WEBVTT

1

00:00:05.870 --> 00:00:13.330

Jisun An: Alright. Thanks, for in joining the lecture today, today's pass code is Fiveram. Please mark your attendance.

2

00:00:13.630 --> 00:00:15.654

Jisun An: A few announcement.

3

00:00:17.810 --> 00:00:36.260

Jisun An: so so on this Thursday we will have a lab session. So please bring your laptop we will use call Lab. So Web Browser will be basically what you will need. But I mean it'll be. I mean, you can just see it, but it'll be nice to do it together. But then so. But we will do

4

00:00:36.260 --> 00:00:49.089

Jisun An: super simple lab like doing basically text classification with different models that we've been talking about. And even though we haven't really talked about degenerative models. But I think just it'd be fun to

5

00:00:49.090 --> 00:01:12.460

Jisun An: use. I mean, there are also like fast AI who which believes in doing in practice without any knowledge. If you can run the model you can run, you can do so. I also kind. I kind of like that aspect as well. So this. And and I assume that you have some basic knowledge about these models as well. So we will have some practice on building doing the text classification with different models.

6

00:01:12.890 --> 00:01:23.261

Jisun An: But then, but then I also understand that there may be some of you have already experienced doing text classification. So for some of you it may be a little boring. So I I keep this

7

00:01:23.550 --> 00:01:47.389

Jisun An: this Thursday, plus as an optional, so we will not check the attendance, so feel free to attend or not. And also I will share the recordings after the course, and also I will share the canvas as well. So maybe if if you feel that you're already familiar with the like text classification. Then you can just review the code on your own and just check back the recordings.

8

00:01:47.440 --> 00:01:52.830

Jisun An: So that was just a note. But so that's the something that we will do this Thursday.

9

00:01:53.490 --> 00:02:00.209

Jisun An: And also I mean, I realized that the the material that I that I prepared was

10

00:02:00.210 --> 00:02:25.189

Jisun An: far more than what I expected, so I made a bit of adjustment in the schedule, but the major dates, especially. The exam dates, will not change, but like when the t 1 i mean the theoretical assignment is released and the p. 1. Those dates might be changing over time. But I will keep you updated and also update the canvas. So, and also there was a minor adjustment as well. So now

11

00:02:25.480 --> 00:02:42.680

Jisun An: we will do the attentions and the modeling and the prompting all together. In a 4 sequential lectures, and then we will. The week 7. We were going to do the lab, 4 prompting and the fine tuning as well. So just for your reference.

12

00:02:43.330 --> 00:02:56.040

Jisun An: and finally a small note. So unfortunately, I need to leave immediately after the class to pick up my baby girl. So I I know that everyone has some questions. I can take a quick questions, but if it will take longer.

13

00:02:56.040 --> 00:03:16.379

Jisun An: more than 2 min, then I would be I mean, you can. You're welcome to come to my office hour or just ping me via email. I just wanted to let you know. So I I really need to leave, and there's a traffic as well. So and thanks a lot for your understanding. But I will still be able to take few quick questions before and after the class. So thanks a lot for your understanding for that.

14

00:03:16.862 --> 00:03:23.559

Jisun An: So for those who just arrived. Today's attendance passcode is 5 grand. So please mark your attendance.

15

00:03:24.860 --> 00:03:26.560

Jisun An: All right. So

16

00:03:26.710 --> 00:03:49.010

Jisun An: now let's move on to today's lecture. So we talk about the language modeling today, and we already started. And you probably familiar with the initial parts of this slide. But I will just go through again. So once again, in terms of building an Nfp system, we started with the very simple representation which is like bag of words

17

00:03:49.377 --> 00:04:04.082

Jisun An: it. Essentially, you are using individual word as a feature, and there's no interactions. I mean, there's no other information about the word themselves. Then it's identity. But you are using them as a feature and use probably some

18

00:04:05.142 --> 00:04:29.460

Jisun An: like very simple one layer network to build a classification model. And in those models there are some of the things that are missing from this pow model, which are basically the conjugations or the compounds, was not able to be handled, which is then handled by the subword models, and they cannot really handle the word similarity which can be tackled by the word embeddings.

19

00:04:29.460 --> 00:04:47.250

Jisun An: and also the combination features was also something missing which can be reserved by the neural networks. And this part also I will, I mean today. I will also talk a little bit more about this one. And finally, one of the most important thing is missing. Was this sentence structure that it really, if the

20

00:04:47.460 --> 00:05:13.989

Jisun An: if the data is now, the words are getting long. I mean, anything is getting longer. Then there's no like sharing of the interactions among these words, or just in the sequence so, and which can be tackled by the sequence model. So we will talk mostly about how to use the neural network to model the language and also how to use the I mean, and introduce, like various sequence models that has been used in this model in the language.

21

00:05:14.701 --> 00:05:18.079

Jisun An: Yeah, there are a few more chairs in. And here as well.

22

00:05:18.680 --> 00:05:22.089

Jisun An: there's a 3. There's a 1 3 here. Yeah.

23

00:05:24.370 --> 00:05:29.280

Jisun An: alright. So so what's the language modeling?

24

00:05:30.870 --> 00:05:40.884

Jisun An: So the goal of modeling a language is assign a probability to a sentence. So we given a sentence, we want to

25

00:05:41.390 --> 00:05:47.989

Jisun An: find out what's the probability of that sentence and the reason that we want to do this is because, if

26

00:05:47.990 --> 00:06:12.219

Jisun An: a if we have a model that assigned a probability to a sentence, then it can be really useful for various applications, and one of such application is like machine translation. So if we know that the high winds tonight has a more probable than large winds tonight. Then, whenever you had to translate like a sentence in other language into English, then you.

27

00:06:12.220 --> 00:06:21.950

Jisun An: and also, if the words can be translated in either high or large. Then, given the other kind of context words, you're probably you would choose

28

00:06:22.515 --> 00:06:46.314

Jisun An: the 1st one, the over, the second one, and also for like spell correction. Example. If you've seen the office is about 15 min from my house, then you would see that is like 15 min from or 15 min from you see that the 15 min from would be far more probable than the other one. So you will. You will know that there might be some

29

00:06:47.450 --> 00:07:14.280

Jisun An: The sentence might be incorrect, and then you may need to kind of correct the spellings, and also for the speech recognition. Even for some sentence that sounds really similar to each other. I so of N, basically, you can see that the probability of seeing this sentence, I so of N, if it's higher than the ION, then it'll be much easier to choose that one over the other. So so that's the basically.

30

00:07:14.280 --> 00:07:19.139

Jisun An: what is it? Why, we want to model the language, and what is the language modeling?

31

00:07:20.400 --> 00:07:28.490

Jisun An: And and more? And then there are many other applications that you can also, I mean, get benefit from this language modeling.

32

00:07:31.860 --> 00:07:54.019

Jisun An: So more formally, basically, we want to compute a probability of a sentence, or like a sequence of the words. So the x here is the sentence, the the capital X is the sentence. And then we assume that X basically has n words as a like sequential of the words. So these are the words that are composing a sentence.

