WEBVTT

1

00:00:05.290 --> 00:00:11.829

Jisun An: All right. Thanks for joining. So today's passcode is feedback.

2

00:00:14.330 --> 00:00:16.560

Jisun An: Please mark your attendance.

3

00:00:20.640 --> 00:00:29.950

Jisun An: So one announcement. So I asked you to the team up for the practical assignment.

4

00:00:30.924 --> 00:00:33.369

Jisun An: And I I

5

00:00:33.620 --> 00:00:40.770

Jisun An: may. You may have forgot. If you have forgotten to do this, then let me know immediately.

6

00:00:40.770 --> 00:01:04.250

Jisun An: So I randomly just assigned for those who haven't put their team name. So please check your team member and if you forgot to add your team, please let me know immediately. Then I will update this spreadsheet. So maybe you can check again tomorrow if there's any updates. But basically you can find this team roster under the assignment.

7

00:01:04.942 --> 00:01:12.920

Jisun An: So if you go to canvas under the practical assignment, there's a link to this roster. So if you haven't done so, please.

8

00:01:13.440 --> 00:01:17.370

Jisun An: yeah, basically, I match the one from top to the bottom, sir.

9

00:01:19.475 --> 00:01:21.549

Jisun An: So that's the one thing.

10

00:01:28.070 --> 00:01:32.739

Jisun An: Okay? So once again, today's pass code is feedback.

11

00:01:40.910 --> 00:02:01.649

Jisun An: Right? So I my, I updated the practical assignment a little bit. So please read it thoroughly, but the basic is so you will team up with another person, but just to just to create the data so that you can use it for your own reports.

12

00:02:01.800 --> 00:02:10.490

Jisun An: But your optimal goal is to build and propose a method for getting the best performance on your teammates data.

13

00:02:10.580 --> 00:02:37.179

Jisun An: and but you will also need to use your own to to demonstrate the general liability of the method that you are proposing. So so individual, you will need to submit individual report individual reports independently. It's it's you can discuss, like various details of this assignment with your teammates, but your report should be coming from your own. So once again, it's a team based. But it's the individual assignment. So make sure that

14

00:02:37.180 --> 00:02:49.680

Jisun An: that you are doing. And hopefully you can cooperate well with your teammates, so that, firstly, getting the good data set will be actually good for getting the good returns. And also you can discuss about

15

00:02:49.700 --> 00:02:59.760

Jisun An: various details on the assignment themselves, so hopefully that can help you to like brainstorm the ideas, or how you will approach to it.

16

00:03:00.892 --> 00:03:10.067

Jisun An: Yeah, let me know if you have any questions about the practical assignments. For those who just arrived arrived. Today's pass code is the feedback

17

00:03:10.740 --> 00:03:16.050

Jisun An: And so today we will talk about the reinforcement learning from human feedback.

18

00:03:16.641 --> 00:03:20.409

Jisun An: And I realized that this has a lot

19

00:03:20.730 --> 00:03:29.044

Jisun An: of content. So this may take at least 2 h. So yeah, this week, we will talk about this issue.

20

00:03:30.310 --> 00:03:54.910

Jisun An: So today I will start from something very high, level and overview of what is the reinforcement learning and how this can be used in the language modeling, and then we will go into a bit more details, and how this rath is actually done. But since the Ra itself is already a big issue, and just to discuss about it, you will probably need another course.

21

00:03:54.910 --> 00:04:07.212

Jisun An: So if you're interested in learning rl, I, unfortunately, it's out of the scope of this class, so you will need to find some other way. But hopefully, this will give you a good guide or understanding. What is this

22

00:04:07.580 --> 00:04:11.430

Jisun An: reinforcement? Learning is is used in the language modeling.

23

00:04:12.460 --> 00:04:17.390

Jisun An: So so where we are at is so we we talked about

24

00:04:17.640 --> 00:04:39.279

Jisun An: it requires multiple steps to have a high quality. Lrm, the lrm, that you are currently using. Is this something that went through different steps here? So, firstly, we did a like language, modeling from the like large, large Internet data. And then the next step was so. But these models are

25

00:04:39.350 --> 00:04:55.060

Jisun An: has some kind of knowledge, but it needs some guidance to generate the output as as is easy to like interact with the users. And that was done by through the 1st fine-tuning, supervised fine-tuning. So for that.

26

00:04:55.060 --> 00:05:18.109

Jisun An: We created different companies are creating like a big data instruction data where it is a set of the input prompts and the outputs. And these outputs are generated by the users. So they are experts are writing basically possible solutions or the outputs for different prompts. And using this instruction data, the Rrms are fine-tuned.

27

00:05:18.920 --> 00:05:42.345

Jisun An: and this instruction tuned model is now is at the level where you can actually interact well with this one. But this instruction, instruction tuned model has a certain limitations, or have some way to improve further. And that's this. That's the this finer steps of like fine tuning to, or, in other words, code as a preference tuning.

28

00:05:42.820 --> 00:06:09.880

Jisun An: Well, it started as a preference tuning, but they also realized that this last step also helps to improve the general ability of the atom. So I will talk about that. And this last step is usually done via the reinforcement learning. And recently they found that this step was very important for them to have this emergent ability of reasoning as well. So once again the reinforcement learning started from

29

00:06:10.160 --> 00:06:21.930

Jisun An: learning more from the human preference, but now they found that the reinforcement to learning itself is good for improving the performance of the Rnm. Especially in their reasoning skill.

30

00:06:22.830 --> 00:06:23.640

Jisun An: So

31

00:06:25.220 --> 00:06:39.430

Jisun An: maybe some of you are familiar with the reinforcement learning, but I think there are also majority of the students here is not very familiar with the Rl. So I will start with the Rl. But a very, very brief introduction once again. Rl. Itself is a big

32

00:06:39.450 --> 00:06:55.759

Jisun An: topic that deserved to a separate course, but I found that the in the later of this lecture we will. I mean I will use some different terms, and without the knowledge of the Ra. It'll be difficult to understand this. So we'll start from here.

33

00:06:56.540 --> 00:06:58.659

Jisun An: So what is reinforcement learning?

34

00:07:00.780 --> 00:07:12.659

Jisun An: so it's the one types of the machine learning where we have an environment X and available ability to make on actions and get some delayed reward. R.

35

00:07:12.660 --> 00:07:27.230

Jisun An: So we have an agent that learns by interacting with an environment, and the agent receives yours for a good actions and penalties for the bad actions, and the goal is to maximize the accumulative rewards over time.

36

00:07:27.230 --> 00:07:50.569

Jisun An: So this would be the typical definition of the Rl. If you are searching for what is rl, but I think it's much easier to understand from the example. So I will show you some example. And these examples are coming from this Youtube video that we have a link here. If you are interested in this domain, these are very, very basic introduction to the Rl. And the Ppo. So I recommend you watching

37

00:07:51.025 --> 00:07:51.480

Jisun An: later.

38

00:07:52.080 --> 00:08:05.389

Jisun An: So, so, assuming that this grid word is now our environment. And our agent is this thought, where this thought is just like moving around within this grid to to reach to a different cells.

39

00:08:05.390 --> 00:08:25.170

Jisun An: And if this agent moves and reach to one of these 3 block sales, then they will get some rewards, some points, so in this case the upper offer right one. If the agent reaches to that box, then it will earn like 5 points and maybe 4 points, or it also have some

40

00:08:25.710 --> 00:08:49.009

Jisun An: sales where it actually get the minus 1 point, meaning that it's the penalty. So in this case. And also, I mean, this environment may have some blocks. So there are some paths that you can go or you cannot go. But but these are the environment where the agents interact. Within this environment, it can have a different actions, where it can go ups, or to the downs, or to the left or

41

00:08:49.010 --> 00:09:07.029

Jisun An: rights, and once it reaches to a certain cell, it will earn these points, and the goal is, this agent is kind of playing a game where they will earn more points, and lose less points, and etc. And the goal of the Rl. Is to learn the the courses of the actions.

