



UPDATING BAYESIAN NETWORKS USING CROWDS

H. Sarma^{*‡}; A.B. Samaddar^{*}; R. Porzel[‡]; J.D. Smeddinck[‡]; R. Malaka[‡]

Abstract: Physical exercise instruction sheets are difficult to understand. In general, considerable information is hidden in these types of instruction sheets, which also makes them difficult for machines to understand. Major missing information types include the source and destination location of a human movement. Here we present a Bayesian network to extract the implicit or missing information from typical exercise instruction sheets. We proposed two different kind of Bayesian networks which consists of three and four variables respectively. The network with three variable are designed to for single exercise instruction with single action or pose and the other one designed for single or multiple sentence with two actions or poses. The conditional probability table (CPT) is the backbone of the Bayesian network. At the start, the CPT is updated from our physical exercise instruction sheet corpus (PEISC). Keeping the *Action* and *Bodypart* fixed, we have developed our CPT using a unique approach, i.e., crowdsourcing, where we have developed a CPT update system using 13 different exercises consisting of 44 different exercise videos. Using this system based on the rating of a participant of the video the specific variable of that CPT is updated automatically in the Bayesian network. We also updated the *Action* variable, which consists of 14 different values (action verbs) using crowdsourcing with a human computation approach.

Key words: *CPT, Bayesian network, crowdsourcing, human computation, exercise instructions, collective intelligence, automatic update.*

Received: September 9, 2016

DOI: 10.14311/NNW.2017.27.028

Revised and accepted: October 21, 2017

1. Introduction

Physical exercises are an important part of many peoples' daily activities and are performed using many different instruction modalities. Physical exercise instruction sheets are one of the major exercise mediums, exercise instruction mediums.

^{*}Himangshu Sarma – Corresponding author, Arun Baran Samaddar; Computer Science & Engineering, National Institute of Technology Sikkim, India, E-mail: himangshu.tezu@gmail.com, absamaddar@yahoo.com

[†]Jan D. Smeddinck; International Computer Science Institute, Berkeley, USA, E-mail: jandavid@icsi.berkeley.edu

[‡]Himangshu Sarma, Robert Porzel, Jan D. Smeddinck, Rainer Malaka; Digital Media Research Lab, TZI, University of Bremen, Germany, E-mail: himangshu.tezu@gmail.com, porzel@tzi.de, smeddinck@tzi.de, malaka@tzi.de

However, these instruction sheets are often difficult to interpret and the field has seen surprisingly little adoption of digital technology. Developing a system for automatically generating animations from textual instructions has great potential to improve the current state of exercise instruction modalities. Formally described, well-formed and reproduced exercise instructions through animations would help individuals who struggle with text-based instructions and they would be of great use beyond the immediate application in exercise instructions. For example, the same system design can be used to generate movement-pattern executions for training robots. Instruction sheets contain substantial amounts of implicit information that is comparatively easy for humans to understand but can be a very complex challenge for automated machine interpretation. Natural language understanding has been the subject of a large array of research efforts, but remains an open problem. While most work has focused on understanding declarative utterances and interrogatives, instructions and the ensuing imperative forms have received less attention. Different syntactic and semantic parsers are typically used to understand languages, but it is impossible to extract implicit information from the sentences or instructions using typical, non-contextual parsers. The human brain succeeds at this challenge by filling the implicit gaps through contextualization based on experience and simulation.

If an automated language understanding system is tasked to understand the simple exercise instruction “*Lift your left arm.*”, it can only extract the *Action* and *Bodypart*. A human can also understand the implicit information, such as from where and to where the arm should be lifted. Therefore, we have developed a Bayesian network to extract the implicit information with the support of human computation for validation and refinement. The conditional probability table (CPT) is the main part of the Bayesian network. It consists of some discrete random variables, where the probability of a variable is calculated with the help of the other variables [17].

Bayesian networks are also known as Bayes or belief networks. They follow a probabilistic directed acyclic graph (DAG) model [12, 18] and typically consist of the following parts [13]:

- a DAG;
- a CPT;
- variables and directed edges between variables;
- every variable has a mutually exclusive state.

