Towards Automated Generation of Semantic Annotation for Activity Recognition Problems

Kristina Yordanova
Department of Visual and Analytic Computing
Department of Computer Science
University of Rostock
Rostock, Germany
kristina.yordanova@uni-rostock.de

***Abstract*—Ground truth is essential for activity recognition problems. It is used to apply methods of supervised learning, to provide context information for knowledge-based methods, and to quantify the recognition performance. Semantic annotation extends simple symbolic labelling by assigning semantic meaning to the label and enables reasoning about the semantic structure of the observed activity. The development of semantic annotation for activity recognition is a time consuming task, which involves a lot of effort and expertise. To reduce the time needed to develop semantic annotation, we propose an approach that automatically generates semantic models based on manually assigned symbolic labels. We provide a detailed description of the automated process for annotation generation and we discuss how it replaces the manual process. To validate our approach we compare automatically generated semantic annotation for the CMU grand challenge dataset with manual semantic annotation for the same dataset. The results show that automatically generated models are comparable to manually developed models but it takes much less time and no expertise in model development is required.**

***Index Terms*—activity recognition, annotation, ground truth, model generation**

I. Introduction and Motivation

The quality of annotation of sensor datasets describing human behaviour plays a central role in the performance of the recognition system. The annotation serves as a ground truth both for training data-driven models for activity recognition and for evaluating the performance of the activity or plan estimation procedure. It also provides the context information needed for developing knowledge-based activity recognition systems [27]. In a previous work we presented a model-based approach to semantic annotation of human behaviour [24], [26]. In model-based annotation the labels assigned to the data provide an underlying semantic structure that contains information about the actions, goals, and plans being executed. This semantic structure is represented in the form of a model of the behaviour's state in terms of collection of state variables. Actions are then defined as effects that change the state of the model. This form of annotation provides structured knowledge of the concepts in the data being annotated and enables the reasoning over underlying behaviour changes, their causal relations, and contextual dependencies. Such annotation is important for evaluating activity and plan recognition approaches that aim to recognise the actions being executed, the goal of the plan as well as the context and causes behind the observed actions.

At the time the approach was developed Kristina Yordanova was funded by the German Research Foundation, grant number YO 226/1-1.

A serious drawback in providing semantic annotation is that it is very time consuming and requires many iterations before a causally correct annotation is developed. For example, annotating the Carnegie Mellon University Multi-Modal Activity Database (CMU-MMAC) with semantic annotation took about two years of work [26]. To address this problem, in this work we propose an automated method for generating semantic model-based annotation. The approach makes use of natural language processing techniques to generate semantic models based on the textual labels provided by the annotator. In that manner, the annotator only needs to provide textual labels and the semantic structure of the labels is automatically generated. Additional checkins ensure that the annotation is causally correct and that errors in the textual labels are not transferred to the automatically generated model. The approach seriously reduces the time needed for annotating a sensor dataset. It also allows generating a semantic structure that can be manually extended by the annotator in case additional implicit context needs to be embedded in the models.

The paper is structured as follows. Section II discusses the state of the art in semantic annotation and automatic model generation. Section III describes the manual process for semantic annotation and the test dataset. Section IV provides detailed information about the proposed approach for automatic generation of semantic annotation. In Section V we discuss the results of comparing automatically generated models for semantic annotation with manually developed models. Finally we conclude the paper with discussion about the results and the future work in Section VI.

II. Related Work

A. Types of annotation

Annotation can roughly be divided into three types: label-based annotation, plan-based annotation and semantic annotation [26]. In label-based annotation the annotator assigns a string as a label to each observation. This string does not have any underlying meaning beside the equality between two strings. This is the most common annotation in activity recognition because even providing a string as a label requires a lot of effort.

The second type of annotation is plan annotation and it is divided into goal labelling and plan labelling [5]. In goal labelling, a label is assigned to each goal the user is pursuing through his or her actions [1], [4]. In plan labelling, apart from the goal, also the steps towards achieving it are annotated (in other words, the sequence of actions) [2]. Goal labelling is usually preferred to plan labelling as it requires less time and effort. In the case of plan labelling, often synthetic data is used, so that the annotation is automatically generated together with the data [13], [17].

