Asking 'Why' in AI: Explainability of Intelligent Systems — Perspectives and Challenges

Alun Preece
Crime and Security Research Institute, Cardiff University
Friary House, Greyfriars Road, Cardiff, CF10 3AE, UK
Email: PreeceAD@cardiff.ac.uk
Telephone: +44 29 2087 4653
Fax: +44 29 2087 4598

Short title: Asking 'Why' in AI

Sponsor

This research was sponsored by the U.S. Army Research Laboratory and the UK Ministry of Defence under Agreement Number W911NF-16-3-0001. The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the U.S. Army Research Laboratory, the U.S. Government, the UK Ministry of Defence or the UK Government. The U.S. and UK Governments are authorized to reproduce and distribute reprints for Government purposes notwithstanding any copyright notation hereon.

Summary

Recent rapid progress in machine learning (ML), particularly so-called 'deep learning', has led to a resurgence in interest in explainability of artificial intelligence (AI) systems, reviving an area of research dating back to the 1970s. The aim of this article is to view current issues concerning ML-based AI systems from the perspective of classical AI, showing that the fundamental problems are far from new, and arguing that elements of that earlier work offer routes to making progress towards explainable AI today.

Keywords: artificial intelligence, explainability, machine learning, interpretability

1 Introduction

An explanation is commonly defined as a reason or justification given for an action or belief. Typically, an explanation provides new information linked to the thing that it is intended to explain and, as with all information, is subject to interpretation by its recipients. In psychological terms, explanations are characterized by a variety of models and schemas, including causal structures, domain-specific patterns (e.g., scientific explanations), and cultural schemas (Keil (2006)).

Artificial intelligence (AI) is concerned with the creation of computer systems (or 'agents') that take actions or express beliefs based on processes that, if exhibited by a natural agent, would be considered as 'intelligent' (Russell and Norvig (2010)). It therefore follows that the generation of explanations has always been a key issue in AI: developers and users of AI systems need to be able to obtain reasons or justifications for the actions or outputs of the machine, and often expect the system to generate explanations that exhibit traces of 'intelligent processing'. As with all explanations, those from an AI system are subject to interpretation, and therefore need to use communicable representations such as mathematical, logical, linguistic, or visual forms.

The interest in explainability of AI systems is naturally linked to surges of interest in AI. The 'classical' period of progress in AI — from the 1970s to early 1990s — featured a corresponding phase of interest in methods for explanation generation in largely symbolic reasoning systems, including so-called 'expert systems' (Jackson (1999)). Significant progress was made on explainability during this period, with solid principles established, but the problem was not considered to have been completely solved.

The recent rapid progress in machine learning (ML), particularly so-called 'deep learning' (LeCun et al. (2015)), has led to a resurgence in interest in explainability.[1] Issues of transparency and accountability have been highlighted as specific areas of concern (Diakopoulos (2016)). A high-profile instance of this issue was seen in the case of Google's Flu Trends (GFT) system, exemplifying both the strength and weakness of 'big data' approaches (Lazer et al. (2014)). After an initial period where the GFT algorithm performed extremely well using data generated from web searches in comparison with official flu statistics, a subsequent failure in performance naturally begged an explanation: *why* did GFT fail? The conclusions of Lazer et al. (2014)'s analysis identify lack of transparency as the key issue: the workings of the GFT algorithm were opaque (a 'black box'), as was its sensitivity to particular patterns in the data (e.g., specific search terms). In short, not only was there no identifiable answer to that *why* question, the more significant problem was that there was no identifiable means of asking the question of the GFT system.

Algorithmic transparency is increasingly viewed from a legal and ethical standpoint as well as a technical one. There is growing concern around issues of fairness

[1] Notably, the recently-announced DARPA program in Explainable Artificial Intelligence (XAI) is largely focused on interpretability of ML approaches:
https://web.archive.org/web/*/https://www.darpa.mil/program/explainable-artificial-intelligence

in machine decision making, particularly arising from biases in the data on which machine learning or statistical decision-support algorithms are trained (Olhede and Rodrigues (2006)). These issues are particularly problematic from a societal perspective where the algorithmic biases relate to characteristics associated with equality and diversity, e.g., gender, race, or religion (Caliskan et al. (2017)). Moreover, there are international efforts to enshrine algorithmic decision making within legal frameworks; for example, the European Union's proposed General Data Protection Regulation is due to come into force in 2018, creating a 'right to explanation' entitling an individual to receive an explanation of any decision made by an algorithm about them (Goodman and Flaxman (2016)).

The aim of this article is to view these current issues concerning ML-based AI systems from the perspective of classical AI, showing that the fundamental problems are far from new, and arguing that elements of that earlier work offer routes to making progress towards explainable AI today. Section 2 reviews progress in explanation generation during the 1970s–1990s knowledge-based systems era. Section 3 examines current thinking around explainability (now more commonly termed interpretability) in ML. Section 4 argues that solutions to the interpretability in the modern context can draw on classical approaches to the original explainability problem. Section 5 concludes the paper by suggesting an agenda and way forward.