33

00:07:54.350 --> 00:08:01.090

Jisun An: and and we want to compute the probability of of these drawing probability of these words.

34

00:08:01.720 --> 00:08:05.472

Jisun An: and also the related work would be so

35

00:08:06.110 --> 00:08:21.651

Jisun An: Instead of like computing the probability of the whole sentence. If you compute the probability of an upcoming word given your previous word. So this this one also is very related task, because eventually you are kind of

36

00:08:22.300 --> 00:08:37.659

Jisun An: considering. I mean, you're basically predicting, like the next kind of word predictions. And this will give you the idea of what is the probability that you will observe from these 5 words as a sentence.

37

00:08:38.860 --> 00:08:52.550

Jisun An: So either computing, like px or PI. Xi. Given x. 1 to x, minus y minus one, which are the next words, prediction, kind of task. Both either of them are called as a language modeling.

38

00:08:53.240 --> 00:09:09.999

Jisun An: So the question here, then, is now so and here I found that we missed the commas. So these are actually 5 different words. So it's comma word comma water comma is comma so comma transparent. So we have 5 words here. So then, how can we compute this Px

39

00:09:10.180 --> 00:09:19.280

Jisun An: and we kind of briefly talked about this, so we can use the chain rule of the probability to compute the joint probability.

40

00:09:19.910 --> 00:09:49.220

Jisun An: And here the chain rule in the probability is that when you have 2 events, and if you want to. I mean the chain rule is coming from the conditional probability where, if you want to compute a given BA probability of events, a given event. B, then you can use this condition probability equation where basically, it would be the probability that both A and B occurs at the same time divided by the probability that the event. B, we're going to happen.

41

00:09:49.220 --> 00:09:50.050

Jisun An: So

42

00:09:51.020 --> 00:10:01.300

Jisun An: and if you just like, change this formula, then we will. We will get this particular equation where, the joint probability using the conditional probability.

43

00:10:01.600 --> 00:10:26.399

Jisun An: So if we use this conditional probability, and also this is applicable when there are like more variables. So if we want to compute the joint probability of events, A, B and C and D, then you can kind of doing that with probability of events, a multiplied by probability. B. Given A and probability. C, given A and B

44

00:10:26.520 --> 00:10:41.100

Jisun An: multiply by P. Probability, d. Given A, B, and C, so this is like, I mean, some basic knowledge from the probability. And we can use literally this particular occasion to our own

45

00:10:41.430 --> 00:10:59.639

Jisun An: problem, which is so if we want to compute. So consider each of these words as a event. So we are kind of looking at the probability to observe each of these words, and if we want to compute, what is the probability to observe this sentence that's composed of these 5 words.

46

00:10:59.640 --> 00:11:12.680

Jisun An: by applying the chain rule, we 1st compute the probability of observing it. One word multiply by probability, water given it, and probability of ease given its water.

47

00:11:12.770 --> 00:11:18.123

Jisun An: and so on. So you can kind of combining, adding this, join

48

00:11:18.700 --> 00:11:33.549

Jisun An: conditional probability to compute the joint probability. So that will be kind of, I mean, if you can compute each of those probability, then you should be able to compute this Px, which is P. Probability of its water, is so transparent.

49

00:11:34.930 --> 00:11:41.560

Jisun An: Any, any, any question up to here is hopefully, it's clear.

50

00:11:42.720 --> 00:12:11.709

Jisun An: So and to be more formally defining once again, the language modeling is similar to like next token prediction, and it can be computed by the joint probability of words in sentences. So to compute the px this pi is the the the symbol for the multiplication. So basically, starting from PI mean, so these should be

51

00:12:11.850 --> 00:12:18.348

Jisun An: the the formula may not be very correct, I mean, I mean, starting from the 1st words,

52

00:12:19.412 --> 00:12:27.099

Jisun An: if you just multiply, multiply them. They are conditional probabilities. Then you can compute this Px.

53

00:12:28.850 --> 00:12:40.570

Jisun An: And now the question is then so how can we compute this conditional probability? So how can we know that, like what is likelihood of seeing water after it? For example?

54

00:12:44.459 --> 00:13:13.099

Jisun An: So once again, the Px. Equal this multiplication of the different conditional probability. And then how can we compute each of these? Conditional probability is, and the most common or simple way to compute this probability is simply using counts and divide. So within your corpus, like training data, basically, you count the times that you observe is water is so transparent you count them. Maybe you've seen them, maybe

55

00:13:13.120 --> 00:13:22.069

Jisun An: unlikely, but maybe 2, and then, divided by its water is so maybe 5. So maybe this probability could be 2 divided by 5

56

00:13:22.290 --> 00:13:32.770

Jisun An: 0 point 4. So for each of these you can compute the conditional probability for each of these. next token kind of probability.

57

00:13:32.970 --> 00:13:34.730

Jisun An: But is this feasible?

58

00:13:36.350 --> 00:13:40.999

Jisun An: Do you think that this would be a good way to estimate the probability?

59

00:13:41.890 --> 00:13:44.260

Jisun An: Probably not? And why not?

60

00:13:50.950 --> 00:14:05.522

Jisun An: I mean, basically, there will be so many different combinations that you need to look for. And also there will be only very few cases that you will observe very long sentences as well, so it would be just impossible to compute or estimate the probability in this way.

61

00:14:07.470 --> 00:14:10.100

Jisun An: so there are just too many possible sentences.

62

00:14:10.761 --> 00:14:22.219

Jisun An: And also I mean to. And basically, we were not going to have enough data to estimate them. Stably, if you are using simple like count and divide.

63

00:14:23.640 --> 00:14:26.057

Jisun An: So what would be the alternative?

64

00:14:26.760 --> 00:14:51.110

Jisun An: and that's where this Engram language model is coming in. And I hope I assume that maybe you've heard of like Unigram or Bigram language modeling here and there, and the engram modeling is based on the assumption of the Markov Assumption. And here so, rather than when to compute or estimate the probability of transparency given, its war is so.

65

00:14:51.110 --> 00:15:08.409

Jisun An: so here we assume that the next token is depending on all your all, your previous tokens right? But instead of using all of your previous tokens, you assume that it only depends on like one or 2 so, and your previous context words.

66

00:15:08.803 --> 00:15:23.769

Jisun An: so if I mean, if you are using like diagram, then you can assume so instead of the probability of these transparents given is water is so, is simply is approximate to the probability of transparency given so.

67

00:15:23.850 --> 00:15:32.750

Jisun An: or if you are assuming, like trigram, then this could be approximate. The transparency probability transparency given is so

68

00:15:32.890 --> 00:15:36.669

Jisun An: so. In other words, you assume that I mean 2 words are always

69

00:15:37.896 --> 00:15:52.240

Jisun An: I mean you. You kind of take the context window as one, and you consider like one previous word, and and that can determine what will come next. And in the second one you assume that the 2 words can determine what comes next.

