Real to H-space Autoencoders for Theme Identification in Telephone Conversations

Titouan Parcollet, Mohamed Morchid, Xavier Bost, Georges Linarès, Renato de Mori

Real to H-space Autoencoders for Theme Identification in Telephone Conversations

Titouan Parcollet, *Student Member, IEEE,* Mohamed Morchid, *Member, IEEE,* Xavier Bost, Georges Linarès, and Renato De Mori, *Life Fellow, IEEE*

***Abstract*—Machine learning (ML) and deep learning with deep neural networks (DNN), have drastically improved the performances of modern systems on numerous spoken language understanding (SLU) related tasks. Since most of current researches focus on new neural architectures to enhance the performances in realistic conditions, few recent works investigated the use of different algebras with neural networks (NN), to better represent the nature of the data being processed. To this extent, quaternion-valued neural networks (QNN) have shown better performances, and an important reduction of the number of neural parameters compared to traditional real-valued neural networks, when dealing with multidimensional signal. Nonetheless, the use of QNNs is strictly limited to quaternion input or output features. This paper introduces a new unsupervised method based on a hybrid autoencoder (AE) called real-to-quaternion autoencoder (R2H), to extract a quaternion-valued input signal from any real-valued data, to be processed by QNNs. The experiments performed to identify the most related theme of a given telephone conversation from a customer care service (CCS), demonstrate that the R2H approach outperforms all the previously established models, either real- or quaternion-valued ones, in term of accuracy and with up to four times fewer neural parameters.**

***Index Terms*—Features extraction, quaternion autoencoder, quaternion neural networks, spoken language understanding**

I. Introduction

Spoken language understanding (SLU) is a major subdomain of artificial intelligence (AI). It defines the human-machine interactions and allows the machine to interpret the human language. Recent advances in the domain have been pushed by the resurgence of neural networks (NN), and more precisely deep neural networks (DNN). Interesting solutions have been therefore proposed for SLU in human-computer dialogues [1], [2], [3], [4], [5]. Also, numerous DNNs have been employed for the extraction of semantic contents in textual documents. Noticeable examples are the recognition of implicit discourse relations [6], information distillation for compressive summarization [7], and extraction of document-level context for machine translation [8]. Extracting semantic contents in spoken conversations is complicated due to the localisation and characterisation of acoustic correlations of relevant acoustic features as reported in [9]. An important component of this research area is the task of topic identification, for which an ample review of the state-of-the-art can be found in [9].

The task of inferring the topics discussed in a real-life human-human telephone conversation is particularly difficult. This paper deals with customer care services (CCS) in which an agent interacts with a customer to address her/his concerns and to provide a solution. An automatic system is developed for automatically detecting the topic of concern, called here the *theme of the conversation*, allowing the agent to concentrates only on solving problems and retrieving necessary information. Automatically annotated themes are employed to perform useful statistics. Human-human telephone conversations are a type of spoken documents involving at least two speakers, suggesting that topics and other semantic contents may have multiple views in a conversation. For example, there may be a view for each speaker and a global view including both speakers. Another view may be the voice of other speakers briefly consulted for obtaining external information, such as a different service of the CCS. Another possibility can be to consider views that describe the formulation of a problem, the solution, or simply views of different segments of a conversation such as speech turns. Indeed, one can consider that the agent is following a very specific path that evolves during the conversation. Unfortunately, appropriate views of a conversation cannot always be inferred with prior knowledge. This paper therefore proposes to learn these views without supervision from any real-valued representation of the document. As views cannot be localized with prior knowledge, it is appropriate to start with real vector features computed with latent Dirichelet allocation (LDA). The use of LDA is motivated by the observation that accurate topic classification for human-human telephone conversations has been obtained with a sophisticated integration of features in a large number of hidden spaces [10]. The possibility of learning, without any supervision, nor manual data segmentation, latent view representations from any real-valued representation of the document, in a single hidden topic space, is investigated in this paper. A satisfactory result would be to achieve performances close to the best results obtained with multiple topic spaces or manually segmented views. With the purpose of reducing computational complexity in a test phase, staked denoising autoencoders using LDA features were proposed on the same task in [11]. To represent these multiple views, multidimensional algebras such as quaternion numbers have

T. Parcollet and X. Bost are with ORKIS, Aix-en-Provence, FRANCE, (email: titouan.parcollet@alumni.univ-avignon.fr, xbost@orkis.com).

T. Parcollet, M. Morchid and G. Linarès are with Avignon Université, Laboratoire Informatique d'Avignon, 339 Chemin des Meinajariès, 84000 Avignon, FRANCE (emails: titouan.parcollet@alumni.univ-avignon.fr, {firstname.lastname}@univ-avignon.fr) .

R. De Mori is with McGill University, Montréal, QC, CANADA (email: rdemori@cs.mcgill.ca).

This work was funded by the AISSPER project supported by the French National Research Agency (ANR) under contract AAPG 2019 ANR-19-CE23-0004-01.

been considered with neural networks. Nonetheless, all these multiviews are extracted based on prior assumptions and may differ in their forms depending on the context of the human-human interactions.

In this paper a new neural architecture based on hyper complex numbers is proposed for unsupervised inference of new representations, made of quaternion number vectors, from a global set of real-valued feature vectors, embedding the whole conversation. Each inferred quaternion is estimated to integrate, in an abstract representation, multiple views to solve the task. A list of the contributions of this work is:

- Detail previously investigated quaternion-valued neural networks models in a comprehensive way (Section IV).
- Introduce a new unsupervised method based on a hybrid autoencoder (AE) called real-to-quaternion autoencoder (R2H, Section V), to create and extract a quaternion-valued input signal from any real-valued source in an unsupervised manner.
- Combine this generated quaternion-valued features to the recent quaternion convolutional neural network (QCNN, Section VI) in a R2H-QCNN.
- Compare the R2H-QCNN to previously investigated quaternion, and state-of-the-art real-valued models on a theme identification task of telephone conversations (Section VII).

The experiments show that the R2H-QCNN architecture outperforms all the previous models on the French DECODA framework, highlighting the efficiency of unsupervised quaternion features extraction made by R2H. Furthermore, we show that quaternion-valued models always outperform real-valued comparable architectures with up to four times fewer neural parameters.

II. Related work

Recently, neural architectures have been proposed for extracting semantic content from documents. Interesting results have been obtained in the extraction of global textual document features of latent semantic relations. In [6], a neural model is proposed that gradually focus on more fine-grained parts of a document, after grasping the global information of a relation between two arguments. In [12], a dynamic attention neural model is introduced to capture the interactions between a question and a document. In [13], existing approaches to diarization, an important automatic speech analysis task, are reviewed and novel approaches are proposed to improve automatic speech recognition for this task. A review on non-automatic conversation analysis and models of conversational interactions applicable to speech can be found in [14]. Multiple views are not explicitly mentioned, but the possibility of considering them has motivations in this review. In [15], an end-to-end framework for topic identification using multi-scale convolutional neural networks (CNN) is proposed, with an algorithmic approach for integrating verification and identification tasks, without taking into account the possible expression of semantic contents. The input is raw text, but the approach could also be applied to speech documents. Then, [16] proposed to integrate different types of word and semantic features with the latent Dirichlet allocation (LDA) method, jointly used with a deep neural network for the task of topic identification in telephone conversations. A deep bottleneck features extractor is investigated in [17], on the same task of theme identification of telephone conversations. In the latter, the authors proposed various deep denoising autoencoders to build robust representations of automaticaly transcribed conversations, for better classification performances. In [18] two automatic speech recognition (ASR) systems, based on word and grapheme embeddings, are proposed to transcribe speech for topic identification. Distributed representation of speech are obtained with CNNs, with an average pooling layer to integrate the outputs of the two ASR systems. These hidden representations do not take into account the possibility that topic mentions may express domain dependent and multiple semantic views. In [19], a gating attention model is proposed for separating a set of voices out of a sound mixture, containing an unknown number of sources during a conversation. Individual speaker embeddings are learned to separate a single speaker, while superpositions of the individual speaker embeddings are used to separate sets of speakers. Different speakers in a conversation could be viewed as expressing different views, but this specific aspect is not considered in the paper. For human-human telephone conversations analysis, novel conversation features represented by vectors of quaternions were introduced in [20].