2 Perspective: Explanation in Classical AI Systems

Even in the earliest AI systems of the 1960s and 1970s, the generation of explanations was identified as a key issue. Initially, the focus was on providing mechanisms for users to obtain traces of the reasoning performed by a system. This approach is exemplified by the rule traces generated by the explanation component of the MYCIN expert system (Buchanan and Shortliffe (1984)). Even at this early stage, it was realized that there were two distinct kinds of stakeholder requiring explanations:

- *developers* of an AI system, needing assistance in debugging the software by being able to verify the correctness or otherwise of rule firing sequences leading to a conclusion;
- *users* of the system, seeking assurance that they could trust the output from the software by inspecting the chain of reasoning supporting a particular conclusion.

Both kinds of stakeholder were essentially requiring that the AI system have a degree of *transparency* in its workings, i.e., the opposite of *opacity*. Commonly, in software engineering, opaque systems are referred to as 'black boxes' while transparent systems are called 'glass boxes' (Beizer (1995)).

MYCIN: Asking WHY and HOW

MYCIN offered two mechanisms aimed at promoting transparency, depending on whether the system was in the mode of offering a conclusion (as the result of a chain of rule firings) or asking a question (as part of a backward chaining inference process). In the former case, a user could ask HOW in response to a conclusion, and receive a trace of the rules fired, along with the certainty factors (Buchanan and Shortliffe (1984)). In the latter case, a user could ask WHY in response to being asked a question by the system, in which case MYCIN would provide a trace of the currently-active goal and sub-goals in the backward chaining process.

The early MYCIN work also highlighted other key challenges in generating explanations in AI systems. Firstly, the comprehensibility of explanations in terms of rule traces is lower when chains are long, hindering transparency (generally, MYCIN inference chains were relatively short due to the system having a small search space). From the developer's perspective, the value of HOW explanations proved very limited, as the harder debugging cases involved complex and unexpected rule interactions (Davis (1980)), leading to research in knowledge base verification and validation (Suwa et al. (1982)). Indeed, verification and validation are closely linked to explanation: verification, being concerned with whether the system is implemented correctly, is tied to a developer's need for explanation — e.g., rule traces in the simplest case; validation, being concerned with whether the system correctly meets its requirements, is associated with the user's need for explanation — e.g., assurance that the system properly models its intended problem domain (O'Keefe and O'Leary (1993)).

A second key challenge highlighted by the early MYCIN work on explanation was that transparency was restricted only to particular parts of the system, specifically the rule base containing explicitly-encoded symbolic knowledge acquired from domain experts. This part of the system was specifically engineered to be comprehensible by human experts in the problem domain, at least in terms of relatively small sets of rules and rule interactions as noted above. Other components of the system, e.g., LISP program code designed to produce lists of drug recommendations, were opaque to users and played no part in generating HOW and WHY explanations. These parts encoded knowledge *implicitly* rather than explicitly. Moreover, the more opaque aspects of an AI system often corresponded to artefacts arising from the programming of the system (Swartout (1983)). While improved transparency in these aspects would assist developers in debugging the system, revealing them to users would be confusing and unhelpful.

Using Meta-Knowledge in Explanation Generation

In view of the opacity of parts of the MYCIN system, and in an effort to reduce the role of programming artefacts in system design, the NEOMYCIN project attempted to encode additional types of knowledge explicitly, including meta-rules for control of reasoning and taxonomic information, e.g., of diseases (Clancey (1987)). The former differentiated various kinds of knowledge including causal rules, rules connect-

ing data to hypotheses, and 'screening' rules that restrict the search space under particular conditions. All of these type of knowledge could play useful roles in generating HOW and WHY style explanations in NEOMYCIN. The important lesson here is that explanations require context in terms of either what the system is currently trying to do (WHY) or how it did it (HOW). A key claim for the NEOMYCIN approach was that the approach was intended to simulate human problem solving and was thus a form of *cognitive modelling* in the sense of Newell (1990).

A broader perspective on context in explanation generation was taken in CENTAUR, where frames called 'prototypes' were used in addition to rules to organize the knowledge base of the system in terms of elements of a deductive process. The CENTAUR approach afforded the system explicit representation of the relationship between data and hypotheses including:

- hypotheses consistent with (i.e., suggested by) data items;
- data items inconsistent with hypotheses ('errors' or 'surprises');
- data items unaccounted for by any hypotheses (residuals).

In this respect, CENTAUR was arguably the first AI system to be designed for explainability, rather than explanation being considered as an add-on feature.

The Explainable Expert Systems Project

In the early 1990s, the Explainable Expert Systems (EES) project further developed the theme of using meta-knowledge for explanation generation, focusing on three key principles (Swartout et al. (1991); Paris (1993)): (i) separation of terminological and declarative domain knowledge from procedural problem-solving knowledge that would be compiled into a run-time system using (ii) transformations that explicitly captured design rationale (e.g., enhancing maintainability or human readability of the transformed knowledge) accessible to a user via (iii) a dialogue-based interaction module that could create explanations, justifications and paraphrases of the system actions and corresponding rationale.

The first principle in EES (separation of terminological and declarative domain knowledge from procedural problem-solving knowledge), in line with the previous NEOMYCIN and CENTAUR thinking, was also compatible with the shift in attention in knowledge-based systems work in the 1990s towards a focus on reusable domain ontologies (Gruber (1994)) and problem-solving methods (Schreiber et al. (1999)) though generally those two sub-fields did not focus specifically on explanation generation. It is also worth noting that, while a strength of the EES approach was explicit representation of design rationale, concerns such as maintainability — while undoubtedly aspects of system transparency — are of more relevance to developers than users. Arguably the earlier NEOMYCIN work, emphasizing framing explanations in terms of cognitive models, was a more relevant approach to meeting users' needs for system explanations.

