Tilburg University
Faculty of Law

Reinforcement learning in the field of tax law

– AI obeying tax legislation through self-learning –

Tax & Technology (TiU)

Paper

Student:	Y. Kaya	J. Molenaar MSc
Study direction:	Tax & Technology	Tax & Technology
ANR:	802709	783469
Date:	14-6-2019	

1 Introduction

1.1 Subject and goal

"AI is likely to be either the best or the worst thing to happen to humanity."
– Stephan Hawking[1]

And dr. Hawking is not alone. Other great minds of our century are concerned about *Artificial Intelligence (AI)* and how it might evolve as a threat to humans. People like Bill Gates[2] and Elon Musk have warned against AI stating that if ever weaponized it could lead to humankind's extinction.[3] Given the fact that AI has already entered our daily lives in the form of self-driving cars, autopilot[4], smart chat bots and that it will only increase further in the future this comes as no great surprise. Meanwhile research in AI shows continuous improvements and more capabilities every day. Google's driverless cars were able to teach themselves how to safely drive from point A to B[5] and Google's DeepMind was able to learn an algorithm by itself how to walk through the use of *Reinforcement Learning (RL)*[6] which is a form of *Machine Learning (ML)*. As AI continuously becomes smarter, it is also being used in a more autonomous manner. Within the scientific community this has led to questions about what the legal status is for autonomous AI-agents and who is responsible for their behaviour.[7] In addition, there is also scientific research on how to make sure AI and autonomous robots will not hurt humans[8], by giving them a moral sense in order to align their interest with humans.[9] There are even researchers who are already calling out for AI-guardians, who need to make sure that other AI algorithms are obeying the law.[10]

In the current literature and practise the focus is mainly on making sure the AI-agents will be able to technically become a part of the community and making sure no humans are harmed along the way through giving AI-agents ethics and a sense of morality.[11] AI-agents however, as they become more autonomous, would have to deal with other laws as well, like contract law and tax law. These so-called

[1] R. McMenemy, "*Stephen Hawking says he fears artificial intelligence will replace humans*", Cambridge News, Nov. 1, 2017, https://bit.ly/2EhxPcc [https://perma.cc/M9VK-EKAQ].
[2] C. Clifford, "*Bill Gates: A.I. is like nuclear energy — 'both promising and dangerous'*", CNBC, Mar. 26, 2019, https://cnb.cx/2FDslZb [https://perma.cc/7WDF-L3MV].
[3] K.J. Ryan, "*Elon Musk (and 350 Experts) Predict Exactly When Artificial Intelligence Will Overtake Human Intelligence*", Inc, Jun. 6, 2017, https://bit.ly/2rxRw8t [https://perma.cc/DW4U-FSQ5].
[4] J. Markoff, "*Planes Without Pilots*", The New York Times, Apr. 6, 2015, https://nyti.ms/2iDQkso [https://perma.cc/84T3-W3XA].
[5] A. Ghoshal, "*How Google's Waymo is using AI for autonomous driving*", Techcircle, May 9, 2018, https://bit.ly/2F4qFHi [https://perma.cc/N5JU-DJCJ].
[6] J. Vincent, "*DeepMind's AI is teaching itself parkour, and the results are adorable*", The New York Times, Jul. 10, 2017, https://bit.ly/2tzwgQq [https://perma.cc/ENV5-WJ6V].
[7] R. Leenes, F. Lucivero, "Laws on Robots, Laws by Robots, Laws in Robots", *Law, Innovation and Technology* 2014/6, No. 2, pp. 193-220.
[8] In order to make sure the famous three laws of Isaac Asimov are abided*: i.) A robot may not injure a human being or, through inaction, allow a human being to come to harm. ii.) A robot must obey orders given it by human beings except where such orders would conflict with the First Law. iii.) A robot must protect its own existence as long as such protection does not conflict with the First or Second Law.*
[9] H. Prakken, "On the problem of making autonomous vehicles conform to traffic law", *Artif Intell Law* 2017/25, No. 3, pp. 341-363.
[10] A. Etzioni, O. Etzioni, "Designing AI Systems that Obey Our Laws and Values", *Communications of the ACM* 2016/59, No. 9, pp. 29-31.
[11] H. Prakken, "On the problem of making autonomous vehicles conform to traffic law", *Artif Intell Law* 2017/25, No. 3, pp. 341-363.

law abiding robots can be realised in multiple ways and in this paper we further investigate how the current improvements in ML can realise that AI-agents obey the law through self-learning methods.

