"Truth in fiction" as an unsupervised learning task

Louis Rouillé

1 Description of the problem of truth in fiction

The so-called problem of "truth in fiction", or "fictional truth", consists in explaining the contrast between pairs of sentences like the following:

(1) Hamlet is a human being.

(2) Hamlet is a crocodile.

There is a contrast between (1) and (2). In order to introduce this contrast, one can say that (1) is intuitively *true* in *Hamlet* whereas (2) is intuitively *false* in *Hamlet*. This description suggests that the contrast between (1) and (2) should be reflected in the *truth-value* of the two sentences. Ay, there's the rub! (Hamlet would say). Indeed, the claim that the above contrast should be reflected in the truth-value of (1) and (2) is a highly controversial one. See for instance (Lewis 1978), (Walton 1990) and (Currie 1990). But there is no need to enter this philosophical controversy here, for I want to argue here that the problem of truth in fiction can be caracterised more abstractly as a problem about available *inferences*. This way of putting the problem will enable a fruitful analogy with machine learning tasks.

The play *Hamlet* is a set of sentences written by Shakespeare in England around 1600. It so happens that neither (1) nor the denial of (2) is one of the sentences of *Hamlet*. However, one can find many relevant sentences to back up the contrasted intuitions. For instance, one relevant sentence Shakespeare wrote is that Hamlet is a Prince; indeed, it is even clear from the entire title which runs: *The Tragedy of Hamlet, Prince of Denmark*. With this piece of information in mind, it is very easy to explain the above contrast following this line of reasoning: Hamlet is a prince, and princes are human beings, hence (1) is true in *Hamlet*; and (2) is false in *Hamlet*, since (1) is true in *Hamlet*.

This works well for *Hamlet*, but, in other fictions, inferences can go astray. In some fictions, princes are crocodiles. In such fictions, (2) would be intuitively true... Of course, Hamlet is not such a story. But how do we know this?

In general, then, in order to predict such contrasts as that between (1) and (2), we need to define an inference relation for the relevant fiction against which the sentences are interpreted. Such an inference relation should, in principle, extract all the intuitive truths in fiction from the set of sentences which constitute the relevant fictional text. This what is called the "problem of truth in fiction". A theory of fictional truth should thus explain how actual readers draw the inferences they draw when they read a fictional text. Such a theory should therefore at least predict contrasts like the above one.

Here is another way of saying the same thing, using a simple distinction. I will say that a fictional truth is *explicit* when it is expressed by a fictional sentence. I will say that it is *implicit* when it is inferred from another fictional truth. We can now rephrase the problem in this manner: a theory of fictional truth should give a systematic way of retrieving all the fictional truths, as opposed to the fictional falsehoods, given a fictional text. In particular, such a theory should say how the implicit fictional truths are derived from the explicit ones. This last descpription of the problem is interesting, for it shows that the reader is faced with a kind of sorting problem: the reader's task consists is sorting out the fictional truths from the fictional falsehoods, given a set of sentences.

2 The fruitful analogy

I want to defend the idea that the problem of truth in fiction, as defined in the previous section, is conceptually identical with a problem of unsupervised learning. I conclude from this analogy that both research communities would gain in getting interested in one another, despite the *de facto* compartmentalisation of disciplines like machine learning and philosophy. The following remarks are inspired by elements from Yann LeCun's 2016 course on deep learning and Stephane Mallat's 2018 course on machine learning at collège de France.

The problem of truth in fiction in the abstract, as observed above, consists in sorting out the fictional truths from the fictional falsehoods given a set of explicit fictional sentences. Humans are very good at doing this, from a very young age. And it seems that one can provide general heuristics for this sorting problem. These are the so-called "generation principles", which are so central in the philosophy of fiction.[1] Subsequently, the natural question a computer scientist would raise is the following: can one train a machine (say a neural network) so as to extract the fictional truths of a given a fictional text? Thinking about the problem of fictional truth from this viewpoint is, I suggest, very much the right thing to do. However, I think, the problem of fictional truth is a

[1] For a review of the discussion on these, see (Woodward 2011). See also (Friend 2017) for a recent very interesting contribution to the debate on the generation principles.

difficult problem, as least as difficult as unsupervised learning tasks which are still open problems in computer sciences today.

First, I will explain what supervised learning is. Then, I will explain what unsupervised learning is. Finally, I will give reasons to think that the problem of truth in fiction can be seen as an unsupervised learning task.