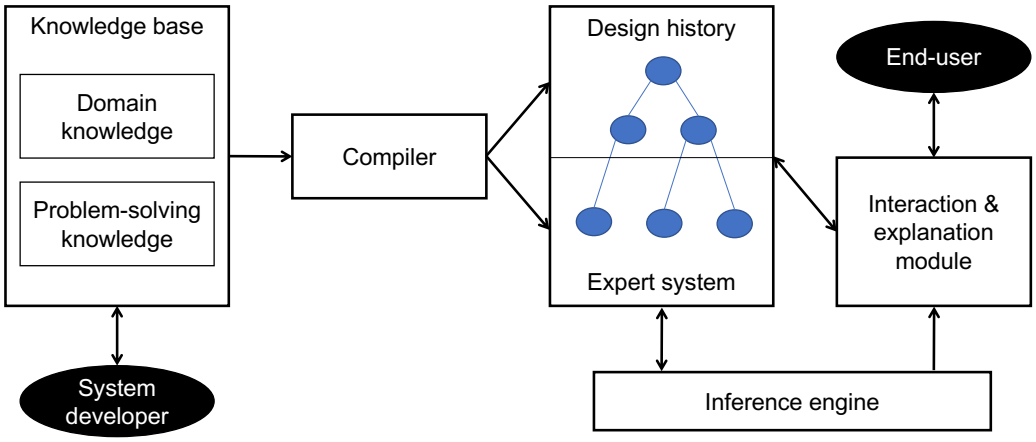

Figure 1: Explainable Expert System framework (adapted from Paris (1993))

The EES work also highlighted the interactive and dialogue-based nature of explanation generation, going far beyond the simple WHY and HOW interactions supported by MYCIN. Using a planner, a set of heuristics, and a natural language (NL) generation system, the EES user interaction module was able to interpret queries such as WHY in context and generate appropriate NL responses based on the design history linked to the expert system.

Summary and Some Desiderata

In conclusion, the classical perspective from the 1960s to 1980s offers a number of desiderata on explanation generation in AI:

1. Explanation generation is an intrinsic designed-in feature of an AI system, not a bolt-on.

2. There are two types of stakeholder requiring explanations from an AI — developers and users — but the needs of these two constituencies are not the same.

3. Interactivity and dialogue is a key element of explanation generation, and a useful distinction can be drawn between what MYCIN termed 'how' and 'why' explanations: the former questions *what the system is doing or intends to do*, the latter questions *what the system did*.

4. Explanations need to cover both the 'know how' and 'know what' of a system — the former is commonly opaque; the latter can be opaque also, especially in a complex system.

5. Cognitive modelling as a basis for explanation generation (framing explanations in terms of reasoning processes that resemble those of human experts)

is seen as a means of promoting system transparency for users, while capturing of software design rationale is a key issue for explanations to developers.

Having reviewed the classical perspective on explanations in AI systems, the following section considers current concerns and approaches in the context of the recent rapid progress in ML and deep learning.

3 Interpretability in ML-based AI Systems

By common definition, an interpretation is the action of explaining the meaning of something. That is, an interpretation generates an explanation, and the two terms are thus closely associated. Since the late 1990s, the term 'interpretable' has been favoured over 'explainable' in the ML context. Where ML was viewed as a 'knowledge discovery' tool, it naturally followed that the discoveries generated by an ML system needed to be interpretable to users in terms of domain knowledge; to this end, researchers focused upon how to exploit domain knowledge as both input to the learning process and in the generation of interpretations of its output (Bratko (1997)). Moreover, as in the classical AI consideration of explanations, interpretations in ML were seen as critical in building user trust in the system (Ridgeway et al. (1998)).

Despite this long-term interest in interpretability of ML-based AI systems, a formal, commonly-agreed definition of the term has remained elusive. Lipton (2017) observes that a key issue in defining interpretability formally is that the concept is not monolithic. This observation essentially echoes the 1970s realization that the kinds of explanations required of an AI system by developers are quite different from those required by end users. Moreover, developers may have differing needs, e.g., verifying the performance or robustness of a system, and end-users will have different perspectives, e.g., better understanding a reported 'discovery' (Bratko (1997)) or determining the fairness of a decision (Diakopoulos (2016); Olhede and Rodrigues (2006)).

Doshi-Velez and Kim (2017) link interpretability to the need for an ML system to satisfy *auxiliary criteria*, i.e., criteria that are in part qualitative and cannot be satisfied by improved training (unlike, say, accuracy). While many examples are given by the authors (and others, e.g., Lipton (2017)) — including being nondiscriminatory (as in fairness), safety (Otte (2013)), and satisfying a user's right to explanation (as in Goodman and Flaxman (2016)) — there does not yet appear to be a comprehensive typology of these kinds of auxiliary criteria.