Specifically, we will investigate if an AI-agent can learn how to comply with tax legislation and court cases by itself with the use of the reinforcement learning method. Through showing recent improvements in RL as well as real-life use cases, we will attempt to answer the main question:

Can an AI-agent learn how to obey tax legislation by itself, based on the reinforcement learning method?

In order to draw a solid conclusion three sub-questions will need to be answered throughout the paper:

i. *What is reinforcement learning?*
ii. *What are the use cases and other achievements this far?*
iii. *What are the characteristics of having to comply with tax legislation and court cases and is reinforcement learning suitable for this problem?*

1.2 Outline paper

To find the answers we seek we have split this paper into three parts. In the introduction we have elaborated on what the purpose of our research is. In chapter 2 we will examine what RL exactly is and how it works by studying its relation to deep learning and its place as a component of AI. In addition, we will examine what sort of problems RL can deal with and look into use cases in the field of (tax) law. In chapter 3 we elaborate on the accomplishments of RL in the field of games and simulations until now and examine whether based on RL an AI-agent can learn itself how to comply with tax legislation and court cases. Finally in chapter four we will summarize our findings and answer the main question of this paper.

2 Reinforcement learning

2.1 Introduction

Humans and animals learn by interacting with their environment. Through experimentation knowledge on cause and effect and the consequences of actions are obtained. Then through trial-and-error the right path towards achieving the desired goal is reached. RL is an algorithm that works in a similar fashion. It makes it possible to teach a computer agent without having to supervise it or giving it a pre-labelled database stating whether something is wrong or right. The idea is that this will allow the machine to eventually take the most suitable action and maximize the rewards in a particular situation. In this chapter we will explore what exactly AI and RL are.

2.2 Artificial intelligence

Intelligence is what distinguishes man from animal. The American psychologist David Wechsler defined intelligence as the capability to behave purposefully, think rationally and deal with your environment effectively.[12] Giving a machine the ability to imitate a human brain is either a noble or a very foolish idea. It frightens many since a machine's raw potential has surpassed ours and is ever increasing.[13] However, in contrast to the fearful voices there are also supportive voices stating that it is humans

[12] D. Wechsler, *The measurement of adult intelligence*, Baltimore: Williams & Wilkins 1944, p. 3.
[13] F. van Paasschen, "*The Human Brain vs. Computers - Should we fear artificial intelligence?*", Medium, Jan. 16, 2017, https://bit.ly/2OSU0Id [https://perma.cc/CKR4-6D8R].

developing AI and that it is also in our hands to implement control mechanisms to teach it ethics. Teaching it ethics and morality will align the interests of AI and humankind.[14]

AI is made up of many components to function well. One of the core components is ML which can roughly be divided in the following methods of learning: *supervised learning, unsupervised learning, semi-supervised learning* and *reinforcement learning*.[15]

2.3 Reinforcement Learning

RL is a method where the agent will try out many different actions to create a predictive model on its own. No pre-defined set of data is fed to the machine at all. Learning from its own experiences it will have to form a policy which takes it towards the desired goal. Whenever the agent performs well it gets a reward, whenever it makes a mistake it will get a penalty. The idea is that the agent analyses what behaviour got it the reward and optimizes its behaviour accordingly. In certain environments the rewards will come frequently, in others the reward is only received at the end. For example in the game of ping-pong a reward comes in the form of a point scored against the opponent. In chess checkmating your opponent is the reward and only comes at the end of the game. When learning to crawl every forward movement is an achievement.
Like animals are hardwired to perceive negative rewards such as pain and hunger and positive rewards when eating food so too must the agent be hardwired to recognize both positive and negative rewards.[16] For RL the mathematical frame-work to tackle this problem is known as the *Markov Decision Process (MDP).* It can be visualized[17] as follows in figure 1:

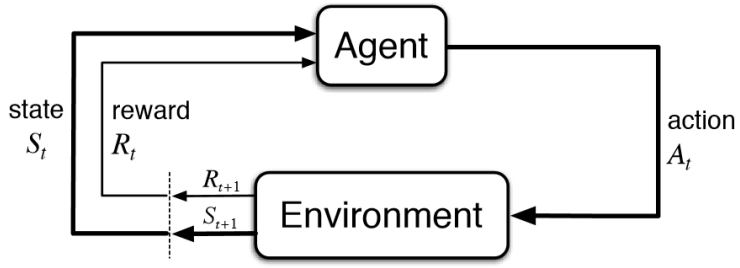

Fig 1. Markov Decision Process (MDP)

