2018 IEEE SmartWorld, Ubiquitous Intelligence & Computing, Advanced & Trusted Computing, Scalable Computing & Communications, Cloud & Big Data Computing, Internet of People and Smart City Innovations

Embedding-level attention and multi-scale convolutional neural networks for behaviour modelling

Aitor Almeida
MORElab
DeustoTech - Deusto Foundation
Bilbao, Spain
aitor.almeida@deusto.es

Gorka Azkune
MORElab
DeustoTech - University of Deusto
Bilbao, Spain
gorka.azkune@deusto.es

Aritz Bilbao
MORElab
DeustoTech - Deusto Foundation
Bilbao, Spain
aritzbilbao@deusto.es

***Abstract*—Understanding human behaviour is a central task in intelligent environments. Understanding what the user does and how she does it allows to build more reactive and smart environments. In this paper we present a new approach to inter-activity behaviour modelling. This approach is based on the use of multi-scale convolutional neural networks to detect n-grams in action sequences and a novel method of applying soft attention mechanisms at embedding level. The proposed architecture improves our previous architecture based on recurrent networks, obtaining better result predicting the users' actions.**

***Index Terms*—behaviour modelling, convolutional neural network, soft attention, embeddings, intelligent environments**

I. Introduction

City4Age [1] is a H2020 research and innovation project with the aim of enabling age-friendly cities. The project aims to create an innovative framework of ICT tools and services that can be deployed by European cities in order to enhance the early detection of risk related to frailty and MCI, as well as to provide personalized interventions that can help the elderly population to improve their daily life by promoting positive behaviour changes [2] [3]. The project has deployed an IoT infrastructure in several european cities [4], with the idea that SmartCities can collect personal data about their citizens' behaviours enhanced through the use of technologies in an unobtrusive and affordable way [5].

City4Age does not offer a single and invariable solution for SmartCities, providing instead a flexible framework composed by multiple methodologies and software/hardware modules that solve specific problems. This way, each city can select the pieces of the framework that fulfil better their own requirements and needs, creating their customized solution. As part of the tools created for the framework, we have developed a series of algorithms for activity recognition and behaviour modelling. The analysis of the users' behaviour is one of the central elements of the City4Age project. The recognized activities and the behaviour variations are then used to ascertain the frailty and MCI risks levels of the users. Once the risk has been detected, users receive meaningful and timely interventions in order to help them improve their behaviour. In the past, we have worked on creating single-user activity recognition algorithms for home environments [6] [7] [8], but in the case of City4Age, we have created algorithms that use a higher level of abstraction to model the user behaviour based on their actions and activities.

In this paper, we present an improved version of our inter-activity behaviour modelling algorithm. As previously defined in [29], the inter-activity behaviour describes how the user chains different activities (e.g. on Mondays after having breakfast, the user leaves the house to go to work, but in the weekends she goes to the main room), while the intra-activity behaviour describes how a single activity is performed by a user at different times (e.g., while the user is preparing dinner, sometimes she may gather all the ingredients before starting, while on other occasions, the user may take them as they are needed). The improved algorithm uses multi-scale convolutional neural networks (CNN) to detect the n-grams present in the input action sequences and predict the next action of the user. We also present a novel method of applying attention for sequence modelling. Instead of applying the inferred attention levels to the hidden states of the recurrent encoder, we apply them to the action embeddings, feeding those adjusted action embeddings to the multi-scale CNNs. Our evaluation shows that this approach can provide better results in the tested scenarios.

II. Related work

There are two main monitoring approaches for automatic human behaviour and activity evaluation, namely, vision- and sensor-based monitoring. For a review of vision-based approaches, [9] can be consulted. When approaching human behaviour and activity recognition in intelligent environments, sensor-based monitoring approaches are the most widely used solutions [10], as vision-based ones tend to generate privacy concerns among the users [11]. Sensor-based approaches are based on the use of emerging sensor network technologies for behaviour and activity monitoring. The generated sensor data from sensor-based monitoring are mainly time series of state changes and/or various parameter values that are usually

978-1-5386-9380-3/18/$31.00 ©2018 IEEE
DOI 10.1109/SmartWorld.2018.00103

processed through data fusion, probabilistic or statistical analysis methods and formal knowledge technologies for activity recognition. There are two main approaches for sensor-based behaviour and activity recognition in the literature: data- and knowledge-driven approaches.