The third type of annotation is the semantic annotation [14]. The semantic annotation is divided into algebraic and model-based annotation [26]. The algebraic annotation provides a semantic meaning for each label. This structure is usually in the form of ontology with the corresponding properties, concepts and relations between the different elements of the annotation [8], [18]. In difference to the algebraic annotation, model-based annotation provides a model of the world and how it changes with the execution of new actions. This is achieved through a collection of state variables that describe the state space of the model [24]. Model-based annotation is very important for reasoning about properties of the world but also how and why the world and the observed behaviour changes. It is, however, the most complex type of annotation, which requires a lot of resources and time. To address this problem, we look at approaches for model generation from textual descriptions.

B. Model generation

The idea behind model generation is to substitute the manual development of behaviour models with models generated from textual sources [21]. The goal of grounded language acquisition is to learn linguistic analysis from a situated context [7], [19]. In other words, texts (often instructional texts) are analysed to discover the action semantics and relationships between actions and the surrounding context.

This could be done in different ways: through grammatical patterns that are used to map the sentence to a machine understandable model of the sentence [6], [29]; through machine learning techniques [3], [11], [15]; or through reinforcement learning approaches that learn language by interacting with an external environment [6], [7], [11], [15], [19].

Models learned through model grounding have been used for plan generation [6], [23], for learning the optimal sequence of instruction execution [7], for learning navigational directions [19], and for interpreting human instructions for robots to follow them [15]. To our knowledge, model generation has not been used for building semantic models for model-based annotation. The very structured text representation of annotation sequences, however, makes them ideal candidate for automatically generating semantic models from the labels. In what follows we adapt a model generation approach initially applied to textual instructions in order to obtain model-based annotation.

III. Metthods and Materials

Before presenting the approach for automated generation of semantic annotation, below we describe the manual process presented in [24] as well as the Carnegie Mellon University Multi-Modal Activity Database dataset and the manual semantic annotation, with which we empirically compare the automated approach to the manual one.

A. A process for providing semantic annotation

Most activity recognition experiments such as the CMU-MMAC [9] could be considered as goal oriented. In other words, the participants in the experiment are performing a sequence of actions that lead to a certain goal (for example, preparing a certain meal [28]). To ensure comparability of different repetitions, usually identical experimental setup is chosen for each trial. This results in an action sequence that resembles a plan, leading from the same initial state (described by the same starting setting of the experiemnt) to a set of goal states (for example the completion of different meals). In the domain of automated planning and scheduling, plan sequences are generated from domain models, where actions are defined by means of preconditions and effects. For example, Figure 1 gives an example of an action in the Planning Domain Definition Language (PDDL) notation.

```
(:action take
 :parameters (?what - takeable ?from - loc)
 :precondition (and
  (= (is-at ?what) ?from)
  (not (= ?from hands))
  ...
 :effect (and
  (assign (is-at ?what) hands)
  (increase (objects_taken) 1)
  (when
   (not (is-clean ?what))
   (not (is-clean hands)))))
```

Fig. 1. An example of an action scheme for the take action in terms of preconditions and effects in PDDL notation. Figure adapted from [26].

A plan is then a sequence of actions generated by grounding the action schemas of the domain leading from an initial state to the goal state. An example of how the execution of one action changes the state can be seen in Figure 2. After the execution of the action take, the values of the functions change, so that the knife is no longer at a specific location and the taken objects are increased with one. After the execution of the action put, now the knife is at the location board and the number of taken objects is decreased.