The advantages of RL are that with the use of such an algorithm relatively unknown situations can be handled whereas tightly programmed robots can only deal with cases they were programmed to handle. It requires no pre-labelled databases and under the right circumstances the agent can master what they are set to do relatively quickly. RL algorithms balance between exploration and exploitation. Explore different actions and exploit what has worked in the past. It is a great learning method to maximize performance. However, RL also has some disadvantages. There is the credit assignment problem where if a set of actions was performed and only the last action resulted in a negative reward then the algorithm assumes the entire set of actions was bad. Another downside is that the aspect of "risk factoring" within RL puts a high strain on systems. Furthermore RL simply assumes that the world it operates in is Markovian and assumes the agent acts alone in the presence of non-learning operators and it assumes all actions are discrete.[18] Yet these downsides can be somewhat mitigated by using select RL algorithms.[19] RL works in a way that is a mathematical abstraction of reality. There are various RL

[14] Boulder, "*The ethics of artificial intelligence: Teaching computers to be good people*", University of Colorado, Mar. 25, 2019, https://bit.ly/2wNrO1h [https://perma.cc/7PES-XDH2].
[15] This paper focusses on reinforcement learning, for more information on the other sorts of learning methods, see: S. Russell, *Artificial Intelligence: A Modern Approach*, New Jersey: Pearson Education 2010, pp. 694-695.
[16] S. Russell, *Artificial Intelligence: A Modern Approach*, New Jersey: Pearson Education 2010, pp. 830-831.
[17] R.S. Sutton, A.G. Barto, *Reinforcement Learning: An Introduction*, Cambridge: The MIT Press 2018, pp. 47-49.
[18] G.J. Laurent, L. Matignon, N.L. Piat, "The world of independent learners is not markovian", *International Journal of Knowledge-based and Intelligent Engineering Systems* 2011/15, No. 1, pp. 55-64.
[19] R.S. Sutton, A.G. Barto, *Reinforcement Learning: An Introduction*, Cambridge: MIT Press 2018, pp. 361-363.

algorithms which each come with their own advantages and shortcomings.[20] They essentially share the same principle as they all work with a feedback loop and are model-free. For example there is a difference in the policies as some algorithms work with on-policy agents, which learn the value based on its current action derived from the current policy, and the off-policy agents, which learn based on actions obtained from another policy.[21] The optimal policy would be one that maximizes the expected total reward. For the purpose of this paper we will not delve further into the technicalities. Instead we will look at RL in general as a model-free feedback loop.

2.4 Use case for (tax) law

For certain circumstances, RL can be of use in means of constructing an algorithm that is able to solve mathematical problems. It is however not suitable for just any sort of problem. The problem and situation should have certain features. Ideally there should be: i) some kind of trial-and-error aspect making feedback loops possible ii) possibility to reward the algorithm and iii) the situation should be able to fit the MDP-model and lastly iv) there should be a control problem.[22] Over the past decade, there have been multiple use cases where RL has proven itself to be equally qualified or even better than its human counterpart in doing the job like with the work of traffic control[23] or mixing chemicals.[24]

Besides these examples, RL is also an use case for several activities performed by law professionals. Due to a convergence of emerging technology and clients that no longer want to pay for 'sorting out activities' by paralegals, law companies have recently shown more interest in applying AI solutions.[25] As a result of machine learning, documents can be automatically scanned and labelled[26] and contracts - to a certain extent - can be reviewed by AI for irregularities.[27] In addition to these features, court cases can now also automatically be summarized and translated.[28] This could be quite useful for law professionals.[29] The underlying technology, which facilitates these features in the field of legal research, is the combination between RL and *natural language processing (NLP).* NLP helps to analyse, understand and get a sense of human language in a clever way by the process of sentence segmentation, tokenization, lemmatization and stemming.[30] A part of the tasks performed by the NLP-algorithm is