42

00:09:07.030 --> 00:09:10.410

Jisun An: to achieve the best scores.

43

00:09:11.160 --> 00:09:33.409

Jisun An: So maybe agents can go to that path or could go to that path, or go to that path by moving to to earn these each of the points. So if we think about at the each of the cells. So if I'm in this current cell, I have these neighbor cells, and each of these cells we call them as a States.

44

00:09:33.840 --> 00:09:49.019

Jisun An: And from that cell we have 4 different cells that we can move to. And, in other words, the action that I can take at that cell is the basically 4. So either going up or left, right or the to the down.

45

00:09:49.910 --> 00:10:02.911

Jisun An: Then assuming that we know that if we go to the right, I mean, then we earn like 4 points, and if we go up it's 1 and left is the minus 3, and if it go to the bottom is the 2, then

46

00:10:03.520 --> 00:10:14.729

Jisun An: then. So these are like the values of each of the cell. So what it mean is that we know that if I'm going to the right, then I earn like the 4 points.

47

00:10:15.064 --> 00:10:38.839

Jisun An: So so these are the the values is in a way that is like the estimated value. So if I move to that particular, send what would be my finer, and the rewards and the value is in a way that these rewards are redistributed across these different cells, so that we know by moving which directions how much reward I will get eventually. Is this something the value represents here?

48

00:10:40.100 --> 00:10:52.470

Jisun An: and then so so given that because my goal is to achieve the higher scores, so my value is also 4, because I know that if I move to the 4, then I will eventually get the 4. So my value is also 4

49

00:10:52.927 --> 00:11:13.469

Jisun An: then, so given these states and the values and the course of the actions. I given all this. I know that my action should be moving to the right right to to get that actual get the that the points, and so deciding like which direction to move is the policy.

50

00:11:14.077 --> 00:11:19.080

Jisun An: So these are some of the terminology that are required to understand these urls

51

00:11:21.007 --> 00:11:38.700

Jisun An: then, now let's so then, but but in in this case each of the the rewards are in some places, and we know that if the current position is here, and if I go to

52

00:11:38.960 --> 00:12:01.396

Jisun An: the move to get to the the cell where we are in the plus 5 so you can kind of go to that. Follow that path. But but but these are the game, so maybe there could be some penalty. So if it costs like one to move one spot, then these path, even though

53

00:12:02.000 --> 00:12:22.749

Jisun An: going through this path and reaching to the cell, would give you the 5 as a point. But if there are cost to move, then starting from this cell, going to that direction to reach it to the 5 may not be the ideal move, because you know that because if you're close to one

54

00:12:23.410 --> 00:12:35.799

Jisun An: basically, it will cost a lot to reach to the plus 5, so it may not be really worthy to move that direction to reach it to the 5. So so, in other words, if we have this path.

55

00:12:35.870 --> 00:12:56.540

Jisun An: then we can reevaluate the values of each of these cell, starting from these 5. So because we know that it goes to want to move 1 1 spot. We know that this cell would be the value of B 4, and this would be the value of B 3 and 2 and one etc. So we can kind of backtrack to assign these values right?

56

00:12:56.540 --> 00:13:11.240

Jisun An: So instead of so if you are, the agency is here, then maybe the idea would be not going to that direction. But maybe instead, we go to to the up and go to the reach to the cell with the number of point 4.

57

00:13:11.750 --> 00:13:36.560

Jisun An: So so the the value of this cell is actually not minus 4. But it would be something similar to 2, because we know that if we move to the up, then we will get the 4 points. So the value of this should be like the 2. So this is the way that how you can learn the value which is the estimated long term rewards of the current cell from from this kind of environments.

58

00:13:37.831 --> 00:13:52.860

Jisun An: And also, if you are simply looking at the cell itself, then from the previous example, then here the value of the current cell would be the 3 given, that if we assume that this gets close to one to move one spot here.

59

00:13:54.360 --> 00:14:22.040

Jisun An: so how? How then, then, eventually? And if you kind of propagate all these rewards to to all the cells. Then you will kind of get these values. So. And the way that how you evaluate this values is, you visit each of the cells, and and try to see and measure the expected rewards from from the current state to to reach to each of the

60

00:14:22.490 --> 00:14:27.919

Jisun An: rewards. So these are like the optimal values that you can get from these environments.

61

00:14:28.594 --> 00:14:33.860

Jisun An: Any any question up to here makes sense right

62

00:14:34.190 --> 00:14:54.270

Jisun An: now, once you have these values then? Now now you can find out what is the best place to move, because for each at each of them. So now we we are assuming that we know all the values for each, all the States. So if you are here.

63

00:14:54.620 --> 00:15:21.840

Jisun An: then you know that like so either. So if if the same, if the 2 States or 2 actions have the same value, then you can just randomly choose. But, for example, here you know that the going to the right is the best, and going to the up is the best to the right and to the right, so you will now know that a sequence of these move actions to reach to the the last cell, and then you will earn a reverse of the 5.

64

00:15:24.440 --> 00:15:52.933

Jisun An: And so this kind of courses of the action is can be also propagated. So based on these values. So I mean, it looked or similar. But basically, if you look at the 1st cell, the only applicable action is going to be down because there's no other way to go, and the the right cell is also having the lower value. So you are supposed to kind of go to the down. And if you go to here, then

65

00:15:53.420 --> 00:16:17.290

Jisun An: so basically, when you reach to this cell. Then there's no way that you will go to the right, because you know that if you go to the right, then you're you will end up. You're likely to end up in a cell with the minus one, which gives you a penalty, so you will not going to go there. But these are the now, the policies or the actions that you know from this environment, like what is the better policy, what is the like worst policy?

66

00:16:20.650 --> 00:16:37.419

Jisun An: So these are the 2, the optimal values and the optimal policy from this environment. So given an agent given an action and given a new word, you the Rl. School is to learn these optimal values or the optimal policies or run them together.

67

00:16:37.930 --> 00:16:50.000

Jisun An: So and this is the very basics of the rl, any question is clear, right?

68

00:16:51.260 --> 00:16:59.780

Jisun An: So why? Rl, is important? Because this helps in optimizing long term decision making. So the here.

69

00:16:59.920 --> 00:17:24.109

Jisun An: So as an agent. You are at one state, and you only have 4 different actions, and you only know the what we're gonna happen immediately in the next step. But your eventual goal is to get some higher score over time. So these rewards are delayed, and you only notice the rewards in a very later at at the each of the actions. So

70

00:17:24.640 --> 00:17:49.589

Jisun An: by knowing all these values and the policies, you will know that a lot of the best path to go to reach to the final goal, which is the earning more point within this game. So eventually, this Rl helps to find out the best solution for the long term kind of goals. So it optimizes for the long-term decision makings. So it has been used in various domain.

71

00:17:49.590 --> 00:17:56.789

Jisun An: especially in the games like Arpago or Dora. 2D. You probably heard, like a few years ago, that there was a

72

00:17:57.720 --> 00:18:12.859

Jisun An: very something going on the Alphago, which I will talk in very, very soon, and also in the robotics. You can also use this Rls. And also autonomous drivings and the Chatbot optimizations. All this has been used with the rl.