In the manual annotation process, we manually create plans that reflect the participants' actions, then we define a planning domain, which describes the causal and semantic connections of the actions to the state of the world. Figure 3 shows the steps in the manual annotation process. The first step is the definition of a dictionary containing all possible actions and entities that describe the given problem. In the second step the action relations are defined. This includes defining the

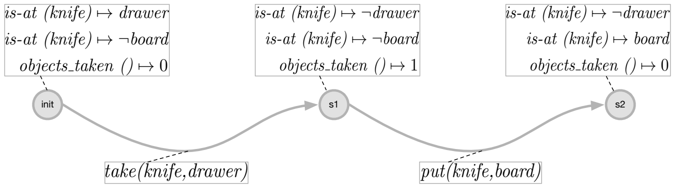

Fig. 2. Change in the state space after the execution of the actions take and put. Figure adapted from [24].

role and type of involved objects as well as how they are related to the action. In the third step the state properties are defined. Basically, we define a set of state properties as a function of a tuple of entities to an entity of the domain. The state space is then defined by each combination of possible mappings of entity tuples. In the fourth step we define the preconditions and effects for each action as shown in Figure 1. In step five we manually annotate the video logs of the sensor dataset according to the dictionary we have defined in step one. Finally, in the sixth step we validate the annotation by comparing the validity of the annotation sequence against the manually developed model. The detailed procedure is also described in [24].

B. The CMU-MMAC

The Carnegie Mellon University Multi-Modal Activity Database (CMU-MMAC) is a collection of kitchen activities [10]. 55 subjects were recorded by multiple sensors such as cameras, accelerometers, and RFID sensors. The CMU-MMAC consists of five sub datasets. These are Brownie, Sandwich, Eggs, Salad, and Pizza. Each dataset contains data from one food preparation task. Annotation for 16 subjects can be downloaded from the CMU-MMAC website[1]. This label sequence is missing semantics, which if present would allow reasoning about context information such as object locations and relations between actions and entities. To address this problem, in a previous work we newly annotated three of the five datasets (Brownie, Sandwich, and Eggs) following the manual annotation process described above. The annotator identified the action classes (11 for the Brownie, 12 for the Eggs, and 12 for the Sandwich), entities (30 for the Sandwich dataset, 44 for the Brownies, and 43 for the Eggs), and all valid action instantiations (119 unique labels where identified for the Sandwich dataset, 187 for the Brownies, and 179 for the Eggs). All in all, 90 action sequences were annotated. The complete annotation can be downloaded from [25]. In this work, we use the manually created model for the “Brownie” dataset to compare it with the model generated through our automated approach.

IV. Approach

In a previous work, we showed that our approach to semantic annotation provides a high quality annotation with an interrater reliability of about 80% overlapping [24]. To produce such high quality, however, it takes a lot of time and effort not only for the annotators, but also for the annotation designers who have to manually build the semantic model used for validating the annotation. To address this problem, we extend the approach so that it automatically generates the underlying semantic model needed for validating the annotation. The method adapts the idea of learning planning operators from textual instructions proposed in [23]. It differs from existing works for model generation in the source of data from which the model is learned. While existing works use textual instructions written by persons, the proposed approach uses such instructions only for the initial model generation. To update the model with newly discovered elements, the model is automatically updated based on the produced annotation. Novelty is also the iterative extension of the model based on newly annotated data. The proposed method automates steps two to four in the manual annotation process.

Step one from the annotation process (actions and entities dictionary definition) is extended so that in addition to the dictionary definition, the domain experts provide a step by step description of the executed action in natural language using the names of the actions and entities from the dictionary. These textual instructions are then parsed as proposed in [24] in order to obtain the relations between the model elements such as relations between entities and actions and entities (*Step two* from the annotation process). Figure 5 shows an example of a sentence being parsed in order to extract the relations between actions and entities. Furthermore, the identified properties in the sentences are used to generate the state properties of the model. For example, from the sentence in Figure 5 we extract the state properties *is-clean (bowl)* and *is-at (bowl)* $\rightarrow$ *cupboard*. This procedure replaces *Step three* in our annotation process.