[20] Variations such as Q-learning (off-policy, works with the greedy-policy) and SARSA (on-policy).
[21] S. Huang, "*Introduction to Various Reinforcement Learning Algorithms*", Towards Data Science, Jan. 12, 2018, https://bit.ly/2K66Tje [https://perma.cc/6G4W-ZSL3].
[22] R.S. Sutton, A.G. Barto, *Reinforcement Learning: An Introduction*, Cambridge: The MIT Press 2018, pp.6-7.
[23] I. Arel, C. Liu, T. Urbanik, A. Kohls, "Reinforcement learning-based multi-agent system for network traffic signal control", *IET Intelligent Transport Systems* 2010/4, No. 2, pp. 128-135.
[24] Z. Zhou, X. Li, R.N. Zare, "Optimizing Chemical Reactions with Deep Reinforcement Learning", *ACS Central Science*, 2017/3, No. 12, pp. 1337-1344.
[25] D. Wittenberg, "Rise of the Machine: Artificial Intelligence in the Practice of Law", *ABA Litigation News* 2017/42, No. 2, p. 27.
[26] For instance, 'ROSS Intelligence' is being used by big law companies, for information, see: https://bit.ly/2X9ReoN [https://perma.cc/2VQ7-ZXVY].
[27] Different AI-tools who make use of natural language processing are already used commercially in order to review contracts like 'Kira Systems', 'LawGeex' and 'eBrevia'.
[28] Deloitte is in the process of creating a tool called 'Tax-I' for VAT cases by the European Court of Justice, which can be used for summaries and predicting outcomes of future cases. See also: https://tax-i.deloitte.nl/
[29] If the court cases would be automatically summarized, law professionals would be able to read these quickly after the court made its decision. Furthermore, being able to understand court cases from other – foreign language speaking – countries creates several possibilities which can be advantageous in future cases.
[30] For further information, please see the following article for an overview of what Natural Language Processing is and how it works: https://bit.ly/2XHdaVg [https://perma.cc/K3S3-AG6B].

done with deep RL models, mainly machine translation, text summarization and sentiment analysis.[31]

3 Reinforcement learning for obeying tax legislation

3.1 Introduction

RL has been successfully used by corporations like Google to teach an agent how to walk, how to navigate without using a map, create its own images and even how to play and master certain games better than a human can.[32] In this section we will explore how those accomplishments were achieved and whether a similar design can be used to have an AI-agent learn how to obey tax legislation by itself.

3.2 Reinforcement learning in videogames

Video games have a common feature: they have a restricted amount of actions which can be performed and rewards are allocated to those actions. In addition, the environment is also restricted, as programmed by the developers of the game. It has been shown in the past that with the help of RL can learn itself how to play certain games.[33]

One example of the accomplishments of an AI-agent while using RL is Atari's Breakout.[34] The purpose in this game is to break all bricks as quickly as possible with a ball that will bounce back towards the paddle. If the ball misses the paddle the game is over. The state is a pre-processed image of the screen and the agent can take four possible actions: i.) do nothing ii.) fire ball iii.) move paddle left iv.) move paddle right. The agent receives the image frame from the game and then decides on an action. After every decision the game simulator executes the action and gives the agent a reward: +1 each time the ball hits a brick and it disappears making the score increase, 0 if nothing happens and -1 if the paddle misses the ball. Every action the agent takes it stores in its memory. In Google's DeepMind project the agent mastered the game in two hours. Eventually the AI-agent was able to beat human scores in 70% of the Atari games.[35] Not just videogames were mastered through RL, also games like the Chinese board game Go. With AlphaGO DeepMind beat European Go champion Fan Hui by winning 5 games to 0.[36] Since Go works with a so called optimal value, which can be determined from every board state/position, the agent can master which positions are best for its pieces through RL.

3.3 Google's walking AI-agent

As shown earlier in this paper, RL has made impressive progress. What these accomplishments have in common is the relatively simple environment with a well-defined reward function as is typical in games.