70

00:15:52.440 --> 00:16:14.639

Jisun An: So this is the what the and I mean. In reality, it's it's so, I mean, basically, if you enlarge the context window, then it will be more and more similar to the true probability. And if you have like small context probability, then obvious small context window. Then obviously, I mean, this will not be very realistic, but so there will be some balance. But this is like very

71

00:16:14.930 --> 00:16:19.069

Jisun An: like easy and very initial way to model our language.

72

00:16:19.910 --> 00:16:47.699

Jisun An: And once again, if we just make it more formal equation, then this Px now can be presented as the products of conditional probability where the next token is depending on K previous tokens. So if you look at the equation, then, instead of using all the previous tokens in the sentence, you're only looking at the k previous tokens to predict the next tokens here.

73

00:16:51.050 --> 00:17:11.350

Jisun An: So and and I mean, this can be. I mean, this is like generalized version, and the easiest, simplest language model using the N-gram model would be pound based unigram model. And here even we don't consider any previous context. But here we assume that the probability of a sentence

74

00:17:11.733 --> 00:17:28.219

Jisun An: or probability of the next token, is just depending on simply by itself. So basically, we assume that independent, the words are independent to each other. So the the occurrences of the words doesn't depend on any context word, they just independently exist.

75

00:17:28.890 --> 00:17:36.400

Jisun An: So simply this, this particular probability that we wanted to compute can be approximate to just a pxi.

76

00:17:38.780 --> 00:17:49.379

Jisun An: And if we want to compute the probability of a sentence which is once again the joint probability of x. 1 to Xn. That could be just multiplication of these pis.

77

00:17:53.000 --> 00:18:02.899

Jisun An: and I mean, if you think about this unigraph, I mean, it'd be very dumb. Right? You simply know, basically the cons of each of the words in your training corpus and then

78

00:18:02.970 --> 00:18:32.969

Jisun An: and then now given, you have that model. If you want to generate a sentence, then you basically sample one word by one word from that model which will basically depends on the count of the word. So these are like some random example that from some unigram model. Try to generate sentences by sampling each of the word. And this will be like 5th on all future. The unincorporated inflation result. So and basically, it doesn't make any sense.

79

00:18:33.298 --> 00:18:34.940

Jisun An: But but they are just.

80

00:18:34.940 --> 00:18:50.210

Jisun An: I mean, you will see the words that is more probably existed in the training set, so you will see more of like this, determinants on of and the articles and some of the words that were popular within the training corpus.

81

00:18:50.710 --> 00:19:15.639

Jisun An: But this is a model. And then how you basically estimate the Px. The probability of the I mean the probability of a word could be simply can be estimated from the maximum likelihood, estimation, and in this case I mean, the name is very fancy maximum likelihood, estimation. But this is just a proportion of the words of the training corpus.

82

00:19:15.640 --> 00:19:24.180

Jisun An: So it simply counts how many times you see the Xi. Divided by the the entire Corpus volume.

83

00:19:24.370 --> 00:19:29.719

Jisun An: so that will just which will be now turning into kind of probability that you observe that? Xi.

84

00:19:32.310 --> 00:19:35.080

Jisun An: So any question up to here?

85

00:19:38.390 --> 00:19:39.100

Jisun An: Yes.

86

00:19:40.260 --> 00:19:41.070

Jisun An: Documents.

87

00:19:42.315 --> 00:19:44.580

Jisun An: Oh, the warmth.

88

00:19:45.400 --> 00:19:50.149

Jisun An: Oh, yeah. So x is each each of the word, yeah. So

89

00:19:50.560 --> 00:19:53.839

Jisun An: so number of the words in the training corpus, yeah, different. So

90

00:19:59.040 --> 00:20:08.276

Jisun An: right? So. But then, so there's a like one small problem, and so now

91

00:20:09.010 --> 00:20:32.899

Jisun An: If but but once again the the language model the goal was to predict a probability for a sentence right? And the using the unigram, how you compute the joint probability for a sentence is multiplying these pi's right. So if you had, is water, is still parents transparent. Then you basically compute p it and P. Water, P. Trans.

92

00:20:32.930 --> 00:20:39.799

Jisun An: Pes and P transparents and users multiply them each other. And then

93

00:20:39.930 --> 00:20:54.310

Jisun An: and of course, you are building your model using the training data. So your vocabulary is based on your training data. But if you now want to apply this unigram model to a test data, then there may be a word that doesn't exist.

94

00:20:54.400 --> 00:21:15.360

Jisun An: that in the training data. And then, in that case, this maximum probability that you see that word will become simply 0. And the problem that having this value, 0 is that once again, to compute the joint probability which is the sentence probability. If one of the value is 0, then it will be simply the 0 right.

95

00:21:16.020 --> 00:21:17.280

Jisun An: Does that make sense

96

00:21:18.530 --> 00:21:43.480

Jisun An: for those who may just lost. So here in the Unigram the joint probability, the p. 1 to pn is simply by multiplying the independent individual. Probability. Xi. Right? So you. So if you now. So you build your model using your training data and assuming in your test data, there was a suddenly I'm g 7. And Jason word never appeared in the training data and those

97

00:21:43.480 --> 00:21:50.380

Jisun An: this count became 0. So any unknown word became the will have the Px. As

98

00:21:50.380 --> 00:21:55.149

Jisun An: 0, because it never appeared in the training corpus. So in that case.

99

00:21:55.150 --> 00:22:19.820

Jisun An: so for the test data, I am Jason. You want to compute the joint probability of the Imson. But then, because Jason never appeared in the training copus, the value becomes 0, and this joint probability will 0. So basically, even though I mean, there are some probability I am is very common sentence. So this sentence should have some value, but because of this unknown word I am Jason will become simply 0,

100

00:22:20.000 --> 00:22:29.880

Jisun An: and this will what it means that basically the unicorn model will fail to estimate the proper probability for a sentence because of these unknown words.

101

00:22:30.580 --> 00:22:34.470

Jisun An: So how can we then resolve the issue. Any idea?

102

00:22:35.290 --> 00:22:40.779

Jisun An: I know it's already in the slide. But any idea how you can solve solve?

103

00:22:43.660 --> 00:22:44.330

Jisun An: Yes.

104

00:22:45.220 --> 00:22:59.230

Jisun An: some words. Yes, yeah, exactly. So. I mean, we talked about the subword problem. And one of the reason is that I mean, I mean one of the benefit of using the subword is now you will have. I mean they. They will handle this low frequency words.