Towards Transparency in Deep Learning Systems

As with classical AI concerns regarding explainability, transparency is a key issue in interpretability of ML systems, but the problem is exacerbated with deep learning systems by the sub-symbolic nature of these approaches. For example, while a trace of rule firings in a classical AI system may or may not be informative to a developer

or end-user, a set of weights in a multi-layer neural network is unlikely to be informative to anyone. This has led researchers to argue that *intelligibility* of ML models is a necessary property for transparency (Lou et al. (2012)): the ability for a human to understand how a learned model works. In classical AI, we saw that attention shifted from examining rule traces to focusing on the meta-knowledge that controlled and guided inference. This was an attempt to frame explanations in terms of *algorithmic transparency*, providing confidence that the system was behaving 'sensibly' in general, rather than at the level of specific rule firing sequences. The problem is, however, that algorithmic transparency for deep learning systems is not achievable given our current understanding of these systems, because we cannot prove that they will work on unseen problems (Lipton (2017)).

In the absence of algorithmic transparency for deep learning ML, researchers have instead opted to focus on finding equivalences to 'traces'. The most common example of this approach is in image classification systems, to associate an output class with the parts of an input image that had the greatest weight in determining the classification. For example, the LIME approach identifies 'super-pixels' (contiguous regions of similarly-weighted pixels) in an input image that contribute positive weight towards a particular output class, with the intuition that these regions are significant in making the model predict that the class may be present in the image (Ribeiro et al. (2016)). This approach has the advantage that the super-pixels will be in and of themselves meaningful to a human, especially in relation to the whole of the original image.

Similarly-motivated approaches include the use of *heat maps* to visualize the relative weighting of parts of an image at the pixel level in terms of a 'hot to cold' scale where 'hottest' = most highly weighted (Montavon et al. (2016)) and *class maps* to highlight the parts of an image most associated (in weight terms) with each of several possible output classes (Kumar et al. (2017)). The latter is interesting because, while the classifier will generally output the most highly predicted class, a class map will provide a visualization of the parts of the image that could have led the classifier to predict a different output (i.e., 'I think it's X because of region A; however, region B suggested it might be Y and region C suggested it might be Z'). This richer context to the interpretation provided by a class map arguably gives a user improved intelligibility of how the classifier works, and therefore a greater impression of algorithmic transparency.

These approaches for 'tracing' input to output relationships in deep learning ML systems are not confined to imagery. Similar techniques can be used to identify the most salient (highest weighted or most predictive) text features or fragments. For example, Lei et al. (2016) propose an approach that extracts coherent phrases from input text that are sufficient to predict the same output as the full input. These extracts are offered as *rationale* for the classification and, like the image regions selected by LIME, heat maps, or class maps, are meaningful to humans.

Transparency vs Post-Hoc Interpretations

A key distinction is drawn in current thinking in terms of explaining the classifications of modern ML systems between true transparency and *post-hoc* interpretations or explanations (Lipton (2017)). This distinction was not present in the classical era because algorithmic transparency was seen as achievable for those kinds of AI system. While transparency aims to reveal how a system actually reached its conclusion, a post-hoc interpretation seeks to explain an output without reference to the inner workings of the system. Post-hoc interpretations have become popular as an approach in the context of deep learning because algorithmic transparency is seen as being unachievable for these systems.

Techniques for generating post-hoc interpretations include visualizations, NL explanations, and retrieval of salient examples. Technically, heat maps and class maps are a form of post-hoc interpretation since they visualize (by use of colour) significant parts of the input image. Use of natural language caption generation is also a form of post-hoc interpretation. For example, Hendricks et al. (2016) propose a method that uses a dual neural network system: one sub-system learns to classify images, and a second sub-system generates text explanations on the basis of textual background knowledge that describes discriminating features of each output class. Thus, the explanations generated by the second sub-system tend to include text descriptions of those discriminating features, when they are detected in an input image. This approach attempts to combine aspects of transparency (highly-weighted features leading to a classification) with post-hoc explanation (textual renderings of discriminant features) though, because deep neural networks are employed, there can be no guarantees that the 'right words' are always associated with the 'right features'.

It is, however, worth noting at this point that these kinds of post-hoc interpretation techniques are analogous to what humans do when asked to explain classification decisions. As noted by Lipton (2017), 'To the extent that we might consider humans to be interpretable, it is [post-hoc] interpretability that applies.' In a sense, seeking fully-transparent interpretations from a deep learning based AI system is holding the system to a higher standard than the one to which humans can be held.

Retrieval of examples is another technique employed in generating post-hoc explanations, taking inspiration from the behaviour of human experts, e.g., doctors and lawyers, who often frame explanations by referring to case studies. For example, Caruana et al. (1999) demonstrated how case-based reasoning could be used to generate explanations for a neural network by using the latter as a means of computing the distance metric for case retrieval. Again, the advantage is that the cases are meaningful to humans though, as with all post-hoc approaches, the explanations offer limited insight into how the classifier actually made its decision.

ML Interpretability and Expert Knowledge

As discussed in Section 1, the notion of explanation is often associated with causality, and a significant part of the classical AI explanation work examined in Section 2 fo-

cused on introducing meta-knowledge to capture causal rules and deductive chains. While causality has been highlighted as a desirable feature for interpretability of ML models (Lipton (2017)), relationships learned by ML systems are not assured to be causal. Indeed, the current state-of-the-art in ML is weak at learning causal models of the world that support understanding (Lake et al. (2016)). Moreover, the tension between correlation and causation is a well-known issue in 'big data' work (Diakopoulos (2016)). The problem of deriving causal associations has been extensively studied but establishing causality generally relies on availability of prior background knowledge (Pearl (2009)), which commonly does not feature in 'big data' systems.

