Unsupervised Inference of Object Affordance from Text Corpora

Michele Persiani
Department of Computing Science
Umeå University
Umeå, Sweden
michelep@cs.umu.se

Thomas Hellström
Department of Computing Science
Umeå University
Umeå, Sweden
thomash@cs.umu.se

Abstract

Affordances denote actions that can be performed in the presence of different objects, or possibility of action in an environment. In robotic systems, affordances and actions may suffer from poor semantic generalization capabilities due to the high amount of required hand-crafted specifications. To alleviate this issue, we propose a method to mine for object-action pairs in free text corpora, successively training and evaluating different prediction models of affordance based on word embeddings.

Affordance; Natural Language Processing; Robotics; Intention Recognition; Conditional Variational Autoencoder;

1 Introduction

The term "affordance" was introduced by the American psychologist Gibson (Greeno, 1994) to describe what an animal can do in a given environment. It has since then been extensively utilized, interpreted, and re-defined (see (Çakmak Mehmet R. Doğar et al., 2007) for an overview) in fields such as robotics (Zech et al., 2017), human-computer-interaction (Schneider and Valacich, 2011) or human-robot-interaction (HRI) (E. Horton et al., 2012). Several interpretations for affordance exist in the literature, we use the term in a loose way to denote actions that can be performed with objects. As a simplified first approach we 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 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 and III we give a brief literature review on affordances from different fields. 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.

2 Affordances

When learned, the mapping G can be used in several ways in artificial systems, for example, by visually identifying objects in the environment or in the verbal dialogue with the user, suitable actions can be inferred by applying G to the observed objects. The objects and actions can then be used for shared planning or intent recognition (Bonchek-Dokow and Kaminka, 2014), thus allowing closer cooperations with the user.

For example, the mapping G may be used in a robot to decide how it should act within a given context that affords certain actions. In HRI, a service robot may for example suggest its user to read a book after it being visually detected or mentioned. Affordances may also be useful for object disambiguation. When a robot is told to "pick it up!", the robot only has to consider objects that are "pickable" in the current scene (E. Horton et al., 2012). Alternatively, affordances may be used to infer the human's intention, which may guide the robot's behavior (Bonchek-Dokow and Kaminka, 2014). If a user expresses will of talking to his children, a robot may infer that the user want to call them, and suggest making a phone call. 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 (Hellström and Bensch, 2018), and thereby increase interaction quality (Bensch et al., 2017).

Classical planning require knowledge about the actions that are possible in a certain situation, i.e. its afforded actions. For simple scenarios, it could suffice to enumerate all objects in the current scene, to later score their affordances and finally select the most promising to activate.

Affordances can be organized in a hierarchy, thus exposing relations or subsumptions between actions (Antanas et al., 2017; Zech et al., 2017). Assuming that a door affords the action *open*, it is clear that in order to be opened, several actions must be performed in a precise sequence (e.g. turn the handle, push the handle). Objects that offer the same grouped sequence of actions could then be represented as similar in a latent

Proceedings of the 22nd Nordic Conference on Computational Linguistics (NoDaLiDa), pages 115–120
Turku, Finland, 30 September – 2 October,

space.

Antanas et al. (Antanas et al., 2017) relate affordances to the symbol grounding problem. In the attempt of grounding the object *door*, we could say it is an object affording *open, close,* etc.: it is grounded over those actions. Further stress is also put on describing affordances as relations between objects and qualities of objects. A pear can be cut with a knife because it's soft, while a hard surface could instead be just scraped. The blade of the knife affords cut only if used in conjunction with soft enough objects. This relational hypothesis is supported by neuroscience studies showing how motor cortices are activated faster if a tool is presented together with another contextual object, rather than alone (Borghi et al., 2012).