105

00:23:01.400 --> 00:23:03.480

Jisun An: So that was the 1st option, and

106

00:23:03.770 --> 00:23:28.890

Jisun An: and and also I mean, compared to the number of possible words, number of possible subwords would be far less than the words, so using the subword would be a good option, and so that will make sure that no words in the test, no tokens in the training test data will have 0 value. And there's another simple method to tackle this which is

107

00:23:28.980 --> 00:23:43.470

Jisun An: using the unknown word model. So rather than simply using the 0 value, I mean, even okay, before this unknown word model the even the easier one would be. You can simply add one to every word. So you assume that

108

00:23:43.470 --> 00:24:06.639

Jisun An: there's you assume that any word just appears at least once in the word. So you can just add one when you compute this maximum likelihood, or even for the inference time, but that will actually have some other consequences in the model, and they will prevent you estimating the probability correctly. So that's even though that is like the simple, possible, and also work for some kind of application, but it's usually not used.

109

00:24:06.640 --> 00:24:17.034

Jisun An: And instead, they're using this unknown word model. So in a way, it's very similar to the idea of adding one. But instead of the adding one to all the words, basically, you are

110

00:24:17.620 --> 00:24:45.660

Jisun An: estimating a particular probability for unknown words, and then use use them as your when you finally compute your Px. So this Pmle. X. Is the probability that you are getting from the maximum maximum likelihood estimation which is based on your training data, and then the lambda on could be just the rate. So how much you want to give the importance into these 2. So assuming that we have like 5% importance for the this.

111

00:24:45.870 --> 00:25:03.599

Jisun An: So 5% in importance to this P Unc. Model, and then it means that 95% importance weight will go to this Pmle model. So in normal cases, you will probably take the value from this mle model. And using, I'm basically multiplying it by 0 point 9 5

112

00:25:04.600 --> 00:25:20.809

Jisun An: so that that will be your kind of value. But whenever you've seen this unknown word, then basically, you will use this P. Unc. Probability, which can be very low. So you can. So there are different ways. How you can build this unknown world model. And the simple way is, you can just

113

00:25:21.270 --> 00:25:41.080

Jisun An: assume that this is just a fixed value, which is very small. So P. On xi. Could be simply given any. Xi is just returns like 0 point 0 0 1, so that you just prevent them. The Px will not be ever 0, but it will have at least some small value, and they will handle the unknown board and in in.

114

00:25:41.240 --> 00:26:02.490

Jisun An: And so you can. Just you can just fix a particular value for this P. Unc. But instead of that, you can also. Common practice is that in your training data you just set a particular threshold and find those infrequent words, and then you just change, replace those infrequent words to Unc. So you just add a new token

115

00:26:02.660 --> 00:26:30.070

Jisun An: named Unc. Unk, and you just replace all those words as Unc, and then you just compute the probability, the Unc. And whenever you've seen any word that is unseen from the training set, then you just change it, I mean, consider I mean managing that as ank. Yes, yes, but because you. The purpose here is you just prevent those probability to be, not to be 0. So you just want to replace to something else.

116

00:26:30.550 --> 00:26:45.009

Jisun An: Yeah. So any on the word will be having very low probability. But this low probability can be set as a fixed value, or it can be also estimated from your training data by replacing the low frequency words into the Unc.

117

00:26:45.120 --> 00:26:47.260

Jisun An: So that would be like the 2 different ways.

118

00:26:48.220 --> 00:26:49.070

Jisun An: Yes.

119

00:26:49.900 --> 00:27:02.609

Jisun An: not get rid of it. But you you mean the the unknown word model right? Or the 1st one.

120

00:27:06.400 --> 00:27:08.820

Jisun An: You mean the segment or the subword mode.

121

00:27:08.980 --> 00:27:12.109

Jisun An: Yeah, so that's the whole company is already.

122

00:27:12.820 --> 00:27:13.810

Jisun An: So the

123

00:27:14.220 --> 00:27:21.739

Jisun An: make sure. All the same, data doesn't have the same word during the training, the testing data, the same word during the training.

124

00:27:21.890 --> 00:27:36.150

Jisun An: Oh, instead of I mean, you can consider the the sub tokens, I mean. So if any, our puppet can be represented by the sub tokens, then I think you can cover any

125

00:27:36.240 --> 00:27:56.230

Jisun An: any word. But but this may or may not be really realistic, as you mentioned. I mean, there could be like some emojis that that really never happened in the training. So I I guess there are certain assumptions, so they may also replace some of very low frequency trunk or something. Yeah, but you are right. Yeah.

126

00:27:57.240 --> 00:28:06.290

Jisun An: thanks for the question, and I think that's a good point, and I'm not sure how in reality this option was applied to

127

00:28:06.661 --> 00:28:16.938

Jisun An: because you assume that you are not. See, you actually don't assume that you were going to see the test data set, so it may be not possible to include or tokens

128

00:28:17.260 --> 00:28:39.029

Jisun An: to be in the vocabulary. But if you assume that these are normal, like purpose with the English, then I think it is possible to include all the tokens. But there could be some unexpected tokens, like some random symbol that you've never seen, and for them I guess there you should some special treatment for that as well, but I'm not sure what was the attribute. Yeah, but that's yeah. Thanks for yeah. Pointing that out.

129

00:28:39.360 --> 00:28:40.680

Jisun An: Any other question.

130

00:28:42.950 --> 00:28:57.430

Jisun An: Okay, and that was that, and and and then and one thing to point out is when we compute this, during probability, which is the multiplication of the independent probability, I mean.

131

00:28:57.720 --> 00:29:08.999

Jisun An: So normally, this Px will be very small value. And we already talked about this. But if you multiplying very, very small value, then basically you will overflow.

132

00:29:09.240 --> 00:29:34.309

Jisun An: So instead of multiplying all this probability, we take the log, and then, so that the multiplication can trend it to I mean transport, I mean changing it to the summation. So instead of the multiplying all this probability, we tend to like put all this probability to the log space so that we can just add the low probabilities.

133

00:29:34.688 --> 00:29:46.939

Jisun An: So by doing so, I mean, most of the characteristic will be the same. So if the probability is getting higher, the low probability will also get high, get higher. So so this is one thing that that we do.

134

00:29:47.170 --> 00:29:54.620

Jisun An: and also also, the adding is faster than the multiplying. So computationally is also more efficient.

135

00:30:00.409 --> 00:30:23.790

Jisun An: Yeah. So the diagram model, so now here, basically, you assume that you the the next token is the probability to see a next token is depending on your previous words previous token of words. And that's the the diagram model. So we approximate this probability as the P. Xi. Given p. Xi. Minus one.

136

00:30:24.480 --> 00:30:33.789

Jisun An: and these are some generated words as well from the diagram model. And now, even though these are also still not yet full functioning sentence

137

00:30:34.100 --> 00:30:36.469

Jisun An: be slightly better than

138

00:30:36.740 --> 00:31:02.330

Jisun An: the unigram model, like, Okay, Rose, one in this issue is, so you kind of see that this captures some relationship, maybe, that the like after East there are pursuing. So you you see this kind of word formations, and in or like Mr. Guria, etcetera. So I mean by grade at least like capture some of the things. But still it's not a perfect language model.