The next step of our process (*Step four*) deals with the manual definition of preconditions and effects. We automate this step by first automatically obtaining the implicit causal relations between the actions in the textual instructions. This is done by converting the textual instructions into time series and then performing a time series analysis to discover any causal dependencies between the series as proposed in [20], [23]. We start by representing each unique action in a text as a time series. Each element in the series represents the number of occurrences of the action in the sentence. We then apply the Granger causality test, which is a statistical test for determining whether one time series is useful for forecasting another. It performs statistical significance test for one time series, “causing” the other time series with different time lags using auto-regression [12]. First, we estimate the regression $y_t = a_o + a_1 y_{t-1} + ... + a_p y_{t-p} + b_1 x_{t-1} + ... + b_p x_{t-p}$. Then we use an F-test to evaluate whether the lagged x terms are significant. For example, we generate time series for the words “take” and “put” and after applying the Granger test, it concludes that the lagged time series for “take” significantly improve the forecast of the “put” time series, thus we conclude that “take” causes “put” [23].

Then based on the semantic structure of the textual descrip-

[1] http://www.cs.cmu.edu/~espriggs/cmu-mmac/annotations/

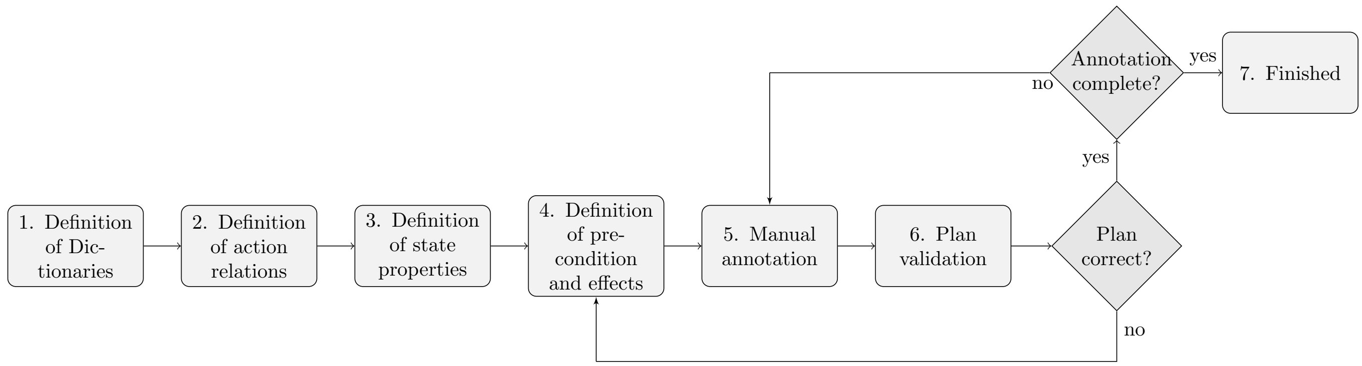

Fig. 3. The manual annotation workflow for model-based annotation as proposed in [24]. Figure adapted from [24].

```
Open the cupboard.
Take the clean bowl form the cupboard.
Put the bowl on the counter.
Take the measuring cup from the cupboard.
Put the measuring cup on the counter.
Take the measuring cup from the cupboard.
```

Fig. 4. Extract of the textual instructions describing the execution sequence of actions in the “Brownie” dataset.

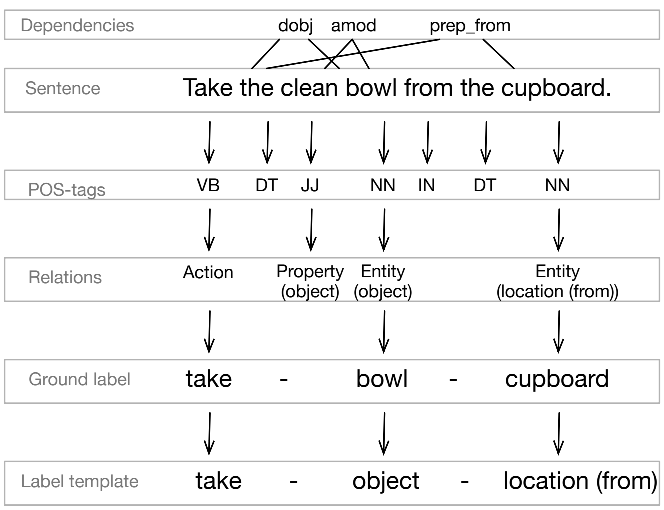

Fig. 5. Parsing a sentence in order to obtain the relations between actions and entities as well as their properties.