Depending on the desired level of abstraction, affordances can be represented on different levels (Zech et al., 2017). We broadly distinct two categories, namely symbolic and sub-symbolic. In symbolic form, affordances are expressed through symbols, and every symbol enjoys certain relations with other symbols. This usually gives rise to the possibility of having a knowledge-base, containing entities such as *affords(knife, cut, pear)*, and organizing them in a graph. Sub-symbolic encodings (such as through neural networks) are instead useful to obtain percepts (Persiani et al., 2018). By clustering the perceptual/procedural space, we obtain entities (the centroids) that may or may not be utilizable as symbols, depending on the nature of the input space and subsequent calculations.

Inference of affordances from images (Zech et al., 2017) is an example of sub-symbolic approach. This is related to object recognition/segmentation, and corresponds to associating afforded actions to different visual regions of the object. Recognized affordance regions can be used for object categorization (Dag et al., 2010). For example, in a kitchen environment objects having two graspable regions could be identified as pans or containers. This is especially useful for robotic manipulation tasks (Yamanobe et al., 2017): a planner for a gripper must have knowledge about the geometric shape of the parts that can actually be grasped.

Ruggeri and Di Caro (Ruggeri and Caro, 2013) propose methodologies on how to build ontologies of affordances, also linking them to mental models and language. If we think at the phrase "The squirrel climbs the tree", we can create a mental image for it, imaging how it reaches the top. If an elephant climbs the tree instead, surely some semantic mismatch will soon arise. The mental model doesn't fit because the tree doesn't afford climbing to the elephant. The opposite might instead apply for scenarios like *"Lifting a trunk"*.

3 Related work

Unsupervised extraction of object-action pairs from free text corpora has been a relevant point in recent Natural Language Processing (NLP) research. Differently from the other methods, corpora can be mined by different techniques with the goal of finding in an unsupervised manner relationships between objects, properties of objects and actions. Chao et al. (Chao et al., 2015) show how in NLP objects and actions can be connected through the introduction of a latent space. They argue that building such a space is equivalent to obtaining a co-occurrence table, referred to as the "affordance matrix". In their approach every object-action word pair is scored through a similarity measure in the latent space, and only the pairs over a certain threshold are retained as signaling the presence of affordance. The affordance matrix, together with other automatically extracted properties and relations (altogether referred to as commonsense knowledge), such as expected location for objects, can be then used to build PKS (Planning with Knowledge and Sensing (Petrick and Bacchus, 2002)) planners (Petrick and Bacchus, 2002; Kaiser et al., 2014).

In (Chen et al., 2019), the authors map semantic frames to robot action frames using semantic role labeling, showing how a language model can yield the likelihood of possible arguments. Their proposed *Language-Model-based Commonsense Reasoning* (LMCR) will give as more probable an instruction such as *"Pour the water in the glass."* rather than *"Pour the water in the plate."*. The LMCR is trained over semantic frames by using mined knowledge about semantic roles and can be used to rank robot action frames by testing the different combinations of the available objects. When searching for an object where to pour water, the LMCR is used to rank the available objects.

4 Method

We trained a generative model for the one-to-many mapping $G : Objects \rightarrow Affordances$ using pairs of the type $<$*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, as is illustrated in Fig. 1. The model allows to rank the different affordances for a given object name, as names of actions that can per performed on it. By employing a neural network model rather than a tabular model we investigate whether wordvectors encoding allows for the generalization the in mapping object-action.

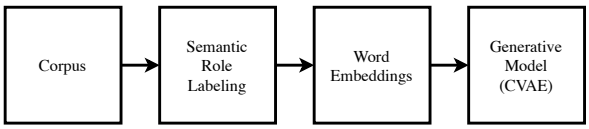

Figure 1: Steps taken to obtain the generative model.

4.1 Corpus

As data source we used the *Yahoo! Answers Manner Questions* (YAMC) dataset[1] containing 142,627 ques-

[1] Obtained at *https://webscope.sandbox.yahoo.com/catalog.php?datatype=l*. Accessed May 16, 2019.

tions 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 all questions, selected for their linguistic properties such as good quality measured in terms of vocabulary and length.

This specific corpus was selected due to the nature of its content. Our hypothesis is that being a collection of QA regarding daily living, the actions and objects being mentioned are more closely related to affordance than the ones in other corpora such as Wikipedia.

