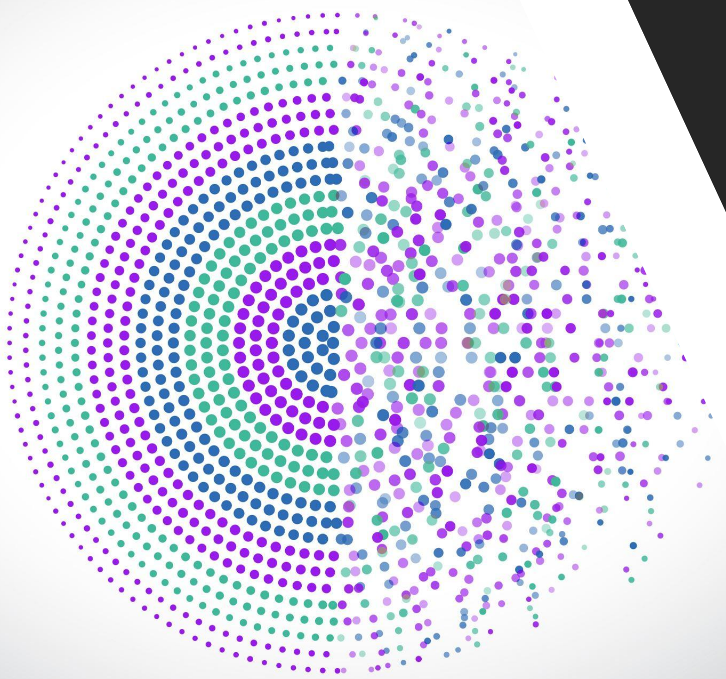


Reinforcement Learning

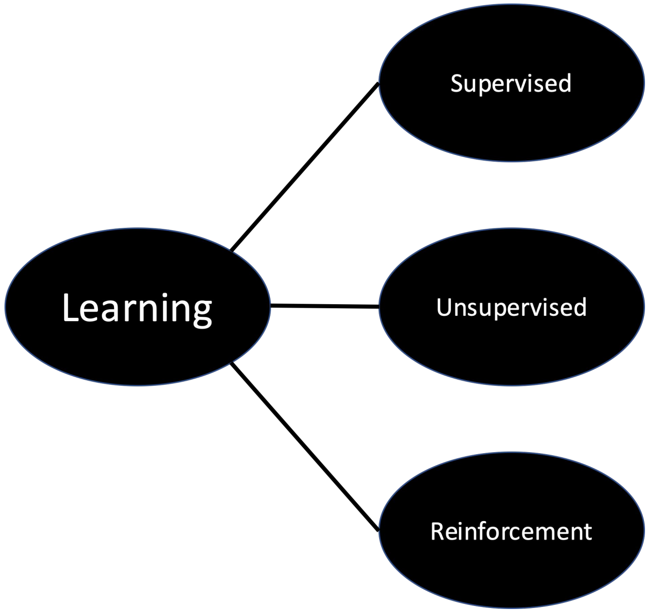
Applications and Future Research

Dr Amita Kapoor,

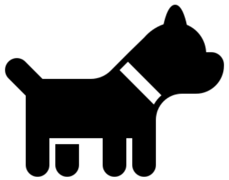
Associate Professor, SRCASW, University of Delhi, India