Bayesian networks (BN) are frequently employed to model uncertainties into decision-making systems, one popular example being action planning and execution in robotics. In a BN, the CPT contains the critical transition probabilities, while the network structure models domain elements to be considered. In order to form reliable and adequate CPTs, researchers usually apply a large dataset to analyze the domain in which they want to build the Bayesian network. However, it is not easy to compile sufficiently large datasets for every domain and it can also be very expensive. Therefore, in this article we present a novel way to build a strong CPT for a BN without relying on large datasets.

We build our CPT using a human computation approach with the help of crowds who participated in our study. To update our CPT, we have developed

a survey system consisting of 13 different exercises involving 13 body parts. The survey consists of 44 different videos, where every exercise is represented through three of four candidate execution rendered videos. Using crowds, we build a strong CPT with this system to obtain accurate implicit information in the same way an individual human would do, based on the physical exercise instruction sheets discussed in Section 5. As mentioned above, the values of our *Action* variable consist of action verbs, but it is impossible to build a Bayesian network with all possible action verbs, so we have built it in a novel way using crowdsourcing as detailed in Section 4.

In our analysis we provide first insights regarding the feasibility of such an approach to improving CPTs in the context of BNs that inform exercise execution renderings.

2. Background

A considerable body of research exists that illustrates approaches to employing BNs for the design of intelligent systems. The ALARM (A Logical Alarm Reduction Mechanism) network is one of the most famous Bayes networks. It is used to send a text message advising of a possible problem [2]. The Coma or Cancer network [7] and the Asia network [15] are some other popular and well-known Bayes networks. As we discussed earlier, in these well-known networks the researchers developed the CPT by analyzing a large dataset. We did not find a single reported Bayesian network that was developed with the help of crowds.

A number of commercial and free academic tools exist for designing BNs that researchers have used to develop their networks. In Tab. I, we list a selection of popular commercial and free tools.

Serial No.	Tools	Commercial	Free
1	AgenaRisk [3]	✓	
2	SamIam [9]		✓
3	GeNIe [10]		✓
4	SMILE [11]		✓
5	Analytica [5]	✓	
6	HUGIN [1]	✓	
7	MSBNx [14]		✓
8	BayesiaLab [6]	✓	
9	Javabayes [8]		✓
10	Netica [16]	✓	
11	PULCINELLA [19]		✓

Tab. I List of tools used to design Bayesian networks.

3. A Bayesian network for physical exercise executions

To extract the implicit information from physical exercise instruction sheets, we developed a BN CPT update system. With the help of a physical exercise instruction sheet corpus (PEISC) [20–22], we first develop the CPT of the Bayesian network. In our application domain, the BNs are designed to extract implicit location in exercise descriptions. We have developed two different Bayesian networks. One Bayesian network is designed for single sentences with single actions, e.g., “*Lift your left arm.*” The second is for multiple actions with single or multiple sentences, e.g., “*Lift your left arm. Move it to your right elbow.*” Both Bayesian networks were designed using *SamIam* (sensitivity analysis, modeling, inference, and more) [4,9], as shown in Figs. 1 and 2.

3.1 Bayesian network for single action

As mentioned earlier, this belief network is designed for a single sentence and models single action executions, as shown in Fig. 1. This network consists of three different variables:

- *Action*: An action variable is designed for action verbs of the instructions. Currently, the variable can take one of 14 common values, i.e., “lift”, “bring”, “bend”, “push”, “keep”, “bent”, “tilt”, “rest”, “lower”, “stretch”, “pull”, “sit”, “reach”, and “raise”.
- *Bodypart*: This Bayesian model is mainly designed for exercise instructions. Human body parts are always involved in an exercise. Therefore, in this variable we include most commonly mentioned human body parts as values, i.e., “l_hip”, “r_hip”, “l_knee”, “r_knee”, “l_ankle”, “r_ankle”, “l_shoulder”, “r_shoulder”, “l_elbow”, “r_elbow”, “l_wrist”, “r_wrist”, “HumanoidRoot”, and “skullbase”.
- *Location*: The “Location” variable is designed for the target or destination locations of the involved body parts, which is used to find the implicit target locations of the exercise. There are 22 values of the “Location” variable as shown in Fig. 3, i.e., from “location1” to “location22”.

