Understanding Each-Other: Engineering Challenges and Opportunities for Users and Systems in the Deep Learning Era

Lucio Davide Spano[0000−0001−7106−0463]

Department of Mathematics and Computer Science, University of Cagliari, Italy
davide.spano@unica.it

Abstract. In this paper, we discuss the impact of Deep Learning (DL) techniques in the present and future of the interactive system engineering. On the one hand, the support for more complex vocabularies offers opportunities in better shaping the communication between the user and the system. On the other hand, we identify challenges related to the lack of transparency and explainability in the trained models, which have a negative impact on system understanding for both developers and users.

Keywords: user interface engineering · deep learning · explainable user interface · intelligent user interface · classification · training · input · output

1 Introduction

In the later years, Deep Learning techniques provided the research community with robust solutions for challenging problems in different fields, such as object and speech recognition, text generation and analysis, shape understanding etc. [9, 14, 7, 15]. A combination of factors supported their effectiveness, including the advancement of processing techniques for big data, the availability of public dataset, the technological evolution of GPUs and the development of Neural Networks. All the wicked problems that found an acceptable solution through such advancements share a common trait: the variability of the input. Such characteristic makes really hard for developers to code functions returning a robust output.

DL techniques contributed to fix many usability problems, for instance in the field of speech recognition [10]. However, from an interaction engineering point of view, their strength against the variability is also their weakness as part of a system: Deep Learning models are either complete black boxes or components that programmers cannot completely predict.

Using a pre-trained model for classifying or analysing data is an example of a black-box component. Indeed, many highly accurate and robust pre-trained models are available for different tasks, and they relieve the development team from the burden of creating their own Neural Network model. However, a pre-trained model does not support the inspection by developers or by end-users.

This may cause unwanted behaviours, for instance when the considered application uses frequently a rare word (e.g., a jargon word in speech recognition) or by a specific user.

Another common case is having a model developed or trained by the same team that creates the application. Developers have more control over the input distribution and they may fine-tune the model on the application at hand. However, the learnt function remains difficult to understand for developers, and this creates difficulties in tracking unwanted behaviours. The problem is even worse if we consider approaches based on reinforcement learning: the system learns guided by human feedback (reward), which may be completely out of the engineer's control.

In this paper, we will refer to the interaction framework proposed by Abowd and Beale [1], one of the most famous interaction frameworks in the HCI literature, as a guide for identifying opportunities and challenges in including such intelligent modules in an interactive application. We will first focus on the process of passing information from the *User* to the *System* and then on the reverse path.

2 Background

In this section we summarise the main concepts introduced in the interaction framework by Abowd and Beale [1]. The framework consists of four components depicted in Figure 1: the *System* (S), the *User* (U), the *Input* (I) and the *Output* (O). Each component has its own language: the *core* language of the System, the *task* language of the User and the two languages of components representing the interface: the *input* and *output*.

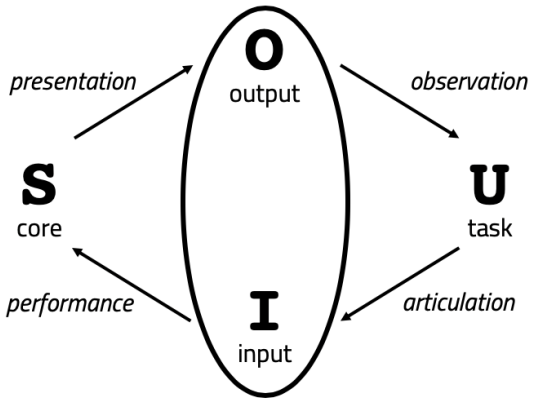

Fig. 1. The interaction framework by Abowd and Beale [1].

The interaction process requires four translations between these languages. Each translation corresponds to one of the four phases of an interaction cycle (*articulation, performance, presentation, observation*), which we can summarise as follows. The *User* formulates her goals in the task language and *articulates* them in the *input* language. The interface translates such information into operations to be *performed* by the System. After the computation, the System changes its state and it *presents* such information using concepts or features of the *output* language. Finally, the User *observes* such output and assesses the result according to her original goal.

In this paper, we analyse the impact of exploiting Deep Learning techniques in each one of these phases.

3 Understanding the user

In this section, we will analyse the engineering challenges and opportunities in the *articulation* and *performance* phases.

On the one hand, we are positive that Deep Learning nears the expressive gap between the *Input* and the *Task* language. Indeed, DL is the technology that grounds the communication between users and the system using natural language, hand or body gestures or a combination of modalities. This is an important opportunity that is currently fostering the development of conversational interfaces, guided both by speech recognition and/or natural language processing.