Quaternions are an extension of complex-numbers, and are defined as hypercomplex numbers that contain a real and three separate imaginary components, perfectly fitting to three and four dimensional feature vectors [21], such as for image processing, robot kinematics or computer graphics [22], [23], [24]. Contrary to traditional homogeneous representations, quaternion networks bundle sets of features together. Thereby, quaternions allow neural network based models to code latent inter-dependencies between groups of input features during the learning process with fewer parameters than real-valued neural networks, by taking advantage of the *Hamilton product* as the equivalent of the mere dot product. Early applications of quaternion-valued backpropagation algorithms [25], [26] have efficiently solved quaternion functions approximation tasks. More recently, neural networks of complex and hyper complex numbers have received an increasing attention [27], [28], [29], [30], and some efforts have shown promising results in different applications. In particular, a deep quaternion network [20], [31], [32], a deep quaternion convolutional network [33], [34], and a quaternion recurrent neural network [35], [36] have been employed for challenging tasks such as images classification, compression and reconstruction, or speech recognition and natural language processing.

Motivations to use quaternion neural networks over real valued ones are numerous and well documented [37], [35]. The major advantage of QNNs is the more efficient and natural representation of the multidimensional input information, compared to traditional real-valued NNs. Indeed, a good neural network model has to represent, encode, and learn the relations charactarizing the input data. Many realistic tasks involve multi-dimensional features, with input elements composed with multiple internally related components. These

applications are image processing, with three channels describing a single pixel, or speech recognition, with three values (Mel filter banks and deltas) for a unique time frequency, or human motion recognition, with outputs and inputs to be represented in the 3D space with 3D coordinates. To succeed in such multidimensional tasks, the employed model needs a specific representation and processing of the features. Indeed, traditional real-valued NNs consider internal relations at the same level than contextual and global dependencies. As an example, relations between different pixels are considered equivalently to the relation that links the three channels composing each pixel, and it is not clear if both relations are accurately learned. More precisely, composed entities are processed by traditional real-valued NNs as a bag of independent and smaller components, while a natural solution is to process such entities as multidimensional and internally related elements with quaternion neural networks. QNNs have shown to be effective to learn internal relations describing the multidimensional features, with much less neural parameters [38]. Indeed, a 4-number quaternion weight linking two 4-number quaternion units only has 4 degrees of freedom, whereas a standard neural net parametrization has 4x4 = 16, *i.e.*, a 4-fold saving in memory.

While quaternion neural networks have been used in a wide range of domain applications, they have mostly been limited to tasks that are easily interpretable in the three and four dimensional spaces. In the latter context, it has been demonstrated that QNNs offer superior performances with a reduction by a factor of four of the number of neural parameters over real-valued neural networks. In this paper, we propose to bridge the gap of the input representation, by introducing an unsupervised technique to extract meaningful quaternion features from any type of input signal, making it feasible to benefit from the better internal representation of QNNs regardless of the space domain of the task.

III. Quaternion algebra

Quaternion numbers $\mathbb{H}$ have been discovered by William Rowan Hamilton [21], and are part of the hyper-complex numbers as a non-commutative extension of complex numbers. One of the remarkable property of quaternion numbers is a solution to the well known Gimbal lock problem for Euler angles [39]. A quaternion Q is written as:

$$Q = r + x\mathbf{i} + y\mathbf{j} + z\mathbf{k}, \tag{1}$$

with 1, **i**, **j**, and **k** the four bases of the four-dimensional vector space, that follow the mathematical relations:

$$\mathbf{i}^2 = \mathbf{j}^2 = \mathbf{k}^2 = \mathbf{ijk} = -1. \tag{2}$$

Such relations are at the heart of quaternions, and make the $\mathbb{H}$ algebra an efficient tool to represent spatial rotations. In a quaternion, r is the real part or scalar part named s while $x\mathbf{i}$, $y\mathbf{j}$, and $z\mathbf{k}$ is the imaginary part or vector part also noted $\overrightarrow{v}$. Therefore, Q can be summarized as:

$$Q = (s, \overrightarrow{v}). \tag{3}$$

One of the quaternion polar representation is:

$$Q = ||Q||(cos\theta + nsin\theta) = ||Q||e^{n\theta} \tag{4}$$

with

$$cos\theta = \frac{s}{||Q||}, \quad sin\theta = \frac{||\overrightarrow{v}||}{||Q||}, \quad n = \frac{\overrightarrow{v}}{||\overrightarrow{v}||} \tag{5}$$

Considering $\mathbb{R}$ and the complex plane $\mathbb{C}$ as subsets of the hyper-complex algebra, Q can be projected back to either of these spaces following:

$$Q_{mat} = \begin{bmatrix} r & -x & -y & -z \\ x & r & -z & y \\ y & z & r & -x \\ z & -y & x & r \end{bmatrix}, \tag{6}$$

for the real plane and:

$$Q_{mat} = \begin{bmatrix} (r + x\mathbf{i}) & (y + z\mathbf{i}) \\ (-y + z\mathbf{i}) & (r - y\mathbf{i}) \end{bmatrix}, \tag{7}$$

for the complex one. The real-valued matrix representation turns out to be particularly efficient for computations, due to the parallelization capabilities of modern graphic processing units (GPU). The conjugate Q^* is an involution of Q and is noted as:

$$Q^* = r1 - x\mathbf{i} - y\mathbf{j} - z\mathbf{k}. \tag{8}$$

The norm of Q denoted $||Q||$ is the same as Euclidean norm in $\mathbb{R}$ but in the four dimensional space $\mathbb{H}$:

$$||Q|| = \sqrt{r^2 + x^2 + y^2 + z^2}. \tag{9}$$

Therefore, a normalized quaternion or unit quaternion $Q^\triangleleft$ can be easily expressed as:

$$Q^\triangleleft = \frac{Q}{||Q||}. \tag{10}$$

In the same manner as for complex numbers, unit quaternions lie on a unit sphere and are used to precisely describe rotations. The inverse Q^{-1} of Q is written as:

$$Q^{-1} = \frac{Q^*}{||Q||^2}. \tag{11}$$

The unit quaternion $Q^\triangleleft$ represents a rotation by an angle θ around a unit axis vector (of length 1) $\overrightarrow{v}$ as:

$$Q^\triangleleft = (cos\frac{\theta}{2} + \overrightarrow{v}sin\frac{\theta}{2}). \tag{12}$$

The conjugate p' of p by $Q^\triangleleft$ is the new position vector of the point after the rotation of $p = (x, y, z)$ by $Q^\triangleleft$ defined as:

$$p' = Q^\triangleleft \otimes p \otimes Q^{\triangleleft -1}, \tag{13}$$

with $\otimes$ the *Hamilton product* between two quaternions Q_1 and Q_2 defined as:

$$\begin{aligned} Q_1 \otimes Q_2 =& (r_1r_2 - x_1x_2 - y_1y_2 - z_1z_2) + \\ & (r_1x_2 + x_1r_2 + y_1z_2 - z_1y_2)\boldsymbol{i} + \\ & (r_1y_2 - x_1z_2 + y_1r_2 + z_1x_2)\boldsymbol{j} + \\ & (r_1z_2 + x_1y_2 - y_1x_2 + z_1r_2)\boldsymbol{k}. \end{aligned} \tag{14}$$

The Hamilton product is used to performs composition of rotations in $\mathbb{R}^3$. Consequently, the quaternion $Q_3 = Q_1 \otimes Q_2$ is equivalent to the rotation Q_1 followed by the rotation Q_2. It is also worth underlying that the product commutativity does not hold in the quaternion space:

$$Q_1 \otimes Q_2 \neq Q_2 \otimes Q_1. \tag{15}$$

Indeed, an object rotated by Q_1 and Q_2 will not be in the same position as it was rotated by Q_2 and Q_1.

IV. Quaternion neural networks

Since the initial proposal of a quaternion-valued multilayer perceptron neural network [25], many new quaternion-based architectures have been developed to benefit from modern, more efficient and powerful neural networks. This section proposes to summarize all the milestones needed to build the proposed R2H-QCNN in a comprehensive way. First, the fundamentals of a basic quaternion-valued neural network (Section IV-A) are presented. Then, the quaternion encoder-decoder neural network (QAE), also known as autoencoder neural network is introduced (Section IV-B), while a quaternion convolutional neural network (QCNN) used for classification is depicted (Section IV-C). Finally, a discussion on the computational complexity of quaternion neural networks is provided (Section IV-D).

A. Basics of quaternion multilayer perceptrons

The multilayer perceptron (MLP) was the first artificial neural network architecture extended to the quaternion algebra (QMLP) [25]. Since then, all the proposed quaternion-valued NNs have been derived from these fundamental equations. [25] have first proposed an adapted backpropagation algorithm that takes into account the specificities of the *Hamilton product* (Eq. 14). This section introduces the basic building blocks of a QMLP, including activation functions (Section IV-A1), the forward (Section IV-A2), backward and update (Section IV-A3) steps.