73

00:18:14.010 --> 00:18:20.020

Jisun An: so and so the outside row. I think it was 2,000

74

00:18:20.180 --> 00:18:25.229

Jisun An: 16. I think something around that when

75

00:18:25.720 --> 00:18:30.830

Jisun An: so everyone knows the goal right? Any the game goal. Anyone doesn't know the goal.

76

00:18:31.917 --> 00:18:45.852

Jisun An: So so goal is a game where I I'm not sure whether I can explain this in English. But you you have these pebbles and

77

00:18:50.200 --> 00:19:10.099

Jisun An: and your your goal is to surround all the pebbles of the the opponents right? But but anyhow, I I really don't think I can explain this. Well, but so for those who knows the goal in in this, what would be the actions, and what would be the state here?

78

00:19:13.580 --> 00:19:24.008

Jisun An: So once again, so in the previous grid word, the action and the state was very clear. Right. The state was the each of the

79

00:19:24.650 --> 00:19:41.049

Jisun An: the cells, and then the action was for the agents to move like to the go up or left and right to the right, to the bottom. But in the art, in the goal games, if you are familiar with what would be the action. And what would be the state here?

80

00:19:41.550 --> 00:19:43.289

Jisun An: Any any idea?

81

00:19:45.870 --> 00:19:47.629

Jisun An: It might be one of the better.

82

00:19:47.970 --> 00:19:48.670

Jisun An: So

83

00:19:49.300 --> 00:20:04.429

Jisun An: moving the pebbles. Do do you know the goal. Or you are okay. I see. So because goal is not moving something. But you have this grid of 16 by 16, and you put one pebble in one of those place.

84

00:20:04.850 --> 00:20:12.369

Jisun An: and then so it it could be similar to the Oslo but far more complicated game.

85

00:20:16.070 --> 00:20:22.149

Jisun An: But that's the goal, right? So what would be the action here?

86

00:20:24.730 --> 00:20:40.169

Jisun An: Just yes, just put the pebble in the grid. Is the action. So in terms of the action, and these are 16 by 16 grid. So you will have 16. Multiply by 16 number of actions possible actions that you can. You can do

87

00:20:40.560 --> 00:20:42.599

Jisun An: then what would be the State

88

00:20:47.070 --> 00:20:49.149

Jisun An: which position they are.

89

00:20:49.360 --> 00:20:54.429

Jisun An: So the entire grid themselves would be the States.

90

00:20:55.400 --> 00:21:21.389

Jisun An: So so the reason that I'm emphasizing here is because in the greater world is, it's easier to understand the action and the state, because it is visualized and is working as it is. But then, in other cases it may not be like 2 dimensional visualization. But still anything can be turning into the actions and the state. So I and once again I'm sorry that I was not able to explain the goal game

91

00:21:21.390 --> 00:21:29.110

Jisun An: very well. But for any other examples, I think it'd be nice if you can think about what would be the actions and the

92

00:21:29.350 --> 00:21:44.680

Jisun An: state in there, but because once once these are the maps into the action and state and the reward, then then everything will be the methodology themselves will be applied in a similar way. And then what would be the reward here?

93

00:21:48.370 --> 00:21:53.198

Jisun An: Yeah, get the points. But then, in the in the goal,

94

00:21:53.970 --> 00:22:16.520

Jisun An: in the goal. It would be very hard, because so so once again, in the in the greedy example, it was clear that someone reaches to a particular cell, you will earn that point, but then go. It may not work. So you, so, having one action, will not give you the or reaching to somewhere will not give you the point

95

00:22:19.940 --> 00:22:36.715

Jisun An: right right at the end of the game, right? So in us, that's that's actually the right answer, but in a simpler term the goal, the reward would be just whether you will win this game or not simply, and which will be done by

96

00:22:37.080 --> 00:23:00.299

Jisun An: How much pebbles you took out from your opponents, and how basically more pebbles left on the table. So that would be your reward, which will, and if you think about this game, go, then the reward will far, far behind. Right? So it will be like at the end of the game, and you are optimizing and finding the best course of the sequences

97

00:23:00.330 --> 00:23:14.349

Jisun An: of the actions. And the once again, action is the like where to put these pebbles, and the goal is basically won this game. And you are redistributing that rewards back to the course of these actions, to the state space.

98

00:23:14.440 --> 00:23:19.290

Jisun An: And the Rl. Is the running model to

99

00:23:19.440 --> 00:23:26.930

Jisun An: learn the best, learn the best possible sequence of the actions to achieve that goal.

100

00:23:30.300 --> 00:23:42.109

Jisun An: so I don't know whether you know it. But so they they had the the Google firstly came out with this rm, and once again, the goal is like very, very complex game, with

101

00:23:42.470 --> 00:24:07.200

Jisun An: in almost not infinite, but like a very big number of these States, and of course a long sequence, and it previously was impossible to beat the human so human was far better at this particular game. But then Google used the Rl model to train for the goal game, and then in 2,006, they compete.

102

00:24:07.200 --> 00:24:18.293

Jisun An: The archive will compete with the human, which is the Korean who is the best player for the game goal at that time, and probably even now

103

00:24:18.790 --> 00:24:34.250

Jisun An: one of the best top players, and if I remember correctly, I think they run like 3 games. And then he said, all won one and all, 5 games, and then he said, won once right. And then the Arpago won like 4 times.

104

00:24:34.910 --> 00:24:39.772

Jisun An: So and this I think this was the very critical moment of the

105

00:24:40.980 --> 00:24:47.633

Jisun An: of the rl, that people found that this is something that could be

106

00:24:48.580 --> 00:24:55.380

Jisun An: enable enable model to be better than the humans.

107

00:24:56.480 --> 00:25:22.392

Jisun An: But anyhow, so coming back to this, a simple example, so I just came there to just give you some idea of how this can be applied in other application. But but now that so these are the like optimal values and the optimal policies, then how can we learn these optimal values and policy once again, I will not go like the deep of how we can actually learn this. Just as a very, very high level ideas of it. So for the value

108

00:25:22.960 --> 00:25:52.190

Jisun An: So if we want to know each of the values in each of these cells here, the easiest way is actually fidget all different cells here, each of the States, and just get the rewards, get the estimated values. But then this is only possible. When you have, like small number of states, what if you have a large like states like play? Go, then, it'll be impossible for you to visit all, each of each of the all States.

109

00:25:52.190 --> 00:26:07.100

Jisun An: So that would be even almost impossible and same for the policy. So now, recently, they found that if you are using, like the learning within neural networks, then you can build a model that can estimate the value of these States.

110

00:26:07.770 --> 00:26:23.569

Jisun An: So the simple idea is that we already know that the value is kind of propagated to your neighbor. Of cells. So that's the simple idea. So you are using this very simple neural network where the input is the state and the output is the values.

111

00:26:23.860 --> 00:26:25.080

Jisun An: So here

112

00:26:25.230 --> 00:26:53.449

Jisun An: the state is the the locations of the the agent themselves. So that's the reason that this network has this X and Y as an input. But you can simply think it. That input is, here is a state, and the output is the value. So, and you are training this, the learning model to learn, I mean, the model will be able to estimate the value for each of the cells or each of these States.