139

00:31:04.500 --> 00:31:31.349

Jisun An: And these engram model can be generalizable to N. Gram models. And so basically given a context length, as our N, you call them as our angram model and in reality, you will need to. So to compute those px or pxi given pxi minus one, you need to actually count. And you need to use the count and divide the kind of method. So

140

00:31:31.430 --> 00:31:36.770

Jisun An: so basically, N cannot be really, really long. But I think the trigrams, or like

141

00:31:36.870 --> 00:31:57.220

Jisun An: 4 Gram or 4 Gram models are still possible. And there exists. Google has released these Mgram models. So you can search for Google and Gram. And they actually have list of all trigrams, or even full Gram and the counts as well. So that would be one resources that you could use if you need these ngram models

142

00:31:58.100 --> 00:32:24.749

Jisun An: and also to deal with this general count. Now, they are also doing some kind of interpolation. So if basically, the idea, I'm not going to go detail, because it's not very important nowadays. But the idea is that if because if the probability that you've seen awards are given the previous 4 words, it would be very, very low. So if you cannot find those probability, then you are using the you're only using the last 3 words instead of the last 4 words

143

00:32:24.750 --> 00:32:44.300

Jisun An: so, and also, if the probability of seeing Xi. Given the last 3 words is very, very low, then you are also using the probability of Xi given the last 2 words. So I mean you just I mean, you're kind of using just some information of your historical information, but just not dealing, using the old end context.

144

00:32:44.699 --> 00:32:46.600

Jisun An: to to get those values.

145

00:32:49.806 --> 00:32:52.710

Jisun An: So Ingram is not

146

00:32:52.800 --> 00:33:08.700

Jisun An: a great model, but it's still very, I mean interesting. Because it it gives a fundamental problem. Definition of the what is what, why we are, what is the language modeling is. But still there are many problems. One of them is.

147

00:33:08.700 --> 00:33:29.050

Jisun An: they basically cannot share the strength among these similar words. So all this I mean, if you see that she bought a car. She purchased a car. She bought a bicycle, she purchased bicycles. You kind of know that the car and the bicycles are share some kind of semantic meanings, and also vote and purchase, are also

148

00:33:29.120 --> 00:33:43.540

Jisun An: share some meanings. But then, in the this engram model they tend to miss all these semantics and also they cannot condition on the context with the intervening words. So the Dr. Jane Smith and Dr.

149

00:33:44.130 --> 00:34:01.529

Jisun An: Then through the smith. So I mean, because they have same doctor and the same surname. There may be some connections, but basically, these 2 2 phrase will the the probability of them will be totally different. Because of these intervening words.

150

00:34:02.420 --> 00:34:12.010

Jisun An: and also they cannot handle long distance, dependency. So. And and that's the reason that we will introduce the neural network based language modeling

151

00:34:13.730 --> 00:34:14.670

Jisun An: but

152

00:34:14.940 --> 00:34:24.542

Jisun An: but so so, even though engrams are mostly introduced as to introduce the concept of the language modeling in this space, but still

153

00:34:25.429 --> 00:34:45.940

Jisun An: still, they can be extremely fast to estimate and apply. So it'll be a good practice for you to understand this language modeling techniques. I think it's easy to do. And also they can be better at modeling low frequency phenomena. So I mean your language modeling because they are basically try to

154

00:34:46.429 --> 00:35:12.940

Jisun An: approximate, I mean, capture all this relationship into a set of the parameters. So they miss out all the information, especially for the low frequency words. So, but then diagram is literally the count, so they will never lose those information. But then, in the in the neural network, sometimes this information can be just go away. So for that purposes Engram can be can be useful. But once again, I mean in practice, we don't use this model, but consider it as a like

155

00:35:14.150 --> 00:35:16.970

Jisun An: that helps for you to understand the language, modeling.

156

00:35:19.200 --> 00:35:20.400

Jisun An: Any questions.

157

00:35:24.110 --> 00:35:48.199

Jisun An: Right? So before moving on to the neural network based language modeling. I'd like to talk briefly about how we evaluate this language model. So how do we know which model is good or bad? So the goal was to assign a probability to a sentence. Right? So we need to know whether our model prefer the good sentences over the bad ones and and I mean.

158

00:35:48.360 --> 00:36:12.110

Jisun An: the usually the evaluation is like like typical. As so we train our model on a training set. And then we test our model performance on the data on that that we haven't seen. So we are using, like the test data to evaluate. And we need, we need to use some kind of evaluation metrics is to tell how well our model does on our test. I mean, this is very typical machine learning, right?

159

00:36:12.250 --> 00:36:25.966

Jisun An: But then, Nfp. Especially in the Nfp. I think people have been using extrinsic evaluation of the language model. In other words, I mean, because it was not very easy how to

160

00:36:27.790 --> 00:36:53.460

Jisun An: even though you know that your language model is good. Sometimes it also fails in the intrinsic evaluation, so they tend to focus on the applications, and then and they just use the language model to tackle that particular application and use them as a proxy of the performance of the model themselves. So, for example, if you have some energy tasks, like spelling corrections or translations, or even simple classification.

161

00:36:53.460 --> 00:36:59.220

Jisun An: so you use those tasks as to evaluate the performance of the language model.

162

00:36:59.960 --> 00:37:29.489

Jisun An: So that was actually the one way. But intrinsic evaluation can be time consuming and and those sometimes, I mean within. When you build your model themselves? You could use the intrinsic evaluation. And once again, in practice, you probably need to see whether this actually works, for like different tasks that you want to achieve, but in the internally, or when you do the experiments, you could use intrinsic evaluation to

163

00:37:29.490 --> 00:37:33.439

Jisun An: to see whether the model learns anything and model gets any better or not.

164

00:37:33.450 --> 00:37:36.959

Jisun An: So so what is this intrinsic evaluation?

165

00:37:37.650 --> 00:37:53.130

Jisun An: So one of the most commonly used metrics, especially when you like, build these models and train the models is the likelihood. And now I hope that you this I don't know. Probably you heard this likelihood before, but after all this

166

00:37:53.550 --> 00:38:13.649

Jisun An: probability of joint probability, I hope they this, this equation is a bit more familiar to you. So this is literally compute the joint probability log of joint probability of test set in each for each of the sentences in the test set right and this, and they just some so I mean.

167

00:38:13.810 --> 00:38:21.979

Jisun An: So for each of the example you computed the low probability log joint probability, and you just sum them up. And that's the log likelihood value.

168

00:38:22.150 --> 00:38:51.380

Jisun An: And then and the the problem of the log likelihood would be, and especially if you are comparing like different training corpus, then, and then the training corpus may have different length. Right? So you just normalize it by the number of words existing training corpus, and which we we call this a per word, low likelihood. And that's the notation we have like words, low likelihood and simply, that's just divided by the number of vocabulary in the test

169

00:38:51.710 --> 00:38:54.450

Jisun An: just to to have a fair comparison.