---

[31] T. Young, D. Hazarika, S. Poria, E. Cambria, "*Recent Trends in Deep Learning Based Natural Language Processing*", Nov. 25, 2018, https://arxiv.org/pdf/1708.02709.pdf [https://perma.cc/UJP7-7BAH].
[32] M. Ellis, "*5 Amazing Things Google's DeepMind AI Can Already Do*", MakeUseOf, Oct. 29, 2018, https://bit.ly/2Znu7oc [https://perma.cc/9LX4-8PU6].
[33] V. Mnih, K. Kavukcuoglu, D. Silver, A. Graves, I. Antonoglou, D. Wierstra, M. Riedmiller, "*Playing Atari with Deep Reinforcement Learning*", Dec. 19, 2013, https://www.cs.toronto.edu/~vmnih/docs/dqn.pdf [https://perma.cc/ZTF3-YRQM].
[34] R. Sidhu, "*Rise of Deep Mind — Google's General Purpose AI*", Medium, Apr. 12, 2019, https://bit.ly/2Iujbye [https://perma.cc/TT8N-WNQG].
[35] V. Mnih, K. Kavukcuoglu, D. Silver, A. Graves, I. Antonoglou, D. Wierstra, M. Riedmiller, "*Playing Atari with Deep Reinforcement Learning*", Dec. 19, 2013, https://www.cs.toronto.edu/~vmnih/docs/dqn.pdf [https://perma.cc/ZTF3-YRQM].
[36] R. Williams, "*Fan Hui: What I learned from losing to DeepMind's AlphaGo*", iNews, May. 25, 2019, https://bit.ly/2HWyfEp [https://perma.cc/4XLC-QYKD].

The reality of the world we live in however is less well-defined and environments can easily change. A less defined environment also has an impact on how the reward structure should be for an AI-agent in order for it to perform specific behaviour.

This is also the case with a – for humans – basic task of walking around. Walking around might seem an 'easy' task, but the environment is continually changing and incentivising an algorithm to keep going from point A to point B is quite a challenge[37]. However, programmers at Google's DeepMind were also – besides creating algorithms that can play videogames – able to create an algorithm that used RL for learning how to walk.[38] As the walking AI-agent has resemblance with our Tax AI-agent, we will dig a bit deeper into the specifics. Below in table 1 we've summarized the main characteristics of DeepMind's walking AI-agent, the environment and task and reward system.

Agent	Observations
DeepMind used three different bodies, which were increasingly complex in comparison with each other, in terms of rotatability and number of joints: the *Planar walker* (two attached legs), the *Quadruped* (4 limbs, similarities with a spider) and lastly the *Humanoid* (resemblance with a human).	The agents had sensors with which they could 'feel' certain elements of their body, among others: the angle of their joint, the speed and torque sensors. Furthermore they also received external information about their environment, which made it possible for the AI-agent to 'see' its surroundings.
Environment	**Task & reward system**
The environment was a simulation of different terrains, with multiple obstacles along the way. The obstacles could differentiate from a hole in the ground, to boxes and other obstacles on and above the ground to walls where the AI had to walk around. The terrain itself also could be different levels of skewness. DeepMind furthermore decided to make the training environments increasingly complex in terms of the aforementioned characteristics, based on prior accomplishments.	In trying to keep the challenge as simple as possible, the basic task of AI-agent was to get from point A to point B. The corresponding reward system was also kept straightforward and similar along all the different terrains. The AI-agent would get points if it would get further from point A to point B, so more points if any closer to B. Furthermore it would get (little) negative points if the body would torque, so it wouldn't start walking from the side or backwards.

Table 1. Key model characteristics of DeepMind's walking AI-agents

The AI-agent didn't have any prior knowledge of the terrain or how to walk or jump over obstacles. Through code the AI-agents were simply given the laws of psychics, e.g. the max rotation of each joint, max power which could be produced and max jumping height given the rules of gravity. Given the fact that no prior knowledge was available and purely on the basis of a form of RL, the results[39] are quite

[37] For earlier research on RL-algorithms with continuous control situations see: T.P. Lillicrap, J.J. Hunt, A. Pritzel, N. Heess, T. Erez, Y. Tassa, D. Silver, D. Wierstra, "*Continuous control with deep reinforcement learning*", Feb. 29, 2016, https://arxiv.org/pdf/1509.02971.pdf [https://perma.cc/TXP5-E6XN].

[38] N. Heess, D. TB, S. Sriram, J. Lemmon, J. Merel, G. Wayne, Y. Tassa, T. Erez, Z. Wang, S.M. Eslami, M. Riedmiller, D. Silver, "*Emergence of Locomotion Behaviours in Rich Environments*", Jul. 10, 2017, https://bit.ly/2Xc1Kfh [https://perma.cc/QX3D-86HQ].