113

00:26:54.230 --> 00:26:59.802

Jisun An: And the way that it can learn is, once again, just, I'm I'm doing very brief

114

00:27:00.440 --> 00:27:23.226

Jisun An: explanation here. So so if you are here at the cell, 2 by 3, and then your value is, for example, based on the current neural model. This value was 0 point 2. And if you you knew that the surroundings values are each of these 4 and then given this neighbor cells values, you are updating your

115

00:27:23.860 --> 00:27:49.450

Jisun An: your your own value. So if these were the cases, and if it still follows the the idea that we talk so it goes to like the one for moving down. Then this. And and you are taking the maximum value of your neighbor cell, because you know that if you go down, then you will reach. You will have like higher points, right? So your policy or the the next course of the action should be going down, and if

116

00:27:49.450 --> 00:28:14.179

Jisun An: if you are going down here, then your value will be 3.9, because your cost one to move down, so your value should be 3.9. But if your neural network say that this is a 0 point 2, then these are the error. Now you can measure right? So you can update this value neural network by consider this as an error, and then you can propagate this error back to your neural network. And then your neural network.

117

00:28:14.180 --> 00:28:24.470

Jisun An: we'll learn what would be the each of the value for each of these States. I mean, that's just the basic, very, very, very easy idea of how you can train the value on neural networks

118

00:28:25.510 --> 00:28:45.620

Jisun An: and similar, I mean for the policy neural networks. It's a similar neural network, but it's a slightly different now. Given a state you're you're predicting. What would be the likely actions that you should take. So the output is now you have the actions. So the input is the state and the output should be the actions and

119

00:28:46.248 --> 00:29:00.430

Jisun An: how will once again you? You don't need to understand everything. I will not going to ask about this part, because I just want to give you some ideas. So how you will learn the policy neural network would be so you will take one path.

120

00:29:00.430 --> 00:29:23.879

Jisun An: So this is the one example path. So you are starting your. We assume that the agents were at 1, 3. And from there, basically, we took one path until it reaches to a cell where it actually gives you the reward. So this is the one example path. So once we have them, then we are computing what was the each of the

121

00:29:23.960 --> 00:29:29.356

Jisun An: what was the thick? What was the all the actions that we had to make to reach to that

122

00:29:29.850 --> 00:29:54.649

Jisun An: reward the cell. So so here we know that if we are at the 5 and 4, so I was saying the the other way around. So if the x, so the X is seeing the X and Y values are 4 and 5, then here we know that our gain is 4, and then the course of the action is the to the right. And here, basically, the gain will be the 3, and the course of action should be the up

123

00:29:54.770 --> 00:29:58.583

Jisun An: and etc, etc, and we we just know that

124

00:30:00.422 --> 00:30:12.143

Jisun An: so going to this direction should be encouraged, and going to this direction should be discouraged. Because

125

00:30:26.150 --> 00:30:30.163

Jisun An: I think I'm missing something.

126

00:30:38.540 --> 00:31:04.959

Jisun An: so I yeah, the thing I mean so is, I kind of missed the intermediate step, so it may not be. Very make sense what each of these probability or the log loss are, makes sense, but just. The the idea is that you are taking each of the path, and from each of the path you are. Consider them as a sequence of the actions and using this as a new data to train these policy neural networks.

127

00:31:05.701 --> 00:31:13.528

Jisun An: so so that's the something that got to happen that to to train this policy. Neural networks?

128

00:31:14.270 --> 00:31:24.490

Jisun An: I think I'm missing something, but I just cannot figure out at the moment. So I will revisit, and probably just talk a little bit more on on Wednesday.

129

00:31:26.310 --> 00:31:39.433

Jisun An: so once again. So the the value and the policy neural networks can be trained in this using team learnings and

130

00:31:40.090 --> 00:32:07.113

Jisun An: so the model we now should be able to tell. Given a State what would be the expected value and given a state, what would be the most likely actions? So that would be the outcome of these neural networks. So even though these values are not the exact, the optimal value or the optimal set of the actions, but these are the estimated approximate values and the policy. And by learning this you should be able to

131

00:32:07.520 --> 00:32:13.129

Jisun An: take the actions to get the better rewards or the better goals.

132

00:32:18.430 --> 00:32:42.400

Jisun An: so once again, sorry for the little confusion, but assuming that you can learn the value and the policy neural networks. So how this Rl applies for the language or the language generation. So if we assume that we use this Rl. To generate the text, what would be the actions, and what would be the States?

133

00:32:52.260 --> 00:32:55.809

Jisun An: So in in language generation? What what would be the action.

134

00:32:59.470 --> 00:33:05.720

Jisun An: or how would you apply the rl, to the language modeling or text generation?

135

00:33:06.310 --> 00:33:09.200

Jisun An: Yes, yeah.

136

00:33:17.140 --> 00:33:17.980

Jisun An: State

137

00:33:21.530 --> 00:33:22.300

Jisun An: good

138

00:33:24.340 --> 00:33:49.909

Jisun An: created. Yes, exactly so. If now we are thinking from the language generation using the Rl, then from the start, you have no text, and your action will be. You have a list of tokens, and you will select one to which one to select. So what would be the next token so selecting the token would be your action.

139

00:33:50.090 --> 00:33:56.719

Jisun An: and the State will be the all the tokens that you have generated. So that would be the state.

140

00:33:57.650 --> 00:33:59.883

Jisun An: And if you think about

141

00:34:01.970 --> 00:34:11.039

Jisun An: available tokens and available language. Then here the number of the actions and number of states will be really, really big.

142

00:34:11.040 --> 00:34:36.550

Jisun An: and also we are still looking at the grid. But this language are not actually at the grid, but for the visualization purposes we are still using the grid. But so this would be how the the language would be look like. So you start from somewhere. So you are starting. And the agents basically pick up or sample one of the tokens from the token, probability, distribution.

143

00:34:36.730 --> 00:35:05.679

Jisun An: So you are picking what? So this, what would be your one state? And then your next token would be color. So now, what color is your another state? And what color is? What color is that? What color is the sky, and what color is the sky question mark? So each of these, even though so here we are. I mean, these are just example. But so, selecting each of the next token would be your actions, and this sequence of the actions will now generate a sentence.

144

00:35:06.630 --> 00:35:35.229

Jisun An: And now, assuming we are at here, what color is the sky? And now we have, you can think of what would be the next tokens? And you can have like different options. It could be. What color is the sky red? What color is the sky blue? What color is the sky banana? These are all possible states that you can go. In other words, you have, like all possible tokens that can be generated. But these 3 could be most likely to be tokens likely tokens.

145

00:35:35.680 --> 00:35:55.839

Jisun An: and then so in. Oh, so these are kind of part of the like the human feedback that we are. We are talking more in details later. But so now, when we have these many different direction that we can take in terms of text generation, and where?

146

00:35:55.840 --> 00:36:14.011

Jisun An: What path, then, would be good path? And to know that the Rl. Requires a reward right? So in this example the reward. Now we assume that it's coming from the human so human can tell like which of these 3 sentences are good. So these rewards are,

147

00:36:14.680 --> 00:36:35.638

Jisun An: coming from this human feedback. So I mean, the blue is the something that obvious. So maybe saying, blue is the highest reward. And the red is. Sometimes when the there's a sunset, we still see the red sky, so maybe it has some reward. So we are giving 2 and banana. These are not very good.

148

00:36:36.700 --> 00:36:45.109

Jisun An: words that are followed by this sentence, but still it's possible. So it's a reward one. But so basically, these are different sentences, and

149

00:36:46.305 --> 00:36:48.779

Jisun An: we may notice rewards from

150

00:36:48.970 --> 00:37:03.030

Jisun An: the human. So now we know that moving to this direction or that direction have different rewards. So you should be able to decide like, which direction would be the best set sequence of the

151

00:37:03.750 --> 00:37:25.609

Jisun An: best action to take so in terms of the language. The Rl. Is essentially is to learn what is the best sequence of the tokens to generate. Given the reward is something. Coming from somewhere like like human feedback, or some other place which I will talk more.

