Text-Based Inference of Object Affordances for Human-Robot Interaction

Michele Persiani[1] and Thomas Hellström[2]

***Abstract*—Affordances denote actions that can be performed in the presence of different objects. In this paper we present a model to generate names of possible affordances for a named object. We use a Conditional Variational Autoencoder as generative model and train it with sentences from a selected corpus. The model can be used in several ways in HRI, for instance by a service robot providing assistance to perform activities of daily living. The preliminary evaluation of the model shows good results compared to a benchmark method.**

***Index Terms*—Affordance, Intention recognition, Human-Robot-Interaction, Generative model, Autoencoder, Natural language processing**

I. Introduction

The term "affordance" was introduced by the American psychologist Gibson [1] to describe what an animal can do with a given object. It has since then been extensively utilized, interpreted, and re-defined (see [2] for an overview) in fields such as human-computer-interaction [3] and human-robot-interaction (HRI) [4]. We focus on its use within HRI, and use the term to denote actions that can be performed with a given object. As a simplified first approach we ignore the influence of different environments, and assume a one-to-many mapping G: *Objects* $\rightarrow$ *Affordances*. The object "door" may, for example, be used to perform the actions "open", "close", and "lock".

This paper presents ongoing work on how G may be learned from free-text corpora. The results show how it is possible to learn a generative model G that, given an object name, generates affordances according to a probability distribution that matches the used training data. Qualitatively results also indicate that the model manages to generalize, both to previously unseen objects and actions.

The paper is organized as follows. In Section II we give motivation for the work from an HRI perspective, followed by a brief review of earlier related work in Section III. The developed method is described in Section IV, and results from the evaluation are presented in Section V. The paper is finalized by conclusions in Section VI.

II. Affordances

Once learned, the function G can be used in several ways by a robot, for instance by a service robot providing assistance to perform activities of daily living. By visually identifying objects in the environment, or in the robot's verbal dialogue with the human, affordances can be inferred through G. The affordances may be used to infer the human's intention, which may guide the robot's behavior [5]. For example, if older adults want to talk to their distant children, a listening robot may infer that the adults wants to call them, and suggest making a phone call. G may also be used by a robot to decide how to act within a given context that affords certain actions. A service robot may, for example, suggest its user to read a book, if a physical book is visually detected. Affordances may also be useful for object disambiguation. When a human tells a robot to "pick it up!", the robot only has to consider objects with the "pick up" affordance in the current scene [4]. Inference of affordances may also be used to design robots that are understandable by humans, since mutually perceived affordances may contribute to explaining a robot's behavior [6], and thereby increase interaction quality [7].

III. Earlier work

Chao et al. [8] mine *semantic affordances* from a combination of crowdsourcing, images, and text. They show how in Natural Language Processing (NLP) objects and actions can be connected through the introduction of a latent space. Narashiman et al. [9] find links between object and action in text using deep reinforcement learning techniques [10]. Antanas et al. [11] relate affordances to the symbol grounding problem. By using image data, they map visual objects to utterances and actions, while through statistical methods they learn ontologies for affordances. Ruggeri and Di Caro [12] explain how affordance is a concept that sits in the middle between objectivity and subjectivity, and propose ontological views for their usage in Computer Science.

IV. Method

A generative model for the one-to-many mapping G : *Objects* $\rightarrow$ *Affordances* was trained with pairs <*object*, *action*>. These pairs were generated by *semantic role labeling* of sentences from a selected corpus. Objects and actions were represented by *wordvectors* throughout the process, which is illustrated in Fig 1 below.

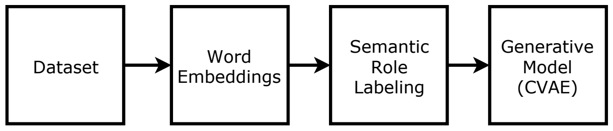

Fig. 1. Steps taken to obtain the generative model.

[1,2] *Department of Computing Science*, *Umeå University*, Umeå, Sweden, michelep@cs.umu.se, thomash@cs.umu.se

A. Corpus

As data source we used the *Yahoo! Answers Manner Questions* (YAMC) dataset[1] containing 142627 questions and corresponding answers. The corpus is a distillation of all questions gathered from the platform *Yahoo! Answers* during the year 2007. It is a small subset of the questions, selected for their linguistic properties such as good quality measured in terms of vocabulary and length.