At the start, the CPT of the network has been built from the data of the PEISC. The implicit *Location* for the exercise instruction *Lift your left arm* is shown in Fig. 1.

3.2 Bayesian network for multiple actions

This belief network is designed mainly for multiple actions expressed in single or multiple sentences. It models four variables, augmenting the selection of the single action network as described above by one extra variable called “*Action2DestLoc*”. This variable has the same values as the variable *Location*. In this network, the *Location* variable is the destination of the first action and source of the second action, whereas *Action2DestLoc* is the destination of the second action.

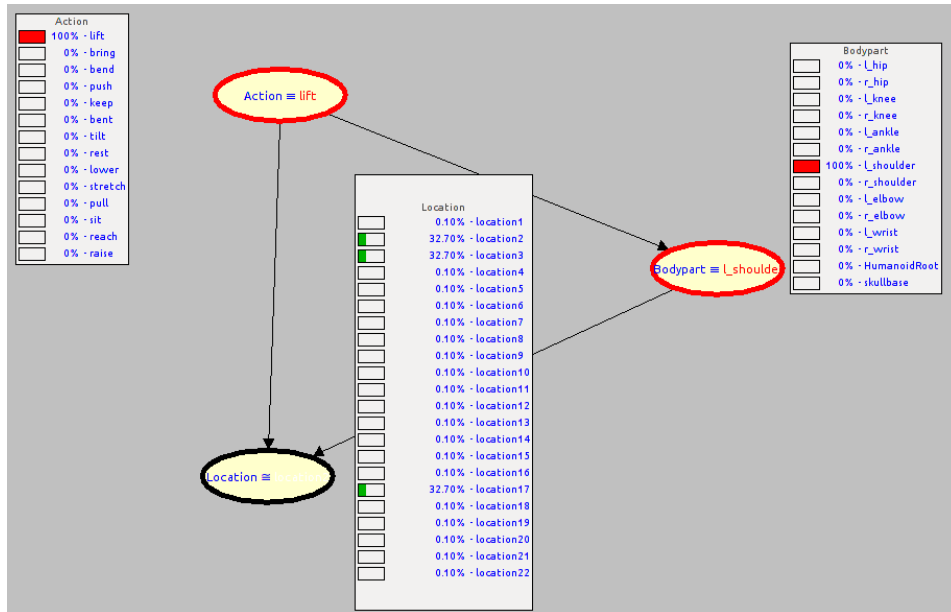


Fig. 1 Bayes network for “Lift your left arm.” before update using crowds.

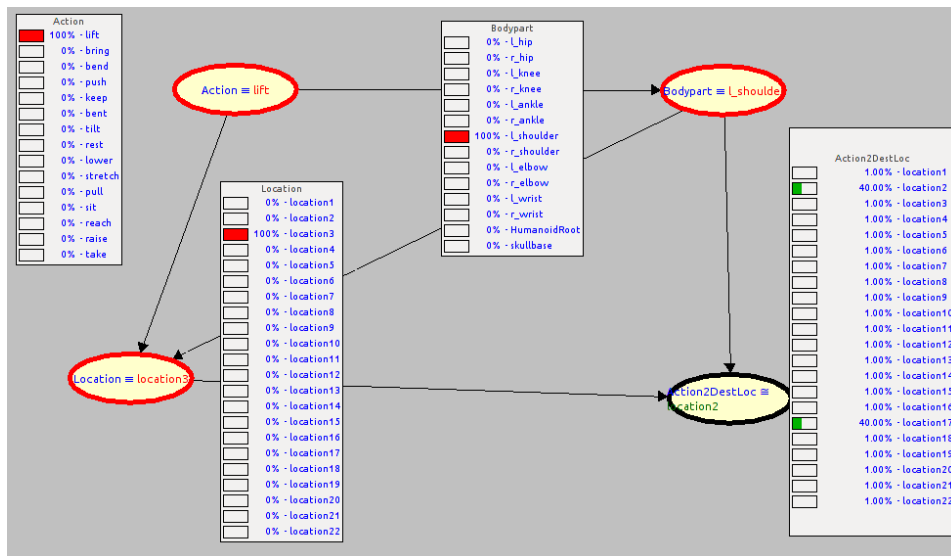


Fig. 2 Bayes network for “Lift your left arm. Move it to your right elbow.” before update using crowds.