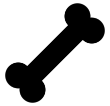
Email: dr.amita.kapoor@ieee.org

Reinforcement Learning

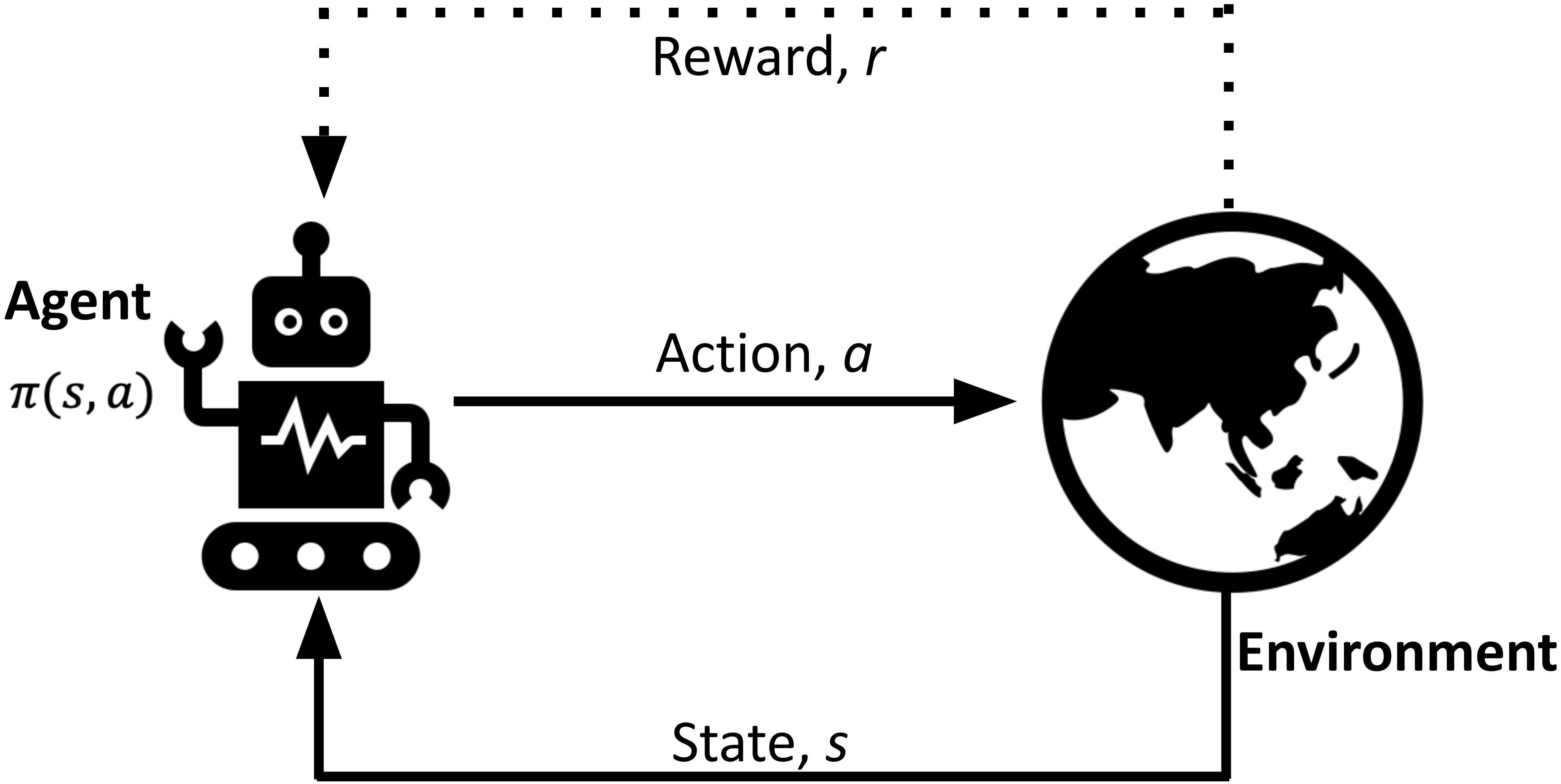


- A type of machine learning
- It is between supervised and unsupervised
- Inspired from how animals learn from experience.