1) Activation functions: Activation functions are at the heart of neural networks. Indeed, non-linearity allows NNs to solve complex tasks. Common activation functions are the sigmoid, the hyperbolic tangent ($\tanh$), or the more recent rectified linear unit (ReLU) [40], [41]. In fact, real-valued activation functions are a well established research area in the machine learning community with an important number of previous researches on various solutions [42], [43]. Quaternion-valued activation functions are not as straighforward as their real-valued counterpart. Indeed, based on the Cauthy-Riemann-Fueter (CRF) equation, the only functions that are analytic in the quaternion domain are either linear or constant. Fortunately, it has been demonstrated that *locally* analytic functions are suitable to train a neural network with the backpropagation algorithm [44]. Consequently, two classes of quaternion-valued activation functions have emerged. *Fully quaternion-valued functions* are extensions to the hypercomplex domain of real-valued activation functions, such as sigmoid or $\tanh$ functions. Our work does not rely on this class of functions that are harder to train due to many singularities [45]. In fact, the most widely spread quaternion activation functions are called *split activation* functions [46]. *Split activations* rely on a simpler implementation and in a more stable training phase due to the decorrelation of the four components composing a quaternion. Unfortunately, they are also reported to obtain slightly worst results [47] depending on the task. Indeed, *split activation* functions map quaternion numbers back to the real-valued space by ignoring the nature of the relation that exists between the components. Let $\alpha(Q)$ be a *split activation* function applied to the quaternion Q:

$$\alpha(Q) = f(r) + f(x)\mathbf{i} + f(y)\mathbf{j} + f(z)\mathbf{k}, \tag{16}$$

with f corresponding to any standard and real-valued activation function (e.g, sigmoid, $\tanh$, ReLU, eLU,...,). The split approach is adopted in the rest of the QMLP equations considering its usage in the first proposition of the QMLP [25] alongside with its higher number of applications.

2) Forward step: In a quaternion-valued neural network layer, all parameters are quaternions, including inputs, weights, biases and outputs. Let us introduce the QMLP first described by [25]. The model is made of M layers of N nodes or neurons, whose number depends on the layer. Let x be the input to a node. Let N_l be the number of neurons contained in the layer l ($1 \leq l \leq M$) and M be the number of layers of the QMLP. b_n^l is the bias of the neuron n ($1 \leq n \leq N_l$) from the layer l. Given a set of P quaternion input patterns x_p ($1 \leq p \leq P$) and a set of labels t_p associated to each x_p, the output γ_n^l ($\gamma_n^0 = x_p^n$) of the neuron n of the layer l is:

$$\gamma_n^l = \alpha(S_n^l), \tag{17}$$

with

$$S_n^l = \sum_{m=0}^{N_{l-1}} w_{nm}^l \otimes \gamma_m^{l-1} + b_n^l, \tag{18}$$

and α any *split activation* function. Finally, the output layer of a QMLP is either quaternion-valued, such as for quaternion approximation [26], or real-valued to obtains a posterior distribution based on a softmax function [20]. Indeed, target classes are often expressed as real numbers.

3) Backward and update steps: The loss function E is a crucial element that quantifies how far the inferred predictions S^M are from the target distribution t_p. Numerous real-valued loss functions have been defined and used for different tasks. In its original paper, [25] proposed an extension of the mean squared error (MSE) [48] to quaternions (QMSE) by solely replacing real-numbers by hyper-complex numbers:

$$E = \frac{1}{N} \sum_{n=1}^{N} (t_{pn} - S_n^M)^2 \tag{19}$$

One may notice that any traditional and real-valued loss function can be applied to a real-valued QMLP output layer. Starting from the QMSE, the backpropagation of QNNs is an extension of the standard backpropagation for real-valued NNs. Its full version is derived by [49], while a brief summary is introduced below. The gradient with respect to the loss E is expressed for each w^l that composes the weight matrix W^l:

$$\Delta^l = \frac{\partial E}{\partial W^l}. \tag{20}$$

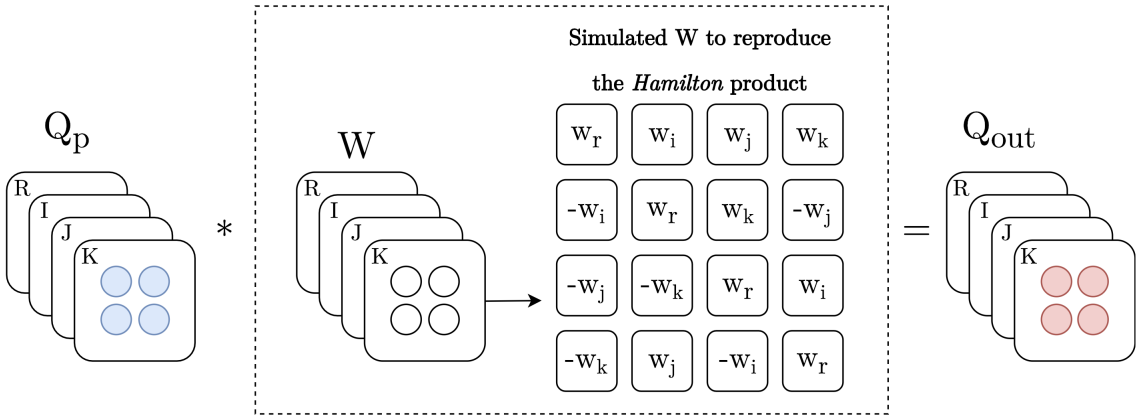

Fig. 1. Illustration of the quaternion-valued convolution process. First, the initial quaternion-valued filter map W is transformed into its real-valued representation. Then, each part of this new representation is convolved with the input signal Q_p. As for the traditional convolution process, rows of the simulated W represents the channels, while columns define the depth of the feature maps.

The output gradient ($l = M$) for each neuron n is expressed as follows:

$$\Delta_n^{l=M} = (t_{pn} - S_n^{l=M}). \tag{21}$$

And the gradients of the hidden layers parameters are derived following the chain rules:

$$\Delta_n^l = \sum_{n=1}^{N^{l+1}} w_{h,n}^{*l+1} \otimes (\Delta_h^{l+1} \alpha'(S_n^{l+1})). \tag{22}$$

Finally, biases and each weight w that connects the neuron n to m of the layer l are updated as:

$$w_{nm}^l = w_{nm}^l + \lambda \Delta_n^l \otimes S_m^{*l-1}, \quad b_n^l = b_n^l + \lambda \Delta_n^l, \tag{23}$$

with λ the learning rate.

B. Quaternion autoencoder neural networks

As described in [50] and [31], the quaternion-valued encoder-decoder or quaternion autoencoder (QAE) is a three-layered ($M = 3$) neural network made of an encoder and a decoder where $N_1 = N_3$, as depicted in Figure 2. The QAE has the same algorithm than the real-valued AE but with quaternion numbers, and with the *Hamilton product* to replace the standard dot product. Given a set of P normalized inputs Q_p ($1 \leq p \leq P$), the encoder computes a hidden representation $h = l_2$ of Q_p following:

$$h_n = \alpha(\sum_{m=0}^{P} w_{nm}^{l_2} \otimes Q_m + b_n^{l_2}), \tag{24}$$

with α any *split activation* function. Then, the decoder attempts to reconstruct the input vector Q_p from this $h_n = l_2$ hidden vector to obtain the output vector $\tilde{Q}_p$:

$$\tilde{Q}_n = \alpha(\sum_{m=0}^{N^{l_2}} w_{nm}^{l_3} \otimes h_m + b_n^{l_3}). \tag{25}$$

The learning phase follows the algorithms previously described in Section IV-A. Indeed, the QAE attempts to reduce the reconstruction error between $\tilde{Q}_p$ and Q_p based on the QMSE loss (Eq. 19).

C. Quaternion convolutional neural networks

The quaternion convolutional neural network (QCNN) [34] is an extension of the widely investigated real-valued CNN to the quaternion algebra. As for QMLP and QAE, the QCNN only contains quaternion numbers and relies on the backpropagation algorithm at learning time. From a formal point of view, the convolution process can easily be extended to the quaternion domain. Let S_{ab}^l be the pre-activation quaternion output at layer l and at the indexes (a, b) of the new feature map, and w the weight filter map of size $f \times f$. The forward equation of a QCNN can be derived based on Eq. 18 as:

$$\gamma_{ab}^l = \alpha(S_{ab}^l), \tag{26}$$

with

$$S_{ab}^l = \sum_{c=0}^{f-1} \sum_{d=0}^{f-1} w^l \otimes \gamma_{(a+c)(b+d)}^{l-1}. \tag{27}$$

with α any *split activation* function. More practically, the quaternion-valued convolution process is implemented in the real-valued space as proposed in [51] and [33]. In this extent, a traditional $2D$ convolutional layer, with a kernel that contains f feature maps, is split into four parts: the first part equal to r, the second one to $x\mathbf{i}$, the third one to $y\mathbf{j}$ and the last one to $z\mathbf{k}$ of a quaternion $Q = r1 + x\mathbf{i} + y\mathbf{j} + z\mathbf{k}$. Then, the quaternion-valued matrix is projected back to the real-valued space following Eq. 6 before being convolved with the quaternion-valued input signal as depicted in Figure 1.