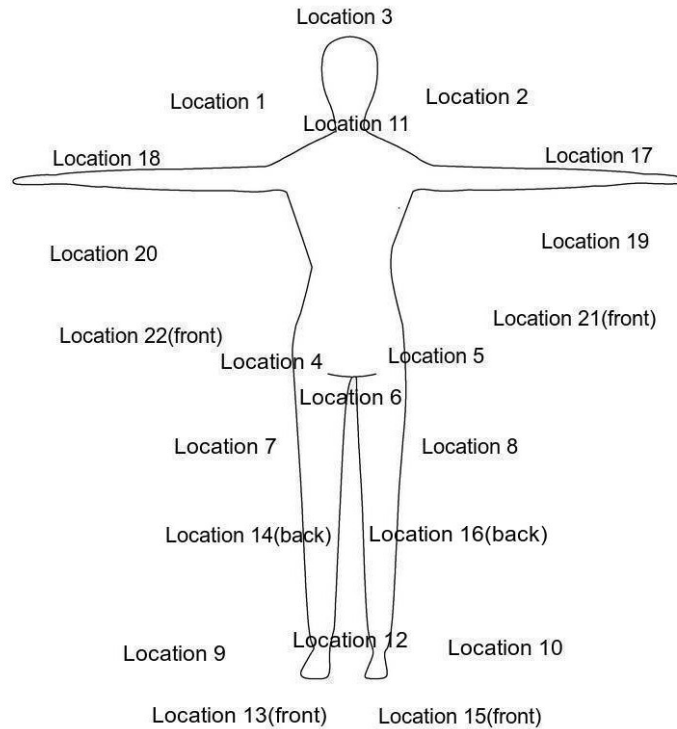


Fig. 3 Different locations or coordinates used in the BN.

Taking “Lift your left arm. Move it to your right elbow.” as an example, the result is shown in Fig. 2.

4. Updating variables in the Bayesian network

As we mentioned earlier, there are three unique variables in our Bayesian networks. It is not possible to change the *Bodypart* and *Location* variables in our system, because we already analyzed the possible body parts and location coordinates in our human body from the PEISC. However, when it comes to action verbs, which are the values in the *Action* variable, they are unlimited and impossible to assume. Therefore, we have started our Bayesian network with 14 different action verbs as shown in Fig. 3. However, we designed our network in such a way that if someone enters an instruction with an action verb that is not a value of the *Action* variable, then the system automatically adds that as a value and uses it as evidence for the next step.

Let us explore an example with the *turn* action verb, because *turn* is not a value that exists in the *Action* variable. Thus, if we enter an instruction *turn your head*, the system will check whether the action verb *turn* is a value in the *Action* variable, and if not then the system will automatically add *turn* as a value in the *Action* variable and set *turn* as evidence for the instruction, as shown in Fig. 6.

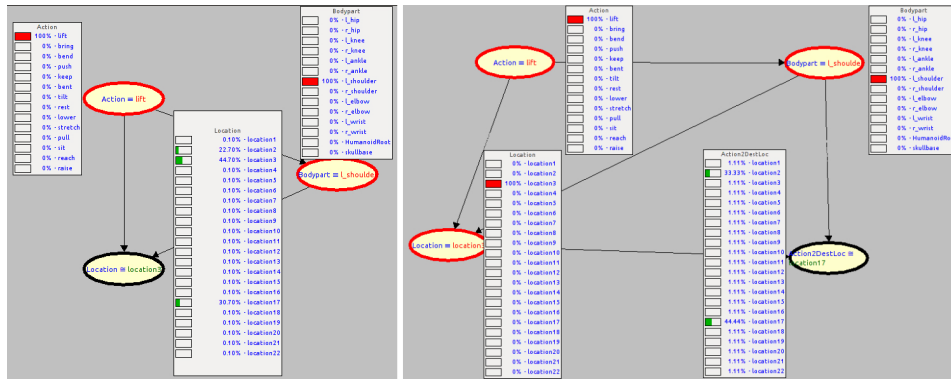


Fig. 4 Bayesian network for “Lift your left arm.” and “Lift your left arm. Move it to your right elbow.” after update using crowds.