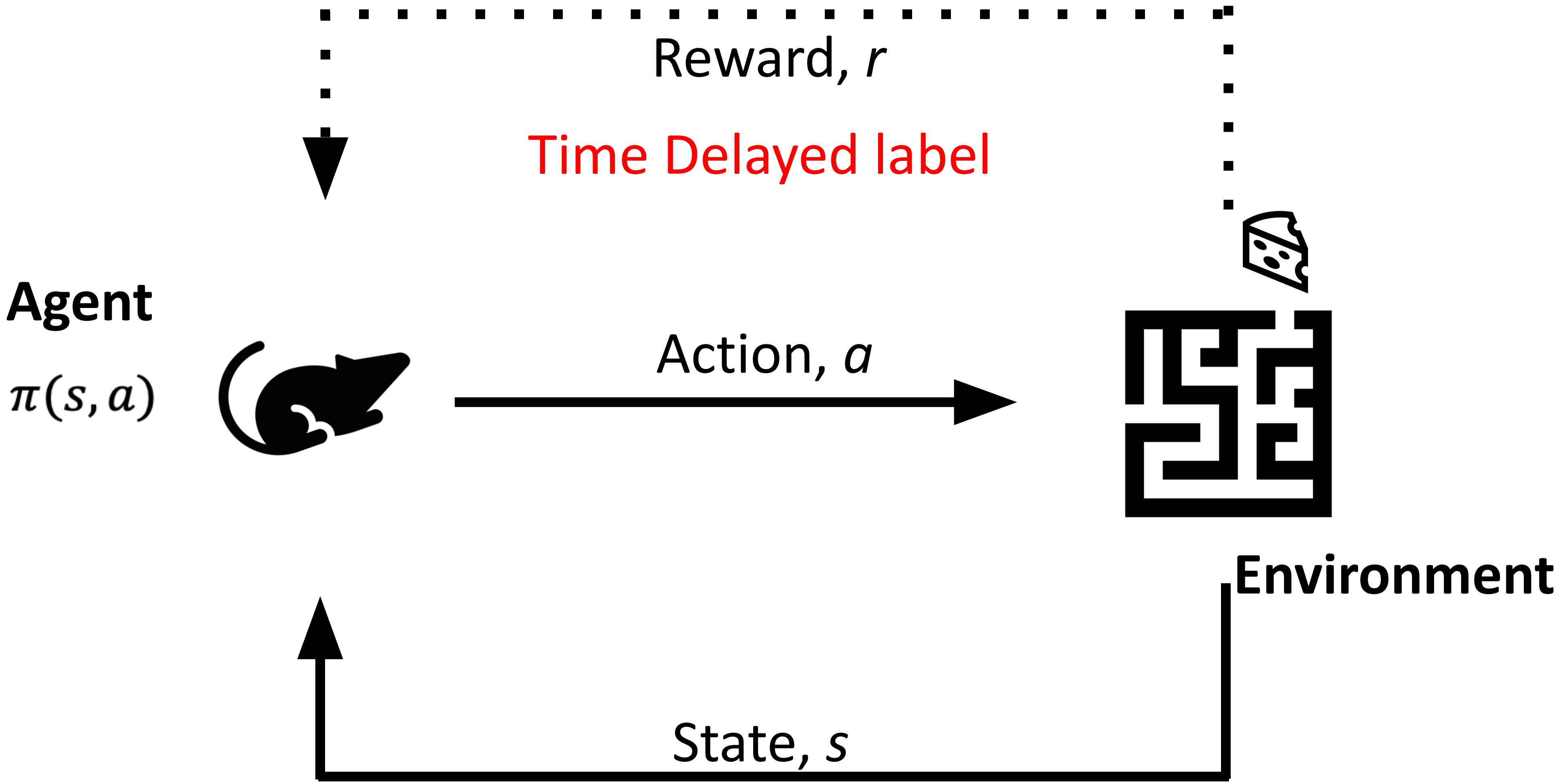




Reinforcement Learning



Reinforcement Learning

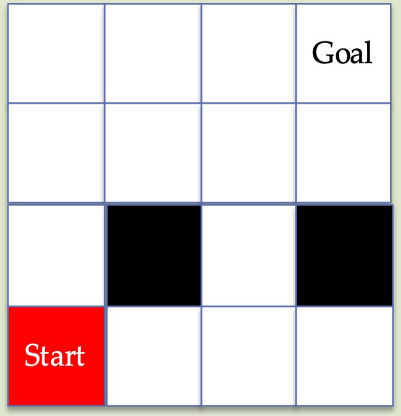


RL Components: State Space S

- Observation of environment.
- Set of all possible states the environment can be in.

$s \in S$

Agent finding path in the maze



s = [[0,0,0,0]
[0,0,0,0]
[0,X,0,X]
[1,0,0,0]]

Agent controlling steering wheel in self-driving car

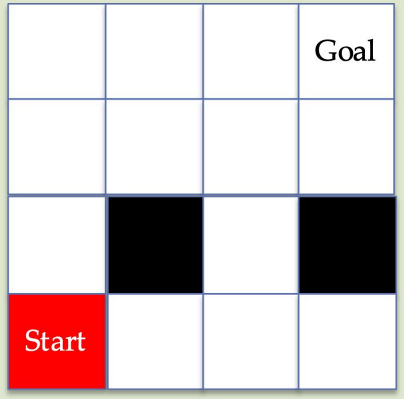
s = The image of the road in-front

Gulli, A., Kapoor, A., & Pal, S. (2019). *Deep learning with TensorFlow 2 and Keras: regression, ConvNets, GANs, RNNs, NLP, and more with TensorFlow 2 and the Keras API*. Packt Publishing Ltd.

RL Components: Action Space *A(s)*

Set of all possible things that the agent can do in a particular state *s*.

Agent finding path in the maze



s = [[0,0,0,0]
[0,0,0,0]
[0,X,0,X]
[1,0,0,0]]

a = [up, down, left, right, no change]

Agent controlling steering wheel in self-driving car

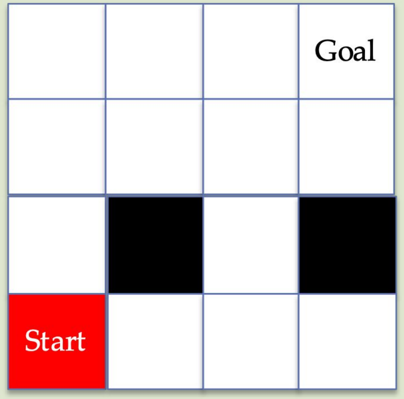
s = The image of the road in-front

a = The angle by which steering wheel is to be rotated

RL Components: Reward *r(s,a,s')*

A scalar value returned by the environment based on the agent's action/s

Agent finding path in the maze



$$s = \begin{bmatrix}[0,0,0,0]\\ [0,0,0,0]\\ [0,X,0,X]\\ [1,0,0,0]\end{bmatrix}$$

a = [up, down, left, right, no change]

