluate the performance in word and sentence recognition of a naturally suited model for single pseudo-syllable recognition [102]. Finally, the CNN and LSTM acoustic models were used to evaluate the performance gained by our psycho-acoustic inspired models in word and sentence recognition.

CMUSphinx [106] and KALDI [31] were used as reference ASRs. These systems use tri-phonetic HMM-GMMs and HMM-DNNs respectively within a reference ASR architecture, and were trained with an open source reference corpus suited for our recognition tasks. The Italian VoxForge corpus [135] was used to train phonetic and tri-phonetic HMMs with 19 h of speech that involved regional inflexions. Although the dimension of VoxForge is generally not sufficient to build a high-performance large-vocabulary ASR, it was sufficient to build high-performance baseline ASRs for spoken digits and numbers. In particular, CMUSphinx and KALDI were trained through the following operations: (i) Pronunciation models for words uttered in the VoxForge recordings were taken from the large database of Cosi [33]; (ii) phonetic acoustic models (and tri-phones) from the pronunciation models were aligned to the recordings through automatic alignment processes; (iii) the language models described in Section 2.2 were integrated to produce two recognisers, one for digit recognition and another one for number recognition. Furthermore, in the single-syllable recognition task (Section 3.2) the phonetic transcriptions from [33] were used to model the pronunciations of the 42 syllables of Table 1.

As a second baseline ASR, the Google Speech-to-Text cloud service was used [19]. This HMM-DNN based ASR, trained with thousands of hours of speech, has top-level performance and high response efficiency, and is used by almost all Google technology. Google constantly improves its performance after periodically collecting users' data and revising acoustic, pronunciation, and language models. This ASR uses phonetic transcriptions of 10 times the words of an entire language dictionary, for each of the 120 languages supported, and also includes a context-specific adaptation process that is able to resize the grammar and to optimise the transcription according the language context [19,136,137]. For example, on number recognition tasks the Google service is able to report the numerical form of the uttered number (e.g. "one hundred three" is reported as 103), while insertions, false starts, and other non-numerical words are deleted if not uttered clearly [19]. Google Speech-to-Text (Dec. 2019 version) was used as a reference state-of-the-art ASR. Indeed, comparing the performance of our method with that of the Google ASR may not be an optimal choice, because of the different training corpora used (voices, data size, data preparation, etc.) and the lack of details about the Google ASR's architecture. Nevertheless, the Google's context-specific adaptation feature and the relatively small testing context (numbers) reasonably allow to use the comparison as a proxy for a quality assessment of our ASR.

3. Results

This section reports a performance comparison between the models described in the previous section. Accuracy is used as a comparison metric, defined as

$$Accuracy = \frac{number\ of\ correctly\ recognised\ units - number\ of\ over - inserted\ units}{total\ number\ of\ units\ in\ the\ manual\ transcription}.$$

In order to make comparisons consistent, the interpretation of "unit" in the accuracy formula changed according to the test case. In fact, the compared ASRs had heterogeneous architectures and used different speech units and output types. For example, the Google ASR reported the entire recognised sentence with numerical symbols (e.g. "1" for a digit and "1350" for a number). Furthermore, the other ASRs used either tri-phones or pseudo-syllables. In this context, a comparison could be consistent only at a final orthographic transcription level. Thus, in the single-syllable recognition task, accuracy was calculated on the number of correctly transcribed orthographic syllables. On digit and number recognition tasks, accuracy was calculated on the orthographic transcription of the entire sentence.

3.1. Acoustic model topologies

The machine-learning models reported in Section 2, were trained to recognise the 44 units reported in Table 1. Multiple parametrisations and implementations of the models were tested. Eventually, the topologies and implementations gaining the highest performance were selected for the comparison. This operation required testing thousands of parameter combinations.

Optimal HMMs for GMM-based tri-phone models were produced with KALDI and used 5 states and 32 mixtures, whereas HMM-DNN phonetic models used 7 hidden layers in the DNN. Out of these HMMs, syllables were represented as concatenations of phonetic HMMs. Optimal syllabic HMMs were produced with JAHMM and had 7 states and 39 Gaussian mixtures. Optimal FHMMs used 2 HMMs with 7 states each and one multivariate Gaussian emission density.

Regarding the deep learning models, cross-validation was used to find optimal parameters and topologies of the CNN and the LSTM acoustic models. Specifically, the optimal CNN topology used windows of 48, 80, 96 and 112 ms to create 64-length feature maps after convolution, and was made up of two FC layers (one hidden layer and one output layer). The feature map was optimally flattened to 1280 elements, the dropout probability was 20%, and the optimal size of the first FC layer was 300. The final FC classification layer had 44 neurons, one for each syllable to recognise. Moreover, the importance of introducing non-linearity in the output of this FC layer through ReLU was tested: All models were also trained without ReLU, and a performance degradation up to 4% was observed, which confirms the positive contribution of this transformation and the need to include it in the acoustic model.

The optimal LSTM was a mono-directional model with one hidden state with 1000 neurons and a final classification layer with 44 neurons, one for each syllable to recognise. The training phases of both models used the Adam optimiser [138] with cross entropy loss criterion and a learning rate of $1.e^{-3}$, reduced of 5 times whenever loss reached a plateau.

As for ASR configuration, the language model of CMUSphinx was trained with mono-grams, bi-grams, and tri-grams. This ASR used HMM phonetic models with 32 Gaussian mixtures, trained with 19 h of annotated recordings from the VoxForge corpus. Our ASR used the exhaustive Viterbi algorithm described in Section 2.3 alternatively combined with HMM, FMM, CNN, and LSTM syllabic acoustic models. The ASR used a language model based on the bi-grams prepared for CMUSphinx. Finally, the Google Speech-to-Text service was used through a Java client that streamed audio files and collected transcriptions.

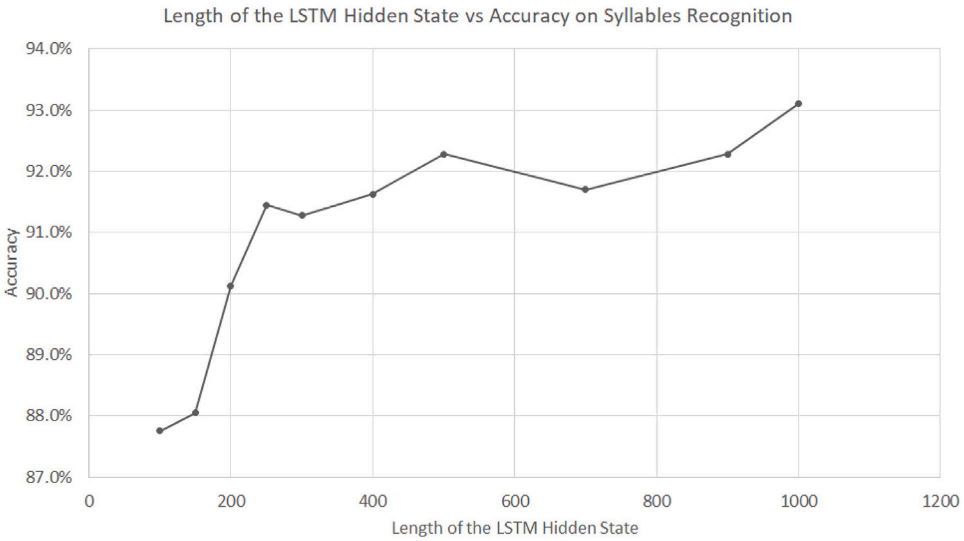

Fig. 3. Variation of the accuracy of our LSTM model on the recognition of syllables of numbers between 0 and 999,999, with respect to the LSTM hidden-state length.