Ross et al. (2017) propose an approach that, while not aiming to assure causal explanations, attempts to avoid offering spurious correlations by applying constraints during training that specify whether or not an input feature is relevant to the classification of that input, according to a human expert. The learned model is thus intended to be 'right for the right reasons': explanations from such a model are optimized for correctness. Again, the approach relies on the availability of background knowledge in the form of human experts' opinion.

Recent work by Doshi-Velez and Kim (2017) proposes a three-level taxonomy for evaluating interpretations in ML systems, where the levels are in descending order of cost:

Application-grounded evaluation involves humans performing real tasks requiring domain expertise, e.g., medical diagnosis or financial decisions. The gold standard for comparison here is with human-produced explanations to assist other humans trying to complete the task. The relative quality and cost of machine-produced vs human-produced explanations is compared.

Human-grounded evaluation involves real humans performing simplified tasks that can be carried out by non-experts therefore making subject recruitment easier at the expense of some external validity. Commonly, the purpose here is to test some aspect of the explanation unrelated to the subject matter, such as speed of reaching a decision, or avoiding cognitive bias.

Functionally-grounded evaluation involves no humans and uses proxy tasks (ideally derived from one or other of the above types of evaluation) together with formal metrics as proxies for explanation quality. This method avoids the need for ethical review and is often used in the earlier stages of assessing an ML approach, where the system has not yet reached maturity. Most of the techniques described in the previous two subsections utilized this kind of evaluation method. Human expertise is implicitly or explicitly factored into the design of the proxy tasks and metric.

Summary and Some Desiderata

In conclusion, current thinking regarding interpretability of ML-based AI systems offers a number of desiderata:

1. As with classical AI systems, transparency is the ideal for interpretability; however, full transparency (particularly algorithmic transparency) is not achievable given the state-of-the-art in deep learning, so post-hoc interpretations will in many cases be the best-available option. Nevertheless, it is important to utilize as much transparency as possible as a basis for generating post-hoc interpretations (for example, generating NL explanations on the basis of salient input image regions or features).

2. Interpretations depend on user's requirements in terms of auxiliary criteria such as safety, legal accountability (e.g., 'right to explanation') or knowledge discovery. As with classical AI systems, different groups of users will have different requirements. The auxiliary criteria are qualitative and resistant to formal definition; nevertheless, works needs to be done to elucidate them better, so that users can specify their interpretation requirements more systematically, and systems can be evaluated more robustly.

3. Interpretations in many cases will depend on domain knowledge, e.g., background or prior knowledge. Examples include meta-knowledge of salient or discriminant features to guide the pertinence of explanations, prior knowledge to frame causal explanations, and knowledge that frames classifications for use in case-based retrieval of examples. Acquisition, curation, and re-use of such domain knowledge needs to be considered more systematically as part of engineering ML-based AI systems.

Having examined perspectives on explanation in AI from the classical era to the present ML-dominated period, the next section draws elements of both together, offering a systems architecture for hybrid ML / knowledge representation and reasoning AI systems.

4 Explanation in AI: A Dual System Approach

The organization of the previous two sections was intended to emphasize distinctions (and some parallels) between the perspectives on explanation in classical AI — with its emphasis on knowledge representation and reasoning — and modern ML — with its emphasis on deep learning. The presentation therefore may have implied a false dichotomy between the two. In actuality, there is significant acknowledgement that AI systems require an integration of reasoning and learning, and there is growing interest in approaches seeking to combine the two. An important motivation for this is to address the weak state of ML in dealing with causal relationships (Bottou (2014)) and to integrate *model-building* with pattern recognition (Lake et al. (2016)).

How exactly to combine reasoning and model-building with ML is a topic of some debate. Some researchers favour approaches that seek to utilize vector-space representations instead of classical manipulation of symbolic expressions (Guha (2015); LeCun et al. (2015)). Others argue for building reasoning capabilities in a bottom-up

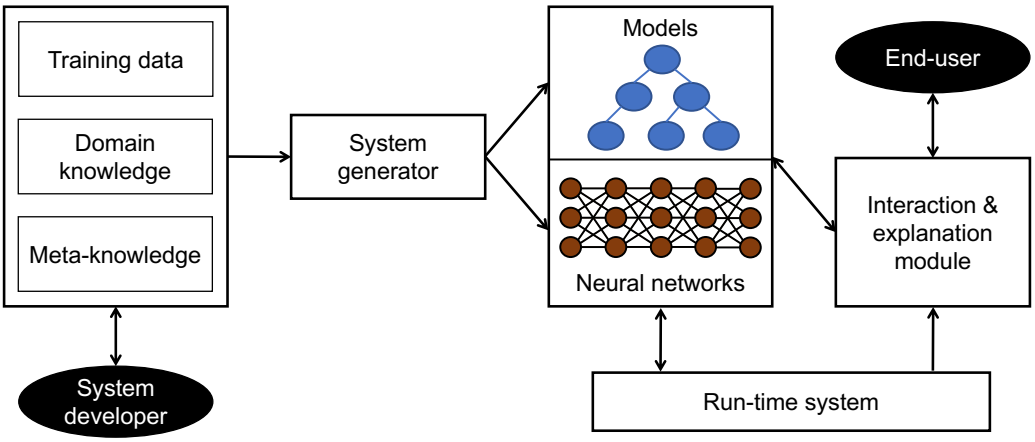

Figure 2: A framework for explainable AI systems

manner, from a rich set of primitive learning operators (Bottou (2014)). A key aspect of these discussions is the issue that, unlike deep learning systems, humans are capable of learning from small amounts of data (Lake et al. (2016)), and knowledge representation-based systems offer this property of what Guha (2015) calls 'teachability'. Recent work aiming to integrate deep learning and Bayesian models within a uniform probabilistic framework appears promising in this context (Wang and Yeung (2016)). Such a model is amenable to user input (i.e., it is teachable), and some initial work has been done in this area, referred to as collaborative deep learning (Wang et al. (2015)).