152

00:37:25.950 --> 00:37:31.609

Jisun An: But that's the how the Rl. Can be mapped with the like. The language modelings or language generations.

153

00:37:31.670 --> 00:37:53.869

Jisun An: And so there could be. Even for the longer path. It'll be possible for any different kind of different sentences you can. You can kind of now tell which one actually has the more about like higher rewards or lower rewards, and the policy also will be now trained and adjusted, based on those rewards.

154

00:37:53.870 --> 00:38:18.810

Jisun An: So once again, rl, once you know these rewards, then these rewards, even though that will happen like much later, so they are delayed reward, but they learn to redistribute all those and rewards back to the other States, and so that, given the current state, you will know which direction to move. So, in language modeling, you will know that, given that, we

155

00:38:18.810 --> 00:38:26.410

Jisun An: that the most preferred sentences in choosing the next tokens, you will learn.

156

00:38:27.122 --> 00:38:34.730

Jisun An: I mean, you will basically incorporate the the definer reward or the preference into the Rl models.

157

00:38:37.120 --> 00:38:57.090

Jisun An: So yeah, I know that this is very still big, because once again, Rl itself is is a big concept, and I think, understanding them fully may not be enough. We may not be able to do it in 1 h. But hopefully, you

158

00:38:57.090 --> 00:39:13.300

Jisun An: understand the concept of the rl, and what is the action? What is the State. What is the reward, and what is the purpose of the Rl. In general? And now I will talk more about the Rl. For the language building.

159

00:39:14.022 --> 00:39:16.909

Jisun An: But any any questions.

160

00:39:23.190 --> 00:39:24.250

Jisun An: All right.

161

00:39:27.890 --> 00:39:40.615

Jisun An: So once again I will start from something high level, and then and then moving on to more more in depth of of the llf, but

162

00:39:42.290 --> 00:39:46.670

Jisun An: so I don't know whether you know the Andre Cathy. He is.

163

00:39:47.490 --> 00:40:12.143

Jisun An: I mean very famous person and he's very, very good educator. I learned so much from him as well, and he recently I shared this whole image from on his twitter. I think he created a video about the Chatgpt. And he was looking at he he tried to.

164

00:40:12.810 --> 00:40:38.818

Jisun An: explain. So why we need like reinforcement learning, and I really liked his analog here. So I brought that to here, and and this 1st part of the the contents is actually coming from his his Youtube videos as well. Because I think this is really good way to understand what what Yrl is really needed in the language modeling. So so that's the something.

165

00:40:39.380 --> 00:40:43.862

Jisun An: just for your notes. But so so, he explained, that is

166

00:40:44.310 --> 00:40:53.279

Jisun An: so basically, the the language modeling. And to to have better language. Modeling is is very similar to how we learn the knowledge.

167

00:40:53.280 --> 00:41:18.239

Jisun An: So when we study a particular course, there are usually 3 steps where or usually this textbook is in 3 different parts. So the one part is the they have these expositions where it just describes all this knowledge that is required for a particular domains. So that's like basically the background knowledge. So you need to understand the different concepts and

168

00:41:18.240 --> 00:41:18.970

Jisun An: drop.

169

00:41:19.670 --> 00:41:30.748

Jisun An: And then, so this is similar to the pre-training of the rrm, so the pre-training they try to understand and absorb all the knowledge, the linguistic knowledge that available

170

00:41:31.440 --> 00:41:37.669

Jisun An: and then so the pre-trained model is basically capable immediately. They know they has a lot of knowledge about the language.

171

00:41:37.670 --> 00:42:02.419

Jisun An: But then they. And then in the textbook, we have these worked problems. So this worked problem is something. So now they, there's like all different description about the particular concept. But then, to help you to understand that better, you have a particular problem, and then the text may also have the solution for that. So by looking at those questions and the answers that you may understand better

172

00:42:02.420 --> 00:42:07.260

Jisun An: about the concept itself, and how you should understand this knowledge.

173

00:42:08.064 --> 00:42:36.429

Jisun An: But then these are the answers that are actually written by the authors of these books. So these are the something that the authors actually want to. be clear about. What is the best solutions. So in a way that these are like the instruction kind of data you have. They have the question, and they have the solution. So this work, the problems is the resemble to the supervised fine tuning where the language is now

174

00:42:36.430 --> 00:42:43.799

Jisun An: have all the knowledge. But then, now it's it's tuned to this question, and the answer set so that it can generate something.

175

00:42:44.480 --> 00:42:49.390

Jisun An: Imitate the the knowledge of the experts of these textbooks.

176

00:42:49.520 --> 00:43:14.999

Jisun An: and then then we have also the other. This parts. Where has the practice problems? So in this practice problems, they have only have the the problem itself, and usually doesn't have any solutions. Right? And solutions comes at the end of the textbook or some other places. And the the reason that they have this practice problem is for you to now think about different ways to solve and tackle that

177

00:43:15.000 --> 00:43:38.100

Jisun An: problem. And so, and then, by thinking about those different paths or different solutions, they hope you to learn better about the content themselves. Right so, and now and then, this practical problem is now something that resemble to the reinforcement learning. So what reinforcement learning is trying to do is given a particular question.

178

00:43:38.100 --> 00:43:51.380

Jisun An: They try, like all different possible solutions, and they try to find the best solutions out of those different possibilities. So in a way that how human learns a particular

179

00:43:51.550 --> 00:44:02.811

Jisun An: concept is and then how the the language models are learned. These concepts are very similar, and and, you know, resembles in in this kind of way.

180

00:44:03.980 --> 00:44:09.989

Jisun An: so the big differences. Here once again, the pre-training is just like

181

00:44:10.140 --> 00:44:36.419

Jisun An: trying to observe all background knowledges, and then for the supervised fine tuning which was the instruction tuned that we discussed. These are the set. So given a set of questions and answers as a human created. The model learns to mimic these answers. And now there's a more possibility. If you explore more or possible generations and outputs and try to find what would be the best

182

00:44:36.918 --> 00:44:50.450

Jisun An: course of the actions of best solutions. That is the what reinforcement learning try to do, and hopefully, that can also help the to be more intelligent in doing what what they do?

183

00:44:54.680 --> 00:45:01.888

Jisun An: So and so why, why, how? The what? So?

184

00:45:02.860 --> 00:45:20.360

Jisun An: going to what reinforcement learning will actually do. So, looking at this example. So this is a very simple math problems. Emily, buy 3 apples and 2 oranges. Each orange cost $2. The total cost of all the fruit is $13. What is the cost of each apple?

185

00:45:20.800 --> 00:45:31.932

Jisun An: So when we have this kind of question. And the the actual answer is, $3 and then these below. Here we we show like 4 different

186

00:45:32.770 --> 00:45:47.080

Jisun An: possible responses. So assuming that these are I mean, there could be different way to solve these problems. And these are 4 different responses that you can think of. Then out of these. So

187

00:45:47.220 --> 00:45:56.140

Jisun An: what would be the best response to include to the instruction data? So if we

188

00:45:56.506 --> 00:46:19.343

Jisun An: so once again, the instruction data is something that human generates. And we are kind of giving a particular way to solve this particular problem. Right? So once you add these examples to the instruction data, then your model will fine tune based on that. So you it will mimic one of these responses. So what do you think? Would this? The 1st one has

189

00:46:19.820 --> 00:46:36.777

Jisun An: as some kind of they set it up as all like equations, and then they kind of get the answers, the second one they describe it, and 3, rd one and 4th one. It is very short and concise, and then get to the kind of point, so which one would be actually better. For

190

00:46:37.460 --> 00:46:43.990

Jisun An: Better to include to the instruction data any idea?