Agent controlling steering wheel in self-driving car

s = The image of the road in-front

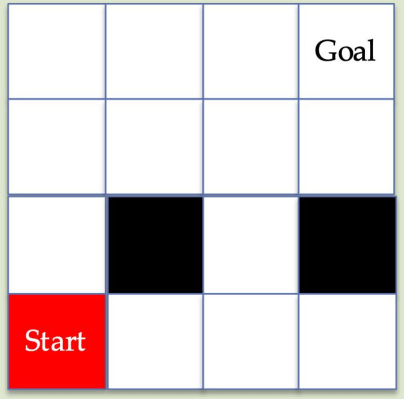
a = The angle by which steering wheel is to be rotated

RL Components: Policy $\pi(s)$

Mapping between each state and the action to take in that state

Optimal Policy $\pi^*(s)$

Agent finding path in the maze



s = [[0,0,0,0]
[0,0,0,0]
[0,X,0,X]
[1,0,0,0]]

a = [up, down, left, right, no change]

Agent controlling steering wheel in self-driving car

s = The image of the road in-front

a = The angle by which steering wheel is to be rotated

RL Components: Return G_t

Discounted sum of all future rewards starting from current time

$$G_t = \sum_{k=t}^{\infty} \gamma^k r_k = \gamma^t r_t + \gamma^{t+1} r_{t+1} + \ \dots + \gamma^{t+n} r_{t+n} + \ \dots$$

Discount factor $\qquad 0 < \gamma < 1$

Discounted total return

RL Components: Q function

Discounted sum of all future rewards starting from current time

$$G_t = \sum_{k=t}^{\infty} \gamma^k r_k = \gamma^t r_t + \gamma^{t+1} r_{t+1} + \ldots + \gamma^{t+n} r_{t+n} + \ldots$$

Discount factor $\quad 0 < \gamma < 1$

$$Q(s_t, a_t) = \mathbb{E}[G_t | s_t, a_t]$$

Expected total future reward an agent in state, *s*, can receive by performing action, *a*

RL Components: Q function

Discounted sum of all future rewards starting from current time

$$G_t = \sum_{k=t}^{\infty} \gamma^k r_k = \gamma^t r_t + \gamma^{t+1} r_{t+1} + \ \dots + \gamma^{t+n} r_{t+n} + \ \dots$$

Discount factor $\quad 0 < \gamma < 1$

$$Q(s_t, a_t) = \mathbb{E}[G_t | s_t, a_t]$$

$$\pi^*(s) = \arg\max_a Q(s, a)$$

RL Components: Value function

Discounted sum of all future rewards starting from current time

$$G_t = \sum_{k=t}^{\infty} \gamma^k r_k = \gamma^t r_t + \gamma^{t+1} r_{t+1} + \ \ldots + \gamma^{t+n} r_{t+n} + \ \ldots$$