[39] Which can be found through a link on: https://deepmind.com/research/publications/emergence-locomotion-behaviours-rich-environments/

amazing. The AI-agents taught themselves to walk from point A to point B within the given limits and obstacles of the environments and adapted to the increasingly complex terrain.[40]

3.4 TAX-AI agent

We've seen that with the current state of technology, AI-agents are able to learn how to move from point A to point B, learning how to deal with obstacles, blockades and continuously changing environments. The setup of DeepMind's AI-agent might also be applicable for an AI-agent who needs to comply with tax law and cases. Whilst investigating this we will explore the conditions and characteristics of how our TAX-AI model should look like in order for the TAX-AI agent to be able to learn how to comply with the tax legislation. We do this by defining the key parts of the Markov Decision Process model which was depicted earlier in this paper for our TAX-AI case. Subsequently we will discuss in more detail whether obeying tax legislation can be learned through RL given the specific characteristics of tax legislation and known limitations of the RL-method.

Markov's Decision process model modified for TAX-AI.

In the table below an overview is given with the characteristics of how the TAX-AI RL model could have the characteristics as shown below in table 2.

Agent	Actions
The TAX-AI agent could have the status of a company or a natural person, who needs to obey the tax law. The agent is a software simulation and in the future could be part of a software package or an autonomous creation.	Every legal binding action can have tax consequences. For this reason, the TAX-AI agent can perform all sorts of legal actions, for instance: transactions with clients and suppliers. [41]
Environment	**Task & reward system**
The environment consists of tax legislation and cases. This means that for each state the Agent has a couple of actions it can take. For each action there are related tax laws and past court cases that limit give a legal bandwidth. Although several actions might be possible, not every action is always legal given the current laws and court cases. In a way the environment looks like an intersection with multiple legal possibilities.	The task of the TAX-AI agent would be to obey the law. Each possible action in the tax law environment will lead to a state (legal or not legal given the current law and cases) and reward positive if legal and negative if illegal.[42] For stimulating the TAX-AI agent to performing actions, instead of doing nothing which could be legal, there will be a negative reward when not undertaking any actions.

Table 2. The RL-model characteristics for the TAX-AI agent

[40] N. Heess, D. TB, S. Sriram, J. Lemmon, J. Merel, G. Wayne, Y. Tassa, T. Erez, Z. Wang, S.M. Eslami, M. Riedmiller, D. Silver, "*Emergence of Locomotion Behaviours in Rich Environments*", Jul. 10, 2017, https://bit.ly/2Xc1Kfh [https://perma.cc/QX3D-86HQ].

[41] Every action in a way is like a new 'Atari' game, where different actions with tax complications can follow and different rules apply. Every action, as with every level in Atari, should be trained with training data and should be configured with the related tax legislation.

[42] The bandwidth of the reward is usually between -1 and 1, as is followed in our example.

These characteristics of Markov's decision model are visualized below, in figure 2.

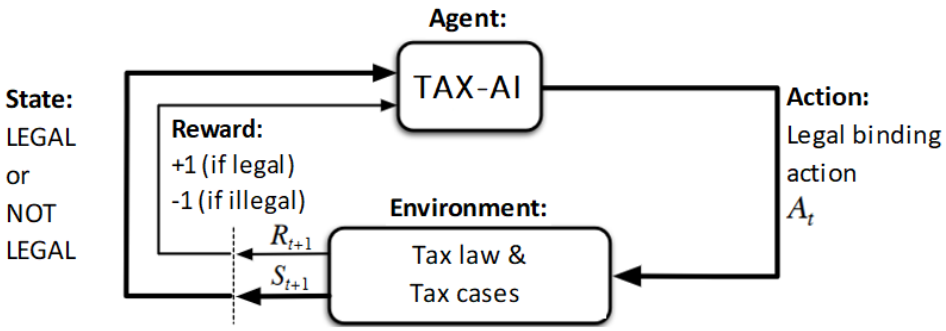

Fig 2. Markov's Decision process model modified for TAX-AI

An example of how this might look like is as follows. Visualize a (software) TAX-AI Agent who needs to learn how to apply the correct VAT rate given a certain state. The environment would be the tax legislation which states that there are 3 tariffs: 0%, 9% and 21%: using another tariff will (already) result in a negative rewards. Then based on the circumstances and facts the TAX-AI agent will need to learn which rate to use for different transactions according to their characteristics. It therefore needs to know the characteristics of the good/service and the available actions. Based on the legislation it can than further be deducted that there is a 'normal' tariff of 21% (lex generalis) and everything else is an exception (lex specialis). It would also need to learn from the related court cases when it is allowed to charge 0%.[43]

VAT law should be relatively simple to learn, given its deductive nature and several steps which need to be taken in a certain order. But the logical reasoning part alone isn't necessarily the issue when it comes to RL. In the last past of this section we will elaborate further on the possible issues by discussing the characteristics of tax legislation and cases, related to each key aspect of RL.