User behaviour in intelligent environments builds on user activities to describe the conduct of the user. Modelling user behaviour entails an abstraction layer over activity recognition. Behaviour models describe how specific users perform activities and what activities comprise their daily living. User behaviour prediction is an important task in intelligent environments. It allows us to anticipate user needs and to detect variations in behaviour that can be related to health risks. In the Mavhome project [12], the authors created algorithms to predict the users' mobility patterns and their device usage. Their algorithms, based mainly on sequence matching, compression and Markov models [13], allowed the intelligent environments to adapt to the user needs. Other authors have used prediction methods to recognize the user activities in smart environments [14]. An analysis of the importance of prediction in intelligent environments can be found in [15]. Prediction has also been used in intelligent environments for the control of artificial illumination [16] using neuro-fuzzy systems or for the control of climate parameters on the basis of user behaviour [17]. A more in-depth analysis of the use of user behaviour prediction for comfort management in intelligent environments can be found in [18]. As explained in the following section, we identify two types of behaviours: intra-activity behaviour (which describes how the user performs activities) and inter-activity behaviour (which describes the actions and activity sequences that compose the user's daily life). To model and predict the inter-activity behaviour, we use action sequences, as this allows us to have a fine-grained description of the user conduct while abstracting the model from specific sensor technology.

To be able to work with a more flexible representation of the information in the intelligent environments, we map the raw sensor data to actions like proposed by Chen et al. [19]. Other authors have shown that actions are a good approach to model behaviours [20] [21]. Using actions to model behaviours has also been tackled in the domain of plan and goal recognition. Hoey et al. [22] use the same definition of actions (short and conscious muscular movements) to analyse the hand-washing process of patients with dementia. In their case, the mapping is done from a video to a set of actions. Krüger et al. [23] use actions to model activities using computational state-space models. In their case, the mapping is done from inertial measurements to actions, and the actions are grouped in different classes. Although different types of sensors (video by Hoey et al., inertial measurements by Krüger et al., and binary sensors in our evaluation) and mapping methods are used, the same concept is present, working on the action-space instead of the sensor-space. In our approach we use semantic embeddings to represent the actions. Embeddings have been used successfully in recent years in very varied tasks, such as knowledge path extraction [24] or sentiment analysis [25].

Multi-scale CNNs having been used by several authors for NLP related tasks. [26] studies the use of multi-scale CNNs for sentence classification, using the convolutional operation as we do to identify the n-grams present in the sentence. [28] analyses in-depth the multi-scale CNN architecture proposed in [26] to study the influence of the different factors and hyper-parameters in the accuracy of the architecture. [27] makes use of multi-scale CNNs in order to match medical questions and answers in Chinese.

III. Behaviour modelling network

To create a probabilistic model for behaviour prediction, we used a deep neural network architecture (see Figure 2). This model shares some characteristics with our previous approach [29] to user behaviour modelling. It also works on the action-space instead of the sensor-space, mapping specific sensor readings to user actions. The action-based approach was initially proposed in [19] and has the advantage of reducing the hypothesis space, as different sensor types may detect the same action. The proposed system is specifically tailored to detect inter-activity behaviour, as defined in [29](see Figure 1). The new approach presented in this paper offers two improvements over the previous one. First, instead of using recurrent neural networks (RNN) to model the sequence, we use multi-scale CNNs to identify the relevant n-grams in the sequence, validating that it improves the previous results. Secondly, we propose a novel way to apply attention: instead of applying attention to the encoder's hidden states as usual [32] [33] [34], we apply it to the action embeddings to identify the most relevant actions in the sequence. Those two contributions allowed us to significantly improve the results of our previous behaviour modelling system.

The behaviour modelling network is divided in four main sections:

- The *input module*, which takes the sequence of user actions as input and transforms it to a sequence of action embeddings. It is composed by the input layer and the embedding matrix in Figure 2.
- The *attention mechanism*, which evaluates the sequence to identify the most important actions in it and weights the action embeddings according to the action importance. It is composed by the Gated Recurrent Unit (GRU) encoder, the hyperbolic tangent (tanh) fully connected layer and the softmax fully connected layer in Figure 2. The result of the attention is applied to the output of the embedding matrix.
- The *multi-scale convolutional feature extractor*, which takes the adjusted action embeddings and detects different length n-grams, extracting the most relevant features of the sequence. It is composed by the CNNs and the 1-max pool layers. There are several of these convolution operations done in parallel, depending on the n-grams that the model wants to identify in the actions.
- The *prediction module*, which takes the identified features and predicts the probability that each individual action has to follow the input sequence. It is composed by two