3.2. Syllable recognition

Acoustic models' performance was first compared on the recognition of the syllables and silence units of Table 1, without the interference of the language model and the decoding process. This performance comparison (Table 2-a) showed that our LSTM outperforms phonetic HMMs by 8.91% absolute accuracy and the second optimal model (HMM-DNN-nnet2) by 2.63%. A Chi-squared test confirmed that this discrepancy was highly significant with our test set size (with p-value of non-significant discrepancy null hypothesis lower than 0.0001) [139]. The accuracy of our LSTM increased non-linearly with the vector length of the LSTM hidden state (Fig. 3), which indicated that a ~1000 length was really required to model the complexity and variability of the syllables. Interestingly, our multi-temporal CNN had comparable performance with HMM-DNN models, and the HMM-DNN model using Natural Gradient had a slightly higher performance than the other HMM-DNN implementation. FHMMs and syllabic HMMs outperformed phonetic HMMs, in agreement with other studies [24,102,115], but had lower performance than the deep learning models. Since HMM-DNN-nnet2 was the second optimal model, it was selected to be used in the KALDI ASR for the comparison on digit and number recognition tasks.

3.3. Digit recognition

Although digits are made up of a maximum of two non-silence syllables, spoken digit recognition involves the issue of aligning sequences of silence models, short pauses, and (one or two) syllables to the signal. Thus, a performance comparison on digit recognition highlighted how much the slight misalignment of syllabic models to the uttered syllables influenced word recognition. In this case, accuracy was calculated on the recognition of entire words directly, especially for a fair comparison with the Google Speech-to-Text service. The context-specific adaptation of the Google ASR made the reported comparison meaningful because it restricted the grammar to the particular task and deletes non-numeric insertions that were not loudly uttered.

Also, due to the moderately-high signal-to-noise ratio, no model reached 100% performance on this "simple" task (Table 2-b). The exhaustive Viterbi algorithm optimised the alignment of the various acoustic models to the speech signal and made syllabic HMMs and FHMMs outperform a standard-approach ASR. Performance was generally high for all ASRs, but the accuracy discrepancy between the CNN and the LSTM models (4%) was higher than in the syllable recognition case (2.76%). Indeed, the CNN model was sensitive to syllable alterations (e.g. stretching and reduction) since it processed signal segments as they were static images. Generally, the difference between the CNN and the LSTM models in accounting for syllabic alterations is more and more evident as long as speech tends to be continuous and spontaneous. For example, alterations of "kwa ttro" as "kwa tro" and of "o tto" as "o to" are more probable within long numbers but also exist with digits. Overall, the high performance of the LSTM-based ASR indicated that our LSTM was a suitable acoustic model for an ASR, and the recogniser also slightly outperformed the Google service (98% vs 97.5% accuracy). A Chi-squared test confirmed that this discrepancy was significant (with p-value of non-significant discrepancy null hypothesis lower than 0.05). Finally, the KALDI ASR using the best tri-phone models (HMM-DNN-nnet2) gained 1.1% higher relative accuracy than the CNN-based model, but lower relative accuracy than the Google ASR (2.5%) and the LSTM-based ASR (3%).

Table 2
Performance comparison between alternative speech recognition models on the recognition of (a) the 44 units involved in our corpus of data, (b) spoken numbers from 0 to 9 (digits), (c) spoken numbers between 0 and 999,999.

Model name	Accuracy (%)
a - Syllable recognition	
LSTM	93.01
HMM-DNN-nnet2	90.38
CNN	90.25
HMM-DNN-nnet1	89.91
FHMMs	86.53
Syllabic HMMs	85.79
Phonetic HMMs	84.10
b - Digit recognition	
LSTM + Exhaustive Viterbi	98.00
Google Speech-to-Text	97.50
KALDI - HMM-DNN-nnet2	95.06
CNN + Exhaustive Viterbi	94.00
FHMMs + Exhaustive Viterbi	93.30
Syllabic HMMs + Exhaustive Viterbi	92.00
CMUSphinx	87.74
c - Number recognition	
LSTM + Exhaustive Viterbi	85.00
Google Speech-to-Text	81.81
KALDI - HMM-DNN-nnet2	81.20
CMUSphinx	79.00
CNN + Exhaustive Viterbi	76.60
FHMMs + Exhaustive Viterbi	72.00
Syllabic HMMs + Exhaustive Viterbi	70.00

3.4. Number recognition

The performance comparison on the recognition of [0,1M) numbers further highlighted the differences between the ASRs (Table 2-c). The main difference with respect to the digit recognition case was the higher performance of CMUSphinx with respect to the CNN-, FHMM-, and syllabic HMM-ASRs. This enhancement was due to the higher amount of training material used to build the CMUSphinx ASR, and also to the lower flexibility of the other models to work on the more continuous and spontaneous speech of the uttered numbers, which presents a large variability due to omissions, false starts, and uncertain speech. In this context, the acoustic syllabic structures can be very different from those of the training set. Models like CNN, FHMMs, and Syllabic HMMs would require more training material to handle this structural variability. In particular, our CNN model was re-adapted from image processing and does not fully capture the unfolding of information in time and its variability across the training set. Instead, the phonetic CMUSphinx and the KALDI ASRs had comparable performance with Google (3.4% and 0.7% relative accuracy, respectively) due to a training material suited for the task. In particular, KALDI demonstrated the high quality and performance that HMM-DNN-based ASRs can reach.

Interestingly, when our ASR used the LSTM acoustic model, it outperformed all other ASRs. Indeed, our ASR had a 3.7% higher relative accuracy than the Google ASR and a 7% higher accuracy than CMUSphinx. A Chi-squared test confirmed that these discrepancies were significant (with p-value of non-significant discrepancy null hypothesis lower than 0.001). The generalisation capability of the LSTM-based ASR and its flexibility to account for syllable alterations was impressive. The LSTM had optimal performance in all presented cases, and the acoustic models used only ~1 h of training material, which was much lower than the 19 h used for CMUSphinx and KALDI and the thousands of Google.

3.5. Issues with large-vocabulary speech recognisers

A one-million-number sentence-set was used instead of a large vocabulary because building a large vocabulary speech recogniser (LVSR) requires solving other additional research questions that were out of our scope. Generally, it is nearly impossible to build a state-of-the-art LVSR for a low-resource language like Italian using publicly available corpora for acoustic and language model training. To better highlight this aspect (and also produce a reference for our future studies), we trained and compared several LVSRs – based on KALDI and CMUSphinx – using alternative open (or low-cost) textual and audio corpora (Table 3). The aim of this comparison was principally to highlight some intrinsic practical difficulties in building LVSRs.