Agent & actions

First of all there needs to be a TAX-AI agent, this could be either a software programme who makes transactions for a company, an autonomous vehicle or a simulation in a game. Given the recent developments as shown by Google's DeepMind, this no longer should be an issue. Next, the agent needs to be able to sense what the facts are. For instance which goods or services it will sell or buy.[44] However, depending on the sort of agent (robot or software) sensing what sort of products need to be sold or purchased could be quite a challenge. In addition, as obeying with tax legislation is not a simple videogame where you can only press a few buttons, programmers would have to program all possible legal binding actions in the algorithm of the agent. Given the subsequent actions that normally need to take place when buying or selling products or wider: obeying with CIT law for instance. This wider domain of actions could be complex[45], but possible actions could also be kept narrow in the beginning.

[43] For instance in case of a delivery of goods between entrepreneurs in different Union states, also known as intra-Community transactions.
[44] Perhaps this can be done through a link with the European Customs manual where are goods are categorized.
[45] Wallach, W., Franklin, S., & Allen, C. (2010). "A conceptual and computational model of moral decision making in human and artificial agents.", *Topics in cognitive science*, *2*(3), pp. 454-485.

Environment & State

Formalizing law

In order for the TAX-AI agent to learn from its environment every tax law, exemptions and cases should be formalized meaning: translated into code. Although tax laws and cases have the big advantage that a lot of situations are described and can therefore be translated into *if* and *then* statements, earlier attempts to do so with traffic law [46] and privacy legislation[47] have proven that this is complex task. This is especially the case with tax legislation where it is common that a lot of exemptions exist and apply in many cases. Moreover, even with more principle based moral codes, literature shows that it is troublesome to let AI-agents learn these principles with RL.[48]

Consistency & changing laws

In our case we also examine whether a TAX-AI agent can obey with tax cases. It is a known fact that court cases, from national level to for instance European level, are not always consistent. This could either be at the same time due to different judges or across time, due to changing paradigms. Consistency is an important factor for RL in order for an AI-algorithm to learn which actions yields a positive reward and which don't. Inconsistency and noise, in combination with a scarce amount of court cases[49], make it hard and troublesome to learn the appropriate behaviour for a RL-agent. The cases however, could also be used for training purposes: a kind of detailed examples for how the law should work.[50]

If it would be possible to overcome the previously mentioned complexities, and you eventually go "live" you want the algorithm to be complete, close to perfect and so it will not make any – or hardly any - mistakes when it comes to compliance with laws and regulations. However, it is a known fact that the tax law legislation and cases are constantly changing. This would lead to an algorithm that is always a few steps behind, and needs to retrain itself first with the changes, in order for it to perform the appropriate – lawful – action in the future. The time needed for trial-and-error leads to a TAX-AI agent that does not fully obey the current tax legislation and cases, which could be a negative side for possible use-cases.[51] This problem would not exist if the new laws could be directly programmed and formalized into the 'brain' of an AI-agent.

[46] H. Prakken, "On how AI & law can help autonomous systems obey the law", *AI4J–Artificial Intelligence for Justice* 2016/42, pp. 42-46.
[47] R. Leenes, F. Lucivero, "Laws on Robots, Laws by Robots, Laws in Robots", *Law, Innovation and Technology* 2014/6, No. 2, pp. 193-220.
[48] J.F. Bonnefon, A. Shariff, "The social dilemma of autonomous vehicles." *Science* 2016/352, pp. 1573-1576.
[49] According to Deloitte's Tax-I, the European Court of Justice ruled in 1153 cases, https://bit.ly/2Igs3bS [https://perma.cc/47DS-EDGF].
[50] H. Prakken, "On the problem of making autonomous vehicles conform to traffic law", *Artif Intell Law* 2017/25, No. 3, pp. 341-363.
[51] For example: reaching compliance with AI agents which make transactions for companies, or as a tool to offer companies guidance on the steps to be taken by companies.