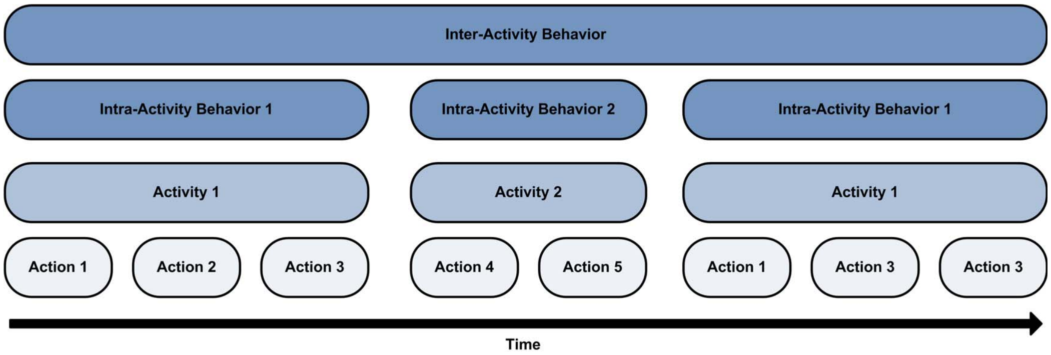

Fig. 1. User behaviour model as defined in [29]. The behaviour modelling network presented in the paper models the inter-activity behaviour.

Rectified Linear Unit (ReLU) fully connected layers and a final softmax fully connected layer in Figure 2.

A. The input module and the attention mechanism

The *input module* works similarly to the one used in [29]. The behaviour modelling network receives the user actions as one-hot-encodings and converts them to semantic embeddings. As validated in [29], using a more rich representation of the actions (i.e. the semantic embeddings) provides better results while predicting the user's next actions. In our model, we use the Word2Vec algorithm proposed by Mikolov et al. [30] to calculate the embeddings, a widely used embedding model inspired originally by the neural network language model developed by Bengio et al. [31]

Given a sequence of actions $S_{act} = [a_1, a_2, ..., a_{l_a}]$, where l_a is the sequence length and $a_i \in \Re^{d_a}$ indicates the action vector of the ith action in the sequence, we let $Context(a_i) = [a_{i-n}, \ldots, a_{i-1}, a_{i+1}, \ldots, a_{i+n}]$ be the context of a_i, where $2n$ is the length of the context window. We let $p(a_i|Context(a_i))$ be the probability of a_i to be in the i^{th} position of the sequence. The target of the model used to create the embeddings is to optimize the log maximum likelihood estimation (logMLE):

$$L_a(MLE) = \sum_{a_i \in S_{act}} \log p(a_i|Context(a_i)) \qquad (1)$$

In our model, we use the Word2Vec implementation in Gensim to calculate the embedding values for each action in the dataset. Gensim[1] is one of the most popular Python vector-space modelling libraries. We represent each action with a vector of 50 float values. Instead of providing the values directly to our model, we have included an embedding matrix layer as the input to the model. In this layer, we store the procedural information on how to transform an action ID to its embedding. Adding this layer allows us to train it with the rest of the model and, in this way, fine-tune the embedding values to the current task, improving the general accuracy of the model [29].

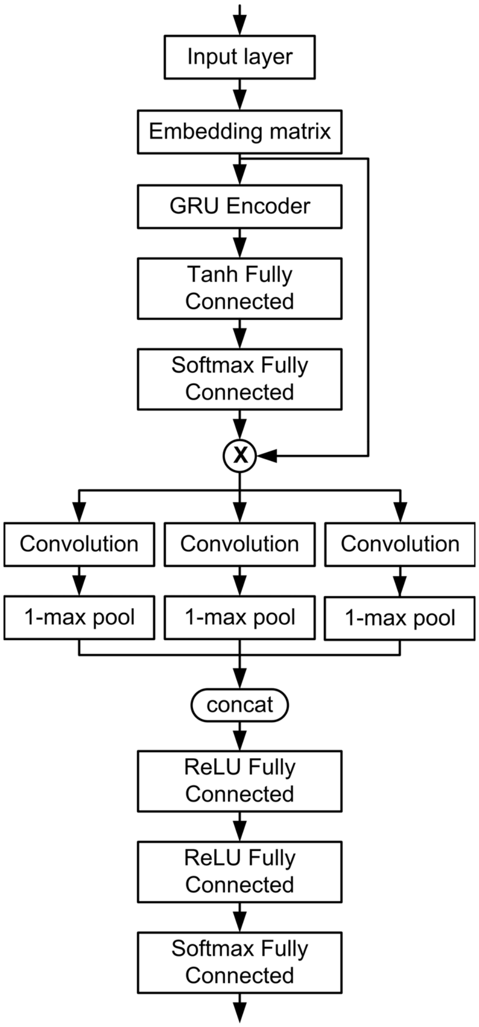

Fig. 2. Behaviour modelling network architecture. The input layer and the embedding matrix form the input module. The GRU encoder, tanh fully connected and softmax fully connected layers form the attention mechanism. The convolution and 1-max pool layers form the multi-scale convolutional feature extractor. Finally the two ReLU fully connected and the final softmax fully connected layers form the prediction module. The number of convolution layers can very depending on the specific task.