D. Notes on computational complexity

The computational complexity remains unchanged with quaternion neural networks compared to real-valued ones, due to the fact that QNNs do not add any depth to the computation

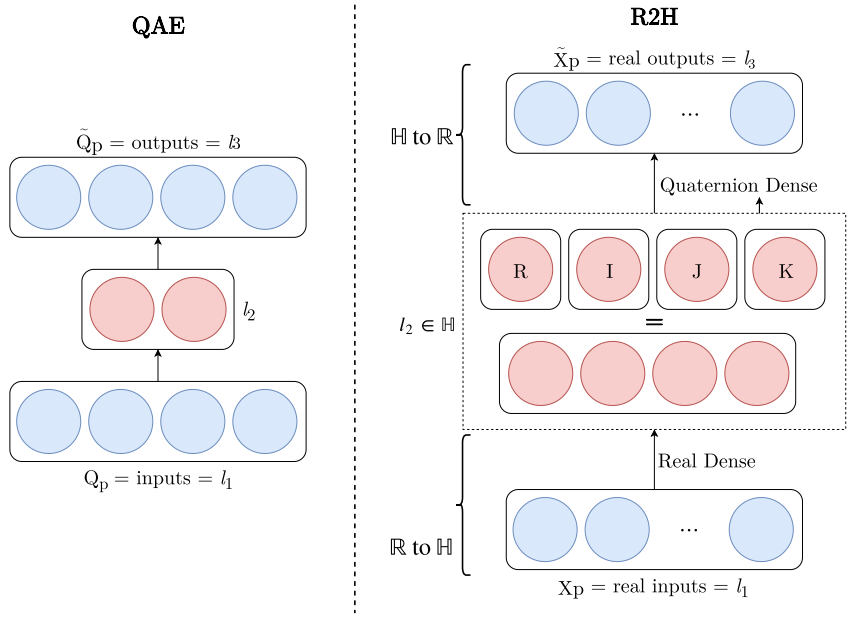

Fig. 2. Illustration of a quaternion autoencoder (QAE, left) and a real-to-quaternion autoencoder (R2H, right).

graph. Indeed, quaternions are manipulated following the real-valued matrix representation (Eq. 6). Nonetheless, and due to the *Hamilton product*, a single forward propagation between two quaternion neurons uses 28 operations, compared to a single one for two real-valued neurons, implying a longer training time (up to 3 times slower). However, such worst speed performances could easily be alleviated with a proper engineered cuDNN kernel implementation for the *Hamilton product*, that would helps QNNs to be more efficient than real-valued ones. Indeed, a well-adapted CUDA kernel would allow QNNs to perform more computations, with fewer parameters, and therefore less memory copy operations from the CPU to the GPU.

V. R2H AUTOENCODER

For many real-life problems, prior knowledge suggests that there are input features that have a limited number of related components. In certain cases, decomposition of input documents can be precisely defined by prior knowledge. For example, it is possible to assume that there are two different speakers for certain types of telephone services, suggesting that an input token could be represented with separate but related speaker views. Furthermore, the semantic content of a conversation may be represented by a limited number of high level abstractions, such as problem types, expressed by a limited number of views that are not precisely defined. In this case, it is appropriate to represent related view features with quaternion numbers, but it is necessary to learn to generate each component composing a quaternion. In this section, a method is proposed for learning this mapping with a real-to-quaternion autoencoder (R2H).

The R2H is a fusion of a real-valued AE and a QAE. The computation of the R2H is composed with two simple steps: 1) map any real-valued input vector to a hidden space with a real-valued encoder; 2) reconstruct the input data with a quaternion-valued decoder. The quaternion-valued decoder is expected to force the hidden representation of the real-valued encoder to project the input features into the quaternion space. Formally, the R2H has the same equations as QAEs, except for the encoder (Eq. 24) that is redefined in the real valued space as:

$$h_n = \alpha\left(\sum_{m=0}^{P} w_{nm}^{l_2} \times X_m + b_n^{l_2}\right), \tag{28}$$

with X the real-valued input vector of size P, and α any *split activation* function. Note that contrary to the QAE, all the parameters but h are real numbers. Indeed, h_n is processed by a standard QAE decoder (Eq. 25) and is therefore defined in the quaternion space. The learning process of the R2H follows the QAE (Section IV-B), with the optimization of the QMSE loss. It is important to notice the use of the *split activation* function in the output layer, since it is mandatory to allow the R2H to reconstruct a real-valued signal with decorrelated components. More practically, and as for QCNNs, the hidden layer h (l_2) is split into four parts that are equivalent to the four components of quaternion numbers as depicted in Figure 2. Therefore, the real-valued encoder uses different weight matrices to generate the real and the imaginary components of the quaternion hidden vector, while the quaternion-valued decoder will use a unique quaternion weight for each generated quaternion. Finally, the number of neurons of h and the number of input features must be divisible by four to be intepreted as quaternion vectors. Nevertheless, it is crucial to distinguish the difference in term of input representation between QAEs and R2H autoencoders. Indeed, QAEs must receive quaternion features as inputs, due to the *Hamilton product* that considers quaternion components to be related. In the case of decorrelated components, QAEs (or any quaternion

neural network) might suffer from bad performances, due to the difficulty induced by the abscence of internal relations. More precisely, QAEs offer an elegant way to generate a quaternion embedding from a quaternion input signal, while the R2H autoencoder allows the generation of quaternion embeddings regardless of the input signal domain, making it feasible to use R2H autoencoders in any real-world task.

VI. Experimental Protocol

As depicted in Figure 4, a textual document is first projected in a basic representation, based on word frequencies (Section VI-C) or latent Dirichlet allocation (LDA, Section VI-D)[52], before being fed to an autoencoder. In the case of a quaternion-valued autoencoder, documents must be first manually segmented following a proper quaternion scheme as detailed in Section VI-B. Different classifiers, AE and real-valued state-of-the-art architectures are proposed in Section VI-E.

A. Spoken conversations dataset

The corpus of spoken conversations is a set of automatically transcribed and annotated human-human telephone conversations of the Paris transportation system CCS (RATP), an example of which is shown in Figure 3. This corpus comes from the first version of the DECODA project [53], [54] and is employed to evaluate the effectiveness of the proposed R2H-QCNN on a conversation theme identification task. The DECODA corpus is composed of $1,242$ telephone conversations recorded during heavy traffics days in the capital, which is equivalent to about 74 hours of signal. The data set was split into 8 categories or dominant themes as described in Table I.

TABLE I
DECODA DATASET.

Class label	*Number of samples*		
	training	**development**	**testing**
problems of itinerary	145	44	67
lost and found	143	33	63
time schedules	47	7	18
transportation cards	106	24	47
state of the traffic	202	45	90
fares	19	9	11
infractions	47	4	18
special offers	31	9	13
Total	**740**	**175**	**327**

The LIA-Speeral Automatic Speech Recognition (ASR) system [55] is used to extract textual content of dialogues from the DECODA corpus. Acoustic model parameters were estimated from 150 hours of speech in telephone conditions. The vocabulary contains $5,782$ words. A 3-gram language model (LM) was obtained by adapting a basic LM with the training set transcriptions.

As conversations are collected in noisy environments the overall word error rate (WER) of the system is of 45.8% on the training set, 59.3% on the development set, and 58.0% on the test set. A "stop list" of 126 words[1] was used to remove unnecessary words (mainly function words) which results in a WER of 33.8% on the training, 45.2% on the development, and 49.5% on the test. These high WER are mainly due to speech disfluencies and to adverse acoustic environments (for example, calls from noisy streets with mobile phones).

[1]http://code.google.com/p/stop-words/

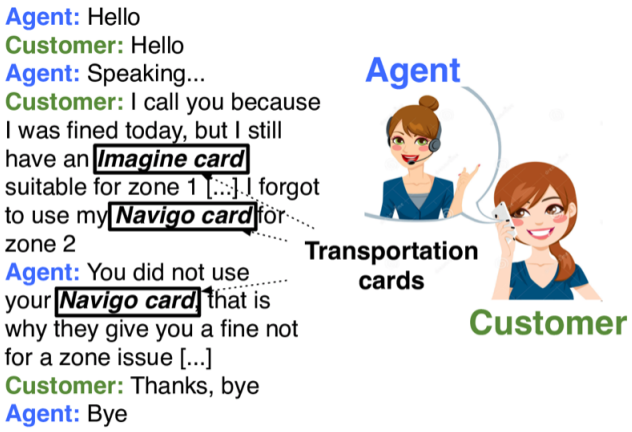

Fig. 3. Example of a *manualy transcribed* dialogue from the DECODA corpus for the SLU task of theme identification.