Rewards

Grey area

As widely known in the field of tax law, it can be a very complex domain where legislation and court cases have passages which are known as 'grey': not fully specified conditions are given when a certain action is allowed. This could raise a problem with RL since in these cases it is not 100% sure what the *state* and therefore the *reward* should be.[52] Nevertheless, it should be remarked that having vague norms isn't necessarily a problem in the field of RL: the TAX-AI agent is not trained to understand and to reason with the legislation. It is purely learning how to obey the law. It could therefore simple form a policy on how to deal with the vague terms.[53] This would however demand a lot from the programmers since these vague norms still need to be formalized in the environment.

Choosing an action always depends on how the TAX-AI agent evaluates its surroundings and is still based on statistical choosing of the best action with the highest reward. This by nature has the possibility of including wrong, illegal actions as well.[54] Furthermore there is the exploration versus exploitation balance, where an AI-agent will explore a bit as well. This may lead to illegal actions for the sake of reward optimization in the future.[55] Lastly, Google's DeepMind AI has shown in the past that reward optimizing behaviour could wind up in very aggressive tactics and actions.[56] If the rewards are not be carefully optimized for law abiding behaviour, it could lead to a TAX-AI agent that takes tax actions which could be marked as border exploratory behaviour.

[52] H. Prakken, "On the problem of making autonomous vehicles conform to traffic law", *Artif Intell Law* 2017/25, No. 3, pp. 341-363.
[53] Idem
[54] Idem.
[55] S. Bubeck, N. Bianchi, "Regret analysis of stochastic and nonstochastic multi-armed bandit problems." *Machine Learning*, 2012/5, No. 1, pp. 1-26.
[56] J.Z. Leibo, "*Multi-agent Reinforcement Learning in Sequential Social Dilemmas*", Feb. 10, 2017, https://storage.googleapis.com/deepmind-media/papers/multi-agent-rl-in-ssd.pdf [https://perma.cc/9R7A-NCEZ].

4 Conclusion

In this paper we have systematically examined the *machine learning (ML)* function of *reinforcement learning (RL)* and whether it can prove to be a solution to our main question: "*Can an artificial intelligence (AI) agent learn how to obey tax legislation by itself, based on the reinforcement learning method?"*

First we questioned what RL exactly is. We discovered that this method of ML works by learning through reward optimization and is highly dependent on feedback from its environment. It must know what it needs to interpret as a reward and as a punishment and finally learn what the optimal path towards the desired result is. RL uses the mathematical frame-work known as the *Markov Decision Process (MDP)* for its learning process. The advantages of RL are that it can handle relatively unknown situations since no pre-labelled set of data is necessary for it to learn and that it is able to explore and exploit different paths all the while finding the optimal path towards its objective.

Secondly, we looked at relevant use cases and results that RL has achieved so far. We discovered that RL can be applied in certain situations: there should be a trial-and-error aspect, possibility of rewards, the situation needs to fit the MDP-model and there should be a control problem. We found that RL has already successfully been used for specific cases like traffic control and mixing chemicals and that it performed better than humans. In addition, we also discovered within the field of tax, RL is already being put to use: scanning and labelling documents and screening contracts and automatically summarizing and translating court cases belong to the possibilities of AI now thanks to RL. Through examining Google's DeepMind successful RL framework, we've also shown that RL is currently capable of playing videogames and it learned itself how to walk.

Finally we applied our findings to solving the main question. For this we defined the key parts of the MDP model for our own TAX-AI case. The TAX-AI agent would be a software simulation of a company or natural person who needs to obey the tax legislations. It can perform all sorts of legal actions like transactions with clients and suppliers. The environment consists of tax legislation and relevant court cases. It will look like an intersection with multiple legal actions where the task of the agent is to obey the law. Every action it takes will be deemed either legal or illegal by the environment and rewarded or punished accordingly. Not undertaking any actions is deemed punishable as well.

Our conclusion to the main question of this paper is that it could work if programmers are able to formalize the tax legislation and cases into the environment. However, there are many downsides, amongst others no guarantee of law abiding behaviour due to the constantly changing environment and exploration activities. So, is it desirable to use RL for tax compliance? This remains the question and the answer is dependent on the use case it will be needed for. For example: rewards for tax compliant behaviour could mean the agent becomes extremely conservative in applying tax legislation. Inversely this method could also lead to aggressive tax planning, as a result of reward optimization yet *suboptimal* in the legal sense since it might mean finding ways to minimize taxation. So the question now becomes: *do we really want to risk aggressive tax exploratory behaviour by machines*? Our answer is: let's hope not and let's think twice before we select RL for learning AI-agents to comply with tax legislation.