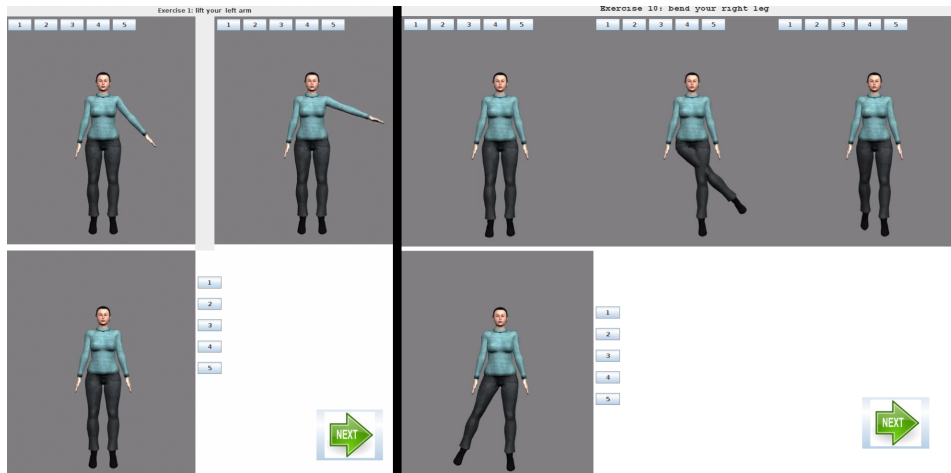


Fig. 5 Survey application for two different exercise with three and four videos.

However, if the action verb is already a value, then the system will not make any change and sets the action verb as evidence.

5. Automatic CPT update system

As mentioned earlier, the CPT is the main part of the Bayesian network. Therefore, to design reliable CPT that accurately models implicit locations, we built our CPT using crowdsourcing that models the actual transition probabilities. We developed this system using Java combined with the SamIam system. In this system, we asked people to rate 13 different exercises consisting of 44 different exercise videos on a scale of 1–5, where 1 was best and 5 was worst. All 13 exercises have been specifically selected as they do not mention the destination location of the body

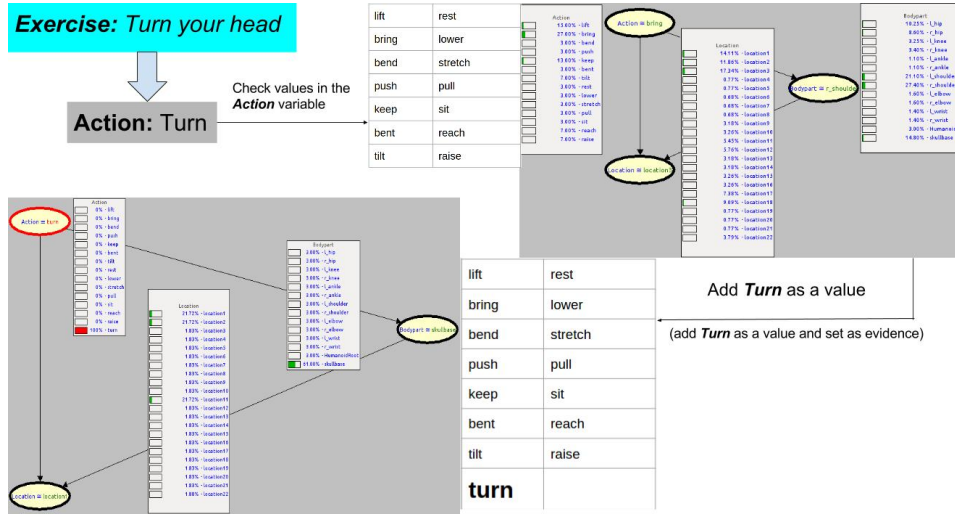


Fig. 6 Automatically adding values in the Action variable.

part. Every exercise consists of three or four videos, and each video defines different coordinates or destination locations of the body part. Put simply, if an exercise has three videos, then the correspondence body part of that exercise has three possible destination locations or coordinates, and if it has four videos, then it will have four possible destination locations or coordinates. The list of 13 exercises and corresponding body parts (which are the values of the *Bodypart* variable) is given in Tab. II.