[1] https://radimrehurek.com/gensim/

After calculating the semantic embeddings we apply a soft *attention mechanism* to identify those actions that are more relevant to predict the next user action. This approach is similar to how the attention mechanism is applied in natural language processing to identify the most important words in a sentence [32] [33] [34], but with a variation regarding the network component to where the attention is applied. Instead of applying the attention to the hidden states generated by the encoder, we apply it to the calculated semantic embeddings. This new approach to applying attention has resulted in better results when predicting user actions, as explained in Section IV.

Action sequences S_{act} are composed by actions a_t with $t \in [1, T]$. Taking a sequence with actions $a_t, t \in [1, T]$, it is fed to the input module to obtain the semantic embeddings of those actions, using an embedding matrix A_e calculated previously with the Word2Vec algorithm. We use a GRU layer [35] like [34] to encode the sequence of action embeddings. This encoder extracts the contextual information of each action in the sequence, with the GRU layer reading the sequence from a_1 to a_T. The GRU has a number of units of 128.

$$\begin{aligned} x_t &= A_e a_t \\ h_t &= \overrightarrow{GRU}(x_t) \end{aligned} \tag{2}$$

The attention mechanism takes the GRU hidden states h_t as input and outputs a vector of weights $\alpha_t \in [0, 1]$ with the importance of each action a_t. To do this, first a fully connected layer with a unit size of 128 is used to get the hidden representation u_t of h_t:

$$u_t = tanh(W_a h_t + b_a) \tag{3}$$

With u_t calculated, a softmax function is used to compute the normalized importance weights (α_t) of each action:

$$\alpha_t = \frac{exp(u_t^\top u_a)}{\sum_t exp(u_t^\top u_a)} \tag{4}$$

Other authors use this weight vector to calculate a weighted sum of the hidden states of the encoder h_t. In our case, we used those weights to estimate the relative importance of each action embedding x_t for the prediction task, obtaining the attention adjusted embedding vectors A_{eadj} (5). As discussed in Section IV, this approach provides better results in the analyzed cases:

$$A_{eadj} = \alpha_t x_t \tag{5}$$

B. The multi-scale convolutional feature extractor and the prediction module

After obtaining the adjusted embedding vectors A_{eadj} in (5), we use a multi-scale CNN architecture as *feature extractor*. This architecture is composed by several parallel CNNs (see Figure 2). Each of these networks perform a 1D convolution, each one with a different kernel size, in order to identify the different action n-grams in the sequence. The sequences are represented by adjusted action embedding sequences, having a fixed length: $A_{eadj} = \{ae_1, ..., ae_{lc}\}$. The dimensionality of the action vectors is represented by d_{ae} and each element being a real number, $ae_i \in \Re^{d_{ae}}$. After calculating the adjusted action embedding vectors, each action sequence is a matrix $A_{eadj} \in \Re^{l_{ae} \times d_{ae}}$. Taking the sequence of adjusted action embeddings, the convolution can be represented as:

$$O_j = f(W_j \circ [ae_1, ..., ae_{l_{ae}-s+1}] + b) \tag{6}$$

$O_j \in \Re^{l_{ae}-s+1}$ is the result of the convolution. $W_j \in \Re^{l \times d}$ and b are the parameters that are being trained. $f()$ is the activation function for the convolution, which in our case is a ReLU activation [36]. Finally, $W \circ A_{eadj}$ represents the element-wise multiplication of the elements. Being the number of filter maps d_o, the output of the convolution is $O = [O_1, ..., O_{d_o}] \in \Re^{(l_{ae}-s+1) \times d_o}$. In our model we use 200 filters for each convolution layer and different kernel sizes depending on the target n-grams: $n_{size} \times d_{ae}$. After each convolution we apply a 1-max pooling layer [37], which filters the extracted features by selecting the maximum value of each filter to reduce the dimensionality. We finally concatenate the results of each parallel 1-max pooling layer and flatten them in order to fed it to the prediction module.