Discount factor $\quad 0 < \gamma < 1$

$$V^{\pi}(s_t) = \mathbb{E}[G_t | s_t]$$

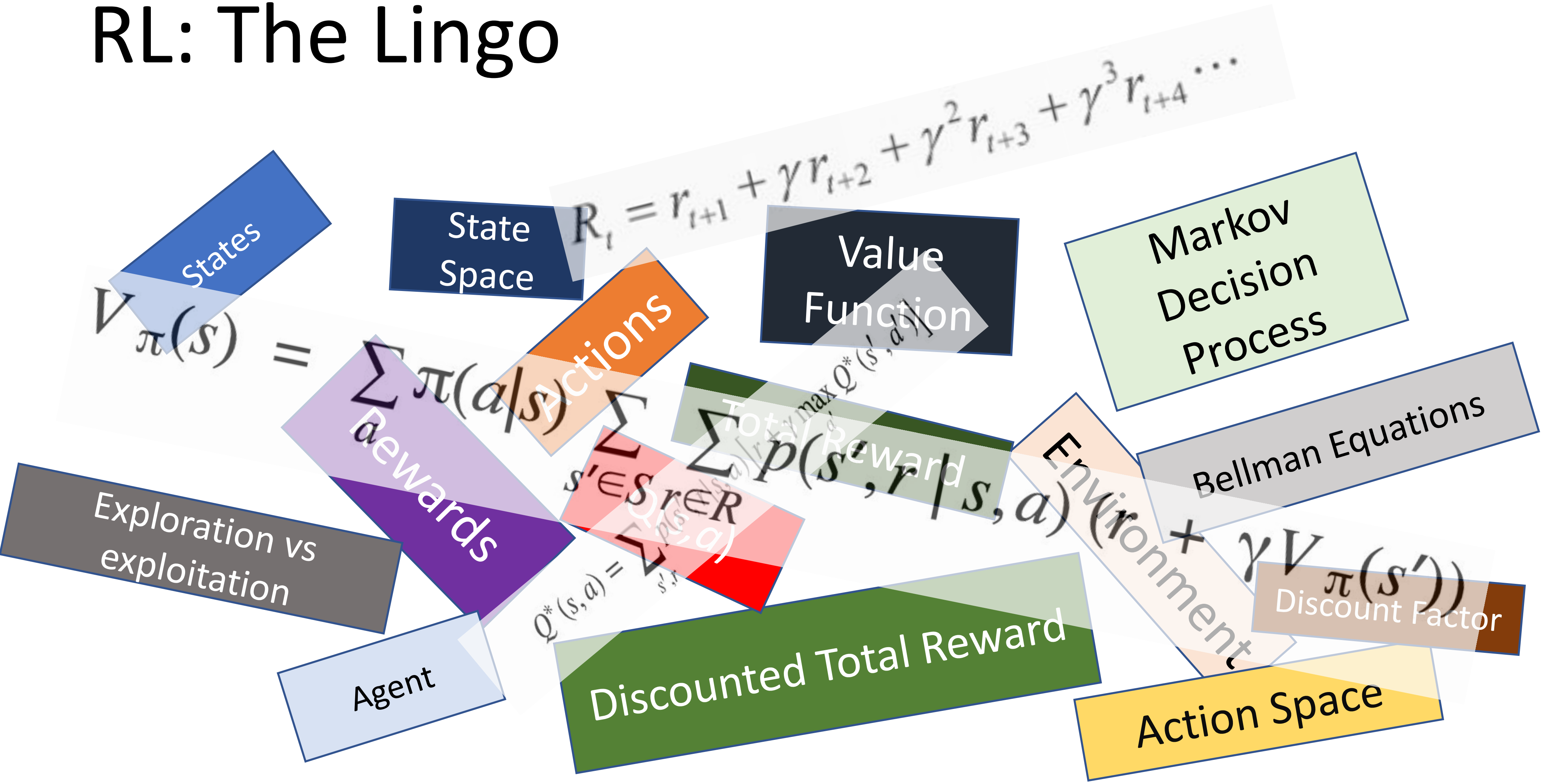
Reinforcement Learning Algorithms

- Value Based Learning
- Agent learns $Q(s,a)$

$$\pi^*(s) = \arg\max_a Q(s,a)$$

- Policy Based Learning
- Agent learn $\pi(s)$
- Samples an action from the policy.

RL: The Lingo



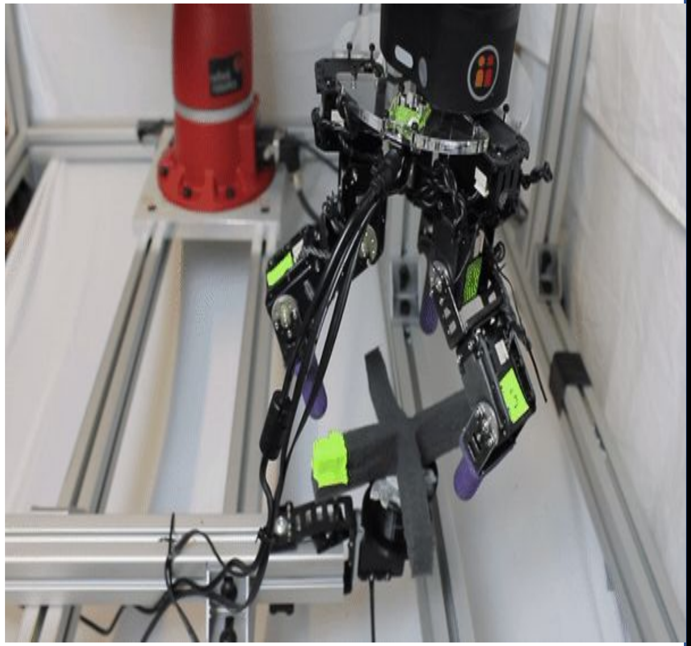
States
State Space
Value Function
Markov Decision Process
Actions
Rewards
Total Reward
Bellman Equations
Environment
Exploration vs exploitation
Q(s,a)
Discount Factor
Agent
Discounted Total Reward
Action Space
$R_t = r_{t+1} + \gamma r_{t+2} + \gamma^2 r_{t+3} + \gamma^3 r_{t+4} \cdots$
$V_\pi(s) = \sum_a \pi(a|s) \sum_{s' \in S, r \in R} p(s', r | s, a)(r + \gamma V_\pi(s'))$