191

00:46:45.020 --> 00:46:45.710

Jisun An: Hmm!

192

00:46:46.240 --> 00:46:49.109

Jisun An: The blue one. And why do you think that would be

193

00:46:51.390 --> 00:47:13.490

Jisun An: more human? So that's actually a good way. So if these were presented to the human, and if we wanted to receive the human feedback, then maybe the blue one would be the something that people would be more preferred. And actually, the current model is fine tuned, based on, I think the blue kind of ones because it's more interpretable for the humans. But then, if you think about from the model

194

00:47:13.590 --> 00:47:37.990

Jisun An: themselves, then the answer is, we actually don't know so even though the blue one is, looks better for us. But does it really better for model? We don't know, because we are not the model. So so what I want to say is that in when you're doing the supervised fine tuning, we are giving the the outputs. We are giving the response right? What is the expected output that we want the model to generate.

195

00:47:38.140 --> 00:47:58.780

Jisun An: But the truth is that there could be actually better or different way to generate this text, and if we know that, then we can create an instruction data for model to fine tune, but because we are not the model, and there could be some different way, or preferred way by the model to get the correct answer.

196

00:47:59.320 --> 00:48:08.529

Jisun An: we just cannot go to that that direction to to create the best data set for model to perform the best

197

00:48:09.012 --> 00:48:22.650

Jisun An: I so so these are the for the same question. These are the answer from the chat. Gpt all 3 mini, which is the another reasoning model, and you already you can see that these are now the

198

00:48:22.820 --> 00:48:47.300

Jisun An: the results that we are getting from them. But and also these are tuned based on the rl, but but I wanted to mention 2 things so actually, there could be 2 different goals of the Rl. In this example, so there could be correctness, and there could be presentation. So, firstly, given this particular question, we want the want to obtain the correct answer. But at the same time we want to

199

00:48:47.300 --> 00:49:05.379

Jisun An: present effectively to the humans as well. So there could be different things that we should consider when generating this output. But for now we are not talking of the presentation. But we are talking only about correctness. So we, if we want to get the correct answer. Then, even though the blue one looks

200

00:49:05.380 --> 00:49:13.360

Jisun An: far more better presentation for the human. But this may not be actually the solution for model to to

201

00:49:13.990 --> 00:49:39.769

Jisun An: get the right answer. And just we don't know. And that's the one of the reason that why people are using Rl to fine tune the model. Further, because there could be something that human don't know. But then, if you explore, or different possible path, which is the sequences of the tokens, then there may be some sequence that is actually for model to get better at answering these questions, even though that may not be

202

00:49:40.130 --> 00:49:50.590

Jisun An: interpretable to the human, were preferable to the human. So so

203

00:49:52.140 --> 00:50:19.689

Jisun An: so when we have this prompt, then then the model language model can generate like different solutions or the outputs. So here each line is the the rollout from these models. And you can think of. We talked about how you can infer the text from these. Rl, so if you are choosing different sampling techniques, then whenever you are running the same, prompt for the from the same model, you will get like the different outputs.

204

00:50:19.690 --> 00:50:38.920

Jisun An: So, assuming that we run this 15, we generated 15 solution, meaning that you just click the new chat button from the chat gpt and prompted this, and then get the output, and you create another new chat and then edit this prompt, and then you get another sample, etcetera. So you you repeat that for 15 times

205

00:50:39.544 --> 00:51:06.310

Jisun An: so and so out of this 15 solution, maybe the 4 of them, which is the the 3 greens and the the one blue one got the right answer, because we, because in this case these are the math problem, and we know the exact answer right. So we know we can tell which one is right or wrong. So assume that we have. 4 of them got the right answer, and we somehow found that this blue one was actually better. So we take that as a top solution.

206

00:51:06.930 --> 00:51:19.420

Jisun An: And then now we can train the the model, and then we just repeat this many, many, many times. Then at some point the the model will learn what would be the best path to

207

00:51:20.060 --> 00:51:22.679

Jisun An: generate to be correct.

208

00:51:23.140 --> 00:51:28.979

Jisun An: and that's the kind of the how, what this rl, we're gonna do with the with the language modeling

209

00:51:31.770 --> 00:51:41.859

Jisun An: and so and and once again. So this will done like many, many times, with many, many different prompts. And

210

00:51:43.530 --> 00:52:02.698

Jisun An: and and so for each of the prompts, maybe you can sample 1,000 different outputs. And then you can also going through all this 1,000. And you see which one actually get the best. So this will Rl usually takes very, very long time to

211

00:52:03.210 --> 00:52:07.080

Jisun An: conduct the experiments, and and also the train as well.

212

00:52:08.232 --> 00:52:20.209

Jisun An: And the. So we we talked about the 3 steps of the Llms, and the 1st step, the pre-training, and the second step the supervised tuning. So these 2 have been very standardized.

213

00:52:20.210 --> 00:52:43.759

Jisun An: So these steps are very straightforward. So many different companies has been also replicated those 2 steps. But this 3rd one, the Rl. Part, has been a bit more black box than the others, because once again, Rl. Is taking really long time to train. And and also the 1st the I mean Gpt, basically Openai was the 1st one who tried this rl.

214

00:52:43.760 --> 00:52:56.230

Jisun An: and then usually they don't disclose much of the information. So everything was quite black box. And and that's the reason that this dipstick that we talked briefly last time was really important, because

215

00:52:56.810 --> 00:53:23.959

Jisun An: it also has many interesting ideas in this paper, and I hope to see talk a little bit more next class. But in terms of that, these are the open model that this model is purely use the Rl. Without the instruction tuned, and they found that just using reinforcement learning, the model was still able to do all the reasonings that the Openai models have been shown

216

00:53:23.960 --> 00:53:39.249

Jisun An: so in and and then so its performance was great, but at the same time it also disclose everything that they have done. What was their experimental steps? And What was your configurations, and how so? What the exact step they went through.

217

00:53:39.550 --> 00:53:44.859

Jisun An: That's the reason that this paper was very informative for, and

218

00:53:45.460 --> 00:53:56.250

Jisun An: assume that it will be very important in the future as well, because from now on, in the next couple of months you will see many other companies that, trying exactly similar to what they have done

219

00:53:56.890 --> 00:54:25.760

Jisun An: so. So one thing that I'd like to highlight a few is that here, this is the the performance chain. Like the learning kind of performance over steps. And the so these are based on this aime test. So these are the math problems where it has. I mean, this kind of like math problems. And they were evaluating the performance of these models based on these math problems. So

220

00:54:26.110 --> 00:54:44.021

Jisun An: something interesting is that they is basically over time as more steps, as more and more you sample the outputs and then run based on them. The model gets like gets the higher accuracy, meaning that they are becoming more and more smart, and their reasoning became better and better.

221

00:54:44.600 --> 00:55:13.285

Jisun An: And then it even more interesting thing was that as the as they train more, the train Rl models longer, then they found that the average length per response was also getting longer and longer. So, in other words, so they were getting more corrects on these math problems. And at the same time they, their responses, were also getting longer and longer. So they eventually they found that

222

00:55:13.880 --> 00:55:19.939

Jisun An: The reason that their response tokens are getting longer is because they found these patterns, that

223

00:55:19.940 --> 00:55:44.929

Jisun An: the they were getting the answers, and they are reflecting whether there's any other way to solve or tackle their particular problem. So they were kind of repeating that process of is there any other way to solve this is this correct? And so they were self reflecting or also exploring the other potential solutions. And then that was the one of the reasons that they were seeing this, like

224

00:55:44.930 --> 00:55:53.910

Jisun An: the length average length was getting longer and longer. So this is one example that we've found in the in the dipsic paper.