We compared recognition performance on a 15-minute corpus extracted from the Italian VoxForge corpus [135] that was not used during ASR training. The textual corpora used for (4-gram) language model training included: (i) the "Italian Web corpus" (itWaC), made up of texts collected from the Internet and including 1.5 billion words [140]; (ii) Paisà, a large and expert-revised collection of Italian texts from the Internet containing ~250 million tokens [141]; (iii) CLEF, a large collection of Italian national newspaper articles from the 90's containing ~1 million words overall [142]; and (iii) the Italian Content Annotation Bank (I-CAB), which

Table 3
Performance comparison between several large-vocabulary automatic speech recognisers at the variation of the corpora used for language model (LM) and acoustic model (AM) training: the Google speech-to-text service (Google ASR), KALDI with deep neural network emission densities used in acoustic models (KALDI - HMM-DNN-nnet2), and the Gaussian-mixture based CMUSphinx.

ASR engine	Corpus for LM	Corpus for AM	Word accuracy (%)
Google ASR	Google	Google	89.10
KALDI - HMM-DNN-nnet2	Paisà	VoxForge	63.64
KALDI - HMM-DNN-nnet2	Paisà	VoxForge+APASCI	67.00
CMUSphinx	Paisà	VoxForge	51.58
CMUSphinx	Paisà	VoxForge+APASCI	54.41
CMUSphinx	Paisà + I-CAB	VoxForge	49.70
CMUSphinx	Paisà + CLEF + I-CAB	VoxForge	49.90
CMUSphinx	Paisà + CLEF	VoxForge	49.90
CMUSphinx	CLEF	VoxForge	42.60
CMUSphinx	CLEF + I-CAB	VoxForge	42.00
CMUSphinx	itWaC	VoxForge	44.00
CMUSphinx	I-CAB	VoxForge	34.40
KALDI - HMM-DNN-nnet2	Paisà	APASCI	57.95
CMUSphinx	Paisà	APASCI	39.35

contains 525 local (Trento province) newspaper articles with ~180,000 words overall [143]. The audio corpora used were VoxForge (~20 h) and APASCI (~2 h), whose audio was based on the same spoken text. The performance across multiple textual corpora was reported only for CMUSphinx for simplicity to highlight performance decrease across the corpora. The following difficulties emerged, which depended on the lack of great effort (and money) investment in data collection and cleaning:

1. The performance gap between the Google ASR and the other LVSRs was very high (from −25.46% to −49.75% word accuracy);
2. Using uncontrolled large textual corpora (e.g. itWaC) may end in lower performance because text from social networks introduces too much noise in the language model and is unsuited for spoken dialogues;
3. Combining different textual corpora (e.g. Paisà+CLEF) may end in lower performance because of too different language structures (e.g. Internet v.s. newspapers);
4. Data cleaning included in Paisà made this corpus the optimal choice to train the language model, but required greater effort by the corpus producers;
5. Generally, using many hours of speech (i.e. VoxForge instead of APASCI) and deep learning models for training acoustic models increases performance, but combining different audio corpora can decrease performance probably because of practical audio-transcription inconsistencies between the corpora;
6. Smaller vocabularies (e.g. CLEF and I-CAB) – even containing thousands of words – may not be sufficient to gain high performance.

Thus, selecting and preparing optimal textual and speech corpora for LVSRs is complex and effort-demanding, especially for low-resource languages. Investigating these issues was outside of this paper's scope, which instead aims at introducing new acoustic models and a new decoding algorithm and comparing them with a state-of-the-art ASR on a common vocabulary. However, our future experiments will investigate the above issues because they call for new ways to achieve state-of-the-art performance with less training material and new decoding strategies that optimally use the language model.

In summary, a one-million-number benchmark sentence-set was used because it corresponded to a non-trivial language model that did not depend on the used training textual corpus and was reasonably comparable with the one used by a reference state-of-the-art ASR. Furthermore, although the lexicon required a short preparation phase, the speech included several features of spontaneous and large-vocabulary speech (e.g. variability, omissions, uncertain speech, and false starts).

4. Discussion and conclusions

4.1. Summary

In this paper, novel syllabic acoustic models and a new ASR have been described and compared with state-of-the-art alternatives. On the single-syllable recognition task, the deep-learning models showed very high performance. FHMMs gained higher performance than syllabic and phonetic HMMs, likely because they recognised both syllabic- and phonetic-scale dynamics associated with different transition speeds in the two parallel HMMs [102]. Also, our multi-temporal CNN model forced a multi-scale analysis (from phonetic to syllabic scales) and gained high performance. However, this model was not able to fully capture the information contained in feature modulations and transitions in the digit and number recognition tasks. Thus, its performance decreased when the acoustic structures of the modelled syllables were not preserved. The properly trained CMUSphinx and KALDI ASRs reached a very high performance on the [0,1M) number recognition task, but still lower than the cutting edge technology of the Google ASR.

In the presented experiment, our LSTM acoustic model reported the highest performance both when used alone and when combined with a decoding algorithm that optimised its prediction capability. In particular, on syllable recognition - i.e. without the presence of the exhaustive Viterbi decoder – the LSTM model outperformed the other models. The performance remained optimal also on digit and number recognition, i.e. when the LSTM was combined with the exhaustive decoding algorithm. This was not the case of the CNN and the other acoustic models, which lost accuracy with respect to the baseline systems as the recognition task became more and more difficult. Thus, the LSTM model both outperformed the other acoustic models and was optimally used by the decoding algorithm. This property indicates that the decoding algorithm was able to use this model at best, although the LSTM performance was already optimal by itself. The LSTM model explicitly accounts for the unfolding of information in time, similarly to HMMs, but also models both long- and short-term information. This likely corresponds to modelling high-rate and low-rate dynamics within one syllable, in agreement with psycho-acoustic studies. At the same time, this behaviour also overcomes the issue of modelling inter-syllabic variability from a small training set, which affected the CNN model's performance. Our results indicate that the LSTM probably learned this variability from the training set and thus was able to manage a more continuous-like speech. Finally, the separation between the speakers in the training and test sets reduced the potential artefact that the model was trained on the same corpus the test set belonged to.

4.2. Using our approach with larger vocabularies and other languages

Extending our approach to an LVSR principally requires the availability of annotations of the syllables actually spoken in the utterances (pseudo-syllables) that allow to develop an effective syllabic language model. The use of the Speecon Italian corpus in the presented experiment was mainly driven by the availability of this information. Given the generality of our approach, the presented results are likely valid for all languages currently managed by the reference ASRs of Fig. 1, as long as pseudo-syllables are used for acoustic modelling. Indeed, pseudo-syllables ensure the stability of the syllable structure and increase acoustic-model performance (Section 2.1.2). Nevertheless, our future work will test the new proposed ASR on other languages. It is worth noting that the obtained results are compliant with those reported by other studies for English. The Google ASR and HMM-DNN-based ASRs can reach over 99% accuracy on clean English spoken digits [4] and ~97% accuracy on noisy speech (with a ~15 dB signal-to-noise ratio) [144]. On a task to recognise ~30,000 English numbers [145], performance can range around a 93% word-recognition accuracy [78,146,147]. As a general reference, with clean dialogue speech the Google ASR can reach ~93% word-recognition accuracy on a ~5000 vocabulary of English words [52], and ~63% word-recognition accuracy on a 7.5 million vocabulary of English words [148]. On the same million-word vocabulary, an ASR based on KALDI and HMM-DNN acoustic models can reach ~63% word-recognition accuracy but is much more sensible to audio noise than the Google ASR [148].