tion, the properties necessary to describe the preconditions and effects of the actions are identified (see Figure 5). Furthermore, based on the identified objects and existing language taxonomies the object hyperonyms are identified. They represent the abstraction hierarchy of the objects and are later used to define the object types in the planning operators. This is done by identifying all hyperonyms of a given object that are defined in a language taxonomy. In our case we use WordNet [16], which is the taxonomy of English language. As some words have different meaning, we take the most often used meaning of a given word.

All information identified until now is then consolidated in a single situation model. This model contains all semantic information needed to build the precondition-effect rules of our model (actions, objects, various causal, spatial and locational relations). Figure 6 shows the situation model generated for the instruction of preparing brownies. For more information about the procedure of generating situation models see [22].

The situation model is then used to define the precondition-effect rules for the set of actions we have identified. Figure 7 shows an example of a generated rule for the action “take” for the Brownie dataset. The name of the action template is the same as the action name. The parameters are then taken from the abstraction hierarchy. In other words, the concrete objects on which the action is applied are replaced with their abstract representations. The precondition is defined through different predicates: default predicates are used to define the execution of the action. Apart from that, any state properties and causal relations are also added to the preconditions and effects. For more details on the procedure see [23].

The generated model is then used in the same manner as the manually built one: each newly annotated video is validated using the model (see *Step six* in our process). Any new entities or actions discovered during the annotation are then automatically added to the list of actions and entities and the model is automatically extended in order to incorporate the new semantic knowledge. This is done by extending the textual instructions used for the initial model with generated from the annotation instructions. The textual instructions are generated by automatically exporting the annotation from ELAN and converting it to natural language where each sentence corresponds to a label in the annotation. Then the model generation procedure is repeated and any actions, entities, state properties, or relations that do not appear in the model are added to it.

V. Results

To evaluate the ability of the approach to generate models for model-based annotation, we first compared the number of action classes, objects, and action templates manually modelled against generated for the “Brownies” dataset. To evaluate the size of the state space model, we compared a model manually built for the CMU “Brownies” dataset with automatically generated one according to our procedure. Table

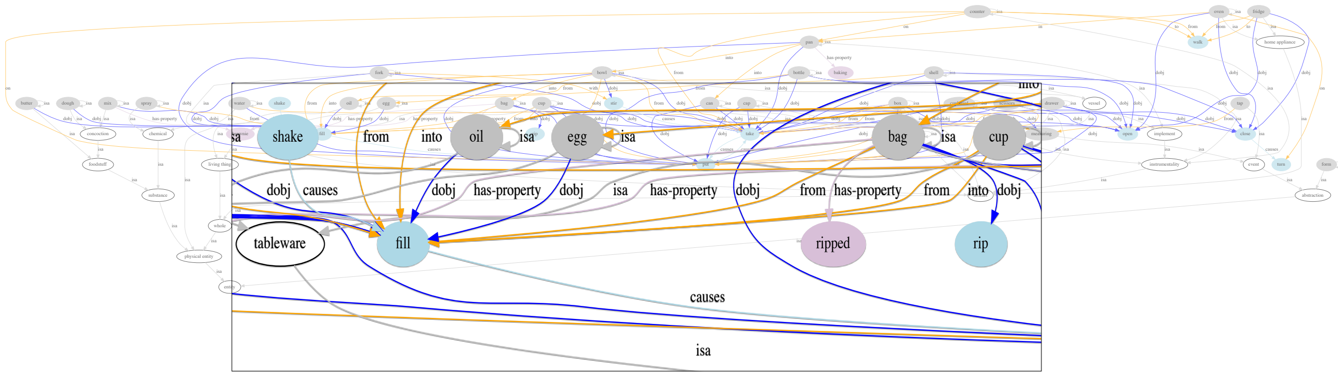

Fig. 6. Extract of the situation model for the brownies instruction. Blue circles indicate actions, grey – objects, lila – properties, white taxonomy of objects. Dark blue relations indicate direct object – verb relation, yellow – different types of relations between indirect objects and verbs or nouns, light blue – causal relations, grey – abstraction hierarchy (figure adapted from [22]).