225

00:55:54.250 --> 00:56:08.280

Jisun An: So they had this response, and then wait. Wait. That's the Aha moments I can select here. Let's reevaluate, reevaluate this step by step to identify if the correct sum can be so so this kind of

226

00:56:08.960 --> 00:56:16.836

Jisun An: the actual, like thinking kind of a process was rebuilt from these. The area based model that the dipstick

227

00:56:17.560 --> 00:56:18.620

Jisun An: came out.

228

00:56:20.140 --> 00:56:43.338

Jisun An: And so the same question the Emily by 3 apples and 2 oranges. So I run it here, using the together that AI so together that AI, by the way, you see another platform where you can use for they have this playground. So you can use, like the different models selecting different models, and then just run them and see how the results are different.

229

00:56:43.840 --> 00:57:08.130

Jisun An: so here, even for this very, very simple questions. By the way, if we run this to the like chatgpt, then they can just answer it, even without any other words. But but like this, it was generating and showing all the this thinking process. So you can see that like, let me try to figure out this problem. Oh, wait! Let me make sure I didn't make any mistakes in my calculation.

230

00:57:08.140 --> 00:57:34.069

Jisun An: or alternatively, maybe using ratios. So you can kind of see that even I mean the Rl. Supposed to explore all different path, and then that reverse, now reflecting back to the outputs themselves. So you can see this kind of why, the way that the just trying to find, like different solutions and reflecting back of their own responses is has been appeared in the in the deep 6.

231

00:57:34.630 --> 00:57:43.169

Jisun An: So once again I I just excluded all the technical details, and I hope I can mention bit more next week. But just this is.

232

00:57:43.510 --> 00:57:52.190

Jisun An: give you some idea. See how what the Rl has been used, and how, and how and how that improves the the

233

00:57:52.420 --> 00:57:54.949

Jisun An: the reasoning aspects of the language models.

234

00:57:55.886 --> 00:58:05.150

Jisun An: I think this was pretty straightforward. But but any question, yes, good morning.

235

00:58:09.990 --> 00:58:12.750

Jisun An: What was the last part? How?

236

00:58:14.720 --> 00:58:15.873

Jisun An: Oh, so

237

00:58:16.580 --> 00:58:41.830

Jisun An: so these were, these did did skip the supervisor learning, but they actually do the post training. So, after all, they did a bit of preference training just to follow the. So they they found that. The instructions are they basically don't follow the instructions. So they had to do a little bit. But after the Rrl, so but then this math problem I think evaluations was done before that. So so

238

00:58:41.900 --> 00:59:00.259

Jisun An: so deep has many different branches, and they tested many things. So one was purely looking at the impacts of the Rl. But then, to be useful for the human, they actually needed the instruction, tuning and the human preference as well. So they did a small preference tuning later after the Ll.

239

00:59:00.640 --> 00:59:02.740

Jisun An: But thanks, thanks for pointing that out.

240

00:59:06.350 --> 00:59:07.710

Jisun An: Any other questions.

241

00:59:11.700 --> 00:59:16.105

Jisun An: Okay? So so in a way that

242

00:59:16.670 --> 00:59:32.160

Jisun An: these particular questions, and also relating to any other like question and answering sets of the data set where they have the verifiable answers, is very straightforward. So, thinking back from the like, the Rl. The Rl. Had

243

00:59:32.320 --> 00:59:45.289

Jisun An: requires the reward right? So out of all these solutions to to pick which one is the best they need to know the rewards. So in this case the getting the correct answer would be the reward.

244

00:59:45.658 --> 01:00:09.989

Jisun An: But then, what if, like some other problems like, write a super super funny jokes about pelicans? So like this kind of problems, there are many different ways to generate the output, so there could be many different jokes. But then, how do we know that which one is actually best or so, which solution should get the reward. So how do we score this?

245

01:00:10.668 --> 01:00:17.030

Jisun An: So in the those domains that is on easily unverifiable.

246

01:00:17.821 --> 01:00:20.828

Jisun An: That then now you need to

247

01:00:21.520 --> 01:00:25.212

Jisun An: to score like each of these solutions.

248

01:00:26.620 --> 01:00:34.849

Jisun An: the. This reinforcement, learning from the human feedback is the one of the solution that proposed for tackling this particular challenge.

249

01:00:35.441 --> 01:01:00.919

Jisun An: So this one is the one. This work was published in 2022 by the open AI. And many of these authors were moving to the anthropic, I think. And so this was the 1st attempt to use the Rl. To use for incorporating the human feedback. And now the

250

01:01:01.000 --> 01:01:16.429

Jisun An: developing from this idea and the methodologically, I think it was very similar. But they increased the basically the preference data set. And then this was the paper that was published for Chat Gpt. Gpt. 3.5. I think so.

251

01:01:17.035 --> 01:01:41.174

Jisun An: The 1st version of the Chat Gpt that you've seen is is basically after this paper. And they literally, these 2. Methodologically, they were very similar. But this one had, like a larger preference data set, which I will talk very soon. And here the their abstracts! They can really say that. Why, the reason that they brought these

252

01:01:42.124 --> 01:02:10.870

Jisun An: rl, hf, is that. So, even though the language models can do many things. But this can generate output that are that may not follow the user's intent, and they can generate output that are untruthful or toxic, or simply not helpful to the users, so they are not aligned with their users. So they are now proposing a new method, that fine tune their model with the human feedback.

253

01:02:12.270 --> 01:02:13.350

Jisun An: So

254

01:02:14.490 --> 01:02:40.052

Jisun An: here the idea is that so once again, given this kind of prompts, it says the doesn't have any correct answers. And if you are now using the typical rl, then you and if you would just want to use the normal. Ll, then, maybe for the 1,000 updates of the 1,000 prompts over the 1,000 rurals you will have.

255

01:02:40.550 --> 01:02:44.340

Jisun An: What's the English of these values? Is it one

256

01:02:44.510 --> 01:02:47.490

Jisun An: trillion, 1 billion and trillion? Is that right?

257

01:02:49.740 --> 01:02:56.865

Jisun An: 1 billion? 0, this is one, yeah, 1 billion scores from humans.

258

01:02:58.640 --> 01:03:27.649

Jisun An: so so if you want to examine all possible ways that one prompt can generate, I mean, 1,000 prompts can generate. Basically, you will need 1 billionth from the human. So it may not be really possible to do it. So instead of like approaching this this way the rlf, the basic idea of the Rlhf is for each of the prompts you just get like 5 different rollouts.

259

01:03:27.650 --> 01:03:32.920

Jisun An: and then you order them from best to the worst using the human.

260

01:03:33.190 --> 01:03:56.489

Jisun An: And then you are now train a new network simulating this human preference, or we call it as a reward model. and then and then and then you run the Rl. As usual, but the reward will come from this reward model rather than the actual human feedback. So that was the like, the basic idea of the Rl, hef.

261

01:04:03.870 --> 01:04:13.175

Jisun An: so so, so once again, the the key point here. So for us to learn.

262

01:04:14.550 --> 01:04:23.209

Jisun An: the policy which is determining the next actions and the in the language generation. The next section is the which token to come next.

263

01:04:25.350 --> 01:04:49.630

Jisun An: It will eventually know which sentence is better than the others, and these ones are can be done relatively easily for those verifiable cases where they have the answers, but for those prompts that doesn't have any like correct answer, then you need some kind of human feedback, so you can imagine that for each of generated outputs.