B. Semantic Role Labeling

In NLP, semantic roles denote the semantic functions that words have in a given phrase [13]. For example, in the phrase "John looks in the mirror", the words "looks in" (denoted V) refer to the action being performed. "John" identifies the agent carrying out the action (denoted $A0$), and "the mirror" is the object (denoted $A1$) over which the action is performed.

Semantic role labeling [14] is the task of assigning semantic roles to words or groups of words in a sentence. A variety of tools exist for this task, with different conventions for the associated roles. As an example, for [15], the SEMAFOR parser [16] was used to infer human intention in verbal commands to a robot. In the current paper we used the parser in SENNA [17], which is a software tool distributed under a non-commercial license for academy[2].

After parsing the corpus using SENNA, phrases with semantic roles $A1$ and V being exactly one word each were selected. Each action V was lemmatized into the basic infinitive form since we were not interested in discriminating temporal or other variants of the verbs.

Finally, all pairs $A1$,V that appeared at least seven times were used to create data samples <*object, action*>. This number was found to be a good trade-off for filtering out spurious pairs.

A few examples of phrases and generated sample pairs <*object, action*> are shown in Table IV-B.

Phrase	**Sample pair**
Add flour.	<flour, add>
Crack the egg.	<egg, crack>
Set the mixer on two steps	<mixer, set>
Whip using the mixer	<mixer, use>
Open the oven.	<oven, open>
Enjoy the cake	<cake, enjoy>

Table IV-B Examples of object-action pairs generated from phrases in a recipe.

C. Word Embeddings

Word embeddings refer to a set of unsupervised methods that allow encoding of words as numeric vectors: *wordvectors*. In the numeric space, semantically and syntactically similar words are close if measured through cosine similarity. This is a desirable property for our generative model, as similar objects should show similar affordances.

GloVe [18] and Word2Vec [19] are common approaches to create word embeddings. We trained Word2Vec over YAMC to get embeddings for words that were most specific for our work. The selected dimensionality for the resulting wordvectors was 100. Qualitative evaluations of the resulting vectorial space showed how the computed embeddings were more suited to encode object-action pairs for common objects, than off-the-shelf embeddings.

[1]Obtained at *https://webscope.sandbox.yahoo.com/catalog.php?datatype=l*. Accessed July 3, 2018.

[2]See https://ronan.collobert.com/senna/. Accessed June 30, 2018.

D. Dataset

The words in each generated pair <*object, action*> were converted to wordvectors using the trained Word2Vec model, to provide numeric data to be used in the subsequent modeling. All data was divided into a training set comprising 15263 pairs, and a test set comprising 5088 pairs. Special care was taken to not include identical pairs in both training and test data sets. The data contained $N_O = 2628$ distinct object names and $N_A = 1167$ distinct action names that were collected into a dictionary.

E. Generative Model

We modelled the one-to-many mapping G: *Objects* $\rightarrow$ *Affordances*, using a *Conditional Variational Autoencoder* (CVAE) [20], illustrated in Fig 2.

A CVAE is a trainable generative model that learns a conditional probability distribution $\mathbf{p}(\mathbf{a}|\mathbf{o})$ while keeping a stochastic latent code in its hidden layers. They can be divided into two coupled layers: an encoder and a decoder. The encoder transforms the input distribution into a certain latent distribution $\mathbf{q}_\phi(\mathbf{z}|\mathbf{a},\mathbf{o})$, while the decoder reconstructs the original vectors from its latent representation $\mathbf{z}$ together with the conditioning input $\mathbf{o}$, with output distribution equal to $\mathbf{p}_\varphi(\mathbf{a}'|\mathbf{z},\mathbf{o})$.