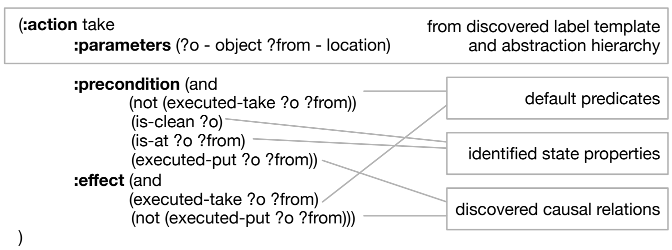

Fig. 7. An example of automatically generated precondition-effect rule for the action "take".

TABLE I

COMPARISON BETWEEN A GENERATED MODEL FOR THE BROWNIE DATASET ($PDDL_g$) AND THE HANDCRAFTED MODEL $PDDL_h$. OPERATORS INDICATES NUMBER OF GROUND OPERATORS IN THE MODEL; PREDICATES NUMBER OF GROUND PREDICATES; BR. FACTOR IS THE BRANCHING FACTOR OF THE MODEL (I.E. HOW MANY STATES CAN BE REACHED FROM A GIVEN STATE IN THE MODEL); AND STATES INDICATE THE SIZE OF THE STATE SPACE WHEN USING ITERATIVE DEEPENING DEPTH FIRST SEARCH WITH MAXIMUM DEPTH OF 5.

Metrics	$PDDL_g$	$PDDL_h$
actions	10	11
objects	28	44
operators	421	257
predicates	339	89
min/mean/max br. factor	1/231.19/421	5/30.82/55
states (depth 5)	10 000 227	1 785 896

I shows comparison between the manual model ($PDDL_h$) used for validating the annotation of the Brownie dataset in [24] and a model, generated from description of the Brownie experiment ($PDDL_g$). On the one side, the generated model identified less actions and objects than the manual one. The one missing action in the generated model is the "other" action, which the approach does not recognise as an action. Regarding the smaller number of objects, it is due to the fact that the manual model contains many composite objects such as "empty_egg_shell". The generated model will consider the object to be "shell" with properties "empty" and "egg".

On the other hand, when we look at the state space model, the generated model produces more elements than the handcrafted one and it has a larger state space and branching factor. This is in part because of lack of common sense knowledge in the instructional text. The manually developed model contains implicit knowledge, the model designer used to reduce the model complexity. The automatic approach did not have access to this knowledge, thus it was not encoded into the model. Nevertheless, as the purpose of the model is not activity recognition but rather encoding and validating the semantic structure of the annotation, the branching factor is not a serious issue (i.e. we still can validate the annotation even with a model that contains many possibilities). Another option is interactively asking for the designer's input to optimise the generated model.

VI. DISCUSSION

In this work we presented an approach for automatic generation of models for model-based annotation of activity recognition datasets. We compared the automatically generated models with manual models. The results showed that the model is comparable to the handcrafted but that it does not encode a common sense knowledge that we humans might include in the manual model. On the one hand, this removes some context information we might be interested in. On the other hand, thinking about the time required to develop a manual model, the automatically generated one saves months of manual work. The model can be used as initial model, which is later extended with additional context information. Even if input from the designer is required, the automatic generation of the semantic model seriously reduces the time needed for manual model development. It also reduces the effort required to produce high quality semantic annotation as the annotator is automatically provided with the semantic structure and validation tool without the need to wait for the model designer to develop or extend the model.