4.2 Semantic Role Labeling

In NLP, semantic roles denote the semantic functions that words have in a given phrase (Carreras and Márquez, 2004). 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 $A1$) being target of the action.

Semantic role labeling (Gildea and Jurafsky, 2002) 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 (Sutherland et al., 2015), the SEMAFOR parser (Das et al., 2010) was used to infer human intention in verbal commands to a robot. In the current paper we used the parser in SENNA (Collobert et al., 2011), which is a software tool distributed with a non-commercial license.

After parsing the corpus using SENNA, phrases with semantic roles $A1$ and V of size one 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 filter out spurious pairs. A fictional example illustrating possible generated sample pairs *<object, action>* is shown in Table 4.2.

Phrase	**<object, action>**
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 4.2 Examples of object-action pairs generated from phrases in a recipe.

Objects and actions are further filtered based on a *concreteness* value (Kaiser et al., 2014), that correspond to how close they are to being physical entities rather than abstract ones. To do so, for every sense of every object we navigate the WordNet entity hierarchy and retain that sense only if it is a child node of *physical entity*. Only objects with a ratio of physical senses above a certain threshold are kept. We apply the same procedure to actions but regarding them as physical if they are child of *move, change, create, make*.

4.3 Dataset

The words in each generated pair *<object, action>* were converted to wordvectors to provide numeric data to be used in the subsequent experiments. All data was divided into a training set comprising of 734,002 pairs, and a test set comprising 314,572 pairs. Special care was taken to include different objects in training and test data sets. This would allow us to test in a more aggressive way the generalization capabilities of the trained models. The data contained $N_O = 33{,}655$ distinct object names and $N_A = 11{,}923$ distinct action names.

4.4 Word Embeddings

Word embeddings (Collobert et al., 2011) model every word x as a dense vector W_x. Words that co-occur often in the corpus have similar associated vectors, and enjoy linear or non-linear properties reflecting semantic or syntactic relationships such as analogies(Drozd et al., 2016). $W_{king} - W_{man} \approx W_{queen} - W_{woman}$ (semantic analogy), or $W_{lift} - W_{lifted} \approx W_{drop} - W_{dropped}$ (syntactic analogy). Similarity of words is often measured though cosine distance of the vectors. For a review on analogy tests see (Finley et al., 2017).

GloVe (Pennington et al., 2014) and Word2Vec (Mikolov et al., 2013) are common approaches to create word embeddings. We trained Word2Vec over YAMC to get embeddings for words that were most specific for our dataset. The selected dimensionality for the wordvectors was 100.

4.5 Generative Model

We compare three different models in how good they are in predicting $P(A|O)$ provided the evidence in the data. A *Conditional Variational Autoencoder* (CVAE) (Doersch, 2016) trained on off-the-shelf GloVe embeddings with dimensionality 200, a CVAE trained on word2vec embeddings fitted on the YAMC dataset, a *K-NN* model.

4.5.1 Conditional Variational Autoencoder

A CVAE is a trainable generative model that learns a conditional probability distribution $P(A|O)$ 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 $Q_\phi(z|A, O)$, while the decoder reconstructs the original vectors from its latent representation z together with the conditioning input o, with output distribution equal to $P_\varphi(A'|z, o)$.

The encoder's latent layer is regularized to be close to certain parametric prior $Q_\vartheta(z|O)$. 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 (Bowman et al., 2016).

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 (Chollet et al., 2015) library for Python.

In order to search for a most direct relationship between objects and actions in wordvectors space, we keep the autoencoder with one hidden layer in both encoder and decoder. Nevertheless, nonlinearity of the output function of the hidden units proved necessary to yield a high accuracy. We set the dropout value for the hidden layers of the autoencoder to 0 (no features are dropped during the training phase), as this setting proved better performance in all of the experiments.

4.5.2 Nearest Neighbor

For a given input object o, the Nearest Neighbors model predicts $P(A|o)$ as $P(A|o')$, where o' is the closest object in training data. o' is found by cosine similarity of the wordvectors o and o'. $P(a'|o')$ is computed as $N(a', o')/N(o')$, where $N(.)$ is the counting of occurrences in training data.