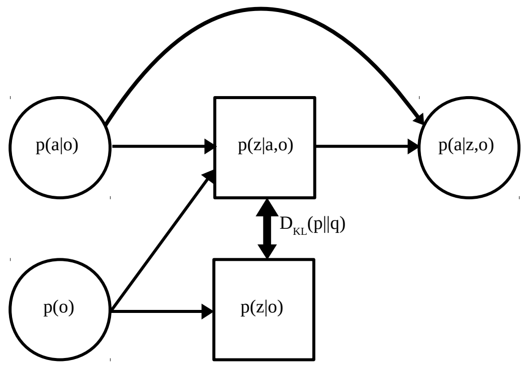

Fig. 2. CVAE with the addition of a parametric prior. Squares represent involved latent distributions.

The encoder's latent layer is regularized to be close to certain parametric prior $\mathbf{q}_\vartheta(\mathbf{z}|\mathbf{o})$. The lower-bound loss function for the CVAE is:

$$L_{CVAE} = \mathbb{E}[\log p_\varphi(a'|z,o)] - \lambda D_{KL}(q_\phi(z|a,o)||q_\vartheta(z|o)) \quad (1)$$

The first term accounts for how good the autoencoder reconstructs the input given its latent representation. The second term regularizes the hidden latent space to be close to a certain posterior distribution. The factor λ balances how regularization is applied during learning. Starting from zero it is linearly grown up to one as the learning epochs advance. This technique addresses the *vanishing latent variable problem* and is referred to as KL annealing [21].

φ, ϕ, ϑ denotes the three disjoint sets of parameters of the components that are simultaneously involved in learning. More specifically, they represent set of weights for the three neural network composing the CVAE. The CVAE was trained using the training set generated as described above, and was implemented using the Keras [22] library for Python.

V. Evaluation

By inputting the name O of an object, and repeatedly sampling the CVAE, we obtain the same number of names for possible actions A. As described above, the sampling follows the estimated conditional probabilities $p(A|O)$. Hence, actions with high probability are output more frequently than actions with low probability. Since the CVAE outputs actions in numeric wordvector format, all output actions are "rounded" to the closest action word appearing in the dictionary. This is equivalent to a *K-NN* classification with $K = 1$, and the final output is the dictionary word belonging to the nearest neighboring wordvector. A few examples of the most probable generated actions for a given input objects is shown in Table V.

Input	Output
door	open, pull, put, loosen, grab, clean, leave, get, slide, shut
egg	hatch, poach, implant, lay, crack, peel, spin, whip, float, cook
wine	pour, add, mix, dry, rinse, melt, soak, get, use, drink
book	read, get, write, purchase, find, use, sell, print, buy, try
cat	declaw, deter, bathe, bath, spay, pet, scare, feed, attack
money	loan, inherit, double, owe, withdraw, save, waste, cost, earn, donate
knife	scrape, cut, brush, chop, use, roll, pull, remove, slide, rub
information	review, request, access, verify, present, obtain, identify, provide, submit, retain
body	trick, adapt, tone, adjust, recover, starve, cleanse, respond, flush, exercise
place	switch, prepare, hide, rent, start, own, guess, travel, avoid, suggest

Table V Examples of actions generated by the CVAE. For every input object we show the 10 most probable outputs, sorted from high to low probability.

Evaluation of generative models is in general seen as a difficult task [23]–[25], and one suggestion is that they should be evaluated directly with respect to the intended usage [23]. In that spirit we evaluated how often our model produced affordances that were correct in the sense that they matched test data.

Since the CVAE produces different results each time it is sampled, it was first sampled several times to estimate a probability distribution $p(A|O)$, to be compared with a similar estimation based on relative frequencies for the test data. A performance measure *Accuracy*, with values between 0 and 1, quantifies the similarity between these two distributions and was computed by the following algorithm.

1) $M \leftarrow 0$.
For each of the N test samples $< O_j, A_j >, j = 1, \ldots, N$, repeat Steps 2-4:
2) Input object O_j to the CVAE and sample it 1000 times to estimate a probability distribution over possible actions for object O_j. Denote the set with the L most probable actions A_C.
3) As ground truth compute, for all actions A, $p(A|O_j) = N(A, O_j)/N(O_j)$, where $N(A, O_j)$ is the number of test data samples $< object, action >$ with $object = O_j$ and $action = A$, and $N(O_j)$ is the number of samples with $object = O_j$. For the resulting distribution, denote the set of the L most probable actions A_F.
4) If any action in A_C appears in A_F : $M \leftarrow M + 1$ (this corresponds to a notion of the CVAE output being "correct" for object O_j).
5) *Accuracy* $\leftarrow M/N$.