2.1 Supervised learning

Machine learning is a term denoting the field of research aiming at studying and developing algorithms and software which automatically and dynamically organise data. These algorithms are of interest to computer scientists and applied mathematicians as well as engineers working in the multifarious domains of digital industry. The tasks and problems solved by learning algorithms can be grouped according to their complexity: problems are "harder" than others in the sense that more powerful algorithms are needed to solve them. In the same spirit, an algorithm is said to be "better" than another when the former solves a problem faster or using fewer resources than the latter. The point of comparison with humans (or living animals) is important to define "being good enough" for an algorithm: given a task in an empirical setting, an algorithm is said to be "good enough" when it solves the problem in a comparable amount of time (or faster) with a comparable score as a representative group of human beings.

For instance, suppose the task is to distinguish spoken language from mere noise, given an audio input; humans are very good at this task, meaning that they rarely err and they give their response very quickly; an algorithm is good enough if it does at least as well as a human being both in success and time response. Another very famous example is that of the game of Go for which some algorithms were good enough for a long time and one algorithm (implemented in AlphaGo, developed by DeepMind) became much better than the best humans quite recently.

There is an important distinction between tasks of *supervised* learning and *unsupervised* learning. Very recently, algorithms became very good at most supervised learning tasks, but the best algorithms are still very bad when it comes to unsupervised learning tasks.

Supervised learning tasks are situations in which the machine is given some feedback. The algorithm can thus use the feedback from the environment to correct itself by estimating and reducing to a minimum its predicted error. There are two kinds of supervised learning which we can call *pure reinforcement* and *reinforcement learning*.

Pure reinforcement are cases when there is little feedback, so the machine needs to be trained a lot. The paradigm example is "playing algorithms", like

AlphaGo. The only feedback AlphaGo gets is whether the game was won or lost. The machine then plays billions of games against itself to improve.

Reinforcement learning are cases when there is as much feedback as there is data. The paradigm example is recognition algorithms.[2] Each training input is annotated. During the training, the machine predicts some relevant features of the input and receives feedback in each trial. At the end of each trial, it corrects itself and "learns" that way. To be properly trained, the algorithm needs *a lot* of annotated inputs.

In the recent years, artificial neural networks have become a very popular method of solving supervised learning problems. Reinforcement learning tasks like object recognition or speech recognition have been solved very efficiently by neural networks. Convolutional network, also known under the name "deep learning" because they contain multiple hidden layers, were the first to achieve results comparable to humans in recognition tasks in 2012. Yann LeCun is the first to propose an algorithm based on the method of back-propagation which can be implemented efficiently in convolutional networks.

Typical cases of object recognition tasks are the following: find all the letters in a given document. The convolutional network is trained on a huge set of labelled photographed (labelled by humans). It somehow gets some characteristic features of "being a letter in a document". Then, it is tested on a set of unlabelled documents. Convolutional networks have been shown to be better than humans in this task. They are reliable to the point that they are widely used industrially. For instance, softwares taking any document as input and giving a full texted document as output are legion (they are called Optical Character Recognition software). Another example: machines which automatically "read" cheques are now widespread in banks.

Interestingly, there is a lot of mathematical research to understand how these networks actually work. They *somehow* get the characteristic features. The recent performances of convolutional networks are difficult to explain mathematically, although there are many heuristics which are robust enough to put them into practice in the industry.

2.2 Unsupervised learning

By contrast, *unsupervised* learning are situations in which no feedback is given to the algorithm. The algorithm's task is to find out the underlying structure of the input without the help of labelled feedback. One of the difficult problems facing computer scientists is to give a precise evaluation of the possible outputs.

[2] Like those used to distinguish between mere noise and spoken language, or algorithms used to do object (or facial) recognition in photographs.

Since there is no *a priori* "correct answer"[3], there is no straightforward way of evaluating and comparing the underlying structure of the data the algorithm could propose.

An abstract way of seeing this problem consists in trying to find the probability density function of the space in which the pieces of data "live". For instance, pictures can be seen as enormous sets of pixels, thus defining a space that can be defined and studied mathematically. The algorithm "observes" many pictures, but the density function (the predicted distribution of each and every one of the pixels) is unobserved: one would need to have a gigantic number of disconnected examples. The task is thus to infer, from the data, the function according to which the population of examples constituting the data is distributed. The difficulty lies in the fact that the distinctive features which explain the distribution of the examples is not known *a priori*.

For example, if the task is to distinguish the foreground from the background of a moving picture, the algorithm will have to find out what the distinctive features of "being background" (and conversely "being foreground") are. But these can be very diverse in nature, here are some clues: being often out of focus, moving against a stable foreground (or the opposite), containing smaller ordinary objects than the foreground, etc.

There is no canonical way of solving unsupervised learning tasks at the moment. The underlying mathematics falters over the scarcity of possible examples compared to the huge diversity of potential distinctive features.