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
body	trick, adapt, tone, adjust, recover, starve, cleanse, respond, flush, exercise

Table 4.5.2 Examples of actions generated by the CVAE. For every input object the 10 most probable outputs are sorted from high to low probability.

5 Evaluation

By sampling the model, we obtain names of possible actions A. As described above, the sampling follows the estimated conditional probabilities $P(A|O)$. Hence, actions with high probability are generated more frequently than actions with low probability. Since the CVAE outputs actions in numeric wordvector format, these actions are "rounded" to the closest action word appearing in the dictionary. This is equivalent to a *K-NN* classification with $K = 1$. A few examples of the most probable generated actions for CVAE are shown in Table 4.5.2.

Evaluation of generative models is in general seen as a difficult task (Theis et al., 2015; Hendrycks and Basart, 2017; Kumar et al., 2018), and one suggestion is that they should be evaluated directly with respect to the intended usage (Theis et al., 2015). In that spirit we evaluated how often our models produced affordances that were correct in the sense that they exactly matched test data with unseen objects. For a model $P_k(A|O)$ we define an accuracy measure as follows:

Algorithm 1 Accuracy computation of a model $P_k(A|O)$

procedure ACCURACY($P_k(A|O), l, m$,test_set)
  $s \leftarrow$ size(test_set)
  $x \leftarrow 0$
  for $(o_i, a_i) \in$ test_set **do**
    $A_o \leftarrow P_k(A|o_i)$ ▷ Output of the k-th model, sampled m times, with $m >> 1$
    SORT(A_o) ▷ The list of actions is sorted in descending order
    $sel_{il} \leftarrow$ FIRST(A_o, l) ▷ The most frequent actions up to l are kept
    if $a_i \in sel_{il}$ **then**
      $x \leftarrow x + 1$ ▷ x is increased when a_i is contained in sel_{il}
    end if
  end for
  $accuracy_k \leftarrow \frac{x}{s}$
end procedure

This measure tests how good a model replicates test data, and is meant to be a quantitative evaluation. Two different CVAEs are evaluated, the first with data encoded with GloVe-200 embeddings, the second with word2vec embeddings obtained over YAMC. We evaluated CVAE, K-NN and a baseline model by the described procedure. As baseline model we used a prior $P(A|O) = P(A)$, that is the probability distribution of actions over all objects. For every action a, $P(a) = N_a/N_{tot}$, where N_a is the number of times a appeared in the dataset. Accuracy computed on the test set for the different models are presented in Figure 2.

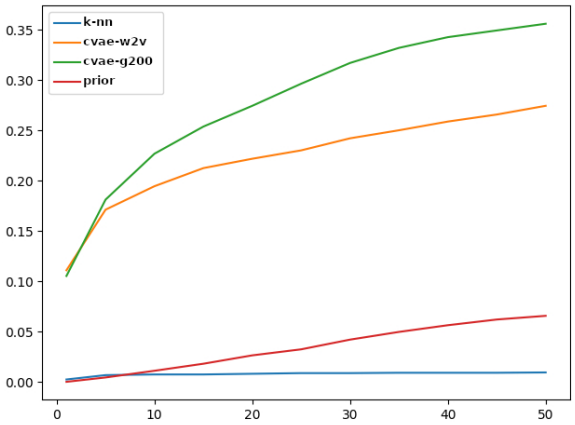

Figure 2: Computed accuracy for the different models. The X axis shows different percentages of retained output actions, starting from the most probable ones (parameter L). The Y axis shows the obtained accuracy.

The K-NN model fails to generalize the task: jumping to the closest object and outputting the empirical probability for it yield performances just above zero, also lower to the baseline.

We explain the K-NN performance as being this low due to the fact that similarity of objects (using cosine distance) does not encode similarity of associated actions. Supporting this hypothesis there is also the necessity of having nonlinear layers in the autoencoder in order to achieve high accuracy values. From this consideration we conclude that in word embedding space the mapping object-action is non-linear using the off-the-shelf embedding features.