In the future we plan to integrate the proposed approach in the ELAN annotation tool, which is often used for annotation of video logs. We also plan to include different automated tests such as checking for gaps, ensuring that the label is part of the dictionary, and ensuring that the sequence of labels is possible in the model. In case the sequence is not possible, the intervention of the annotator is required either by adapting the annotation sequence, so that it is causally valid according to the model or by adjusting the structure of the model to allow the annotation sequence.

References

[1] D. W. Albrecht, I. Zukerman, and A. E. Nicholson. Bayesian models for keyhole plan recognition in an adventure game. *User Modeling and User-Adapted Interaction*, 8(1-2):5–47, 1998.

[2] M. Bauer. Acquisition of user preferences for plan recognition. In *Proceedings of the Fifth International Conference on User Modeling (UM96*, pages 105–112, 1996.

[3] L. Benotti, T. Lau, and M. Villalba. Interpreting natural language instructions using language, vision, and behavior. *ACM Trans. Interact. Intell. Syst.*, 4(3):13:1–13:22, Aug. 2014.

[4] N. Blaylock and J. Allen. Statistical goal parameter recognition. In *4th International Conference on Automated Planning and Scheduling (ICAPS'04)*, pages 297–304, June 2004.

[5] N. Blaylock and J. Allen. Hierarchical goal recognition. In G. Sukthankar, R. P. Goldman, C. Geib, D. V. Pynadath, and H. H. Bui, editors, *Plan, activity, and intent recognition*, pages 3–32. Elsevier, Amsterdam, 2014.

[6] S. R. K. Branavan, N. Kushman, T. Lei, and R. Barzilay. Learning high-level planning from text. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Long Papers - Volume 1*, ACL '12, pages 126–135, Stroudsburg, PA, USA, 2012. Association for Computational Linguistics.

[7] S. R. K. Branavan, D. Silver, and R. Barzilay. Learning to win by reading manuals in a monte-carlo framework. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies - Volume 1*, HLT '11, pages 268–277, Stroudsburg, PA, USA, 2011. Association for Computational Linguistics.

[8] H.-S. Chung, J.-M. Kim, Y.-C. Byun, and S.-Y. Byun. Retrieving and exploring ontology-based human motion sequences. In O. Gervasi, M. L. Gavrilova, V. Kumar, A. Laganà, H. P. Lee, Y. Mun, D. Taniar, and C. J. K. Tan, editors, *Computational Science and Its Applications – ICCSA 2005*, volume 3482 of *Lecture Notes in Computer Science*, pages 788–797. Springer Berlin Heidelberg, 2005.

[9] F. de la Torre, J. Hodgins, J. Montano, S. Valcarcel, R. Forcada, and J. Macey. Guide to the carnegie mellon university multimodal activity (CMU-MMAC) database. Technical Report CMU-RI-TR-08-22, Robotics Institute, Carnegie Mellon University, July 2009.

[10] F. de la Torre, J. K. Hodgins, J. Montano, and S. Valcarcel. Detailed human data acquisition of kitchen activities: the CMU-Multimodal Activity Database (CMU-MMAC). In *Workshop on Developing Shared Home Behavior Datasets to Advance HCI and Ubiquitous Computing Research, in conjuction with CHI 2009*, 2009.

[11] D. Goldwasser and D. Roth. Learning from natural instructions. *Machine Learning*, 94(2):205–232, 2014.

[12] C. W. J. Granger. Investigating Causal Relations by Econometric Models and Cross-spectral Methods. *Econometrica*, 37(3):424–438, August 1969.

[13] L. M. Hiatt, A. M. Harrison, and J. G. Trafton. Accommodating human variability in human-robot teams through theory of mind. In *Proceedings of the Twenty-Second International Joint Conference on Artificial Intelligence*, IJCAI'11, pages 2066–2071, Barcelona, Spain, 2011. AAAI Press.

[14] A. Kiryakov, B. Popov, D. Ognyanoff, D. Manov, A. Kirilov, and M. Goranov. Semantic annotation, indexing, and retrieval. In D. Fensel, K. Sycara, and J. Mylopoulos, editors, *The Semantic Web - ISWC 2003*, volume 2870 of *Lecture Notes in Computer Science*, pages 484–499. Springer Berlin Heidelberg, 2003.