Serial No.	Exercise	Bodypart	Videos
1	Lift your left arm	l_shoulder	3
2	Lift your right arm	r_shoulder	3
3	Bend your left ankle	l_ankle	3
4	Bend your right ankle	r_ankle	3
5	Bend your left knee	l_knee	4
6	Bend your right knee	r_knee	4
7	Bend your left wrist	l_wrist	3
8	Bend your right wrist	r_wrist	3
9	Bend your left leg	l_hip	4
10	Bend your right leg	l_hip	4
11	Bend your left elbow	l_elbow	3
12	Bend your right elbow	r_elbow	3
13	Tilt your head	skullbase	4

Tab. II List of exercises with corresponding Bodypart of the Bayesian network.

In this system, when a participant rated a video as 1, the CPT of the other *Location* variables corresponding to that body part decrease by 25%, and added the total decreased amount to the clicked *Location* variable. If a participant clicks 2, 3, or 4, then the CPT decreases by 20%, 15%, and 10%, respectively, and if they click 5, then the CPT remains the same without any change. Fig. 5 and shows how the system looks for three and four videos.

If we take the example of “*Lift your left arm.*”, then there are three different possible destination locations with same probability, as shown in Fig. 1. With these three possible locations, we have developed the system for “*Lift your left arm.*” as shown in Fig. 5. In the application, every video is rated on a scale of 1–5: when a participant rated a video as 1, 2, or 3, the other two video location CPTs decrease by 25%, 20%, and 15%, respectively, and we add those as rated videos; for ratings of 4 and 5, there will be no change. However, if there are four possible locations as shown in Fig. 5, then for a rating of 4 this changes to a 10% decrease and there is still no change for a rating of 5. Using this approach, the probability changes depending on the the crowd input.

A total of 22 people participated in our survey. Relating to the system that is represented in Figs. 1, 2 and 5, the probable locations before and after survey for exercise instructions “*Lift your left arm.*” and “*Lift your left arm. Move it to your right elbow.*” are shown in Tab. III. The probability of location3 for “*Lift your left arm.*” and location17 for “*Lift your left arm. Move it to your right elbow.*” are increased to 44.7% and 44.44% respectively after updating the CPT using crowds.

Lift your left arm		Lift your left arm Move it to your right elbow	
<i>Before Survey</i>	<i>After Survey</i>	<i>Before Survey</i>	<i>After Survey</i>
location2 (32.7%)	location2 (22.7%)	location2 (40%)	location2 (33.33%)
location3 (32.7%)	location3 (44.7%)	location17 (40%)	location17 (44.44%)
location17 (32.7%)	location17 (30.7%)		

Tab. III Results before and after the survey.

The results of all 13 exercises are shown in Fig. 7.

6. Conclusion

Bayesian networks are very important tools for modeling uncertainty about the relationship between variables. These types of uncertainty are easy for the human brain to extract, even from small sample real-world observations, but machine learning systems face considerable challenges when attempting to emulate such small sample learning based on information with implicit elements. In this article, we have proposed two Bayesian networks to extract the implicit information from text-based physical exercise instruction sheets, which is not possible with typical semantic parsers. These two networks consist of three and four variables, respectively, for single sentences with a single pose and multiple-sentence instructions with two poses. The network consists of 50 unique values for variables *Action*, *Bodypart*, *Location*, and *Action2DestLoc*. A strong CPT is very important when designing