On the other hand, an increase in the *input* language expressiveness increases also its complexity and, consequently, it causes challenges in the *performance* phase (see Figure 1). Higher complexity means a larger vocabulary for expressing the user's intent, and higher complexity in the translation to the *core* language. Deep Learning may help again in this task (e.g. in Natural Language Understanding), but this reduces the developer control over the overall process and the interface ability to provide intermediate feedback.

The following sections provide two sample modalities that had a different level of success in applying Deep Learning techniques for their intrinsic characteristics.

3.1 Conversational interfaces

In the last years, the development of conversational interfaces such as text-based chat-bots or vocal assistant applications (e.g., Alexa or Google Home) dramatically decreased their complexity. Different services exist that provide both speech recognition and the natural language interpretation. They require an easy configuration if we take into account the complexity of natural language itself. Instances of such services are Facebook Wit.ai [4] or Juji [17].

A conversation naturally requires communication turns. Therefore, it is acceptable for both the users and the system to wait until the entire sentence is available before starting the processing phase. People do not need to learn

which word or phrases are available in a system since we reached good language coverage. This allows using libraries based on DL for both speech recognition and interpretation. The *Core* receives a simplified representation of the user's commands including part of speech tagging, handling synonyms etc.

In this field, the DL techniques fit particularly well, since the pipeline does not require any intermediate guidance and turns are strictly defined. Applying a black box approach is acceptable for both developers and users as long as the accuracy and the robustness to the input fluctuation of the model is good enough. In addition, using a pre-trained model does not require sharing the training set, so the big data owners such as Google, Amazon and Facebook provide APIs and software components that exploit them, increasing the overall usability of the applications built on top of such components.

3.2 Gesture interfaces

Deep Learning techniques do not have the same seamless integration in gestural interaction. First of all, in this field, the tracking sensors limit the interaction vocabulary, if we compare it against the natural expressiveness of the human body. Therefore, users need guidance for discovering which movements the system is able to interpret and how to perform them.

This requires an accurate design for both the feedback and feedforward components in a gestural interface. They respectively support the users in understanding the effect of their previous actions (feedback) and to foresee the effect of future actions before performing them (feedforward). The design space for both components has been widely investigated in the literature, (e.g., by Luyten et al. [13]). It is very difficult to implement such guidance systems using DL or classification techniques in general. They require the entire gesture sequence for assigning a label. If they support partial gesture recognition, they usually provide the final gesture label, without supporting sub-part identification. The research in gesture description languages proved that the latter is a key requirement for creating effective feedback and feedforward components in gestural interfaces [11].

However, how to map the structure of highly-accurate classifier into a gesture description language is an open challenge. Solutions for specific techniques exist (e.g. Hidden Markov Models [2]), but research is still needed for obtaining similar results for Neural Networks.

Finally, the lack of a common vocabulary and of a clear winner among the tracking sensors hinders the availability of big training datasets for reaching the same interface design flexibility we have for natural language. In general, datasets are tailored for the interface at hand and, consequently, their size is much smaller.

4 Understanding the System

While the *Input* elements may employ Deep Learning techniques for supporting the user in controlling the application, easing the *articulation* and the *per-*

formance phase, the *System* and the *Output* components may exploit DL for performing computations on data or for presenting the results to the user.

4.1 System features with Deep Learning

In the *System* component, Deep Learning techniques help in solving highly-dimensional problems, approximating functions that are difficult to define through the training data. DL provides a robust solution to difficult problems, but it creates challenges when the user requires information on the computation procedure. This affects the *observation* phase, where the user assesses the results according to her goals. Without proper explanations, users may perceive the results as unreliable. However, the lack of information lays on the *System* component rather than on the *Output* since usually DL models do not support straightforward ways for creating labelling explanations.

A typical example is supporting transparency and explainability in Recommender Systems. The user may increase her trust in the system when it provides a list of suggested items together with the explanation of why the system considers them useful [16]. Providing such insights sets different engineering challenges: the simplification of the model, its predictability, its representation (e.g., through a metaphor). A good amount of literature exists on this topic, but it provides guidelines and solutions in specific domains. The general engineering challenge is still open [3][12].

As already discussed for gesture interfaces, such fragmentation into domain-specific solutions requires also domain-specific and high quality data. This narrows the applicability of the techniques to datasets available for the research community and the general public. In other domains, it is a complete prerogative of big data owners.

4.2 Supporting Output with Deep Learning

At the *Output* component, Deep Learning techniques may be employed for generating human understandable representations of the internal system state, or to provide information on how to perform specific tasks through the interface.

Such advice remains underinvestigated in the literature, to the best of our knowledge. In our opinion, it represents an opportunity for lowering the barrier for systems requiring a relevant initial learning phase: DL techniques may be employed for acquiring knowledge on the difficulties in the *observation* phase (for instance analysing help tickets) and generating suggestions when the problems are detected.