B. User-Agent-Document segmentation

The user-agent-document segmentation (UAD) proposed in [20] provides speaker dependent features encoded in quaternions. The initial idea is to construct a purely imaginary quaternion based on the speech turns of the different speakers. In this way, a dialogue quaternion $Q = r + x\mathbf{i} + y\mathbf{j} + z\mathbf{k}$ can be composed with the speech turns of the user in $x\mathbf{i}$, the speech turns of the agent in $y\mathbf{j}$ and the whole document in $z\mathbf{k}$. As for image processing [37] the real component r is set to zero, and will change to any other value after the computation performed with the *Hamilton product* to obtain the first layer latent features. The quaternion features obtained with UAD feeding a quaternion classifier have shown superior theme identification accuracies for the DECODA corpus compared with results obtained with other types of features [20], [32].

C. Word frequency-based representations

In information retrieval, textual documents are often considered as bag-of-words in which many basic statistical methods can be applied [56]. In this work, we propose to consider both a straightforward representation of the DECODA spoken documents called binary representation, and the more sophisticated and investigated term frequency–inverse document frequency also known as TF-IDF [56], as first baselines. First, the 500 most occurring words are extracted from the training set of DECODA to compose the reference vocabulary. The binary representation is obtained by composing a binary vector of size 500 (vocabulary size), denoting the occurrence (1) or the absence (0) of each word for a given spoken document. Such binary representation does not consider any information about the relevance of a given word considering a given document. In this extent, we propose to also use the frequency–inverse document frequency (TF-IDF) features [57]. Indeed, the TF-IDF is intended to reflect how important a word is to a

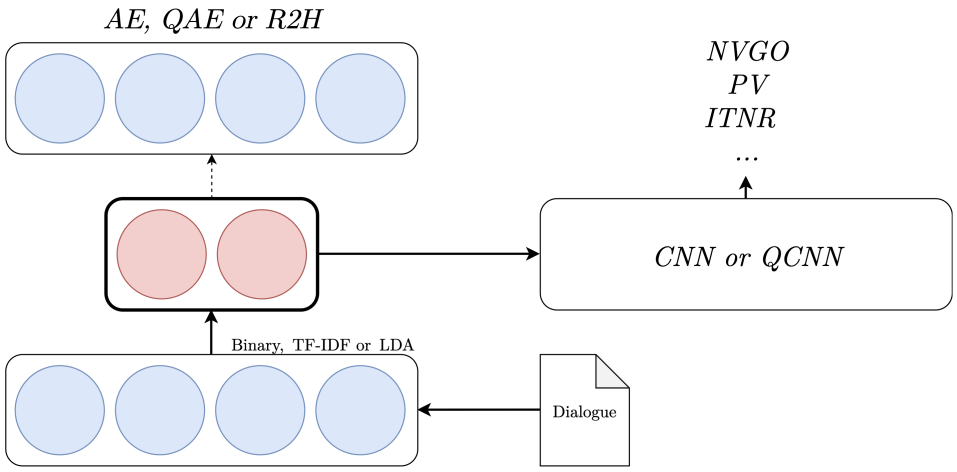

Fig. 4. Illustration of the complete architecture including the autoencoder (AE, QAE or R2H, left) and the classifier (CNN or QCNN, right).

document considering a corpus. Let w_i be a word contained in the document d_j containing $|d_j|$ words of a corpus C of size $|C|$, $f(w_i, d_j)$ is the frequency of w_i in the document d_j, and $n(w_i)$ is the number of document containing w_i. The $tf(w_i, d_f)$, and $idf(w_i)$ are calculated following:

$$tf(w_i, d_j) = \frac{f(w_i, d_j)}{|d_j|}, \quad idf(w_i) = log\frac{|C|}{n(w_i)}. \tag{29}$$

Finally, the TF-IDF is expressed as:

$$tfidf(w_i, d_j) = tf(w_i, d_j) \times idf(w_i). \tag{30}$$

During the experiments, documents are projected according to the segmentation. In the case of the UAD segmentation, each part is projected and concatenated to obtain a $500 \times 3 = 1500$ input vector. If no segmentation is applied, the final input vector only contains the projection of the entire document and is of size 500.

D. Latent Dirichlet Allocation

The latent Dirichlet allocation (LDA) is a more complex and commonly used method, to represent documents in an unsupervised manner, as probability distributions of hidden topics in a document [52], [58]. For the experiments described in this section, the LDA model is trained over the training set of DECODA following the standard hyper-parameters heuristic [52]. Consequently, $\alpha = \frac{50}{T}$, with T the number of topics, and $\beta = 0.01$. The number T has been previously investigated for this task in [20], [32], and is set to 25. More precisely, 10 runs of the $T = 25$ LDA model are concatenated to obtain a final vector of size $25 \times 10 = 250$, to alleviate any variation. It is important to notice that the input vector size in the case of a LDA inference of the UAD segmentation for a given document is 750. Indeed, each part is project in 10 differents $T = 25$ LDA models. Finally, 250 zeros are added for a quaternion-valued processing to build purely imaginary quaternions as described in section VI-B.

E. Models architectures

As depicted in Figure 4, the model architecture is made of two core components: 1) An autoencoder to extract the input features; 2) A convolutional classifier. Autoencoders are real-valued (AE), straightforward quaternion-based (QAE) or hybrid (R2H) and trained during 100 epochs with the Adam learning rate optimizer with a learning rate of 0.005, a momentum of 0.9, and a weight decay of 0.001. This number of epoch ensure that all models have converged to a minimal loss. Then, CNNs and QCNNs are trained using the features extracted with these autoencoders (AE-CNN, QAE-QCNN, and R2H-QCNN) with the same optimizer and hyperparameters during 10 epochs. The ReLU activation function is used across all the layers for both the autoencoders and the classifiers, except for the output layers that are based on the $\tanh$ function for the autoencoders, and the softmax function for the classifiers. No regularization techniques are employed. It is important to notice that the $\tanh$ function is a better fit than ReLU in the case of a bottleneck layer, due to its bounded nature. Indeed, the ReLU might produces very high and positive values, while projecting the negative space to zero, creating a poorly distributed internal representation to be used as input features. Conversely, the $\tanh$ function offers a smooth distribution of the features in a bounded interval, suiting particularly well to classifiers. At the end of the training, the results on the test dataset are saved with respect to the best results observed on the validation dataset. It is worth noticing that the results are averaged over five runs (5-folds), to alleviate any variation due to the parameters initialization.

1) AE & QAE: Baseline models involve straightforward autoencoders. Due to the quaternion representation of the basic QAE, the latter needs the input features to already represent quaternion numbers. Therefore, AE and QAE are trained using the UAD segmentation followed by binary, TF-IDF or LDA representations. The hidden layer of both autoencoders is fixed to a size of 100 neurons, that are equivalents to 25 quaternion numbers.

2) R2H: For a fair comparison, the hidden dimension of the R2H is also set to 100 values that are equivalent to 25 quater-

nion numbers. Due to the nature of the R2H, the input features representation does not require any particular segmentation, and can be defined in the real-valued space. The encoder part is initialized randomly following the Glorot criterion [40], while the quaternion-valued decoder parameters are initialized following the scheme described in [34], alongside with the quaternion version of the Glorot criterion.

3) CNN and QCNN: It is first important to note that CNNs and QCNNs are designed so the internal dimensions are equivalents. Consequently, a real-valued convolutional layer of 256 filter maps (FM) corresponds to a quaternion-valued convolutional layer of 64 quaternion filter maps. In the case of a QCNN, these 64 filter maps will represent $64 \times 4 = 256$ real values. Both CNNs and QCNNs contain three 1D convolutional layers of size 256 FM (64 for the QCNN) followed by two dense layers of size 512 and 8, corresponding to the number of classes. The last dense layer of the QCNN is real-valued to classify the document in the real-valued space.

4) Word2Vec baselines: For a fair comparison, the experiments incorporate results obtained with the same conditions but with traditional and state-of-the-art words representations, known as words to vectors (word2vec). In this extent, a continuous bag-of-words (CBoW) [59], a skip-gram [60], and a fastText [61] models are tested with DECODA. More precisely, these models assume that it is feasible to learn the meaning of a word, by looking at the words that tend to appear near it. The CBoW model trains each word against its context, while skip-gram trains each context against the word. Consequently, skip-gram is often prefered with small datasets, or corpora that contain a lot of rare words (*i.e.* not occurring frequently). FastText [61] is an extension of CBoW and skip-gram, that proposes to decompose the vocabulary with smaller entities, named n-grams. The intuition is that a very rare word might be better represented if we consider smaller pieces as fundamental units. Indeed, while the word *elephant* may appears once in the DECODA framework, the elements *el*, *le*, or *ph* might be more frequent and thus be used to better train the context.