Applications

Games

- Mnih, V., Kavukcuoglu, K., Silver, D., Graves, A., Antonoglou, I., Wierstra, D., & Riedmiller, M. (2013). Playing atari with deep reinforcement learning. *arXiv preprint arXiv:1312.5602*.
- Mnih, V., Kavukcuoglu, K., Silver, D. *et al.* Human-level control through deep reinforcement learning. *Nature* **518,** 529–533 (2015). https://doi.org/10.1038/nature14236
- Silver, D., Schrittwieser, J., Simonyan, K., Antonoglou, I., Huang, A., Guez, A., Hubert, T., Baker, L., Lai, M., Bolton, A., Chen, Y., Lillicrap, T., Hui, F., Sifre, L., van den Driessche, G[redacted] Graepel, T. and Hassabis, D. 2017. Mastering the game of Go without human knowledge. Nature, 550, 7676 (Oct. 2017), 354--359

Robotics



- Kumar, V., Todorov, E., & Levine, S. (2016, May). Optimal control with learned local models: Application to dexterous manipulation. In *2016 IEEE International Conference on Robotics and Automation (ICRA)* (pp. 378-383). IEEE.
- Gupta, A., Eppner, C., Levine, S., & Abbeel, P. (2016, October). Learning dexterous manipulation for a soft robotic hand from human demonstrations. In *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)* (pp. 3786-3793). IEEE.
- Rajeswaran, A., Kumar, V., Gupta, A., Vezzani, G., Schulman, J., Todorov, E., & Levine, S. (2017). Learning complex dexterous manipulation with deep reinforcement learning and demonstrations. *arXiv preprint arXiv:1709.10087*.

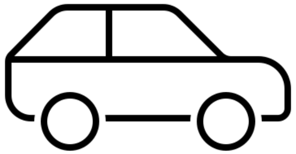
Finance

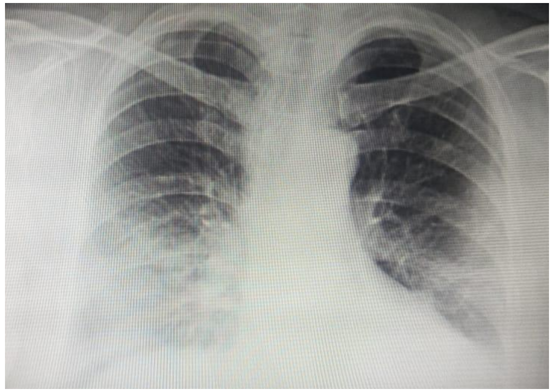
- Portfolio Optimization
- Optimal trade execution
- Pricing strategy in insurance agency

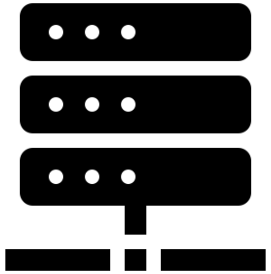
- Jiang, Z., Xu, D., & Liang, J. (2017). A deep reinforcement learning framework for the financial portfolio management problem. *arXiv preprint arXiv:1706.10059*.
- Zhang, Z., Zohren, S., & Roberts, S. (2020). Deep reinforcement learning for trading. *The Journal of Financial Data Science*, *2*(2), 25-40.
- Krasheninnikova, E., García, J., Maestre, R., & Fernández, F. (2019). Reinforcement learning for pricing strategy optimization in the insurance industry. *Engineering applications of artificial intelligence*, *80*, 8-19.

Applications Contd.

- Ridesharing order dispatching
- Medical Image report generation
- Data center cooling