As benchmark method we generate actions using a random distribution, with $p(A|O_j) = \frac{1}{N_A}$ for all actions belonging to the training set, 0 otherwise.

We evaluated the benchmark in a similar fashion as described above, by replacing the CVAE for generation of the probability distribution in Step 2. Accuracy computed on the test set for CVAE and the benchmark are presented in Table V, for varying values of L.

L	CVAE	Random
1	0.099	0
5	0.552	0.09
10	0.781	0.322
15	0.862	0.498
20	0.903	0.773

Table V *Accuracy* for the CVAE and Random distributions, calculated as described above.

VI. Conclusions

We presented a novel generative model for text-based affordance generation by employing a Conditional Variational Autoencoder (CVAE). The presented preliminary results show that the model outperforms the benchmark method in generating possible actions for an input object.

For a given object, the action with highest probability to be generated by CVAE was most probable in test data 10% of the time. Given that the data set contained 1167 distinct action names, these results are quite satisfying. Considering more than just the action with highest probability, the accuracy for CVAE increases fast. For example, for $L = 5$, at least one correct action was output in 9% of the cases for the random distribution, and in 55% of all cases for CVAE.

As future work we will address several open questions:

- The used CVAE model is a complex architecture with several meta parameters and design choices. We will further investigate alternative designs, and use the performance measures to, possibly automatically, find optimal parameters.
- The relevance of using corpora like YAMC to generate affordances for HRI has to be investigated further. The difference between usage of language in human-robot

dialogue and in general corpora may affect accuracy, and alternative corpora could be considered.

- The generalization ability, i.e. performance for objects not present in the training data, will be investigated further. Successful generalization ability means that the method has true predictive power and does more than memorizing training data.
- Training with domain specific data will be investigated. As mentioned in the introduction, real object affordances typically depend on the environment, and better performance may be achieved by implicitly defining a specific domain (such as kitchen environments) and learn affordances with objects and actions relevant for that domain only.
- Alternative performance measures will be examined, for example based on a distance metric applied to the distributions in step 2 and 3 above. Explicit ways to assess the model's generalization ability will also be developed.
- Finally, envisioning robots as embodied agents, we will explore how affordances generation can be biased by taking into account specific perceptual and bodily abilities. The YAMC corpus expresses actions available to animals and humans, but can a robot, after a long day, just relax on the sofa?

Acknowledgement

This work has received funding from the European Union's Horizon 2020 research and innovation program under the Marie Skłodowska-Curie grant agreement No 721619 for the SOCRATES project.

References

[1] J. Gibson, *The theory of affordances in Perceiving, Acting, and Knowing. Towards an Ecological Psychology.* Hoboken, NJ: John Wiley & Sons Inc., 1977.

[2] M. Çakmak Mehmet R. Doğar, E. Uur, and E. Şahin, "Affordances as a framework for robot control," 2007.

[3] C. Schneider and J. Valacich, *Enhancing the Motivational Affordance of Human–Computer Interfaces in a Cross-Cultural Setting*. Heidelberg: Physica-Verlag HD, 2011, pp. 271–278. [Online]. Available: https://doi.org/10.1007/978379082632631

[4] T. E. Horton, A. Chakraborty, and R. St. Amant, "Affordances for robots: A brief survey," vol. 3, pp. 70–84, 12 2012.

[5] E. Bonchek Dokow and G. Kaminka, "Towards computational models of intention detection and intention prediction," vol. 28, 01 2013.

[6] T. Hellström and S. Bensch, "Understandable robots - what, why, and how," *Paladyn, Journal of Behavioral Robotics*, Accepted for publication.

[7] S. Bensch, A. Jevtić, and T. Hellström, "On interaction quality in human-robot interaction," in *International Conference on Agents and Artificial Intelligence (ICAART)*, 2017, pp. 182–189.