Differently from the *soft alignment* process used in end-to-end models [149], our ASR aligns acoustic models to the signal exhaustively and explicitly, and can re-use statistical language models of standard ASRs. Overall, with respect to end-to-end models, our ASR presents a clear separation – as modules – between the phases of feature extraction, language and acoustic modelling, and decoding. This property allows improving the ASR by substituting alternative processes to these modules. One drawback of our ASR is its high computational complexity that mainly depends on the pre-calculation of a large number of likelihoods (Section 2.3). However, this complexity does not compromise efficiency for practical usages of the ASR: Using a parallel implementation of the decoding algorithm on 8 cores with HMM syllabic models, the recognition of a spoken number requires averagely ~5 s on a machine with an Intel Core i7-7700HQ CPU and 16 GB of Random Access Memory. Parallelising the computation on multiple cores or machines would make computational time manageable also if a large number of syllables were involved, e.g. a large vocabulary (~500 syllables) would require ~5 s on a distributed computation using ~100 cores/machines on a cloud computing platform. Furthermore, the search space of the decoding algorithm could be drastically reduced through "islands' recognition", i.e. by focusing the process on those portions of the speech signal that are (i) acoustically relevant (prominent) compared to the surrounding units, (ii) pronounced with reasonable accuracy, and (iii) more clearly recognisable [150]. We will explore also this research direction in the future.

4.3. Concluding remarks

In summary, a completely new ASR architecture has been presented. Our approach's main novel characteristics are the inspiration by studies on the multi-temporal processing of speech in human beings and the use of pseudo-syllables instead of tri-phones as acoustic models. These features have produced high-quality results compared to the Google ASR and the KALDI tri-phonic HMM-DNN ASR. Furthermore, our approach has the technical advantage that it can be applied to new acoustic models and can re-use statistical language models of classic ASRs. The reported results suggest that taking into account the results of psycho-acoustic studies - i.e. including also non-phonetic dynamics — and questioning the standard-used ASR approaches may produce effective solutions. This observation positively answers to our original research question. Furthermore, our results show that the high performance of our ASR and acoustic models is likely due to the use of pseudo-syllables instead of tri-phones on the reported recognition tasks, i.e. a perceptual syllable definition has direct benefits for both the acoustic models and the ASR. One open question is if the multi-temporal processing included in our acoustic models was crucial to increase ASR performance. Indeed, our CNN explicitly modelled multi-temporal processing but did not gain top performance. In contrast, the LSTM model implicitly accounted for different time-scale dynamics and gained very high performance.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgements

The authors want to thank Daniela Burba for proofreading and correcting the paper. Coro, Origlia and Cutugno want to thank and remember Prof. Renata Savy for having inspired the foundations of this research.

Software

Our ASR is Open Source to allow for comparisons and verification. The source codes of the decoding algorithm and of the HMM-based syllabic acoustic models, and the DL trained models are available on the GitHub at

https://github.com/gianpaolocoro/AutomaticSpeechRecognitionResearch

The source code for training the deep learning acoustic models is available at

https://github.com/fvmassoli/deep-acoustic-modeling

References

[1] Hawkins S, Smith R. Polysp: A polysystemic, phonetically-rich approach to speech understanding. Ital J Linguist 2001;13:99–188.

[2] Pieraccini R. The voice in the machine: Building computers that understand speech. MIT Press; 2012.

[3] Markowitz JA. Robots that talk and listen: Technology and social impact. de Gruyter Berlin; 2015.

[4] Li J, Deng L, Haeb-Umbach R, Gong Y. Robust automatic speech recognition: A bridge to practical applications. Academic Press; 2015.

[5] Mustafa MK, Allen T, Appiah K. A comparative review of dynamic neural networks and hidden Markov model methods for mobile on-device speech recognition. Neural Comput Appl 2019;31(2):891–9.

[6] CBInsights. How big tech is battling to own the $ 49B voice market. 2019, https://www.cbinsights.com/research/facebook-amazon-microsoft-google-apple-voice/.

[7] Szaszák G, Tündik MÁ, Beke A. Summarization of spontaneous speech using automatic speech recognition and a speech prosody based tokenizer. In: 8th international conference on knowledge discovery and information retrieval (KDIR 2016), Porto, Portugal. 2016, p. 221–7.

[8] Sahu P, Dua M, Kumar A. Challenges and issues in adopting speech recognition. In: Speech and language processing for human-machine communications. Springer; 2018, p. 209–15.

[9] Naing SHM, Pa Pa W. Automatic speech recognition on spontaneous interview speech. In: 16th international conference on computer applications 2018 (ICCA 2018), Yangon, Myanmar. 2018, p. 1–5.

[10] Knill KM, Gales MJF, Manakul PP, Caines AP. Automatic grammatical error detection of non-native spoken learner english. In: ICASSP 2019 - 2019 IEEE international conference on acoustics, speech and signal processing (ICASSP). 2019, p. 8127–31.

[11] Padrell-Sendra J, Martín-Iglesias D, Diaz-de Maria F. Support vector machines for continuous speech recognition. In: 2006 14th European signal processing conference. IEEE; 2006, p. 1–4.

[12] CMUSphinx. Training an acoustic model for CMUSphinx. 2017, https://cmusphinx.github.io/wiki/tutorialam/.

[13] Mwiti D. A 2019 guide for automatic speech recognition. 2019, https://heartbeat.fritz.ai/a-2019-guide-for-automatic-speech-recognition-f1e1129a141c.

[14] Greenberg S. Understanding speech understanding: Towards a unified theory of speech perception. In: Proceedings of the ESCA tutorial and advanced research workshop on the auditory basis of speech perception. Keele, England; 1996, p. 1–8.

[15] Ostendorf M, Digalakis VV, Kimball OA. From HMM's to segment models: A unified view of stochastic modeling for speech recognition. IEEE Trans Speech Audio Process 1996;4(5):360–78.

[16] Cutugno F, Origlia A, Schettino V. 7 syllable structure, automatic syllabification and reduction phenomena. In: Rethinking reduction: Interdisciplinary perspectives on conditions, mechanisms, and domains for phonetic variation, Vol. 25. Walter de Gruyter GmbH & Co KG; 2018, p. 205.

[17] Dunning T. Statistical identification of language. Computing Research Laboratory, New Mexico State University Las Cruces, NM, USA; 1994.