The *prediction module* takes the extracted features and uses them to predict the user's next action. It is composed by three fully connected layers. The first two layers (f_{re}) use ReLU activations(and have a unit size of 512), in order to model the abstract representation of the next action.

$$f_{re} = relu(WX + b) \tag{7}$$

The final fully connected layer uses a softmax activation (see (4)) to predict the probabilities of the next action. This layer has a unit size equal to the number of distinct actions that are modelled in the system. It provides a vector with the probability of each action being the next one that the user is going to execute.

IV. Validation

A. Experimental setup

To evaluate the proposed model, we used the dataset[2] published by Kasteren et al. [39]. This dataset has been selected because it is widely used in the activity recognition and intelligent environments literature. This allows other researchers working in both areas to better compare the results of this paper with their own work. The dataset is the result of monitoring a 26-year old man in a three-room apartment where 14 binary sensors were installed. Those sensors were installed in locations such as doors, cupboards, refrigerator, freezer or toilet. Sensor data for 28 days was collected for a total of 2120 sensor events and 245 activity instances. The annotated activities were: 'LeaveHouse', 'UseToilet', 'TakeShower', 'GoToBed', 'PrepareBreakfast', 'PrepareDinner' and 'GetDrink'. In this specific case the sensors were mapped one to one to *actions*, resulting in the following set of *actions*: 'UseDishwasher',

[2] https://sites.google.com/site/tim0306/datasets

'OpenPansCupboard', 'ToiletFlush', 'UseHallBedroomDoor', 'OpenPlatesCupboard', 'OpenCupsCupboard', 'OpenFridge', 'UseMicrowave', 'UseHallBathroomDoor', 'UseWashingmachine', 'UseHallToiletDoor', 'OpenFreezer', 'OpenGroceriesCupboard' and 'UseFrontdoor'.

For the training process, the dataset was split into a training set (80% of the dataset) and a validation set (20% of the dataset) of continuous days. These sets are composed by the raw sensor data provided by Kasteren et al. In order to make the training process more streamlined, we apply the sensor to action mappings off-line. This allows us to train the deep neural model faster while still having the raw sensor data as the input. To do the training we use n actions as the input (as described in the sequence length experiments) to predict the next action (see subsection IV-B for a description on how accuracy is evaluated). That is, the training examples are the sequences of actions and the label is the next action that follows that sequence, being a supervised learning problem. The proposed architectures have been implemented using Keras[3] and executed using TensorFlow[4] as the back-end. Each of the experiments has been trained for 1000 epochs, with a batch size of 128, using Categorical Cross Entropy as the loss function and Adam [38] as the optimizer. After the 1000 epochs we selected the best model using the validation accuracy as the fitness metric. The *action* embeddings were calculated using the full training set extracted from the Kasteren dataset and using the word2vec [30] algorithm and the embedding layer was configured as trainable.

To validate the results of the system, we performed the following experiments (a summary of the experiments can be found in Table I):

- The best results of our previous approach [29] using RNNs (*A3*, *S2* and *S3*).
- An architecture using only multi-scale CNNs without any attention mechanism. In these experiments we varied the length of the n-grams that were detected by the CNNs, using n-grams with length 2,3,4 and 5 in *M1* and 3,4 and 5 in *M2*. Other variations of n-gram size have been tested, but the obtained results were not competitive enough to be included.
- An architecture using only the input module, attention mechanism and prediction module (without the multi-scale convolutional feature extractor). In this case the attention mechanism is applied as usually to the hidden states of the GRU encoder instead of the embeddings (*M3*). This is the traditional approach using attention for sequence modelling.
- The complete architecture, but using the traditional approach of applying attention to the hidden states of the GRU encoder. We then feed those values to the multi-scale CNNs (*M4*). Take into account that the traditional approach is the one used in M3, without using the multi-scale CNNs. We have included this experiment for completeness' sake.
- The complete architecture proposed in this paper, with the novel approach of applying the attention to the embedding values. In these experiments we have varied the size of the GRU and tanh activated fully connected layers, using the embedding size 50 in *M5* and 128 in *M6*. Whenever the attention layer is used in the rest of the experiments, the size of the GRU and tanh activated fully connected layers is also 128.

TABLE I
EXPERIMENT CONFIGURATION SUMMARY

ID	Description
A3, S2 & S3	Previous approach. See [29] for a description of the configurations.
M1	Multi-scale CNN without attention. N-gram lenghts: 2,3,4 & 5.
M2	Multi-scale CNN without attention. N-gram lenghts: 3,4 & 5.
M3	Input module + attention mechanism + prediction module. Without multi-scale CNNs.
M4	Complete architecture. Attention applied to the recurrent hidden states.
M5	Proposed architecture. Attention layer's size: 50.
M6	Proposed architecture. Attention layer's size: 128.