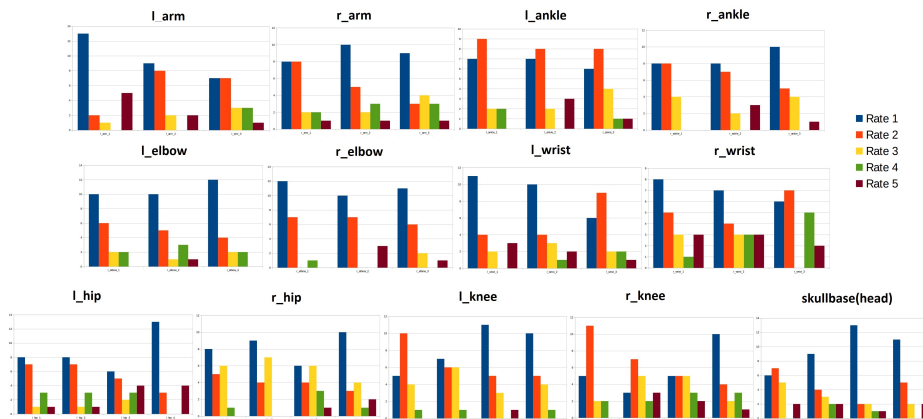


Fig. 7 Results from the crowd.

a strong and accurate Bayesian network to obtain accurate results. Therefore, to design a strong Bayesian network, we have proposed a novel way to design the CPT of the network. We have employed human computation to design the CPT using crowdsourcing. We designed a survey application consisting of all 13 different exercises corresponding to 13 human joints that are the values of the *Bodypart* variable. The participants rated 44 different videos on a scale of 1–5, with the possible destination location of those exercises. From the results, we found that the most probable destination location is very easy to extract from the network, even though it was unclear before the crowd input. We also updated the values of the *Action* variable using crowdsourcing if the value was not an existing value of the *Action* variable. In the future, we will attempt to extend this approach to building a whole Bayesian network with the help of crowds, e.g., with all variables, values, and CPTs defined with the help of crowds.

Acknowledgement

The research reported in this paper has been (partially) supported by the German Research Foundation DFG, as part of Collaborative Research Center (Sonderforschungsbereich) 1320 “EASE – Everyday Activity Science and Engineering”, University of Bremen (<http://www.ease-crc.org/>). The research was conducted in subprojects H2 “Mining and explicating instructions for everyday activities” and P1 “Embodied semantics for the language of action and change”.

References

- [1] ANDERSEN S.K., OLESEN K.G., JENSEN F.V., JENSEN F. HUGIN – A shell for building Bayesian belief universes for expert systems. In: *Proceedings of the 11th International Joint Conference on Artificial Intelligence*, Detroit, MI, USA, 1989, pp. 1080–1085. Available from: <https://www.ijcai.org/Proceedings/89-2/Papers/037.pdf>.