Of course, all of these approaches, while offering means to integrate ML with knowledge representation and reasoning, come with an important caveat: the interpretability problem for such a integrated approach will also need to be addressed.

An Explainable ML Framework

Figure 2 draws on the classical EES framework (Figure 1) to propose an architecture for explainable ML-based AI systems. A *system generator* module (with essentially the same role as the compiler in EES) builds a dual system with two parts — a model part, comprising models of the world, and a neural network part — from a set of inputs including training data, domain knowledge, and meta-knowledge. A run-time module uses the dual system to derive classifications and inferences, while also providing input to the interaction module that provides output to the end-user also allows them to seek explanations.

To show how the framework is intended to operate, we consider a number of example ML-based AI systems using techniques from the previous section:

Basic transparency: super-pixels, heatmaps, class maps Our simplest example is a deep learning image classification system where the only available explana-

tion is in the form of LIME-style super-pixels extracted from the input image. Training examples are provided to a deep learning algorithm that is part of the system generator, along with the meta-knowledge requirement that the generated system be capable of offering super-pixel regions as explanations. Here, the model part of the dual system is empty. The generated system is able to inform the interaction module that it is able to answer basic 'Why do you think it is X?' queries; it does so via the super-pixel computation capability built-into the neural network part of the generated system and captured by the run-time module.

The case is similar where heatmaps are to be generated, but slightly different where class maps are used since then the generated system is able to inform the interaction module that it is capable of answering 'Why do you think it is X?' and also 'Why do you not think it is Y?' queries by drawing on the relative feature weights generated by the neural network at run-time.

'Right for the right reasons' Here, the input to the system generator module must include knowledge acquired from domain experts that specifies whether or not an input feature is relevant to a particular output class. Meta-knowledge includes the requirement that the generated system be trained to optimize explanations for correctness in terms of the background knowledge. The model part of the generated system will include this domain knowledge. When answering 'Why do you think it is X?' queries, the interaction module will draw on both the model part of the system as well as the weights computed by the run-time module from the neural network part.

Note that the interaction module can also allow the end-user to explore the model of relevant features per output class, to gain insights into what the system 'knows', as an additional means of building trust between user and system. Thus, in some cases the interaction module will draw on both the model and neural network parts of the system, while in other cases the model part alone may suffice to provide a user with useful explanations.

Causal explanations Leaving aside the challenges in learning causal relationships discussed in Section 3 and in Pearl (2009), to the extent that progress is likely to be made in this area in future (Lake et al. (2016) offer a detailed discussion of prospects and approaches in relation to deep learning) the dual system approach provides a means of capturing learned causal knowledge in the model part of the system, linked to elements of the neural part of the system. It is unclear at present what is the best approach for capturing such linkages, though the work on Bayesian deep learning appears to be a promising direction (Wang and Yeung (2016)).

In this case, meta-knowledge will capture the requirement to learn causal models, and may specify particular features or classes to constrain the learned relationships (in relation to the domain knowledge). Early 1990s work on knowledge discovery also points to the role in background domain knowledge as an

input to a ML system to support explanation generation (Bratko (1997)). The enriched model parts of a generated system in this case will afford deeper interactions with the end-user in terms of elucidating and justifying causal explanations. The possibilities here resemble the capabilities of systems like CENTAUR in Section 2 where users could explore deductive processes and identify input data that was consistent or inconsistent with hypotheses.

Case-based explanations by example Framing a collection of 'exemplary' classifications as cases for retrieval as explanations would use a similar approach to causal explanations in terms of using meta-knowledge to specify frames for the cases, domain knowledge of pertinent features and ontological relationships, and distance metrics for retrieval of pertinent examples. Cases would form part of the model element of the generated system, and interactions would be further enriched to support queries such as, 'Show me examples of X'. Retrieval of examples will in some cases be subject to privacy requirements, however. For example, it may be acceptable to use imagery from a particular patient's case in offering an explanation in a medical diagnosis system provided that no personal data is revealed.

Satisfying auxiliary criteria The final example concerns auxiliary criteria relating to background knowledge such as safety-critical elements or features relating to characteristics associated with equality and diversity, e.g., gender, race, or religion. The framework supports capturing such criteria in terms of background knowledge for the learning system, incorporating the criteria in the model part of the system, and supporting end-user queries relating to the criteria. However, given the extremely challenging problem of defining many of the auxiliary criteria objectively, how to capture these in the model part of the generated system is an open problem.

Discussion w.r.t. Explanation Desiderata

Considering the above framework against the desiderata for AI systems identified in Section 2:

1. The framework ensures that explanation-generation is an intrinsic, designed-in feature of the generated system, accessible to end-users via the interaction system.