[18] Huang X, Acero A, Hon H-W, Foreword By-Reddy R. Spoken language processing: A guide to theory, algorithm, and system development. Prentice hall PTR; 2001.

[19] Google. Cloud speech-to-text features description. 2019, https://cloud.google.com/speech-to-text/.

[20] Markov AA. An example of statistical investigation in the text of 'Eugene Onyegin' illustrating coupling of tests in chains. In: Proc. of the Academy of Sciences of St. Petersburg, Russia. 1913, p. 153–62.

[21] Rabiner LR, Juang B. A tutorial on hidden markov models. IEEE ASSP Mag 1986;3(1):4–16.

[22] Young SJ, Russell N, Thornton J. Token passing: A simple conceptual model for connected speech recognition systems. Cambridge University Engineering Department Cambridge; 1989.

[23] Ghahramani Z, Jordan MI. Factorial hidden Markov models. In: Advances in neural information processing systems. 1996, p. 472–8.

[24] Logan B, Moreno P. Factorial HMMs for acoustic modeling. In: Proceedings of the 1998 IEEE international conference on acoustics, speech and signal processing, ICASSP'98 (Cat. No. 98CH36181), Vol. 2. IEEE; 1998, p. 813–6.

[25] Cosi P. Auditory modeling and neural networks. In: International summer school: Speech processing, recognition and artificial neural networks. Citeseer; 1998, p. 235–58, Paper Available from http://www.csrf.pd.cnr.it/Papers/PieroCosi/cp-IIASS98.pdf.

[26] Cosi P, Hosom J-P. HMM/Neural network-based system for Italian continuous digit recognition. In: Proceedings of the 14th international congress of phonetic sciences (ICPhS '99). Citeseer; 1999, p. 1669–72.

[27] Ahad A, Fayyaz A, Mehmood T. Speech recognition using multilayer perceptron. In: IEEE students conference, ISCON'02. Proceedings. Vol. 1. IEEE; 2002, p. 103–9.

[28] Abdel-Hamid O, Mohamed A-r, Jiang H, Deng L, Penn G, Yu D. Convolutional neural networks for speech recognition. IEEE/ACM Trans Audio Speech Lang Process 2014;22(10):1533–45.

[29] Hinton G, Deng L, Yu D, Dahl G, Mohamed A-r, Jaitly N, Senior A, Vanhoucke V, Nguyen P, Kingsbury B, et al. Deep neural networks for acoustic modeling in speech recognition. IEEE Signal Process Mag 2012;29.

[30] Swietojanski P, Ghoshal A, Renals S. Convolutional neural networks for distant speech recognition. IEEE Signal Process Lett 2014;21(9):1120–4.

[31] Povey D, Ghoshal A, Boulianne G, Burget L, Glembek O, Goel N, Hannemann M, Motlicek P, Qian Y, Schwarz P, et al. The kaldi speech recognition toolkit. In: IEEE 2011 workshop on automatic speech recognition and understanding. IEEE Signal Processing Society; 2011, p. 1–4, IEEE Catalog No.: CFP11SRW-USB.

[32] Pan J, Liu C, Wang Z, Hu Y, Jiang H. Investigation of deep neural networks (DNN) for large vocabulary continuous speech recognition: Why DNN surpasses GMMs in acoustic modeling. In: 2012 8th international symposium on Chinese spoken language processing. IEEE; 2012, p. 301–5.

[33] Cosi P. A KALDI-DNN-based asr system for Italian. In: 2015 international joint conference on neural networks (IJCNN). IEEE; 2015, p. 1–5.

[34] Sak H, Senior A, Beaufays F. Long short-term memory recurrent neural network architectures for large scale acoustic modeling. In: Fifteenth annual conference of the international speech communication association. 2014, p. 338–42.

[35] Soltau H, Liao H, Sak H. Neural speech recognizer: Acoustic-to-word LSTM model for large vocabulary speech recognition. 2016, arxiv preprint arXiv:1610.09975.

[36] Senior A, Sak H, Shafran I. Context dependent phone models for LSTM RNN acoustic modelling. In: 2015 IEEE international conference on acoustics, speech and signal processing (ICASSP). 2015, p. 4585–9. http://dx.doi.org/10.1109/ICASSP.2015.7178839.

[37] Qu Z, Haghani P, Weinstein E, Moreno P. Syllable-based acoustic modeling with CTC-SMBR-LSTM. In: 2017 IEEE automatic speech recognition and understanding workshop (ASRU). 2017, p. 173–7. http://dx.doi.org/10.1109/ASRU.2017.8268932.

[38] Bengio Y, Simard P, Frasconi P, et al. Learning long-term dependencies with gradient descent is difficult. IEEE Trans Neural Netw 1994;5(2):157–66.

[39] Hochreiter S. Untersuchungen zu dynamischen neuronalen Netzen, Vol. 91. Diploma, Technische Universität München; 1991.

[40] Massoli FV, Carrara F, Amato G, Falchi F. Detection of face recognition adversarial attacks. 2019, arxiv preprint arXiv:1912.02918.

[41] Rao K, Sak H, Prabhavalkar R. Exploring architectures, data and units for streaming end-to-end speech recognition with rnn-transducer. In: 2017 IEEE automatic speech recognition and understanding workshop (ASRU). IEEE; 2017, p. 193–9.

[42] Zhang Y, Chan W, Jaitly N. Very deep convolutional networks for end-to-end speech recognition. In: 2017 IEEE international conference on acoustics, speech and signal processing (ICASSP). IEEE; 2017, p. 4845–9.

[43] Chiu C-C, Sainath TN, Wu Y, Prabhavalkar R, Nguyen P, Chen Z, Kannan A, Weiss RJ, Rao K, Gonina E, et al. State-of-the-art speech recognition with sequence-to-sequence models. In: 2018 IEEE international conference on acoustics, speech and signal processing (ICASSP). IEEE; 2018, p. 4774–8.

[44] Weng C, Cui J, Wang G, Wang J, Yu C, Su D, Yu D. Improving attention based sequence-to-sequence models for end-to-end english conversational speech recognition. In: Interspeech. 2018, p. 761–5.

[45] Watanabe S, Hori T, Karita S, Hayashi T, Nishitoba J, Unno Y, Soplin NEY, Heymann J, Wiesner M, Chen N, et al. Espnet: End-to-end speech processing toolkit. 2018, arxiv preprint arXiv:1804.00015.

[46] Zeghidour N, Usunier N, Synnaeve G, Collobert R, Dupoux E. End-to-end speech recognition from the raw waveform. 2018, arxiv preprint arXiv:1806.07098.

[47] Zeghidour N, Xu Q, Liptchinsky V, Usunier N, Synnaeve G, Collobert R. Fully convolutional speech recognition. 2018, arxiv preprint arXiv:1812.06864.

[48] Jaitly N, Zhang Y, Chan W. Very deep convolutional neural networks for end-to-end speech recognition. 2019, US Patent 10, 510, 004.

[49] Sainath TN, Pang R, Rybach D, He Y, Prabhavalkar R, Li W, Visontai M, Liang Q, Strohman T, Wu Y, et al. Two-pass end-to-end speech recognition. 2019, arxiv preprint arXiv:1908.10992.