2.3 The problem of truth in fiction as an unsupervised learning task

Here are two examples taken from LeCun's lecture on April 8th 2016 which illustrate the challenges of unsupervised learning.

One is the continuation of video shots. Take as input the very short video of someone letting go of a ball down to an inclined plane, and predict the next few seconds. Humans are very good at doing this: we have strong intuitions that the ball will fall and bounce in the direction roughly perpendicular to the inclined plane. But maybe the ball is very flat and it does not bounce, or it is full of helium and it would go up. So there is a bunch of possible continuations of the movie which are very different. A good algorithm will have to define what "possible" precisely means here. But you can see how this becomes extremely difficult when you have to predict all the pixels on the screen (which is what machines do) according to the "possibles". The intuitive physics we, humans,

[3] This is what labelled data provides you with.

use does not predict such low level properties, but rather abstracts away from the negligible features of the video and emphasises the crucial elements of it.

The second example is the modelling of "common sense". Take as input a natural language sentence like: "Gérard took his hat from the table and left the room", and predict all the inferences one can draw from this sentence. There are many. Several inferences are plausible: Gérard extended one of his arm to take his hat; if he was sitting, then he would have stood up; he opened the door before leaving the room; etc. Several inferences are implausible: Gérard used a fishing rod; he crawled to the door; he destroyed the door open; etc. Several inferences are very implausible: Gérard put his hat on using telekinesis, and dematerialise out of the room. Note that any combination of inferences can actually by met by an *ad-hoc* scenario: very implausible inferences will naturally yield explicitly fictional scenarii. Interestingly, these three sets of inferences cannot be distinguished by their cardinality: there are just as many plausible inferences as there are implausible ones. The underlying metric of the plausible needed here is very difficult to formalise so as to make an algorithm out of it.

Neural networks are nowhere near the abilities of humans in these tasks of unsupervised learning yet.

Here is, then, the problem of truth in fiction presented as a task of unsupervised learning. The input for the reader is the text, that is a set of sentences organised in the form of a book. From this data, the reader manages to extract all the fictional truths, or rather a *substantial* part of them. The precise definition of "substantial" here is just as difficult as the definition of "possible" and "plausible" in the above examples. Note that the *a priori* knowledge for this kind of fiction-reading is very large and diverse in nature. First, the reader has linguistic competence, meaning they can read and understand sentences of a natural language, thus mastering in an integrated manner all the strata of linguistic meaning: phonology, syntax, semantics, pragmatics, rhetoric, etc. But they also need to know enough of intuitive physics, intuitive behavioural science, psychology, etc. to imagine the possibility and plausibility of the fictional events. And the reader should also know some things about the distinction between fiction and nonfiction, some things about the medium, and about the genre conventions necessary to make the good inferences.

As can be seen from the examples, the problem of truth in fiction is more complex in practice in that it integrates a great many other unsupervised learning tasks. The solution to the problem of "common sense" is, in a clear sense, presupposed in the problem of fictional truth, just as understanding natural language is presupposed by reading. But this does not imply that the problem of fictional truth is *conceptually* more complex. It may simply be the coordination of several problems of the same complexity. For instance, letter recognition

is as complex a problem as word recognition, though recognising words clearly presuppose the recognising of letters in alphabetical languages. Consequently, it is possible that the problem of fictional truth is as complex as an unsupervised learning task. My stating this, of course, betrays my optimism which the above analogy is trying to back up.

I think it is helpful to see fiction-reading as a problem of unsupervised learning for two reasons: First it helps clarify the kind of cognitive tasks it relates to, even though the complexity of fiction reading may be higher than "simple" unsupervised learning tasks. Second it suggests both that the philosopher of fiction should get interested in machine learning, and that machine learning should get interested in fiction reading practices as studied by philosophers and literary theorists.

References

Currie, Gregory (1990). *The nature of fiction*. Cambridge University Press. DOI: 10.1017/CBO9780511897498.

Friend, Stacie (2017). "The real foundation of fictional worlds". In: *Australasian Journal of Philosophy* 95.1, pp. 29–42. DOI: https://doi.org/10.1080/00048402.2016.1149736.

Lewis, David (1978). "Truth in fiction". In: *American philosophical quarterly* 15.1, pp. 37–46. URL: https://www.andrewmbailey.com/dkl/Truth_in_Fiction.pdf.

Walton, Kendall (1990). *Mimesis as Make-believe: On the Foundations of the Representational Arts*. Harvard University Press.

Woodward, Richard (2011). "Truth in fiction". In: *Philosophy Compass* 6.3, pp. 158–167. DOI: 10.1111/j.1747-9991.2010.00367.x.