2. A variety of stakeholders are catered-for by the framework: their distinct requirements in terms of kinds of explanation (transparency and ad-hoc) can be specified as meta-knowledge input, and they can access explanations via the interaction system.

3. The interaction system supports a variety of dialogues, appropriate to the explanation mechanisms built into the generated system, specified by the meta-knowledge inputs.

4. The dual system distinguishes between 'know what' (model) and 'know how' (neural network) knowledge levels. As with classical systems, the former is less opaque (its opacity is more a function of system complexity than representation) while the latter is far more so; nevertheless, transparency-based and post-hoc explanations can be generated for both parts.

5. The question of to what extent cognitive modelling plays a useful role in explanation generation for deep learning-based AI systems is an interesting one, and is bound up with the discussion of to what extent such systems are biologically-inspired (see Eliasmith (2015) for detailed discussion on this matter). At the simplest level, the framework caters for addressing causal relationships and model-based explanations, which provides at least a basis for explanation in terms of 'higher-level' reasoning processes.

Next, considering the framework against the ML system explanation desiderata from Section 3:

1. The framework is designed to support the generation of both transparency-based and post-hoc explanations as shown in the example cases above. The first two cases (*Basic Transparency* and *'Right for the right reasons'*) are focused more on transparency, while the latter three have significant post-hoc aspects (though with a basis in transparency).

2. The framework, while not solving the issue of formally defining auxiliary criteria, makes them a designed-in feature in terms of meta- and domain knowledge.

3. Similarly, the framework is explicitly designed to exploit domain knowledge in generating a system with explanation capabilities.

5 Conclusion and Future Work

The framework described in the previous section is largely conceptual at present, but we are building its various components in current research (Chakraborty et al. (2017); Nottle et al. (2017)). The key stages in a roadmap for realizing the framework are:

- Development of a service-oriented approach to providing explanation mechanisms for ML systems. Most of the state-of-the-art approaches described in Section 3 are available as open source software, so the next stage would be to wrap them as APIs.

- Systematic definition of a typology of auxiliary criteria for assurance of ML systems, encompassing fairness, transparency, and accountability aspects, along with robust metrics for each.

- Research and development of protocols to support explanation-seeking dialogues between users and AI systems.

An area that has not been considered in the preceding discussion is the need for machine-to-machine explanation, which is becoming a more important issue in the Internet of Things (IoT) context, especially where IoT technologies are to be deployed in safety-critical application domains (Fraga-Lamas et al. (2016)). The difference between explanations generated for human consumption and those generated for machine consumption is an area that researchers are beginning to consider (Dhurandhar et al. (2017)); ultimately, however, it is likely that future AI systems will need to provide both kinds of explanation.

References

Beizer, B. (1995). *Black-Box Testing: Techniques for Functional Testing of Software and Systems*. Wiley.

Bottou, L. (2014). From machine learning to machine reasoning. *Machine Learning*, 94(2):133–149.

Bratko, I. (1997). Machine learning: Between accuracy and interpretability. In Della, R. G., Lenz, H., and R., K., editors, *Learning, Networks and Statistics (International Centre for Mechanical Sciences (Courses and Lectures), vol 382)*, volume 382, pages 163–177. Springer.

Buchanan, B. and Shortliffe, E. (1984). *Rule Based Expert Systems: The MYCIN Experiments of the Stanford Heuristic Programming Project*. Addison-Wesley.

Caliskan, A., Bryson, J. J., and Narayanan, A. (2017). Semantics derived automatically from language corpora contain human-like biases. *Science*, 356(6334):183–186.

Caruana, R., Kangarloo, H., Dionisio, J., Sinha, U., and Johnson, D. (1999). Case-based explanation of non-case-based learning methods. In *Proceedings of the AMIA Symposium*, pages 212–215.

Chakraborty, S., Preece, A., Alzantot, M., Xing, T., Braines, D., and Srivastava, M. (2017). Deep learning for situational understanding. In *20th IEEE International Conference on Information Fusion*.

Clancey, W. (1987). *Knowledge-based Tutoring: The GUIDON Program*. MIT Press.

Davis, R. (1980). Applications of meta-level knowledge to the construction, maintenance and use of large knowledge bases. In Lenat, D. and Davis, R., editors, *Knowledge-Based Systems in Artificial Intelligence*, pages 229–490. McGraw-Hill.

Dhurandhar, A., Iyengar, V., Luss, R., and Shanmugam, K. (2017). TIP: Typifying the interpretability of procedures. *arXiv preprint arXiv:1706.02952*.

Diakopoulos, N. (2016). Accountability in algorithmic decision making. *Communications of the ACM*, 59(2):56–62.

Doshi-Velez, F. and Kim, B. (2017). Towards a rigorous science of interpretable machine learning. *arXiv preprint arXiv:1702.08608*.

Eliasmith, C. (2015). *How to Build a Brain: A Neural Architecture for Biological Cognition*. Oxford University Press.

Fraga-Lamas, P., Fernández-Caramés, T. M., Suárez-Albela, M., Castedo, L., and González-López, M. (2016). A review on internet of things for defense and public safety. *Sensors*, 16(1644).

Goodman, B. and Flaxman, S. (2016). European Union regulations on algorithmic decision-making and a "right to explanation". In *2016 ICML Workshop on Human Interpretability in Machine Learning (WHI 2016)*.