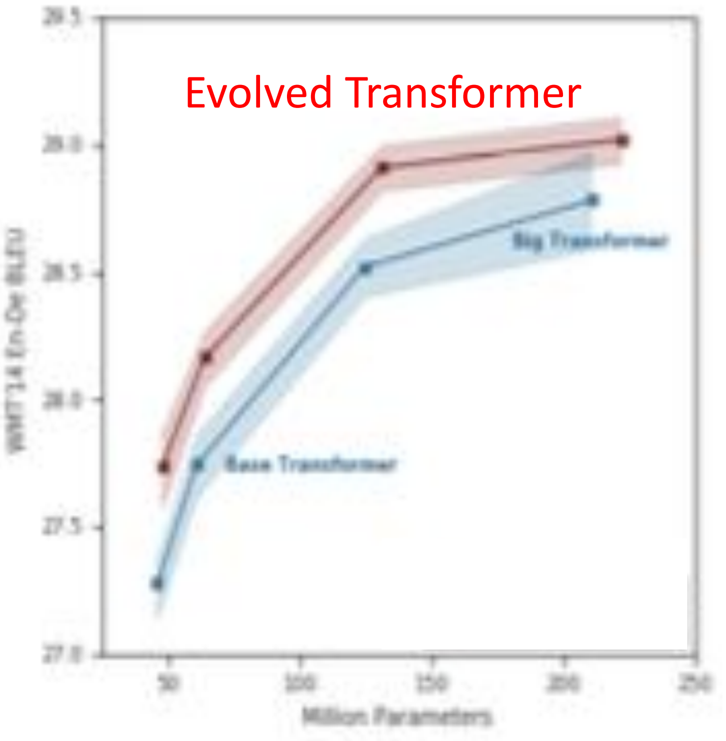
Future Research Directions

Automated Machine Learning

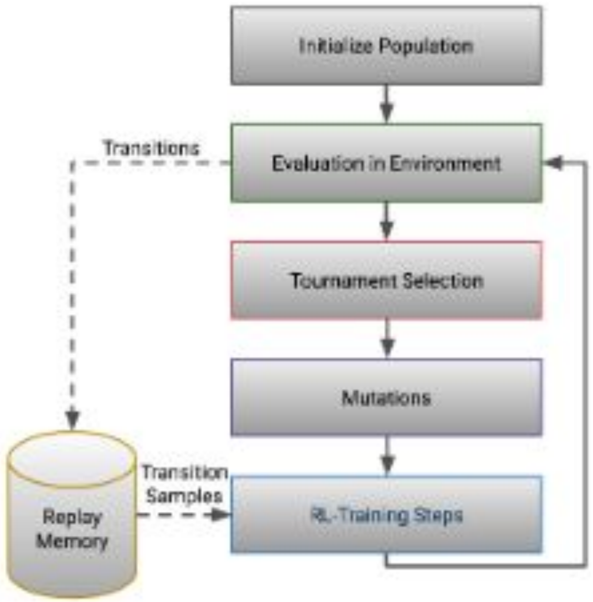
Data Preparation → Feature Selection → Automatic Model

- Model Search
- Hyperparameter Search

- Reinforcement Learning
- Evolutionary Algorithms



AutoRL

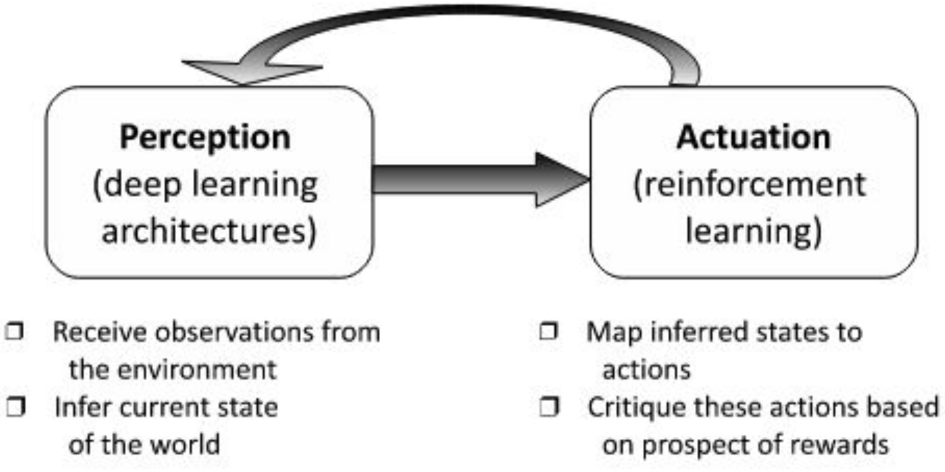


- RL is very sensitive to hyperparameters:
  - Epsilon- exploration/exploitation
  - Discount factor
  - Replay buffer
  - Learning rate
- Moving target problem
- SEARL – population based hyperparameter optimization
- Automated Reward- using Actor – Critic networks

- Franke, J. K., Köhler, G., Biedenkapp, A., & Hutter, F. (2020). Sample-Efficient Automated Deep Reinforcement Learning. *arXiv preprint arXiv:2009.01555*.
- Chiang, H. T. L., Faust, A., Fiser, M., & Francis, A. (2019). Learning navigation behaviors end-to-end with autorl. *IEEE Robotics and Automation Letters*, *4*(2), 2007-2014.