170

00:38:54.955 --> 00:39:24.240

Jisun An: And then sometimes you may hurt the negative low likelihood and it's because simply because you want to use it as a loss, and for the loss you want to minimize the value rather than the maximize, because loss is like error. So I mean, this is just human perception, and I mean human find it better. I mean, associated better when it's something you minimize something. Your error basically losses an error.

171

00:39:24.240 --> 00:39:35.699

Jisun An: So the negative log likelihood is the negative value of the log of joint probability of each of the sentence in the test set.

172

00:39:36.850 --> 00:39:40.859

Jisun An: and then then entropy is

173

00:39:41.030 --> 00:39:59.849

Jisun An: is also called as a forward cross entropy, and if you look at the entropy, then this value is almost looking exactly the same as forward log likelihood, a negative negative log forward likelihood. So value, I mean, the equations are exactly the same.

174

00:40:00.492 --> 00:40:28.050

Jisun An: But but 1 1 difference is here instead of the the log. They are using log base 2, and and I'm not an expert to explain this. But but these are coming from the information theory. Where the reason they are using this log 2 is they want to represent how many bits it would require to compress a data. And the bits is basically represents one and 0. So

175

00:40:28.050 --> 00:40:34.889

Jisun An: so I think that's the what where this is coming from. But the idea of

176

00:40:36.010 --> 00:40:42.970

Jisun An: idea of what it computes is more or less the same as negative log likelihood, forward, low likelihood.

177

00:40:43.090 --> 00:40:53.280

Jisun An: and finally, the perplexity which is the common measure that used for evaluating the atom models is the 2 to the entropy.

178

00:40:53.740 --> 00:41:19.449

Jisun An: So what this does is now you, these are just exponentiate those log value. So you initially took the log to avoid the offload under flow. But then, but then to for the interpretability you want to exponentiate again so that it can became a count value again. And that's the what the perplexity represents.

179

00:41:21.291 --> 00:41:28.630

Jisun An: and yeah. So I mean, and then so so if you just so these 3 will be all the same kind of vacation.

180

00:41:28.860 --> 00:41:53.859

Jisun An: So let's say when a let's say that you see this kind of sentence, when a dog see a score? It will usually. And what would be the next words? How can there could be like the different things. It can be like the B, or jump, or start, or run, or try, and for each of these tokens their probability would be, or different. And then, if you use this probability their perplexity will be something like this.

181

00:41:54.650 --> 00:42:23.310

Jisun An: And here the the perplexity. How you interpret this value is to predict the next token, how many words did you have as a sample space? So so, if the perplexity is like 20 that you, it means that you have 20 different options to choose from, to determine the next token. So, in other words, perplexity, the lower value is better because you for choosing the next token, you basically have less

182

00:42:23.500 --> 00:42:48.459

Jisun An: options to choose from meaning that it is more deterministic. So you are more confident about what token will be coming next, if you have only so compared to perplexity. 20 versus 50, meaning that in determining the next token you will have 50 different options versus 20 different options. So if you have 50, it means that you are uncertain like about the what will come next.

183

00:42:48.460 --> 00:43:03.809

Jisun An: If you have like lower options than smaller options, then you are more confident about what you're predicting. So the lower perplexity means that the model is more confidence about what to predict. As a next token, given your previous context.

184

00:43:04.700 --> 00:43:12.220

Jisun An: So that would be so it take your time to like. Observe how the log pro likelihood.

185

00:43:12.658 --> 00:43:31.221

Jisun An: and the perplexities are related to each other. But the key thing is, the perplexity is the intrinsic evaluation that evaluates how good your language model, and the lower value is better. And this value, and the reason that people like perplexity is because the value themselves can be

186

00:43:31.600 --> 00:43:54.330

Jisun An: can be interpreted in a nice way. And I also saw already post that people were saying that if we are using low likelihood, then your performance improvement looks very small, but if you are using perplexity, then the performance improvement looks much larger, so more impressive. So there was a fun joke there from the academics. So perplexity is one of the intrinsic evaluation that you're using for that

187

00:43:55.230 --> 00:43:56.280

Jisun An: any question

188

00:44:01.272 --> 00:44:05.519

Jisun An: the lower indicator. Yes.

189

00:44:05.750 --> 00:44:13.279

Jisun An: So in this case the word would be one of the boxes as well. He will usually.

190

00:44:15.850 --> 00:44:18.050

Jisun An: Yeah.

191

00:44:18.460 --> 00:44:19.210

Jisun An: 25.

192

00:44:20.140 --> 00:44:27.049

Jisun An: No, I think. Well, it it will be usually be. Yeah, I mean,

193

00:44:30.410 --> 00:44:31.460

Jisun An: Well.

194

00:44:31.720 --> 00:44:50.190

Jisun An: you should believe in the numbers, not to your intuition. Yeah, yeah. I mean it. Could it could be it could be I don't know. Spoken word. It would be a bit weird, but maybe written, it would be using more. And also it depends on the model that computed probability. So the training corpus could be slightly different. I don't know.

195

00:44:50.330 --> 00:44:57.300

Jisun An: Yes, it will be when we are. Probably

196

00:44:57.730 --> 00:45:15.669

Jisun An: complexity is evaluation. It's it's not a technically, it's a loss. A loss is low likelihood. Negative low likelihood value is forward. Low likelihood is more often used as a loss. Actually, even not a loss. It's it's just measuring something. Something

197

00:45:16.320 --> 00:45:22.830

Jisun An: as a evaluation metric is even not a loss, I think. Yeah, thanks. Thanks for the question.

198

00:45:24.410 --> 00:45:44.809

Jisun An: right? We we have a lot to move on. So back to the language modeling so we now come to the neuron language model. So instead of this simple diagram, now, we will build a model that calculates the probability of the next word in a sequence given some history and

199

00:45:45.152 --> 00:46:09.130

Jisun An: so we've seen the engram. But the neural network model far performed better than the Engram language model. And I mean, nowadays we are using like the transformer which we will talk next week. But still the simple fit port also can a good job. And also it can be used for like different systems. So this will be the good intermediate step to understand the the transformer.

200

00:46:09.780 --> 00:46:33.359

Jisun An: So this fit for the neural network neural language model was introduced by the Benjio at L in 2,003. So that's already now more than 20 years ago. But this was, I think, a very exciting kind of start of like developing this entire domain. So here the test is simple, we are predicting the next word wt, given the prior word like

201

00:46:33.360 --> 00:46:49.469

Jisun An: t, minus one t, minus 2 t. Minus 3 and etc. And this would be also depending on the context window. I mean, we call them as a context window. It depends on how many words that you want to consider them. But if you are considering more priority definitely, the model will be more complicated and harder to train and etc.