Gruber, T. R. (1994). Toward principles for the design of ontologies used for knowledge sharing. *Journal of Human Computer Studies*, 43(5/6):907–928.

Guha, R. (2015). Towards a model theory for distributed representations. In *Knowledge Representation and Reasoning: Integrating Symbolic and Neural Approaches: Papers from the 2015 AAAI Spring Symposium*.

Hendricks, L. A., Akata, Z., Rohrbach, M., Donahue, J., Schiele, B., and Darrell, T. (2016). Generating visual explanations. In *European Conference on Computer Vision (ECCV 2016)*, pages 3–19. Springer.

Jackson, P. (1999). *Introduction to Expert Systems*. Addison-Wesley Longman, 3rd edition.

Keil, F. C. (2006). Explanation and understanding. *Annual Review of Psychology*, 57:227–254.

Kumar, D., Wong, A., and Taylor, G. W. (2017). Explaining the unexplained: A CLass-Enhanced Attentive Response (CLEAR) approach to understanding deep neural networks. In *Computer Vision and Patter Recognition Workshop (CVPR-W) on Explainable Computer Vision*.

Lake, B. M., Ullman, T. D., Tenenbaum, J. B., and Gershman, S. J. (2016). Building machines that learn and think like people. *CoRR*, abs/1604.00289.

Lazer, D., King, R. K. G., and Vespignani, A. (2014). The parable of Google Flu: Traps in big data analysis. *Science*, 343:1203–1205.

LeCun, Y., Bengio, Y., and Hinton, G. (2015). Deep learning. *Nature*, 521(7553):436–444.

Lei, T., Barzilay, R., and Jaakkola, T. (2016). Rationalizing neural predictions. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 107–117.

Lipton, Z. C. (2017). The mythos of model interpretability. In *2016 ICML Workshop on Human Interpretability in Machine Learning (WHI 2016)*.

Lou, Y., Caruana, R., and Gehrke, J. (2012). Intelligible models for classification and regression. In *Proceedings of the 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD'12)*, pages 150–158. ACM.

Montavon, G., Lapuschkin, S., Binder, A., Samek, W., and Müller, K.-R. (2016). Explaining nonlinear classification decisions with deep taylor decomposition. *Pattern Recognition*, 65:211–222.

Newell, A. (1990). *Unified theories of cognition*. Harvard University Press.

Nottle, A., Quintana-Amate, S., Harborne, D., Alzantot, M., Braines, D., Tomsett, R., Kaplan, L., Srivastava, M., Chakraborty, S., and Preece, A. (2017). Distributed opportunistic sensing and fusion for traffic congestion detection. In *First International Workshop on Distributed Analytics InfraStructure and Algorithms for Multi-Organization Federations*.

O'Keefe, R. M. and O'Leary, D. E. (1993). Expert system verification and validation: a survey and tutorial. *Artificial Intelligence Review*, 7(1):3–42.

Olhede, S. and Rodrigues, R. (2006). Fairness and transparency in the age of the algorithm. *Significance*, 14(2):8–9.

Otte, C. (2013). Safe and interpretable machine learning: A methodological review. In Moewes, C. and Nürnberger, A., editors, *Computational Intelligence in Intelligent Data Analysis (Studies in Computational Intelligence, vol 445)*, pages 111–122. Springer.

Paris, C. L. (1993). Explainable expert systems: A research program in information processing. Technical report, Information Sciences Institute, University of Soutgern California.

Pearl, J. (2009). *Causality: Models, Reasoning, and Inference*. Cambridge University Press.

Ribeiro, M. T., Singh, S., and Guestrin, C. (2016). "Why should i trust you?": Explaining the predictions of any classifier. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD'16)*, pages 1135–1144. ACM.

Ridgeway, G., Madigan, D., and Richardson, T. (1998). Interpretable boosted naïve bayes classification. In *Proceedings of the Fourth International Conference on Knowledge Discovery and Data Mining (KDD'98)*, pages 101–104. AAAI Press.

Ross, A. S., Hughes, M. C., and Doshi-Velez, F. (2017). Right for the right reasons: Training differentiable models by constraining their explanations. *arXiv preprint arXiv:1703.03717*.

Russell, S. and Norvig, P. (2010). *Artificial Intelligence: A Modern Approach*. Pearson, 3rd edition.

Schreiber, G., Akkermans, H., Anjewierden, A., Hoog, R. D., Shadbolt, N. R., van de Velde, W., and Wielinga, B. J. (1999). *Knowledge Engineering and Management: The CommonKADS Methodology*. MIT Press.

Suwa, M., Scott, A., and Shortliffe, E. (1982). An approach to verifying completeness and consistency in a rule-based expert system. *AI Magazine*, 3(4):16–21.

Swartout, W., Paris, C., and Moore, J. (1991). Explanations in knowledge systems: design for explainable expert systems. *IEEE Expert*, 6(3):58–64.

Swartout, W. R. (1983). XPLAIN: a system for creating and explaining expert consulting programs. *Artificial Intelligence*, 21(3):285–325.

Wang, H., Wang, N., and Yeung, D.-Y. (2015). Collaborative deep learning for recommender systems. *arXiv preprint arXiv:1409.2944*.

Wang, H. and Yeung, D.-Y. (2016). Towards bayesian deep learning: A survey. *arXiv preprint arXiv:1604.01662*.