The two CVAEs performance is higher, reaching a score of 0.35 with the off-the-shelf wordvectors. Additionally, we observed that training word2vec embeddings over the corpus lead to overfitting: performance computed over the test set comprising unseen objects is lower than the performance obtained with general purpose wordvectors.

6 Conclusions

With the goal of mining knowledge about affordance from corpora, we presented an unsupervised method that extracts object-action pairs from text using Semantic Role Labeling. The extracted pairs were used to train different models predicting $P(A|O)$: two Conditional Variational Autoencoders and one K-NN model. The presented results show that, on unseen objects, a CVAE trained on off-the-shelf wordvectors performs significantly better than the other tested models. Furthermore, we show how the K-NN model fails to generalize on our specific benchmark task, having performance even lower than the baseline model.

Knowledge about affordance, even in simple forms such as a object-action mapping, is relevant for applications such as inference of intent or robot planning. In robotics, planning requires a high amount of specifications inserted in the domain description, usually resulting in most of the decision rules being hand-crafted. With this paper, we present an algorithm allowing the leverage of knowledge about affordance present in corpora, thus allowing for a method of generating of at least a part the domain automatically.

Future work related to this research will be about improving the method by which the object-action pairs are mined, followed by reasearch on how this knowledge can be transformed to be used for robotic planning and intent recognition problems.

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

Laura Antanas, Ozan Arkan Can, Jesse Davis, Luc De Raedt, Amy Loutfi, A. Persson, Alessandro Saffiotti, Emre Ünal, Deniz Yuret, and Pedro Zuidberg dos Martires. 2017. Relational symbol grounding through affordance learning : An overview of the reground project. In *International Workshop on Grounding Language Understanding (GLU). Stockholm, Sweden: Satellite of INTERSPEECH.*

Suna Bensch, Alexander Jevtić, and Thomas Hellström. 2017. On interaction quality in human-robot interaction. In *International Conference on Agents and Artificial Intelligence (ICAART)*, pages 182–189.

Elisheva Bonchek-Dokow and Gal A Kaminka. 2014. Towards computational models of intention detection and intention prediction. *Cognitive Systems Research*, 28:44–79.

Anna M Borghi, Andrea Flumini, Nikhilesh Natraj, and Lewis A Wheaton. 2012. One hand, two objects: Emergence of affordance in contexts. *Brain and cognition*, 80(1):64–73.

Samuel R. Bowman, Luke Vilnis, Oriol Vinyals, Andrew M. Dai, Rafal Józefowicz, and Samy Bengio. 2016. 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*, pages 10–21.

Xavier Carreras and Lluìs Márquez. 2004. Introduction to the conll-2004 shared task: Semantic role labeling.

Yu-Wei Chao, Zhan Wang, Rada Mihalcea, and Jia Deng. 2015. Mining semantic affordances of visual object categories. In *CVPR*, pages 4259–4267. IEEE Computer Society.

Haonan Chen, Hao Tan, Alan Kuntz, Mohit Bansal, and Ron Alterovitz. 2019. Enabling robots to understand incomplete natural language instructions using commonsense reasoning. *CoRR*.

François Chollet et al. 2015. https://keras.io Keras.

Ronan Collobert, Jason Weston, Léon Bottou, Michael Karlen, Koray Kavukcuoglu, and Pavel Kuksa. 2011. Natural language processing (almost) from scratch. *J. Mach. Learn. Res.*, 12:2493–2537.

Nilgun Dag, Ilkay Atil, Sinan Kalkan, and Erol Sahin. 2010. Learning affordances for categorizing objects and their properties. In *2010 20th International Conference on Pattern Recognition*, pages 3089–3092. IEEE.

Dipanjan Das, Nathan Schneider, Desai Chen, and Noah A. Smith. 2010. Probabilistic frame-semantic parsing. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, HLT '10, pages 948–956, Stroudsburg, PA, USA. Association for Computational Linguistics.