202

00:46:50.440 --> 00:46:51.780

Jisun An: The testing bow down

203

00:46:51.860 --> 00:47:16.071

Jisun An: so in a in a way that so so given your prior work, which we assume that it now already has the embedding so we are not using like the one hot vector but some kind of trained word to back any embeddings. So these are the kind of the one layer, 2 layer network that we've seen in the last lecture. So the the embedding value will have some weights. And then,

204

00:47:16.390 --> 00:47:33.299

Jisun An: has the neural unit where it do the weighted sum, and then we detain each like the nonlinear activation function and the feeding into one hidden layer where the hidden layer themselves also has the weight which will be now became the output metrics before this optimax

205

00:47:34.660 --> 00:47:35.680

Jisun An: So

206

00:47:36.990 --> 00:47:45.316

Jisun An: this will be like the very basic kind of model of the fit forward letting language model that is using the feed forward network.

207

00:47:45.650 --> 00:48:08.130

Jisun An: and so I've been talking about this combination features. And now the neural network we need enable you to look at the combination of the features. And here's 1 of those example. So I mean, so the word embedding basically can capture some features of the words. So I mean, this is not just a toy example, but assuming that the 1st feature of the word embedding indicate the verb.

208

00:48:08.130 --> 00:48:21.780

Jisun An: and the second feature indicate the determinant. Then, like these, vectors, like giving will have higher value for the 1st value, and they all have will have higher, like positive value for the the second value.

209

00:48:22.050 --> 00:48:39.079

Jisun An: And then now you have this weight metrics that now capture the combination of these features. So, for example, the maybe the 34th row in the weight matrix can. Look at the feature one and the

210

00:48:39.080 --> 00:48:58.369

Jisun An: and the feature one in the second to previous world, and the feature 2 in the previous word and then somehow, that's the what and and basically if these, both words are the positive, then this also will lead to the more positive value in this network. So

211

00:48:58.660 --> 00:49:04.420

Jisun An: so neural network kind of will enable this kind of combinations, features.

212

00:49:04.810 --> 00:49:12.060

Jisun An: and to give you how it works, and also it relating to the neural network that we talked in the last lecture.

213

00:49:12.080 --> 00:49:40.570

Jisun An: So we have the input of a N words and previous words. And each of these words will be, we already have some embeddings. So the W. Before the W. The projection layer embedding will be our 1st kind of layer 0, and for each of these embeddings we will have some weights. That is connecting to the hidden layer. So these rated sum of W and the words embeddings, plus some

214

00:49:41.700 --> 00:50:09.857

Jisun An: that will kind of lead input into the hidden layer. And then the hidden layer is also connected to the output layer where the software access is applied to and that will result in the some kind of prediction probability for each of the words. So because we are predicting the next token, you can see this as a prediction classification kind of problem.

215

00:50:10.610 --> 00:50:33.170

Jisun An: but then the output labels, basically, you have, like like N labels, which N is the number of the vocabulary. So you are. You have all for all, each of the old words, you are producing some probability that, given this 4, all the which is the 3 words ought to be the next probable words, so you are doing kind of classification. Here.

216

00:50:35.020 --> 00:50:36.270

Jisun An: Does that make sense?

217

00:50:37.840 --> 00:50:38.809

Jisun An: Don't know.

218

00:50:39.140 --> 00:50:46.559

Jisun An: Oh, alright so why do we use those Max here?

219

00:50:50.670 --> 00:50:53.139

Jisun An: So fix?

220

00:50:54.920 --> 00:51:17.429

Jisun An: Oh, which one? Yes. So so yeah. So we have the sigmoid where it can be used as a like binary classification, and the softmax is the generalized version of the sigmoid. So it will basically turn any value into the probability. So before this soft Max. The value that we will have will be log count, so it will be some

221

00:51:17.490 --> 00:51:38.081

Jisun An: of the weighted some of these previous layers. So eventually you will have some value which is not from like ranging from like minus plus I don't think there's a like particular range there. And then given those values. So you simply by adopting the software, you will now have the probability for each of the word

222

00:51:38.780 --> 00:51:39.770

Jisun An: So

223

00:51:40.217 --> 00:51:55.049

Jisun An: after the Softmax. Then my head is our predicted values. Right? So assuming that for order this model we're resulting in okay for this other work, the probability will be 0 point 0 0 1 fish would be 0 point 6 5 is

224

00:51:55.050 --> 00:52:09.950

Jisun An: too high, but I just exaggerate a little bit, and then maybe 4 4 or the 4 would be very weird, so maybe lower probability, etc. Etc. So while I mean, basically, our model will predict some probability for each of the words, given these inputs.

225

00:52:10.040 --> 00:52:21.269

Jisun An: then, how we train is, basically, we have our beer value from our sentence. Right? So assuming that our actual training sentence was 4 or the fish.

226

00:52:21.560 --> 00:52:39.750

Jisun An: It means that so we had the true label, One in the fish. So now this became once again the classification problem. So you can compute the loss. And in this case using cross entropy loss, which is commonly used for the classification. So once you have the loss, what can we do is basically we do that propagation.

227

00:52:39.910 --> 00:52:53.310

Jisun An: So you will update these U and W. According to this loss value and using the derivatives and the gradients of all these parameters. So you can see that these, the parameters, will change their values according to these values.

228

00:52:53.790 --> 00:53:01.299

Jisun An: And they will do like example by example. But usually they just do Mini batching where you combine like few examples.

229

00:53:04.750 --> 00:53:10.579

Jisun An: mask and you'd have a bunch of different words, that they would predict. But in this situation, would you be like.

230

00:53:11.580 --> 00:53:34.839

Jisun An: Yeah, yeah, I mean, this is this was once again introduced in 2,003 was very, very simple model. So and and this is the basics. So you can. Now, as the computing power getting better, you can consider more words, and you can also change the task as not to predict the next token. But you can predict the mask

231

00:53:35.123 --> 00:53:43.060

Jisun An: tokens as well. So that would be something that we can also talk next week. Yeah. But this one super simple. Given the 1st n word K word

232

00:53:43.060 --> 00:53:55.930

Jisun An: just predicting the next one and and consider. Keep them as a classification task. But but the the core structure or framework is very similar to this one. I think they share a lot of properties.

233

00:53:57.620 --> 00:53:58.810

Jisun An: Any questions.

234

00:54:02.860 --> 00:54:03.839

Jisun An: Okay, so

235

00:54:04.530 --> 00:54:28.600

Jisun An: one thing to note is that so basically, now, the neural network model has an interesting property where basically shares all these features for the inputs and the output words. So the word embedding. Already we know that the similar input word will get the similar vectors, and also the similar output words will also get similar roles in these metrics, and also the hidden in the hidden state. The similar context also gets similar

236

00:54:28.600 --> 00:54:35.100

Jisun An: hidden state. So altogether, this will help to understand better the which words are more related to each other.

237

00:54:35.180 --> 00:54:37.560

Jisun An: That became the strength of this neural network.

238

00:54:38.430 --> 00:54:51.949

Jisun An: And 1 1 additional thing which is not really necessary, but for practical reason. So so so basically, the output layer before before the softmax.