[15] T. Kollar, S. Tellex, D. Roy, and N. Roy. Grounding verbs of motion in natural language commands to robots. In O. Khatib, V. Kumar, and G. Sukhatme, editors, *Experimental Robotics*, volume 79 of *Springer Tracts in Advanced Robotics*, pages 31–47. Springer Berlin Heidelberg, 2014.

[16] G. A. Miller. Wordnet: A lexical database for english. *Commun. ACM*, 38(11):39–41, Nov. 1995.

[17] M. Ramirez and H. Geffner. Goal recognition over pomdps: Inferring the intention of a pomdp agent. In *Proceedings of the Twenty-Second International Joint Conference on Artificial Intelligence*, volume 3 of *IJCAI'11*, pages 2009–2014, Barcelona, Spain, 2011. AAAI Press.

[18] S. Saad, D. De Beul, S. Mahmoudi, and P. Manneback. An ontology for video human movement representation based on benesh notation. In *Multimedia Computing and Systems (ICMCS), 2012 International Conference on*, pages 77–82, May 2012.

[19] A. Vogel and D. Jurafsky. Learning to follow navigational directions. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, ACL '10, pages 806–814, Stroudsburg, PA, USA, 2010. Association for Computational Linguistics.

[20] K. Yordanova. Discovering causal relations in textual instructions. In *Recent Advances in Natural Language Processing*, pages 714–720, Hissar, Bulgaria, September 2015. RANLP 2015 Organising Committee.

[21] K. Yordanova. From textual instructions to sensor-based recognition of user behaviour. In *Companion Publication of the 21st International Conference on Intelligent User Interfaces*, IUI '16 Companion, pages 67–73, New York, NY, USA, 2016. ACM.

[22] K. Yordanova. Automatic generation of situation models for plan recognition problems. In *Proceedings of the International Conference Recent Advances in Natural Language Processing*, pages 823–830, Varna, Bulgaria, September 2017. INCOMA Ltd.

[23] K. Yordanova. Extracting planning operators from instructional texts for behaviour interpretation. In *German Conference on Artificial Intelligence*, pages 215–228, Berlin, Germany, Sptember 2018.

[24] K. Yordanova and F. Krüger. Creating and exploring semantic annotation for behaviour analysis. *Sensors*, 18(9), 2018.

[25] K. Yordanova, F. Krüger, and T. Kirste. Semantic annotation for the CMU-MMAC dataset (version 2). University Library, University of Rostock, 2018. http://purl.uni-rostock.de/rosdok/id00002273.

[26] K. Yordanova, F. Krger, and T. Kirste. Providing semantic annotation for the cmu grand challenge dataset. In *2018 IEEE International Conference on Pervasive Computing and Communications Workshops (PerCom Workshops)*, pages 579–584, March 2018.

[27] K. Yordanova, A. Paiement, M. Schrder, E. Tonkin, P. Woznowski, C. M. Olsson, J. Rafferty, and T. Sztyler. Challenges in annotation of user data for ubiquitous systems: Results from the 1st arduous workshop. Technical Report arXiv:1803.05843, arXiv preprint, March 2018.

[28] K. Yordanova, S. Ldtke, S. Whitehouse, F. Krger, A. Paiement, M. Mirmehdi, I. Craddock, and T. Kirste. Analysing cooking behaviour in home settings: Towards health monitoring. *Sensors*, 19(3), 2019.

[29] Z. Zhang, P. Webster, V. Uren, A. Varga, and F. Ciravegna. Automatically extracting procedural knowledge from instructional texts using natural language processing. In N. Calzolari, K. Choukri, T. Declerck, M. U. Doğan, B. Maegaard, J. Mariani, J. Odijk, and S. Piperidis, editors, *Proceedings of the Eighth International Conference on Language Resources and Evaluation (LREC-2012)*, pages 520–527, Istanbul, Turkey, May 2012. European Language Resources Association (ELRA). ACL Anthology Identifier: L12-1094.