The word2vec baselines rely on a training phase of 10 epochs, based on the non-segmented automatic transcriptions of the train set of the DECODA dataset. The context window is fixed to 10, while a number of negative samples of 20 is used to smooth the training. The fastText model relies on the skip-gram approach, due to better results observed over CBoW, and processes bi-gram units. Finally, the word embedding size is fixed to 100, to match the dimensions of our previously introduced techniques.

VII. Experiments

To evaluate the performances of the R2H, we first introduce the results observed with a simple QAE compared to the real-valued AE using the manual UAD segmentation (Section VII-A). Then the R2H is compared to the AE when no segmentation is applied (Section VII-B), before being compared to the QAE that uses the UAD segmentation (Section VII-C). In this extent, the performances of the quaternion-valued neural networks are investigated in all the testing conditions. Finally, a summary of all the results obtained so far on the DECODA framework with various neural architectures, including word2vec, are reported in Section VII-D.

A. AE-CNN vs QAE-CNN with UAD document segmentation

Due to the quaternion nature of the QAE and for fair comparison, both QAEs and AEs are trained with the UAD document segmentation to obtain a proper quaternion input representation. Table II shows the results observed for both QAE-QCNN and AE-CNN models.

TABLE II
Results observed with the UAD document segmentation for both QAE-QCNN and AE-CNN models on the DECODA dataset. Expressed percentages represent the accuracy. Results are from a 5-folds average.

Models	Type	Features	Dev. %	Test %	Params
AE-CNN	$\mathbb{R}$	Binary	80.4	78.9	4.2M
QAE-QCNN	$\mathbb{H}$	Binary	81.4	80.0	1.1M
AE-CNN	$\mathbb{R}$	TF-IDF	86.1	81.1	4.2M
QAE-QCNN	$\mathbb{H}$	TF-IDF	86.6	82.1	1.1M
AE-CNN	$\mathbb{R}$	LDA	86.1	83.4	4.0M
QAE-QCNN	$\mathbb{H}$	LDA	86.1	84.0	1.0M

It is worth underlying the important difference with respect to the number of neural parameters between the two models. This is easily explained by considering the content of the quaternion algebra. Indeed, for a fully-connected layer with $2,048$ input values and $2,048$ hidden units, a real-valued NN has $2,048^2 \approx 4.2$M parameters, while to maintain equal input and output dimensions the quaternion equivalent has 512 quaternions inputs and 512 quaternion hidden units. Therefore, the number of parameters for the quaternion-valued model is $512^2 \times 4 \approx 1$M. In fact, quaternion neural networks need four times less neural parameters to deal with the same internal signal dimension, compared to real-valued ones, leading to a lower number of parameters to manipulate and save. Such a complexity reduction turns out to produce better performances and results in a smaller memory footprint while saving models on budget memory systems. It can also have an impact on the generalization ability of the network, since reducing the number of parameters can act as a regularizer.

As expected, the binary representation of the textual document gives the worst results with both models, and the more complex LDA representation the best ones. Nonetheless, the QAE-QCNN consistently outperform the AE-CNN regardless of the input representation. Indeed, best tests of 80.0%, 82.1%, 84.0% are reported for the quaternionic solution with the binary, TF-IDF and LDA representations respectively, compared to 78.9%, 81.1%, 83.4% for the real-valued model. This represent an average gain of 1% in accuracy with four times less neural parameters.

B. AE-CNN vs R2H-QCNN without document segmentation

Maintaining high performance with a drastic reduction of the parameters to be estimated is an important objective, especially when limited resources are available as for the

application considered in this paper. Consequently, the R2H-QCNN is compared to the traditional AE-CNN without any segmentation of the textual documents. Input features of both models are real-valued and correspond to common document representations (Section VI).

TABLE III
RESULTS OBSERVED WITHOUT DOCUMENT SEGMENTATION FOR BOTH AE-CNN AND R2H-QCNN MODELS ON THE DECODA DATASET. EXPRESSED PERCENTAGES REPRESENT THE ACCURACY. RESULTS ARE FROM A 5-FOLDS AVERAGE.

Models	Type	Features	Dev. %	Test %	Params
AE-CNN	$\mathbb{R}$	Binary	80.4	78.5	4.2M
R2H-QCNN	$\mathbb{RH}$	Binary	82.4	82.6	1.1M
AE-CNN	$\mathbb{R}$	TF-IDF	82.9	80.5	4.2M
R2H-QCNN	$\mathbb{RH}$	TF-IDF	86.1	83.6	1.1M
AE-CNN	$\mathbb{R}$	LDA	88.9	84.0	3.7M
R2H-QCNN	$\mathbb{RH}$	LDA	91.6	86.3	0.9M

Table III first shows that the R2H-QCNN substantially outperforms its real-valued counterpart in all the testing conditions. Therefore, results of 82.6%, 83.6%, 86.3% are reported on the test set for the R2H-QCNN with the binary, TF-IDF and LDA representations, compared to 78.5%, 80.5%, 84.0% for the real-valued model. This is an average gain of 3.1% in accuracy with the same factor of reduction of the number of parameters equal to four. Consequently, the R2H-QCNN outperforms the AE-CNN, while processing real-valued input features and with less neural parameters.

C. R2H-QCNN vs QAE-QCNN

While previous experiments have demonstrated the superiority of quaternion-valued neural networks over real-valued ones on the DECODA classification task, Table IV compares the unsupervised R2H to a QAE using the manual UAD segmentation, to highlight the meaningfulness of the latent representation learned by the R2H, compared to a manually designed segmentation.

TABLE IV
RESULTS OBSERVED FOR THE QAE-QCNN WITH THE UAD SEGMENTATION AND THE R2H-QCNN WITHOUT DOCUMENT SEGMENTATION ON THE DECODA DATASET. EXPRESSED PERCENTAGES REPRESENT THE ACCURACY. RESULTS ARE FROM A 5-FOLDS AVERAGE.

Models	Type	Seg.	Features	Test %
QAE-QCNN	$\mathbb{H}$	UAD	Binary	80.0
R2H-QCNN	$\mathbb{RH}$	None	Binary	82.6
QAE-QCNN	$\mathbb{H}$	UAD	TF-IDF	82.1
R2H-QCNN	$\mathbb{RH}$	None	TF-IDF	83.6
QAE-QCNN	$\mathbb{H}$	UAD	LDA	84.0
R2H-QCNN	$\mathbb{RH}$	None	LDA	86.3

It is first important to mention that the R2H-QCNN always outperforms the QAE-QCNN. Indeed, test results of 82.6%, 83.6%, 86.3% are reported for the R2H-QCNN with the binary, TF-IDF and LDA representations, compared to 80.0%, 82.1%, 84.0% for QAE-QCNN and the UAD document segmentation. The average gain is of 2.1% with an almost identical number of free parameters. Indeed, both QCNN classifiers are equivalent and the R2H only has few parameters more due to the real-valued encoder. According to the results, the quaternionic representation learned in an unsupervised manner by the R2H is more meaningful than the one manually created by the user-agent document segmentation. More precisely, the R2H learns a multidimensional projection of the text input features that is optimal in the quaternion space, while the previously introduced UAD segmentation makes a strong assumption by manually providing a representation that might not be a perfect fit to the quaternion algebra, resulting in worst performances.

D. Overview of the DECODA results and discussions

Table V summarizes the recent accuracies observed on the test set of the DECODA corpus with various quaternion and real-valued neural network architectures. Three word2vec models (Section VI-E4) are used as baselines for a fair comparison with respect to modern approaches.

TABLE V
RESULTS OBSERVED FOR VARIOUS NEURAL NETWORKS ON THE TEST SET OF THE DECODA DATASET. EXPRESSED PERCENTAGES REPRESENT THE ACCURACY.