[50] Graves A, Fernández S, Gomez F, Schmidhuber J. Connectionist temporal classification: labelling unsegmented sequence data with recurrent neural networks. In: Proceedings of the 23rd international conference on machine learning. 2006, p. 369–76.

[51] Graves A, Jaitly N. Towards end-to-end speech recognition with recurrent neural networks. In: International conference on machine learning. PMLR; 2014, p. 1764–72.

[52] Novoa J, Wuth J, Escudero JP, Fredes J, Mahu R, Yoma NB. DNN-HMM based automatic speech recognition for HRI scenarios. In: Proceedings of the 2018 ACM/IEEE international conference on human-robot interaction. 2018, p. 150–9.

[53] Audhkhasi K, Saon G, Tüske Z, Kingsbury B, Picheny M. Forget a bit to learn better: Soft forgetting for CTC-based automatic speech recognition. In: Proc. Interspeech 2019. 2019, p. 2618–2622.

[54] Jenkins JJ, Strange W. Perception of dynamic information for vowels in syllable onsets and offsets. Percept Psychophys 1999;61(6):1200–10.

[55] Malaia EA, Wilbur RB. Syllable as a unit of information transfer in linguistic communication: The entropy syllable parsing model. Wiley Interdiscip Rev: Cogn Sci 2019.

[56] Marr D. Vision: A computational investigation into the human representation and processing of visual information, henry holt and co. New York, NY 2: Inc.; 1982.

[57] Scharenborg O, Norris D, Ten Bosch L, McQueen JM. How should a speech recognizer work? Cogn Sci 2005;29(6):867–918.

[58] Norris D. Shortlist: A connectionist model of continuous speech recognition. Cognition 1994;52(3):189–234.

[59] Norris D, McQueen JM. Shortlist B: a Bayesian model of continuous speech recognition. Psychol Rev 2008;115(2):357.

[60] Massaro D. Perceptual images processing time and perceptual units in auditory perception. Psychol Rev 1972;2:124–45.

[61] Ostendorf M. Moving beyond the ‘beads-on-a-string’model of speech. In: Proc. IEEE ASRU workshop. 1999, p. 79–84.

[62] Fujimura O. Syllable as a unit of speech recognition. IEEE Trans Acoust Speech Signal Process 1975;23(1):82–7.

[63] Yule G, Bernini G. Introduzione alla linguistica. Il mulino; 1997.

[64] Martin P. Prominence detection without syllabic segmentation. In: Proc. of speech prosody [Online]. 2010, p. 1–4, URL: http://speechprosody2010.illinois.edu/papers/102010.pdf.

[65] D'Alessandro C, Mertens P. Automatic pitch contour stylization using a model of tonal perception. Comput Speech Lang 1995;9(3):257–88.

[66] Roach P. English phonetics and phonology. A practical course. Cambridge University Press; 2000.

[67] MacNeilage PF, Davis BL. On the origin of internal structure of word forms. Science 2000;288(5465):527–31.

[68] Fujimura O. Syllable timing computation in the c/d model. In: Third international conference on spoken language processing (ICLPS 1994), Yokohama, Japan. 1994, p. 519–22.

[69] Warren RM, Healy EW, Chalikia MH. The vowel-sequence illusion: Intrasubject stability and intersubject agreement of syllabic forms. J Acoust Soc Am 1996;100(4):2452–61.

[70] Arnal LH, Poeppel D, Giraud A-L. A neurophysiological perspective on speech processing in “The Neurobiology of Language”. In: Neurobiology of language. Elsevier; 2016, p. 463–78.

[71] Greenberg S. Speaking in shorthand–A syllable-centric perspective for understanding pronunciation variation. Speech Commun 1999;29(2–4):159–76.

[72] Cutugno F, Leone E, Ludusan B, Origlia A. Investigating syllabic prominence with conditional random fields and latent-dynamic conditional random fields. In: Thirteenth annual conference of the international speech communication association. 2012, p. 2402–5.

[73] Wu S-L, Kingsbury E, Morgan N, Greenberg S. Incorporating information from syllable-length time scales into automatic speech recognition. In: Proceedings of the 1998 IEEE international conference on acoustics, speech and signal processing, ICASSP'98 (Cat. No. 98CH36181), Vol. 2. IEEE; 1998, p. 721–4.

[74] Kahn D. Syllable-based generalizations in english phonology. Routledge; 2015.

[75] Peeva MG, Guenther FH, Tourville JA, Nieto-Castanon A, Anton J-L, Nazarian B, Alario F-X. Distinct representations of phonemes, syllables, and supra-syllabic sequences in the speech production network. Neuroimage 2010;50(2):626–38.

[76] Rong F, Isenberg AL, Sun E, Hickok G. The neuroanatomy of speech sequencing at the syllable level. PLoS One 2018;13(10):e0196381.

[77] Kingsbury BE, Morgan N, Greenberg S. Robust speech recognition using the modulation spectrogram. Speech Commun 1998;25(1–3):117–32.

[78] Wu S-L, Kingsbury ED, Morgan N, Greenberg S. Incorporating information from syllable-length time scales into automatic speech recognition. In: Proceedings of the 1998 IEEE international conference on acoustics, speech and signal processing, ICASSP '98 (Cat. No.98CH36181), Vol. 2. 1998, p. 721–4.

[79] Cutugno F, Coro G, Petrillo M. Multigranular scale speech recognizers: Technological and cognitive view. In: Bandini S, Manzoni S, editors. AI*IA 2005: Advances in artificial intelligence. Berlin, Heidelberg: Springer Berlin Heidelberg; 2005, p. 327–30.

[80] Coro G. A step forward in multi-granular automatic speech recognition (Ph.D. thesis), Naples, Italy: University of Naples, Federico II; 2008.

[81] Baby D, Hamme HV. Investigating modulation spectrogram features for deep neural network-based automatic speech recognition. In: Sixteenth annual conference of the international speech communication association. 2015, p. 2479–83.

[82] Batliner A, Möbius B. Prosody in automatic speech processing. 2019.