- [2] BEINLICH I.A., SUERMONDT H.J., CHAVEZ R.M., COOPER G.F. *The ALARM Monitoring System: A Case Study with two Probabilistic Inference Techniques for Belief Networks*. In: HUNTER J., COOKSON J., WYATT J., eds. *AIME 89. Lecture Notes in Medical Informatics*, vol. 38. Berlin, Heidelberg: Springer, 1989, pp. 247–256, doi: [10.1007/978-3-642-93437-7_28](https://doi.org/10.1007/978-3-642-93437-7_28).
- [3] BIRTLES N., FENTON N., NEIL M., TRANHAM E. Agenarisk manual (version 6.1). 2014. Available from: http://www.agenarisk.com/resources/AgenaRisk_User_Manual.pdf.
- [4] CHAN H., DARWICHE A. When do numbers really matter? In: *Proceedings of the Seventeenth conference on Uncertainty in artificial intelligence*, Seattle, Washington. Morgan Kaufmann, 2001, pp. 65–74. Available from: <https://www.jair.org/media/967/live-967-2041-jair.pdf>.
- [5] CHRISMAN L., HENRION M., MORGAN R., ET AL. Analytica 4.2 user guide. lumina decision system. Inc., Los Gatos, CA, USA, 2010. Available from: http://analyticaonline.com/ana/AdeUsersGuide4_2_3.pdf.
- [6] CONRADY S., JOUFFE L. Introduction to bayesian networks & bayesialab. 2013, 3(201). Available from: http://library.bayesia.com/download/attachments/10092794/Bayesian_Networks_Intro_v16.pdf.
- [7] COOPER G.F. Nestor: A computer-based medical diagnostic aid that integrates causal and probabilistic knowledge. Technical report, DTIC Document, 1984. Available from: <http://www.dtic.mil/get-tr-doc/pdf?AD=ADA152046>.
- [8] COZMAN F.G. Javabayes-bayesian networks in java, 2001. Available from: <http://www.cs.cmu.edu/~javabayes/>.
- [9] DARWICHE A. Samiam, 2010. Software available from <http://reasoning.cs.ucla.edu/samiam>.
- [10] DRUZDZEL M.J. GeNIe: A development environment for graphical decision-analytic models. In: *Proceedings of the AMIA Symposium*, Washington, DC, USA. American Medical Informatics Association, 1999, p. 1206. Available from: <http://www.pitt.edu/~druzdzel/psfiles/amia99.pdf>.
- [11] DRUZDZEL M.J. Smile: Structural modeling, inference, and learning engine and genie: A development environment for graphical decision-theoretic models. In: *Proceedings of the Sixteenth National Conference on Artificial Intelligence and the Eleventh Innovative Applications of Artificial Intelligence Conference Innovative Applications of Artificial Intelligence*, AAAI '99/IAAI '99, Orlando, Florida, USA. American Association for Artificial Intelligence, 1999, pp. 902–903. Available from: <http://www.aaai.org/Papers/AAAI/1999/AAAI99-129.pdf>.
- [12] JENSEN F.V. *An introduction to Bayesian networks*. New York: Springer-Verlag, 1996.
- [13] JENSEN F.V. *Bayesian Networks and Decision Graphs*. Secaucus: Springer-Verlag, New York, 2001.
- [14] KADIE C.M., HOVEL D., HORVITZ E. Msbnx: A component-centric toolkit for modeling and inference with bayesian networks. *Microsoft Research, Richmond, WA, Technical Report MSR-TR-2001-67*, 2001, 28. Available from: <https://www.microsoft.com/en-us/research/wp-content/uploads/2016/02/tr-2001-67.pdf>.
- [15] LAURITZEN S.L., SPIEGELHALTER D.J. Local computations with probabilities on graphical structures and their application to expert systems. *Journal of the Royal Statistical Society. Series B (Methodological)*, 1988, pp. 157–224. Available from: <http://www.eecis.udel.edu/~shatkay/Course/papers/Lauritzen1988.pdf>.
- [16] MANUAL N. Netica v1.05. *Norsys Software Corp*, 1997. Available from: https://www.norsys.com/downloads/old_versions/NeticaMan_Win_105.pdf.
- [17] MURPHY K.P. *Machine learning: a probabilistic perspective*. MIT press, 2012.
- [18] RUGGERI F., KENETT R.S., FALTIN F.W. *Encyclopedia of statistics in quality and reliability*, 2007.

- [19] SAFFIOTTI A., UMKEHRER E. Pulcinella: A general tool for propagating uncertainty in valuation networks. In: *Proceedings of the Seventh conference on Uncertainty in Artificial Intelligence*, Los Angeles, CA, USA. Morgan Kaufmann Publishers Inc., 1991, pp. 323–331. Available from: <https://pdfs.semanticscholar.org/767a/a03075337bf74e79defb24f61ce4407044c9.pdf>.
- [20] SARMA H., PORZEL R., MALAKA R. A Step Toward Automated Simulation in Industry. In: *Dynamics in Logistics: Proceedings of the 5th International Conference LDIC, 2016 Bremen, Germany*, Bremen, Germany. Springer International Publishing. 2016, pp. 99–105, doi: [10.1007/978-3-319-45117-6_9](https://doi.org/10.1007/978-3-319-45117-6_9).
- [21] SARMA H., PORZEL R., MALAKA R., SAMADDAR A.B. A Step towards Textual Instructions to Virtual Actions. In: *2017 IEEE 7th International Advance Computing Conference (IACC)*, Hyderabad, India. IEEE, 2017, pp. 239–243, doi: [10.1109/IACC.2017.0060](https://doi.org/10.1109/IACC.2017.0060).
- [22] SARMA H., PORZEL R., SMEDDNICK J., MALAKA R. Towards Generating Virtual Movement from Textual Instructions: A Case Study in Quality Assessment. In: *Proceedings of The Third AAAI Conference on Human Computation and Crowdsourcing (HCOMP-2015)*, San Diego, USA. AAAI, 2015. Available from: http://www.humancomputation.com/2015/papers/35_Paper.pdf.