Models	Type	Features	Test %
MLP[32]	$\mathbb{R}$	LDA	83.4
QMLP[32]	$\mathbb{H}$	LDA	84.6
CBoW-CNN	$\mathbb{R}$	Binary	77.8
skipgram-CNN	$\mathbb{R}$	Binary	82.6
fastText-CNN	$\mathbb{R}$	Binary	83.4
AE[11]	$\mathbb{R}$	TF-IDF	81
DSAE[11]	$\mathbb{R}$	TF-IDF	82.0
DAE[31]	$\mathbb{R}$	TF-IDF	83.0
R2H-QCNN	$\mathbb{RH}$	TF-IDF	83.6
QAE-QCNN	$\mathbb{H}$	LDA	84
QDAE[31]	$\mathbb{H}$	LDA	85.2
R2H-QCNN	$\mathbb{RH}$	LDA	86.3
DNN[32]	$\mathbb{R}$	LDA	84.0
QDNN[32]	$\mathbb{H}$	LDA	85.2

As for previous experiments, QNNs consistently outperform equivalent real-valued neural networks in the DECODA framework. Input features representations also have a great impact on the final accuracies of all the tested models. Indeed, models fed with LDA features outperformed by far models fed with word frequency representations. This is explained by the nature of the LDA algorithm. LDA is much more complex than the straightforward statistical TF-IDF method. It provides information about latent relations between the words composing a corpus. In fact, LDA performs an unsupervised classification over each document by giving the probability of each topic with respect to the document.

It is worth underlying the lower standard deviation over the five runs observed with the R2H model compared to QAE and AE architectures. Indeed, the standard deviation of R2H-QCNN experiments with LDA input features is 0.004 compared to 0.005 and 0.008 for the QAE and AE respectively. In fact, all the models are very robust to the random initialization of the parameters, due to the nature of the input representation induced by the embedding transformation.

Then, it is important to notice that word2vec models are trained with the binary representation of the vocabulary, and manage to achieve significantly better results that the ones reported with traditional real-valued autoencoders on the same input features (Table III). This is easily explained by the fact that word2vec methods are an improved version of the mere autoencoder, to specifically take into consideration the meaning of a word. Then, it is well-known that the skip-gram variation is preferable over CBoW with small datasets, or with documents that contain a lot of rare words. Both conditions are validated with the DECODA dataset, and the initial assumption is confirmed with an average accuracy of 82.6% reported for the skip-gram, compared to 77.8% for CBoW. In the same manner, the more robust representation obtained by decomposing words into smaller n-grams allows fastText to reach an accuracy of 83.4%, while operating with binary input features, yielding the best accuracy observed so far with the latter representation. Interestingly, such performances are also superior to the ones observed with the TF-IDF input representation and real-valued classifiers (83.0%). Nevertheless, transformations obtained with LDA offer higher accuracies across all the tested models. An explanation comes from the size of the vocabulary considered. Indeed, the vocabulary size of DECODA is barely superior to $3,000$ words, with many meaningless words for the classification task. Consequently, word2vec approaches might suffer from both the lack of training samples, and the lack of diversity in the vocabulary.

Based on the LDA input features, the proposed R2H obtains the best observed accuracies so far on the DECODA dataset, outperforming the word2vec baselines, and the previously investigated manual and quaternion-valued document segmentations. Overall, the R2H provides a bridge to project any real-valued source to a quaternion-valued representation that fit perfectly to QNNs. Indeed, the conduced experiments have demonstrated that the latent multi-view representation learn by the R2H, allows a quaternion-valued classifier to performs better than real-valued representations, or manually adapted quaternion segmentations.

VIII. Conclusion

This work addresses an important issue raised by previous researches on quaternion-valued neural networks. Indeed, it provides an efficient and unsupervised tool based on a hybrid autoencoder called real-to-quaternion autoencoder (R2H), to extract a quaternion-valued representation from any real-valued input vector, for a better processing by quaternion-valued neural networks (QNN). This is an important step forward for researchers that are not able to benefit from the efficiency of QNNs due to a non-adapted input vector representation. Indeed, the R2H allows the use of any QNNs with any task, regardless of the input features domain. The conduced experiments on the classification task of the DECODA dataset, have shown that the R2H outperformed all the previously investigated models, even when compared to an adapted manual segmentation of textual document for quaternion neural networks. Finally, it is also demonstrated that QNNs always outperform equivalent real-valued NNs on this task with four times fewer parameters.

References

[1] R. Sarikaya, G. E. Hinton, and A. Deoras, "Application of deep belief networks for natural language understanding," *IEEE/ACM Transactions on Audio, Speech and Language Processing (TASLP)*, vol. 22, no. 4, pp. 778–784, 2014.

[2] I. V. Serban, A. Sordoni, Y. Bengio, A. Courville, and J. Pineau, "Building end-to-end dialogue systems using generative hierarchical neural network models," in *Thirtieth AAAI Conference on Artificial Intelligence*, 2016.

[3] Y.-N. Chen, D. Hakkani-Tür, G. Tür, J. Gao, and L. Deng, "End-to-end memory networks with knowledge carryover for multi-turn spoken language understanding." in *Interspeech*, 2016, pp. 3245–3249.

[4] P. Haghani, A. Narayanan, M. Bacchiani, G. Chuang, N. Gaur, P. Moreno, R. Prabhavalkar, Z. Qu, and A. Waters, "From audio to semantics: Approaches to end-to-end spoken language understanding," in *2018 IEEE Spoken Language Technology Workshop (SLT)*. IEEE, 2018, pp. 720–726.

[5] D. Serdyuk, Y. Wang, C. Fuegen, A. Kumar, B. Liu, and Y. Bengio, "Towards end-to-end spoken language understanding," in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2018, pp. 5754–5758.

[6] Y. Liu and S. Li, "Recognizing implicit discourse relations via repeated reading: Neural networks with multi-level attention," *arXiv preprint arXiv:1609.06380*, 2016.

[7] P. Li, W. Lam, L. Bing, W. Guo, and H. Li, "Cascaded attention based unsupervised information distillation for compressive summarization," in *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, 2017, pp. 2081–2090.

[8] J. Zhang, H. Luan, M. Sun, F. Zhai, J. Xu, M. Zhang, and Y. Liu, "Improving the transformer translation model with document-level context," *arXiv preprint arXiv:1810.03581*, 2018.

[9] G. Tur and R. De Mori, *Spoken language understanding: Systems for extracting semantic information from speech*. John Wiley & Sons, 2011.

[10] M. Morchid, R. Dufour, G. Linares, and Y. Hamadi, "Latent topic model based representations for a robust theme identification of highly imperfect automatic transcriptions," in *Computational Linguistics and Intelligent Text Processing*. Springer, 2015, pp. 596–605.

[11] K. Janod, M. Morchid, R. Dufour, G. Linares, and R. De Mori, "Deep stacked autoencoders for spoken language understanding," *ISCA INTERSPEECH*, vol. 1, p. 2, 2016.

[12] C. Xiong, V. Zhong, and R. Socher, "Dynamic coattention networks for question answering," *arXiv preprint arXiv:1611.01604*, 2016.

[13] N. Ryant, E. Bergelson, K. Church, A. Cristià, J. Du, S. Ganapathy, S. Khudanpur, D. Kowalski, M. Krishnamoorthy, R. Kulshreshta, M. Liberman, Y.-D. Lu, M. Maciejewski, F. Metze, J. Profant, L. Sun, Y. Tsao, and Z. Yu, "Enhancement and analysis of conversational speech: Jsalt 2017," *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 5154–5158, 2018.

[14] E. M. Hoey and K. H. Kendrick, "Conversation analysis," *Research methods in psycholinguistics: A practical guide*, pp. 151–173, 2017.

[15] R. Pappagari, J. Villalba, and N. Dehak, "Joint verification-identification in end-to-end multi-scale cnn framework for topic identification," in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2018, pp. 6199–6203.

[16] Y. Esteve, M. Bouallegue, C. Lailler, M. Morchid, R. Dufour, G. Linares, D. Matrouf, and R. De Mori, "Integration of word and semantic features for theme identification in telephone conversations," in *Natural Language Dialog Systems and Intelligent Assistants*. Springer, 2015, pp. 223–231.

[17] K. Janod, M. Morchid, R. Dufour, G. Linares, and R. De Mori, "Denoised bottleneck features from deep autoencoders for telephone conversation analysis," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 25, no. 9, pp. 1809–1820, 2017.

[18] J. Sun, W. Guo, Z. Chen, and Y. Song, "Topic detection in conversational telephone speech using cnn with multi-stream inputs," in *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2019, pp. 7285–7289.

[19] S. Mobin and B. Olshausen, "Auditory separation of a conversation from background via attentional gating," *arXiv preprint arXiv:1905.10751*, 2019.

[20] T. Parcollet, M. Morchid, P.-M. Bousquet, R. Dufour, G. Linarès, and R. De Mori, "Quaternion neural networks for spoken language understanding," in *Spoken Language Technology Workshop (SLT), 2016 IEEE*. IEEE, 2016, pp. 362–368.

[21] W. R. Hamilton, "Ii. on quaternions; or on a new system of imaginaries in algebra," *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science*, vol. 25, no. 163, pp. 10–13, 1844.

[22] S. J. Sangwine, "Fourier transforms of colour images using quaternion or hypercomplex, numbers," *Electronics letters*, vol. 32, no. 21, pp. 1979–1980, 1996.