Carl Doersch. 2016. Tutorial on variational autoencoders.

Aleksandr Drozd, Anna Gladkova, and Satoshi Matsuoka. 2016. Word embeddings, analogies, and machine learning: Beyond king-man+ woman= queen. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pages 3519–3530.

Thomas E. Horton, Arpan Chakraborty, and Robert St. Amant. 2012. Affordances for robots: A brief survey. In *Avant. Pismo Awangardy Filozoficzno-Naukowej, 2*, volume 3, pages 70–84.

Gregory Finley, Stephanie Farmer, and Serguei Pakhomov. 2017. What analogies reveal about word vectors and their compositionality. In *Proceedings of the 6th Joint Conference on Lexical and Computational Semantics (*SEM 2017)*, pages 1–11. Association for Computational Linguistics.

Daniel Gildea and Daniel Jurafsky. 2002. Automatic labeling of semantic roles. *Comput. Linguist.*, 28(3):245–288.

James G Greeno. 1994. Gibson's affordances. *Psychological Review*, 101(2):336–342.

Thomas Hellström and Suna Bensch. 2018. Understandable robots - what, why, and how. *Paladyn, Journal of Behavioral Robotics*, 9(1).

Dan Hendrycks and Steven Basart. 2017. A quantitative measure of generative adversarial network distributions.

Peter Kaiser, Mike Lewis, Ronald PA Petrick, Tamim Asfour, and Mark Steedman. 2014. Extracting common sense knowledge from text for robot planning. In *2014 IEEE International Conference on Robotics and Automation (ICRA)*, pages 3749–3756. IEEE.

Ashutosh Kumar, Arijit Biswas, and Subhajit Sanyal. 2018. ecommercegan : A generative adversarial network for e-commerce. *arXiv preprint arXiv:1801.03244*.

Maya Çakmak Mehmet R. Doğar, Emre Uur, and Erol Şahin. 2007. Affordances as a framework for robot control. In *Proceedings of the 7th international conference on epigenetic robotics, epirob07*.

Tomas Mikolov, Kai Chen, Gregory S. Corrado, and James A. Dean. 2013. Efficient estimation of word representations in vector space. *CoRR*.

Jeffrey Pennington, Richard Socher, and Christopher D. Manning. 2014. Glove: Global vectors for word representation. In *In EMNLP*.

Michele Persiani, Alessio Mauro Franchi, and Giuseppina Gini. 2018. A working memory model improves cognitive control in agents and robots. *Cognitive Systems Research*, 51:1–13.

Ronald PA Petrick and Fahiem Bacchus. 2002. A knowledge-based approach to planning with incomplete information and sensing. In *AIPS*, volume 2, pages 212–222.

Alice Ruggeri and Luigi Di Caro. 2013. How affordances can rule the (computational) world. In *AIC@AI*IA*.

C. Schneider and J. Valacich. 2011. *Enhancing the Motivational Affordance of Human–Computer Interfaces in a Cross-Cultural Setting*, pages 271–278. Physica-Verlag HD, Heidelberg.

Alexander Sutherland, Suna Bensch, and Thomas Hellström. 2015. Inferring robot actions from verbal commands using shallow semantic parsing. In *Proceedings of the 17th International Conference on Artificial Intelligence ICAI'15*, pages 28–34.

Lucas Theis, Aäron van den Oord, and Matthias Bethge. 2015. A note on the evaluation of generative models. *CoRR*.

Natsuki Yamanobe, Weiwei Wan, Ixchel G Ramirez-Alpizar, Damien Petit, Tokuo Tsuji, Shuichi Akizuki, Manabu Hashimoto, Kazuyuki Nagata, and Kensuke Harada. 2017. A brief review of affordance in robotic manipulation research. *Advanced Robotics*, 31(19-20):1086–1101.

Philipp Zech, Simon Haller, Safoura Rezapour Lakani, Barry Ridge, Emre Ugur, and Justus Piater. 2017. Computational models of affordance in robotics: a taxonomy and systematic classification. *Adaptive Behavior*, 25(5):235–271.