B. Metrics

To validate the predicting capabilities of the proposed model we have evaluated its performance using the top-k accuracy. The top-k (acc_at_k) accuracy is a standard metric in different prediction and modelling tasks, and is defined as:

$$acc\_at\_k = \frac{1}{N}\sum_{i=1}^{N} b[a_i \in C_i^k] \tag{8}$$

Where a_i is the expected action and C_i^k is the set of the top k predicted actions. $b[.] \rightarrow \{0, 1\}$ represents the scoring function, when the condition in the first part is true, the function value is 1, otherwise, the value is 0. In our case, if the ground-truth action is in the set of k predicted actions, the function value is 1. To evaluate our models we provide the accuracy for k values of 1, 2, 3, 4 and 5.

C. Results and discussion

Table II shows the results of the performed experiments. As can be seen, the proposed architecture (M5 and M6) improves the results obtained with our previous approach using RNNs (A3, S2 and S3), except for a single prediction, where the results are tied. Having the attention layers' size tied to the embeddings' size (M5) does not improve the results when comparing it to an arbitrary size (M6). Embedding level soft attention (M5 and M6) considerably improves the performance of multi-scale CNNs (M1 and M2). When comparing our approach to applying attention at embedding level and combining

[3] https://github.com/fchollet/keras

[4] https://www.tensorflow.org/

it with multi-scale CNNs, we can see that it offers better results for behaviour modelling against the traditional approach of applying it to the hidden states of the recurrent encoder (M3). M4 shows that combining the traditional approach with multi-scale CNNs is not a viable option, producing the worst results.

TABLE II
EXPERIMENT RESULTS: ACCURACY FOR DIFFERENT NUMBER OF PREDICTIONS

ID	acc_at_1	acc_at_2	acc_at_3	acc_at_4	acc_at_5
A3	**0.4744**	0.6282	0.7179	0.7905	0.8589
S2	0.4255	0.6255	0.7021	0.8085	0.8382
S3	0.4658	0.6452	0.7264	0.7948	0.8504
M1	0.4358	0.6324	0.7264	0.7905	0.8418
M2	0.4230	0.6068	0.7094	0.7692	0.8333
M3	0.4487	0.6452	0.7307	0.7905	**0.8632**
M4	0.3931	0.5726	0.6837	0.7649	0.8076
M5	**0.4744**	**0.6624**	0.735	0.8034	0.8589
M6	**0.47**	0.6538	**0.7436**	**0.8162**	**0.8632**

Regarding the better results using multi-scale CNNs than RNNs for behaviour modelling, we expect them to be related on how the multi-scale CNNs model the relations between the actions. With the multi-scale CNNs we are able to model relations between groups of actions of different lengths (in our specific case, n-grams with lengths of 2, 3, 4 and 5). This allows the model to be more flexible and to model different types of relations between the actions, while the RNNs would converge into an specific relation type. We also suspect that the relevant relations to predict the next action have a short duration, not allowing the RNNs to take advantage of the long term memory and the longer sequences. This also can explain why our novel way to applying attention offers better results in the studied cases. Being the n-grams the focal point for the predictions, it is important for the multi-scale CNNs to receive the action vectors as input. By applying the attention to the GRU Encoder hidden states h_t and feeding them to the multi-scale CNNs, all individual action information is lost and no n-grams can be identified. In the other hand, adjusting the action vectors x_t with the calculated importance weights α_t allows to maintain the individual action information while also using the attention mechanism.

We believe that the results presented in this paper validate our approach of combining the use of multi-scale CNNs with attention mechanisms in order to model the user behaviour. It also validates our novel approach to applying the attention mechanisms to the embedding vectors instead of the hidden states of the recurrent encoder.

V. CONCLUSION AND FUTURE WORK

In this paper we have presented an improved approach to inter-activity behaviour modelling. This approach improves the results of our previously presented behaviour modelling architecture [29]. While our previous approach was based on RNNs, this new model combines the usage of multi-scale convolutional neural networks and soft-attention mechanisms to predict the user's actions. We also present a novel approach to how to apply the soft attention mechanism when combining it with multi-scale CNNs. Instead of adjusting the recurrent encoder's hidden states with the calculated attention percentages, we use them to adjust the embedding vectors that are fed to the multi-scale CNNs. As validated with the performed experiments using a widely-used dataset in intelligent environments, our new approach provided better results than both using multi-scale CNNs without attention or using the traditional approach to attention.