[8] Y.-W. Chao, Z. Wang, R. Mihalcea, and J. Deng, "Mining semantic affordances of visual object categories." in *CVPR*. IEEE Computer Society, 2015, pp. 4259–4267. [Online]. Available: http://dblp.uni-trier.de/db/conf/cvpr/cvpr2015.html

[9] K. Narasimhan, T. D. Kulkarni, and R. Barzilay, "Language understanding for text-based games using deep reinforcement learning," in *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, EMNLP 2015, Lisbon, Portugal, September 17-21, 2015*, 2015, pp. 1–11. [Online]. Available: http://aclweb.org/anthology/D/D15/D15-1001.pdf

[10] V. Mnih, K. Kavukcuoglu, D. Silver, A. A. Rusu, J. Veness, M. G. Bellemare, A. Graves, M. Riedmiller, A. K. Fidjeland, G. Ostrovski, S. Petersen, C. Beattie, A. Sadik, I. Antonoglou, H. King, D. Kumaran, D. Wierstra, S. Legg, and D. Hassabis, "Human-level control through deep reinforcement learning," *Nature*, vol. 518, no. 7540, pp. 529–533, Feb. 2015. [Online]. Available: http://dx.doi.org/10.1038/nature14236

[11] L. Antanas, O. A. Can, J. Davis, L. D. Raedt, A. Loutfi, A. Persson, A. Saffiotti, E. Ünal, D. Yuret, and P. Z. dos Martires, "Relational symbol grounding through affordance learning : An overview of the reground project," 2017.

[12] A. Ruggeri and L. D. Caro, "How affordances can rule the (computational) world," in *AIC@AI*IA*, 2013.

[13] X. Carreras and L. Márquez, "Introduction to the conll-2004 shared task: Semantic role labeling," 2004.

[14] D. Gildea and D. Jurafsky, "Automatic labeling of semantic roles," *Comput. Linguist.*, vol. 28, no. 3, pp. 245–288, Sep. 2002. [Online]. Available: http://dx.doi.org/10.1162/089120102760275983

[15] A. Sutherland, S. Bensch, and T. Hellström, "Inferring robot actions from verbal commands using shallow semantic parsing," in *Proceedings of the 17th International Conference on Artificial Intelligence ICAI'15*, H. Arabnia, Ed., July 2015, pp. 28–34.

[16] D. Das, N. Schneider, D. Chen, and N. A. Smith, "Probabilistic frame-semantic parsing," in *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, ser. HLT '10. Stroudsburg, PA, USA: Association for Computational Linguistics, 2010, pp. 948–956. [Online]. Available: http://dl.acm.org/citation.cfm?id=1857999.1858136

[17] R. Collobert, J. Weston, L. Bottou, M. Karlen, K. Kavukcuoglu, and P. Kuksa, "Natural language processing (almost) from scratch," *J. Mach. Learn. Res.*, vol. 12, pp. 2493–2537, Nov. 2011. [Online]. Available: http://dl.acm.org/citation.cfm?id=1953048.2078186

[18] J. Pennington, R. Socher, and C. D. Manning, "Glove: Global vectors for word representation," in *In EMNLP*, 2014.

[19] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient estimation of word representations in vector space," *CoRR*, vol. abs/1301.3781, 2013. [Online]. Available: http://dblp.uni-trier.de/db/journals/corr/corr1301.html

[20] C. Doersch, "Tutorial on variational autoencoders," 2016, cite arxiv:1606.05908. [Online]. Available: http://arxiv.org/abs/1606.05908

[21] S. R. Bowman, L. Vilnis, O. Vinyals, A. M. Dai, R. Józefowicz, and S. Bengio, "Generating sentences from a continuous space," in *Proceedings of the 20th SIGNLL Conference on Computational Natural Language Learning, CoNLL 2016, Berlin, Germany, August 11-12, 2016*, 2016, pp. 10–21. [Online]. Available: http://aclweb.org/anthology/K/K16/K16-1002.pdf

[22] F. Chollet *et al.*, "Keras," https://keras.io, 2015.

[23] L. Theis, A. van den Oord, and M. Bethge, "A note on the evaluation of generative models," *CoRR*, vol. abs/1511.01844, 2015.

[24] D. Hendrycks and S. Basart, "A quantitative measure of generative adversarial network distributions," 2017.

[25] A. Kumar, A. Biswas, and S. Sanyal, "ecommercegan : A generative adversarial network for e-commerce," *CoRR*, vol. abs/1801.03244, 2018.