[83] Chang S. A syllable, articulatory-feature, and stress-accent model of speech recognition (Ph.D. thesis), Berkeley: University of California; 2002.
[84] Pinson MB, Pinson DT. Syllable based automatic speech recognition. 2019, US Patent App. 16/031, 637.
[85] Jespersen O. Lehrbuch der phonetik. Indoger Forsch 1905;18(s1):594.
[86] House D. Differential perception of tonal contours through the syllable. In: Proc. of ICSLP. 1996, p. 2048–51.
[87] Cutugno F, D'Anna L, Petrillo M, Zovato E. APA: Towards an automatic tool for prosodic analysis. In: Speech prosody 2002, international conference. 2002, p. 231–4.
[88] D'Anna L, Cutugno F. Segmenting the speech chain into tone units: human behaviour vs automatic process. In: Proceedings of the XVth international congress of phonetic sciences (icphs). 2003, p. 1233–6.
[89] D'Anna L, Petrillo M. Sistemi automatici per la segmentazione in unità tonali. In: Atti delle XIII giornate di studio del gruppo di fonetica sperimentale (GFS). 2003, p. 285–90.
[90] Origlia A, Abete G, Cutugno F. A dynamic tonal perception model for optimal pitch stylization. Comput Speech Lang 2013;27(1):190–208.
[91] Origlia A, Cutugno F. Combining energy and cross-entropy analysis for nuclear segments detection. In: INTERSPEECH. 2016, p. 2958–62.
[92] Origlia A, Cutugno F, Galatà V. Continuous emotion recognition with phonetic syllables. Speech Commun 2014;57:155–69.
[93] Siemund R, Höge H, Kunzmann S, Marasek K. SPEECON-speech data for consumer devices. In: LREC. Citeseer; 2000, p. 329–33, URL: http://www.lrec-conf.org/proceedings/lrec2000/pdf/63.pdf.
[94] ELRA. Italian Speecon database. 2019, http://catalogue.elra.info/en-us/repository/browse/ELRA-S0213/.
[95] CMU. The carnegie mellon university CLM toolkit. 2019, https://sourceforge.net/projects/cmusphinx/files/cmuclmtk/0.7/.
[96] Davis S, Mermelstein P. Comparison of parametric representations for monosyllabic word recognition in continuously spoken sentences. IEEE Trans Acoust Speech Signal Process 1980;28(4):357–66.
[97] Tyagi V, Wellekens C. On desensitizing the mel-cepstrum to spurious spectral components for robust speech recognition. In: Proceedings.(ICASSP'05). IEEE international conference on acoustics, speech, and signal processing, 2005. Vol. 1. IEEE; 2005, p. I–529.
[98] Parcollet T, Zhang Y, Morchid M, Trabelsi C, Linarès G, De Mori R, Bengio Y. Quaternion convolutional neural networks for end-to-end automatic speech recognition. 2018, arxiv preprint arXiv:1806.07789.
[99] Kim C, Kumar M, Kim K, Gowda D. Power-law nonlinearity with maximally uniform distribution criterion for improved neural network training in automatic speech recognition. In: 2019 IEEE automatic speech recognition and understanding workshop (ASRU). IEEE; 2019, p. 988–95.
[100] Paliwal KK. On the use of filter-bank energies as features for robust speech recognition. In: ISSPA'99. Proceedings of the fifth international symposium on signal processing and its applications (IEEE Cat. No. 99EX359), Vol. 2. IEEE; 1999, p. 641–4.
[101] Tyagi V, McCowan I, Misra H, Bourlard H. Mel-cepstrum modulation spectrum (MCMS) features for robust ASR. In: 2003 IEEE workshop on automatic speech recognition and understanding (IEEE Cat. No. 03EX721). IEEE; 2003, p. 399–404.
[102] Coro G, Cutugno F, Caropreso F. Speech recognition with factorial-HMM syllabic acoustic models. In: Eighth annual conference of the international speech communication association (Interspeech). 2007, p. 870–3.
[103] D'Anna L, Coro G, Cutugno F. EVALITA 2009: Abla srl participant report. In: EVALITA 2009 speech recognition challenge. 2009, p. 1–6, URL: http://www.evalita.it/2009/proceedings.
[104] Viterbi A. Error bounds for convolutional codes and an asymptotically optimum decoding algorithm. IEEE Trans Inform Theory 1967;13(2):260–9.
[105] Francois J-M. JAHMM: An implementation of hidden Markov models in Java. 2019, https://github.com/KommuSoft/jahmm.
[106] Lamere P, Kwok P, Gouvea E, Raj B, Singh R, Walker W, Warmuth M, Wolf P. The CMU SPHINX-4 speech recognition system. In: IEEE intl. conf. on acoustics, speech and signal processing (ICASSP 2003), Hong Kong, Vol. 1. 2003, p. 2–5.
[107] Yu D, Deng L. Deep neural network-hidden markov model hybrid systems. In: Automatic speech recognition. Springer; 2015, p. 99–116.
[108] Serizel R, Giuliani D. Deep-neural network approaches for speech recognition with heterogeneous groups of speakers including children. Nat Lang Eng 2017;23(3):325–50.
[109] Ravanelli M, Omologo M. Contaminated speech training methods for robust DNN-HMM distant speech recognition. 2017, arxiv preprint arXiv:1710.03538.
[110] Maas AL, Qi P, Xie Z, Hannun AY, Lengerich CT, Jurafsky D, Ng AY. Building DNN acoustic models for large vocabulary speech recognition. Comput Speech Lang 2017;41:195–213.
[111] Patel T, Krishna D, Fathima N, Shah N, Mahima C, Kumar D, Iyengar A. Development of large vocabulary speech recognition system with keyword search for manipuri. In: Interspeech. 2018, p. 1031–5.
[112] Smit MP, Virpioja S, Kurimo M. Advances in subword-based HMM-DNN speech recognition across languages. Computer Speech & Language 2021;66:101158. http://dx.doi.org/10.1016/j.csl.2020.101158, https://www.sciencedirect.com/science/article/pii/S0885230820300917.
[113] Chao G-L, Chan W, Lane I. Speaker-targeted audio-visual models for speech recognition in cocktail-party environments. 2019, arxiv preprint arXiv:1906.05962.
[114] Mao S, Tao D, Zhang G, Ching P, Lee T. Revisiting hidden Markov models for speech emotion recognition. In: ICASSP 2019-2019 IEEE international conference on acoustics, speech and signal processing (ICASSP). IEEE; 2019, p. 6715–9.
[115] Gael JV, Teh YW, Ghahramani Z. The infinite factorial hidden Markov model. In: Advances in neural information processing systems. 2009, p. 1697–704.
[116] Florian B, Sepp K, Joshua H, Richard H. Hidden markov models in the neurosciences. In: Hidden markov models, theory and applications. IntechOpen; 2011, p. 169.
[117] Virtanen T. Speech recognition using factorial hidden Markov models for separation in the feature space. In: Ninth international conference on spoken language processing. 2006, p. 89–92.
[118] Tu Y-H, Du J, Dai L-R, Lee C-H. A speaker-dependent deep learning approach to joint speech separation and acoustic modeling for multi-talker automatic speech recognition. In: 2016 10th international symposium on chinese spoken language processing (ISCSLP). IEEE; 2016, p. 1–5.
[119] Ghahramani Z. Matlab implementation of factorial hidden Markov models. 2002, http://mlg.eng.cam.ac.uk/zoubin/software.html.
[120] Krizhevsky A, Sutskever I, Hinton GE. Imagenet classification with deep convolutional neural networks. In: Pereira F, Burges CJC, Bottou L, Weinberger KQ, editors. Advances in neural information processing systems 25. Curran Associates, Inc.; 2012, p. 1097–105, URL: http://papers.nips.cc/paper/4824-imagenet-classification-with-deep-convolutional-neural-networks.pdf.
[121] Massoli FV, Amato G, Falchi F. Cross-resolution learning for face recognition. Image Vis Comput 2020;103927.
[122] Girshick R. Fast r-cnn. In: Proceedings of the IEEE international conference on computer vision. 2015, p. 1440–8.
[123] Deng L, Liu Y. Deep learning in natural language processing. Springer; 2018.
[124] Ortis A, Farinella GM, Battiato S. An overview on image sentiment analysis: Methods, datasets and current challenges. In: Proceedings of the 16th international joint conference on e-business and telecommunications, ICETE 2019 - Volume 1: DCNET, ICE-B, OPTICS, SIGMAP and WINSYS, Prague, Czech Republic, July 26-28, 2019. 2019, p. 296–306. http://dx.doi.org/10.5220/0007909602900300.
[125] Paszke A, Gross S, Massa F, Lerer A, Bradbury J, Chanan G, Killeen T, Lin Z, Gimelshein N, Antiga L, et al. PyTorch: An imperative style, high-performance deep learning library. In: Advances in neural information processing systems. 2019, p. 8024–35.
[126] LeCun Y, Bengio Y, et al. Convolutional networks for images, speech, and time series. In: The handbook of brain theory and neural networks, Vol. 3361. 1995, p. 1995.
[127] Hochreiter S, Schmidhuber J. Long short-term memory. Neural Comput 1997;9(8):1735–80.
[128] Coro G. Automatic speech recognition: A syllabic approach. 2004, https://sites.google.com/site/gianpaolocoro/ricerca/tesi-di-laurea.