As future work we would like to tackle two problems. First we would like to include temporal information to obtain better predictions. Intuitively, the actions that the users perform during different periods of the day should be more distinguishable (e.g. preparing breakfast versus preparing dinner, or taking usually a shower during the morning versus taking it before going to bed). We expect that including the temporal information in the model will lead to improved results. Secondly we would like to improve the model by including better internal representations both of the inputs and the predictions. We plan to do this by using the action embedding vectors as a prediction target. We expect that the improved representation will offer better result by improving the expressiveness of the network.

ACKNOWLEDGMENTS

This work has been supported by the European Commission under the City4Age project grant agreement (number 689731). Aritz Bilbao's work was funded by the Basque Government's Department of Education's predoctoral grant. We gratefully acknowledge the support of NVIDIA Corporation with the donation of the Titan X Pascal GPU used for this research.

REFERENCES

[1] Paolini, P., Di Blas, N., Copelli, S., & Mercalli, F. (2016, September). City4Age: Smart cities for health prevention. In Smart Cities Conference (ISC2), 2016 IEEE International (pp. 1-4). IEEE.

[2] Almeida, A., Fiore, A., Mainetti, L., Mulero, R., Patrono, L., & Rametta, P. (2017). An IoT-Aware Architecture for Collecting and Managing Data Related to Elderly Behavior. Wireless Communications and Mobile Computing, 2017.

[3] di Blas, N., Paolini, P., & Plotti, G. (2017, September). Combining IOT, open data and messaging for prevention of MCI/frailty. In Software, Telecommunications and Computer Networks (SoftCOM), 2017 25th International Conference on (pp. 1-5). IEEE.

[4] Mulero, R., Almeida, A., Azkune, G., Jiménez, P. A., Waldmeyer, M. T. A., Castrillo, M. P., Patrono, L., Rametta, P. & Sergi, I. (2018). An IoT-aware Approach for Elderly-Friendly Cities. IEEE Access.

[5] Mulero, R. Urosevic, V., Almeida, A., Tatsiopoulos, C.. (2018). Towards ambient assisted cities using linked data and data analysis. Journal of Ambient Intelligence and Humanized Computing. p. 1-19.

[6] Azkune, G., Almeida, A., López-de-Ipiña, D., & Chen, L. (2015). Extending knowledge-driven activity models through data-driven learning techniques. Expert Systems with Applications, 42(6), 3115-3128.

[7] Azkune, G., Almeida, A., López-de-Ipiña, D., & Chen, L. (2015). Combining users activity survey and simulators to evaluate human activity recognition systems. Sensors, 15(4), 8192-8213.

[8] Bilbao, A., Almeida, A., & López-de-Ipiña, D. (2016). Promotion of active ageing combining sensor and social network data. Journal of biomedical informatics, 64, 108-115.

[9] Weinland, D., Ronfard, R., & Boyer, E. (2011). A survey of vision-based methods for action representation, segmentation and recognition. Computer vision and image understanding, 115(2), 224-241.

[10] Chen, L., Hoey, J., Nugent, C. D., Cook, D. J., & Yu, Z. (2012). Sensor-based activity recognition. IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews), 42(6), 790-808.

[11] Yilmaz, A., Javed, O., & Shah, M. (2006). Object tracking: A survey. Acm computing surveys (CSUR), 38(4), 13.

[12] Das, S. K., Cook, D. J., Battacharya, A., Heierman, E. O., & Lin, T. Y. (2002). The role of prediction algorithms in the MavHome smart home architecture. IEEE Wireless Communications, 9(6), 77-84.

[13] Cook, D. J., Youngblood, M., Heierman, E. O., Gopalratnam, K., Rao, S., Litvin, A., & Khawaja, F. (2003, March). MavHome: An agent-based smart home. In Pervasive Computing and Communications, 2003.(PerCom 2003). Proceedings of the First IEEE International Conference on (pp. 521-524). IEEE.

[14] Fatima, I., Fahim, M., Lee, Y. K., & Lee, S. (2013). A unified framework for activity recognition-based behavior analysis and action prediction in smart homes. Sensors, 13(2), 2682-2699.

[15] Cook, D. J., & Das, S. K. (2007). How smart are our environments? An updated look at the state of the art. Pervasive and mobile computing, 3(2), 53-73.

[16] Kurian, C. P., Kuriachan, S., Bhat, J., & Aithal, R. S. (2005). An adaptive neuro-fuzzy model for the prediction and control of light in integrated lighting schemes. Lighting Research & Technology, 37(4), 343-351.

[17] Morel, N., Bauer, M., El-Khoury, M., & Krauss, J. (2001). Neurobat, a predictive and adaptive heating control system using artificial neural networks. International Journal of solar energy, 21(2-3), 161-201.