Artificial General Intelligence

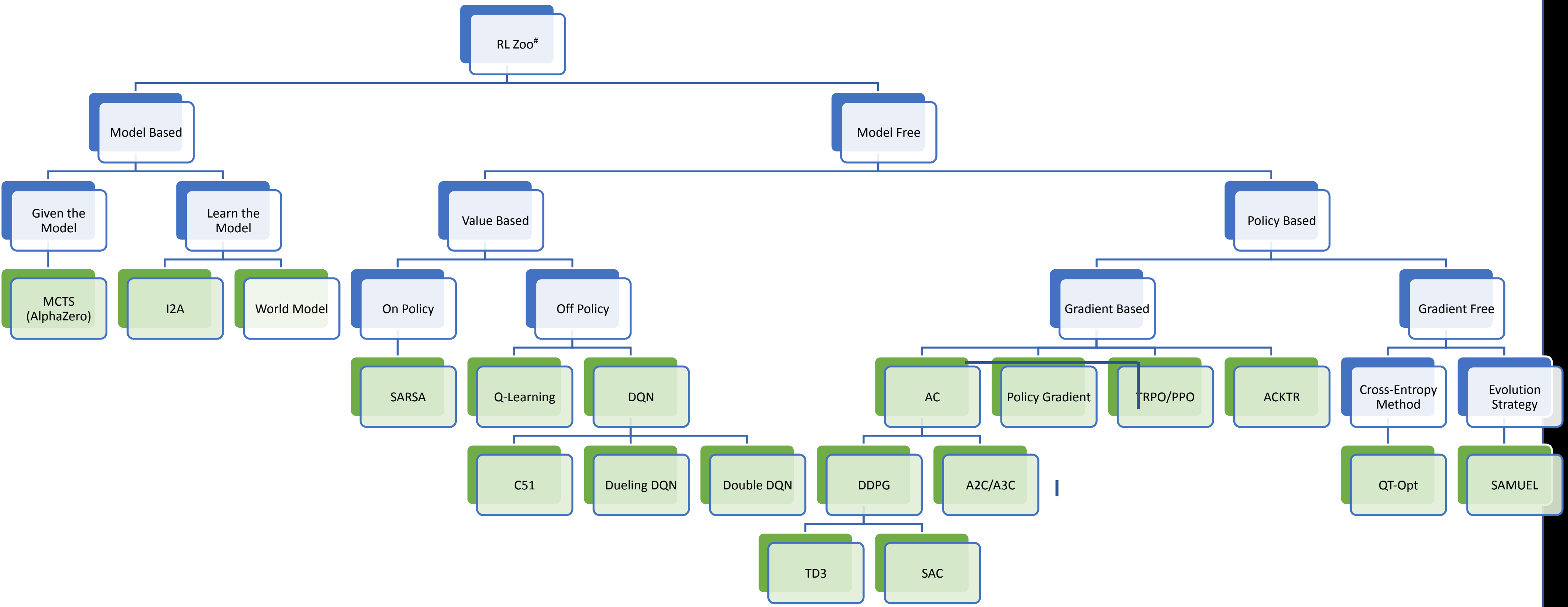
- Build AI systems with goals alignment – RL



- Cognitive Architectures – Clarion a model proposed by Sun

- Arel I. (2012) Deep Reinforcement Learning as Foundation for Artificial General Intelligence. In: Wang P., Goertzel B. (eds) Theoretical Foundations of Artificial General Intelligence. Atlantis Thinking Machines, vol 4. Atlantis Press, Paris.
- Sun, R. (2006). The CLARION cognitive architecture: Extending cognitive modeling to social simulation. *Cognition and multi-agent interaction*, 79-99.

RL Zoo



#Zhang H., Yu T. (2020) Taxonomy of Reinforcement Learning Algorithms. In: Dong H., Ding Z., Zhang S. (eds) Deep Reinforcement Learning. Spring[redacted]


Thank
You
köszönöm
Kiitos
Asante
Shukria
Vinaka
Maake
Dhanyavadagalu
감사합니다
Dankscheen
спасибо
Kam Sah Hammida
اركش
Manana
Dankon
Terima Kasih
Matondo
Dank Je
Mauruuru
Biyan
Diolch i Chi
Taiku
Tack
Blagodaram
Ngiyabonga
Dziekuje
Chokrane
Grazie
Juspaxar
Arigato
Mochchakkeram
Gracias
Khap
Tingki
நன்றி
Bedankt
Dakujem
धन्यवाद
Grazas
cảm ơn bạn
Paldies
Gratias Tibi
Ua Tsaug Rau Koj
Mèsi
Kia Ora
Khun
Obrigado
Dĕkuji
Nuwun
Nirringrazzjak
ありがとう
Suksama
Hvala
Kop
Rahmat
Matur
谢谢
хвала
Welalin
Di Ou
Djiere Dieuf
Misaotra
Danke
Merci
Go Raibh Maith Agat
Tuke
Eskerrik Asko
Salamat
ขอบคุณ
Najis