264

01:04:49.840 --> 01:05:11.779

Jisun An: Human will need to examine like what would be whether that sentence is good or bad. So you can actually have, like the direct assessment as well like giving these scores, but usually it is very hard for human to give the direct scores. So direct assessment is not the ideal way to evaluate these things, because if you have 10 different sentences, and if you

265

01:05:11.780 --> 01:05:30.801

Jisun An: give like the different scores, it would be much harder than simply comparing. So in the Rhf paper. They decided to go with the comparison. So giving different samples. And then they asked to rank the 5 different examples. And then using this human ordering, they

266

01:05:34.390 --> 01:05:50.059

Jisun An: They build a reward model which now is the model that will given a output of the given, the the prompts and the output, and then they will kind of assign a particular score. So that's the what this reward model does.

267

01:05:51.600 --> 01:06:18.748

Jisun An: And here, I was here simply saying that human can like provide their preference. And then that's the reason that we are called them as of human preference. But the human can prefer on different aspects of it. Right? So you can actually ask these different questions like whether it's fluent or whether it's adequate whether it's coherent, whether it's a truthful, helpful, or the homeless. So you can imagine these can be

268

01:06:19.320 --> 01:06:48.670

Jisun An: served as individual rewards, even though we've been talking about only one rewards. But then rewards can be combination of many of these functions as well. So that's the so if you have multiple different goals, meaning that different rewards, then still the combination of them can be served as a 1 reward of the model, and then that will will learn the the policies which is once again the actions to maximize these rewards.

269

01:06:53.930 --> 01:06:55.050

Jisun An: So

270

01:06:55.220 --> 01:07:21.760

Jisun An: the Rl. Hf is once the aim is to align the Rms with the human intents, and there are like 3 steps, and the 1st step is so we usually start with the a large pre-trained, add them so the instruction tuned model that we had in after this the the second step. So for the 3rd step we will start from here.

271

01:07:22.660 --> 01:07:28.270

Jisun An: And so so instruction tuned model

272

01:07:28.710 --> 01:07:33.959

Jisun An: actually is good enough for most of the jobs it may not I mean

273

01:07:34.468 --> 01:08:02.741

Jisun An: if you think about especially to collect the existing knowledge. So if you are asking some some general questions about like searching kind of Google instruction tuned model itself already. Quite good enough. But it's just rl, kind of just improves this instruction. Tuned model to be better at firstly, finding better path to get the correct answer, and secondly,

274

01:08:03.170 --> 01:08:14.564

Jisun An: reflect the human preference but the this instruction tuning model definitely has some limitations, and Rl is partly kind of tackle these limitations.

275

01:08:15.070 --> 01:08:18.352

Jisun An: so this provides the learning. Usually

276

01:08:19.260 --> 01:08:44.189

Jisun An: get the the correct, and I mean they usually have, like the correct outputs. And if but then, if you are able to generate like 15 different examples. But if you are selecting only the ones that you consider to be correct and then use it for the supervised learning, then you are basically wasting all these 14 different examples that was not preferred by the user. But then, so considering these are the negative.

277

01:08:44.189 --> 01:09:02.520

Jisun An: then you can actually make the model to be better aligned with those preferred. And also, you can like, be away from those negative examples as well. But usually the supervisor learning just don't ignore these negative feedbacks, and they just don't learn from this negative feedback.

278

01:09:02.840 --> 01:09:11.802

Jisun An: And also this was some. The point that we already discussed. So tasks like open-ended creative generation have no right answers. And

279

01:09:12.439 --> 01:09:36.677

Jisun An: they can actually have multiple valid responses. But then the training data can only include some of them. And also the 3rd point is about the hallucination. So in the train, in the instruction tuned model because they are simply the model that generate the next tokens, so especially the earlier models they usually

280

01:09:37.109 --> 01:10:00.610

Jisun An: they struggle to recognize when it should say, I don't know. So even though they don't know about something they just say, and they just generate something about. So if you are asking, Is there a particular person? Do you know particular person who doesn't exist in the world, then we know that the person is not existing. So the model should say that I don't know, but because these are the model that just generate.

281

01:10:00.610 --> 01:10:14.480

Jisun An: and they know that this is the human person's name, so they will find some likely tokens that will follow by that sentence, and they will just generate something, and that's the main reason that why these models has the hallucinations.

282

01:10:14.730 --> 01:10:29.373

Jisun An: And so the Rl. In in certain way, you can kind of prevent this hallucination by by creating this preference data, when you confront the something that you don't know, then say, I don't know

283

01:10:30.020 --> 01:10:47.080

Jisun An: and also I mean, there could be, as I showed, there could be, different human feedback or human preferences on different dimensions, and they usually does not optimize for these specific needs or the values, and the Rl. Is to try to capture them.

284

01:10:51.160 --> 01:10:52.270

Jisun An: So

285

01:10:59.040 --> 01:11:01.065

Jisun An: so I mean

286

01:11:03.940 --> 01:11:08.380

Jisun An: So, as we mentioned the second step of the

287

01:11:09.020 --> 01:11:28.669

Jisun An: Rlhf will be creating a reward model. And these reward models are separate model that are trained based on the preference data. And this preference data is coming from the preference rating where for the from the instruction tuned model for the same prompt, we can sample like the different outputs

288

01:11:28.740 --> 01:11:47.751

Jisun An: for this y, 1, y. 2, y. 3, we ask human to label these 3 examples, and then rank, which output is more appropriate given this particular prompts, and we call these as our preference, judgment. And then now we use this preference judgment to build the

289

01:11:48.380 --> 01:11:54.229

Jisun An: the rewards, and the the main reason that that we are building this reward model is because.

290

01:11:54.230 --> 01:12:18.619

Jisun An: if you need to rate all possible generated output from the model, it is very, very expensive. Once again, remember, for the 1,000 updates of the 1,000 prompts of the 1,000 rollout that will require 1 billion human scores, which is extremely extremely expensive. And that's the reason that we are building the reward model so that we can automatically evaluate the human preference on behalf of the human

291

01:12:19.295 --> 01:12:30.709

Jisun An: and so the idea of the reward model is, so can we train a model to predict the human preference judgments. So the input would be the prompt and the output and the

292

01:12:31.080 --> 01:12:37.290

Jisun An: output, the generated text and the output will be some scholar score that represents the value.

293

01:12:37.960 --> 01:12:47.659

Jisun An: And then there could be many different ways to train this model. But the the initial paper was using this readily pairwise preference model.

294

01:12:48.161 --> 01:12:54.030

Jisun An: Which is something that I will do from next on Thursday.

295

01:12:54.376 --> 01:13:04.439

Jisun An: So from here I will go little bit deeper of how what's the the loss for this weird model? And what is the loss for the Rl. Atf.

296

01:13:04.440 --> 01:13:26.160

Jisun An: and we will talk a little bit about Dpo, which is the direct preference optimization. which is the basically overcome. The cumbersomeness of the Rl. And and hopefully talk also a little bit more about the different techniques that used in the deep stick on Thursday. Any questions

297

01:13:26.190 --> 01:13:27.259

Jisun An: up to here.

298

01:13:29.800 --> 01:13:47.289

Jisun An: So I hope so, even though I missed a lot of technical parts. I hope that this was a good overview of what's the Rl. And why the rl, has been used for the language modeling, and especially for building a good quality. Large language models.

299

01:13:47.850 --> 01:13:55.950

Jisun An: e, yeah, that will be all for today. And yeah, I will see you on Thursday.

300

01:13:56.810 --> 01:13:58.170

Jisun An: Thank you.