[18] Dounis, A. I., & Caraiscos, C. (2009). Advanced control systems engineering for energy and comfort management in a building environmentA review. Renewable and Sustainable Energy Reviews, 13(6-7), 1246-1261.

[19] Chen, L.; Nugent, C.D.; Mulvenna, M.; Finlay, D.; Hong, X.; Poland, M. A logical framework for behaviour reasoning and assistance in a smart home. *Int. J. Assist. Robot. Mechatron.* 2008, 9, 20–34.

[20] Schank, R. C. (1983). Dynamic memory: A theory of reminding and learning in computers and people. Cambridge university press.

[21] Schank, R.C. Abelson, R.P. (1977) Scripts, Plans, Goals and Understanding, an Inquiry into Human Knowledge Structures. Lawrence Erlbaum Associates.

[22] Hoey, J., Poupart, P., von Bertoldi, A., Craig, T., Boutilier, C., & Mihailidis, A. (2010). Automated handwashing assistance for persons with dementia using video and a partially observable Markov decision process. Computer Vision and Image Understanding, 114(5), 503-519.

[23] Krüger, F., Nyolt, M., Yordanova, K., Hein, A., & Kirste, T. (2014). Computational state space models for activity and intention recognition. A feasibility study. PloS one, 9(11), e109381.

[24] Wang, Z., Zhang, J., Feng, J., & Chen, Z. (2014, July). Knowledge Graph Embedding by Translating on Hyperplanes. In AAAI (Vol. 14, pp. 1112-1119).

[25] dos Santos, C., & Gatti, M. (2014). Deep convolutional neural networks for sentiment analysis of short texts. In Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers (pp. 69-78).

[26] Kim, Y. (2014). Convolutional neural networks for sentence classification. arXiv preprint arXiv:1408.5882.

[27] Zhang, S., Zhang, X., Wang, H., Cheng, J., Li, P., & Ding, Z. (2017). Chinese Medical Question Answer Matching Using End-to-End Character-Level Multi-Scale CNNs. Applied Sciences, 7(8), 767.

[28] Zhang, Y., & Wallace, B. (2015). A sensitivity analysis of (and practitioners' guide to) convolutional neural networks for sentence classification. arXiv preprint arXiv:1510.03820.

[29] Almeida, A., & Azkune, G. (2018). Predicting Human Behaviour with Recurrent Neural Networks. Applied Sciences, 8(2), 305.

[30] Mikolov, T.; Sutskever, I.; Chen, K.; Corrado, G.S.; Dean, J. Distributed representations of words and phrases and their compositionality. In *Advances in Neural Information Processing Systems*; The MIT Press: Cambridge, MA, USA, 2013; pp. 3111–3119.

[31] Bengio, Y.; Ducharme, R.; Vincent, P.; Jauvin, C. A neural probabilistic language model. *J. Mach. Learn. Res.* 2003, 3, 1137–1155.

[32] Hermann, K. M., Kocisky, T., Grefenstette, E., Espeholt, L., Kay, W., Suleyman, M., & Blunsom, P. (2015). Teaching machines to read and comprehend. In Advances in Neural Information Processing Systems (pp. 1693-1701).

[33] Lin, Z., Feng, M., Santos, C. N. D., Yu, M., Xiang, B., Zhou, B., & Bengio, Y. (2017). A structured self-attentive sentence embedding. arXiv preprint arXiv:1703.03130.

[34] Yang, Z., Yang, D., Dyer, C., He, X., Smola, A., & Hovy, E. (2016). Hierarchical attention networks for document classification. In Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (pp. 1480-1489).

[35] Bahdanau, D., Cho, K., & Bengio, Y. (2014). Neural machine translation by jointly learning to align and translate. arXiv preprint arXiv:1409.0473.

[36] Glorot, X.; Bordes, A.; Bengio, Y. Deep Sparse Rectifier Neural Networks. In Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics, PMLR, Ft. Lauderdale, FL, USA, 11–13 April 2011; Volome 15, No. 106, p. 275.

[37] Boureau, Y. L., Ponce, J., & LeCun, Y. (2010). A theoretical analysis of feature pooling in visual recognition. In Proceedings of the 27th international conference on machine learning (ICML-10) (pp. 111-118).

[38] Kingma, D., & Ba, J. (2014). Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980.

[39] Van Kasteren, T., Noulas, A., Englebienne, G., & Krse, B. (2008, September). Accurate activity recognition in a home setting. In Proceedings of the 10th international conference on Ubiquitous computing (pp. 1-9). ACM.