[129] Coro G, Masetti G, Bonhoeffer P, Betcher M. Distinguishing violinists and pianists based on their brain signals. In: Tetko IV, Kůrková V, Karpov P, Theis F, editors. Artificial neural networks and machine learning – ICANN 2019: Theoretical neural computation. Cham: Springer International Publishing; 2019, p. 123–37.

[130] Kapur R. The vanishing gradient problem. 2020, https://ayearofai.com/rohan-4-the-vanishing-gradient-problem-ec68f76ffb9b.

[131] Mishkin D, Sergievskiy N, Matas J. Systematic evaluation of convolution neural network advances on the imagenet. Comput Vis Image Underst 2017;161:11–9.

[132] Novak R, Bahri Y, Abolafia DA, Pennington J, Sohl-Dickstein J. Sensitivity and generalization in neural networks: an empirical study. 2018, arxiv preprint arXiv:1802.08760.

[133] Srivastava N, Hinton G, Krizhevsky A, Sutskever I, Salakhutdinov R. Dropout: a simple way to prevent neural networks from overfitting. J Mach Learn Res 2014;15(1):1929–58.

[134] Muller MFK-R. Estimating a-posteriori probabilities using stochastic network models. In: Proceedings of the 1993 connectionist models summer school. Psychology Press; 2014, p. 324.

[135] VoxForge. VoxForge free speech recognition corpora. 2012, http://www.voxforge.org/.

[136] Peters J, Matusov E, Meyer C, Klakow D. Topic specific models for text formatting and speech recognition. 2011, US Patent 8, 041, 566.

[137] Ballinger BM, Schalkwyk J, Cohen MH, Allauzen CGL, Riley MD. Speech to text conversion. 2011, US Patent App. 12/976, 972.

[138] Kingma DP, Ba J. Adam: A method for stochastic optimization. 2014, arXiv preprint arXiv:1412.6980.

[139] NIST. SCTK, the NIST scoring toolkit. 2018, https://github.com/usnistgov/SCTK.

[140] Baroni M, Bernardini S, Ferraresi A, Zanchetta E. The WaCky wide web: a collection of very large linguistically processed web-crawled corpora. Lang Resour Eval 2009;43(3):209–26.

[141] Lyding V, Stemle E, Borghetti C, Brunello M, Castagnoli S, Dell'Orletta F, Dittmann H, Lenci A, Pirrelli V. The paisa'corpus of italian web texts. In: 9th web as corpus workshop (WaC-9)@ EACL 2014. EACL (European chapter of the Association for Computational Linguistics); 2014, p. 36–43.

[142] CLEF. The Clef initiative corpus. 2020, http://www.clef-initiative.eu/web/clef-initiative/home.

[143] Magnini B, Pianta E, Girardi C, Negri M, Romano L, Speranza M, Lenzi VB, Sprugnoli R. I-CAB: the Italian content annotation bank. In: LREC. Citeseer; 2006, p. 963–8.

[144] Milde B, Köhn A. Open source automatic speech recognition for german. In: Speech communication; 13th ITG-symposium. VDE; 2018, p. 1–5.

[145] Cole RA, Noel M, Lander T, Durham T. New telephone speech corpora at CSLU. In: Fourth European conference on speech communication and technology. 1995, p. 1–4.

[146] Greenberg S. On the origins of speech intelligibility in the real world. In: Robust speech recognition for unknown communication channels. 1997, p. 1–11, URL: http://http.icsi.berkeley.edu/ftp/global/pub/speech/papers/escarsr97-origins.pdf.

[147] Dimitrakakis C, Bengio S. Phoneme and sentence-level ensembles for speech recognition. EURASIP J Audio Speech Music Process 2011;2011:1–17.

[148] Kimura T, Nose T, Hirooka S, Chiba Y, Ito A. Comparison of speech recognition performance between kaldi and google cloud speech API. In: Pan J-S, Ito A, Tsai P-W, Jain LC, editors. Recent advances in intelligent information hiding and multimedia signal processing. Cham: Springer International Publishing; 2019, p. 109–15.

[149] Wang D, Wang X, Lv S. An overview of end-to-end automatic speech recognition. Symmetry 2019;11(8):1018.

[150] Ludusan B, Origlia A, Cutugno F. On the use of the rhythmogram for automatic syllabic prominence detection. In: Twelfth annual conference of the international speech communication association. 2011, p. 2413–6.

Gianpaolo Coro is a Physicist with a Ph.D. in Computer Science with a focus on Automatic Speech Recognition. His research focuses on Artificial Intelligence, Data Mining, Cloud Computing, and Open Science paradigm applied to the domains of Ecology and Natural Language Processing.

Fabio Valerio Massoli is a PostDoc at the AIMH lab of ISTI-CNR. He has a Ph.D. in High Energy Physics from University of Bologna, in collaboration with the Columbia University (NY), with a thesis on Dark Matter search. Currently, his research interests include deep learning, supervised and unsupervised learning, generative models, and quantum theory and technologies.

Antonio Origlia took his PhD in 2013 with a thesis on Affective Computing, focusing on emotional speech analysis with robotics applications. Then, he concentrated on Human–Computer Interaction topics, mainly focusing on applications for Cultural Heritage also involving the use of speech. His work mainly concentrates on probabilistic dialogue systems and their use in advanced applications developed using game engines.

Francesco Cutugno is associate professor of Computational Linguistics and Human Machine Interaction at University Federico II of Naples, Italy. From 2013 to 2018 he has been the President of the Italian Speech Sciences Association. His main research interests are in the fields of acoustic phonetics; computational linguistics; automatic spoken dialogue systems, technology applications in the cultural heritage sector.