Such a process may be enhanced exploiting explicit models representing the *output* language. We plan to investigate this in the End User Development [8] field. We are currently developing a web-based authoring environment for point and click games [6, 5]. The user defines the gameplay through generic objects (e.g., transitions, switches, keys, etc.) and Event-Condition-Action rules. They represent a simplified and controllable model of the game definition, which still requires a learning effort for the end-user. We plan to apply Deep Learning

techniques for supporting both goal-oriented suggestions and question answering, tailoring the answers in the context of the developed game, as described by its rules.

5 Conclusion

In this paper, we presented a set of challenges and opportunities in engineering interactive systems, analysing the integration of Deep Learning techniques into the Abowd and Beale interaction framework [1]. The open research directions span from the transparency of the trained models, the support of intermediate feedback during the classification to the generation of explanations about the system in natural language.

References

1. Abowd, G.D., Beale, R.: Users, systems and interfaces: A unifying framework for interaction. In: HCI. vol. 91, pp. 73–87 (1991)
2. Carcangiu, A., Spano, L.D., Fumera, G., Roli, F.: Deictic: A compositional and declarative gesture description based on hidden markov models. International Journal of Human-Computer Studies **122**, 113–132 (2019)
3. Eiband, M., Völkel, S.T., Buschek, D., Cook, S., Hussmann, H.: When people and algorithms meet: user-reported problems in intelligent everyday applications. In: Proceedings of the 24th International Conference on Intelligent User Interfaces. pp. 96–106. ACM (2019)
4. Facebook: Wit.ai Natural Language for Developers. https://wit.ai/, online, accessed: 2018-04-25
5. Fanni, F.A., Mereu, A., Senis, M., Tola, A., Spano, L.D., Murru, F., Romoli, M., Blečić, I., Trunfio, G.A.: PAC-PAC: End user development of point and click games. In: Proceedings of the IS-EUD 2019 (to appear). Springer (2019)
6. Fanni, F.A., Mereu, A., Senis, M., Tola, A., Spano, L.D., Murru, F., Romoli, M., Blečić, I., Trunfio, G.A.: PAC-PAC: intelligent storytelling for point-and-click games on the web. In: Proceedings of the 24th International Conference on Intelligent User Interfaces: Companion. pp. 45–46. ACM (2019)
7. Han, J., Zhang, D., Cheng, G., Liu, N., Xu, D.: Advanced deep-learning techniques for salient and category-specific object detection: a survey. IEEE Signal Processing Magazine **35**(1), 84–100 (2018)
8. Lieberman, H., Paternò, F., Klann, M., Wulf, V.: End-user development: An emerging paradigm. In: End user development, pp. 1–8. Springer (2006)
9. Litjens, G., Kooi, T., Bejnordi, B.E., Setio, A.A.A., Ciompi, F., Ghafoorian, M., Van Der Laak, J.A., Van Ginneken, B., Sánchez, C.I.: A survey on deep learning in medical image analysis. Medical image analysis **42**, 60–88 (2017)
10. Munteanu, C., Baecker, R., Penn, G., Toms, E., James, D.: The effect of speech recognition accuracy rates on the usefulness and usability of webcast archives. In: Proceedings of the SIGCHI conference on Human Factors in computing systems. pp. 493–502. ACM (2006)
11. Spano, L.D., Cisternino, A., Paternò, F., Fenu, G.: Gestit: a declarative and compositional framework for multiplatform gesture definition. In: Proceedings of the 5th ACM SIGCHI symposium on Engineering interactive computing systems. pp. 187–196. ACM (2013)

12. Springer, A., Whittaker, S.: Progressive disclosure: empirically motivated approaches to designing effective transparency. In: Proceedings of the 24th International Conference on Intelligent User Interfaces. pp. 107–120. ACM (2019)
13. Vermeulen, J., Luyten, K., van den Hoven, E., Coninx, K.: Crossing the bridge over norman's gulf of execution: revealing feedforward's true identity. In: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems. pp. 1931–1940. ACM (2013)
14. Zhang, Q., Yang, L.T., Chen, Z., Li, P.: A survey on deep learning for big data. Information Fusion **42**, 146–157 (2018)
15. Zhang, S., Yao, L., Sun, A., Tay, Y.: Deep learning based recommender system: A survey and new perspectives. ACM Computing Surveys (CSUR) **52**(1), 5 (2019)
16. Zhang, Y., Chen, X.: Explainable recommendation: A survey and new perspectives. arXiv preprint arXiv:1804.11192 (2018)
17. Zhou, M.X., Chen, W., Xiao, Z., Yang, H., Chi, T., Williams, R.: Getting virtually personal: chatbots who actively listen to you and infer your personality. In: Proceedings of the 24th International Conference on Intelligent User Interfaces: Companion. pp. 123–124. ACM (2019)