[23] S.-C. Pei and C.-M. Cheng, "Color image processing by using binary quaternion-moment-preserving thresholding technique," *IEEE Transactions on Image Processing*, vol. 8, no. 5, pp. 614–628, 1999.

[24] N. A. Aspragathos and J. K. Dimitros, "A comparative study of three methods for robot kinematics," *Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on*, vol. 28, no. 2, pp. 135–145, 1998.

[25] P. Arena, L. Fortuna, L. Occhipinti, and M. G. Xibilia, "Neural networks for quaternion-valued function approximation," in *Circuits and Systems, 1994. ISCAS'94., 1994 IEEE International Symposium on*, vol. 6. IEEE, 1994, pp. 307–310.

[26] P. Arena, L. Fortuna, G. Muscato, and M. G. Xibilia, "Multilayer perceptrons to approximate quaternion valued functions," *Neural Networks*, vol. 10, no. 2, pp. 335–342, 1997.

[27] A. Hirose and S. Yoshida, "Generalization characteristics of complex-valued feedforward neural networks in relation to signal coherence," *IEEE Transactions on Neural Networks and learning systems*, vol. 23, no. 4, pp. 541–551, 2012.

[28] M. Tygert, J. Bruna, S. Chintala, Y. LeCun, S. Piantino, and A. Szlam, "A mathematical motivation for complex-valued convolutional networks," *Neural computation*, vol. 28, no. 5, pp. 815–825, 2016.

[29] I. Danihelka, G. Wayne, B. Uria, N. Kalchbrenner, and A. Graves, "Associative long short-term memory," *arXiv preprint arXiv:1602.03032*, 2016.

[30] S. Wisdom, T. Powers, J. Hershey, J. Le Roux, and L. Atlas, "Full-capacity unitary recurrent neural networks," in *Advances in Neural Information Processing Systems*, 2016, pp. 4880–4888.

[31] T. Parcollet, M. Morchid, and G. Linares, "Quaternion denoising encoder-decoder for theme identification of telephone conversations," *Proc. Interspeech 2017*, pp. 3325–3328, 2017.

[32] ——, "Deep quaternion neural networks for spoken language understanding," in *Automatic Speech Recognition and Understanding Workshop (ASRU), 2017 IEEE*. IEEE, 2017, pp. 504–511.

[33] C. J. Gaudet and A. S. Maida, "Deep quaternion networks," in *2018 International Joint Conference on Neural Networks (IJCNN)*. IEEE, 2018, pp. 1–8.

[34] T. Parcollet, Y. Zhang, M. Morchid, C. Trabelsi, G. Linarès, R. de Mori, and Y. Bengio, "Quaternion convolutional neural networks for end-to-end automatic speech recognition," in *Interspeech 2018, 19th Annual Conference of the International Speech Communication Association, Hyderabad, India, 2-6 September 2018.*, 2018, pp. 22–26. [Online]. Available: https://doi.org/10.21437/Interspeech.2018-1898

[35] T. Parcollet, M. Ravanelli, M. Morchid, G. Linarès, C. Trabelsi, R. D. Mori, and Y. Bengio, "Quaternion recurrent neural networks," in *International Conference on Learning Representations*, 2019. [Online]. Available: https://openreview.net/forum?id=ByMHvs0cFQ

[36] T. Parcollet, M. Morchid, G. Linarès, and R. De Mori, "Bidirectional quaternion long short-term memory recurrent neural networks for speech recognition," in *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2019, pp. 8519–8523.

[37] T. Isokawa, N. Matsui, and H. Nishimura, "Quaternionic neural networks: Fundamental properties and applications," *Complex-Valued Neural Networks: Utilizing High-Dimensional Parameters*, pp. 411–439, 2009.

[38] N. Matsui, T. Isokawa, H. Kusamichi, F. Peper, and H. Nishimura, "Quaternion neural network with geometrical operators," *Journal of Intelligent & Fuzzy Systems*, vol. 15, no. 3, 4, pp. 149–164, 2004.

[39] J. Diebel, "Representing attitude: Euler angles, unit quaternions, and rotation vectors," *Matrix*, vol. 58, no. 15-16, pp. 1–35, 2006.

[40] X. Glorot and Y. Bengio, "Understanding the difficulty of training deep feedforward neural networks," in *Proceedings of the thirteenth international conference on artificial intelligence and statistics*, 2010, pp. 249–256.

[41] V. Nair and G. E. Hinton, "Rectified linear units improve restricted boltzmann machines," in *Proceedings of the 27th international conference on machine learning (ICML-10)*, 2010, pp. 807–814.

[42] B. Karlik and A. V. Olgac, "Performance analysis of various activation functions in generalized mlp architectures of neural networks," *International Journal of Artificial Intelligence and Expert Systems*, vol. 1, no. 4, pp. 111–122, 2011.

[43] L. Trottier, P. Gigu, B. Chaib-draa *et al.*, "Parametric exponential linear unit for deep convolutional neural networks," in *2017 16th IEEE International Conference on Machine Learning and Applications (ICMLA)*. IEEE, 2017, pp. 207–214.

[44] S. De Leo and P. Rotelli, "Local hypercomplex analyticity," *arXiv preprint funct-an/9703002*, 1997.

[45] B. C. Ujang, C. Jahanchahi, C. C. Took, and D. Mandic, "Quaternion valued neural networks and nonlinear adaptive filters," 2010.

[46] B. C. Ujang, C. C. Took, and D. P. Mandic, "Quaternion-valued nonlinear adaptive filtering," *IEEE Transactions on Neural Networks*, vol. 22, no. 8, pp. 1193–1206, 2011.

[47] D. P. Mandic, C. Jahanchahi, and C. C. Took, "A quaternion gradient operator and its applications," *IEEE Signal Processing Letters*, vol. 18, no. 1, pp. 47–50, 2011.

[48] C. J. Willmott and K. Matsuura, "Advantages of the mean absolute error (mae) over the root mean square error (rmse) in assessing average model performance," *Climate research*, vol. 30, no. 1, pp. 79–82, 2005.

[49] T. Nitta, "A quaternary version of the back-propagation algorithm," in *Neural Networks, 1995. Proceedings., IEEE International Conference on*, vol. 5. IEEE, 1995, pp. 2753–2756.

[50] T. Isokawa, T. Kusakabe, N. Matsui, and F. Peper, "Quaternion neural network and its application," in *International Conference on Knowledge-Based and Intelligent Information and Engineering Systems*. Springer, 2003, pp. 318–324.

[51] C. Trabelsi, O. Bilaniuk, Y. Zhang, D. Serdyuk, S. Subramanian, J. F. Santos, S. Mehri, N. Rostamzadeh, Y. Bengio, and C. J. Pal, "Deep complex networks," *arXiv preprint arXiv:1705.09792*, 2017.

[52] D. M. Blei, A. Y. Ng, and M. I. Jordan, "Latent dirichlet allocation," *Journal of machine Learning research*, vol. 3, no. Jan, pp. 993–1022, 2003.

[53] F. Bechet, B. Maza, N. Bigouroux, T. Bazillon, M. El-Beze, R. De Mori, and E. Arbillot, "Decoda: a call-centre human-human spoken conversation corpus." in *LREC*, 2012, pp. 1343–1347.

[54] C. Lailler, A. Landeau, F. Béchet, Y. Estève, and P. Deléglise, "Enhancing the ratp-decoda corpus with linguistic annotations for performing a large range of nlp tasks." in *LREC*, 2016.

[55] G. Linares, P. Nocéra, D. Massonie, and D. Matrouf, "The lia speech recognition system: from 10xrt to 1xrt," in *Text, Speech and Dialogue*. Springer, 2007, pp. 302–308.

[56] V. Van Asch, "Macro-and micro-averaged evaluation measures [[basic draft]]," *Belgium: CLiPS*, 2013.

[57] S. Robertson, "Understanding inverse document frequency: on theoretical arguments for idf," *Journal of documentation*, vol. 60, no. 5, pp. 503–520, 2004.

[58] R. Krestel, P. Fankhauser, and W. Nejdl, "Latent dirichlet allocation for tag recommendation," in *Proceedings of the third ACM conference on Recommender systems*. ACM, 2009, pp. 61–68.

[59] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient estimation of word representations in vector space," *arXiv preprint arXiv:1301.3781*, 2013.

[60] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean, "Distributed representations of words and phrases and their compositionality," in *Advances in neural information processing systems*, 2013, pp. 3111–3119.

[61] A. Joulin, E. Grave, P. Bojanowski, and T. Mikolov, "Bag of tricks for efficient text classification," *arXiv preprint arXiv:1607.01759*, 2016.