239

00:54:51.960 --> 00:55:17.910

Jisun An: the embedding space will be the size will be more or less, you can make it as the same as the input embedding. So in so initially, I told you that you can given the input you get some embedding from like word 2 big or some other model. But you can actually train those embeddings together because you can tie the input and the output embedding, which represents the word, because in this model we are predicting the word in the as an output.

240

00:55:17.910 --> 00:55:38.480

Jisun An: so the output embedding can be exactly the same. I mean the the size of the output. Embedding can be exactly the same as an input embedding. And you can use actually them as our same embedding so that you can update both. Update this embeddings. I mean, only have one in one embedding space for the words. And then you can just update doing training.

241

00:55:38.800 --> 00:56:05.339

Jisun An: And I mean, this may not be really nice, I mean, doesn't sound like a necessary step. But if you think about, if you consider all entire vocabulary, then that the metrics will be really, really large. And if you keep in as an input and open embedding, then basically, the space will be doubled up. So if you just combine them, I'm using the same embedding space. Then you can like, reduce the space memory that you use for the training.

242

00:56:08.320 --> 00:56:15.680

Jisun An: And so what this neural net language based modeling has handled? So now we know that

243

00:56:15.680 --> 00:56:39.640

Jisun An: the similar words will have similar input embeddings. So they will be also reserved and also the condition on context. Also, the combination feature will also solve these kind of particular issues. But still this model cannot really handle the long distance dependence, dependency. So I mean, once again, I mean, you can increase the context window larger, and but then they will really make the model more complicated and

244

00:56:40.038 --> 00:56:44.020

Jisun An: costly to to train them. So and so. So.

245

00:56:44.200 --> 00:57:07.029

Jisun An: That's the reason that what we have introduced is 4 sequence models. And I'm sorry that I'm kind of in a rush a little bit, but I will keep it really short, because this is, some of them are very important concept. But now the modern modern models is not using some of the model that I will introduce today. So just I will keep it very high level. And so I mean, yeah.

246

00:57:07.671 --> 00:57:10.770

Jisun An: and just point out some of the important aspects of them.

247

00:57:11.300 --> 00:57:17.180

Jisun An: So these models are essentially come out to handle this long distance dependency.

248

00:57:17.970 --> 00:57:33.741

Jisun An: And there are like 3 major types of the sequence model is like the recurrence where it conditions representation on the encoding of the history. So they are using now the the outputs of the

249

00:57:34.520 --> 00:57:39.900

Jisun An: so they are using the outputs of the previous inputs

250

00:57:41.510 --> 00:57:46.330

Jisun An: as our input to the next units of the neural units.

251

00:57:46.480 --> 00:58:08.840

Jisun An: And the convolution is basically they are using like the local context. And the attention is they are also using the local context, but they also consider giving some weight to like the different context. So the attention is something that we will talk more and in detail next week. But I will go through the recurrence and the

252

00:58:09.600 --> 00:58:34.570

Jisun An: convolution. So the recurrent network is the the network themselves was introduced in 1996, 9. And and these are once again. These are, I mean, this was used in language modeling and the Nlp. At some point, and also they are used for some other domains, like image processing, or like time, series, analysis, and etc. But but not anymore in the in the Nlp.

253

00:58:34.570 --> 00:58:45.480

Jisun An: Themselves. But so they were introduced as a tool to remember the information. So these are like the example of the feedforward neural network that we just saw.

254

00:58:45.480 --> 00:58:50.196

Jisun An: And they in the recurrent neural networks, basically the output of the

255

00:58:51.176 --> 00:59:12.690

Jisun An: and if you see the equation, I think it would be easier to compare. So your hidden states are in the feed forward neural network. Your hidden states are only depending on your input X right? But then, in the recurrent neural network. You also consider the output of your previous hidden states. So that would be the only difference between these 2 models

256

00:59:13.800 --> 00:59:16.590

Jisun An: and so, and

257

00:59:16.720 --> 00:59:28.949

Jisun An: and the the network that looks like here. If we just unroll them in time, then it will basically. So given, we have these inputs and you start from some hidden states

258

00:59:28.950 --> 00:59:58.209

Jisun An: and then which are initialized as a randomly or with a 0 values. And then you go through this added unit where it has the input from the the 1st words, and then it has this output, and using that output, it predicts a particular label. So in this example, I'm giving you particular, I mean it can be. It can be also like language modeling. Next token, prediction, or it can be like a named entity predictions, or like sequence, labeling kind of task as well. So you can kind of imagine those tasks.

259

00:59:58.716 --> 01:00:01.409

Jisun An: And then, once you have that, then the

260

01:00:01.740 --> 01:00:18.129

Jisun An: so for the next token, you are also using the output of the previous neural unit as an input to the Rnn neural unit. And then you also get the input from the from the actual input. And also you predict, and you just do that over time.

261

01:00:18.220 --> 01:00:33.890

Jisun An: So in in this case, this will learn, then, the order of these sentences, and so I mean, so this long sentence dependency will be partially solved. Here.

262

01:00:35.540 --> 01:01:00.080

Jisun An: but then, but then, actually so once again, I don't expect you to understand this entirely, but there are just some, a few concept that I hope you can kind of catch it. But in terms of the training. If you think about the neural network, the one of the condition or neural network to train, is it need to be directly, acyclically

263

01:01:00.100 --> 01:01:14.458

Jisun An: a cyclic graph. So basically, you need to be able to compute the loss at the end. And then that loss value will be back propagated to all the parameters. But in this example, because you are predicting for each of the

264

01:01:15.715 --> 01:01:38.940

Jisun An: neural units, you will have n predictions, and and those it means that you will have N losses at the same time. So and and this will not be then. Now the directed Acyclic graph. So the solution was, basically, you just combine all these losses of just to sum them up, and then just make them as like Acyclic graph, and then back, provide all these loss to the other end.

265

01:01:42.010 --> 01:01:43.070

Jisun An: So

266

01:01:43.390 --> 01:02:09.769

Jisun An: so. And one other thing that to note is that these parameters are all shared so, even though it looks like they are in different. I mean the visualizing way they they look. They are different, like neural unit. But they are actually just one neural unit. So you will, you will just update this parameter as this sequence or the text orders. So but but these will be just one set of parameter that you will see.

267

01:02:09.900 --> 01:02:13.260

Jisun An: And also it's same for the attentions or the other convolution network

268

01:02:15.240 --> 01:02:20.920

Jisun An: and but then here oh.

269

01:02:21.540 --> 01:02:36.920

Jisun An: so there are like some other types, I mean, just this tie Rnn is now, instead of predicting something from left to the right, they just added another additional layer, so that it can also predict from right to left.

270

01:02:37.060 --> 01:02:56.250

Jisun An: So I mean, even though this may not be really needed for the language modeling. Because if the language anyhow, you need to go from left to the right. But in this case, if you are using like by other than that, it will understand better the context left both direction. So if you using this model for encoding, then it would actually work better.

271

01:02:56.779 --> 01:03:14.709

Jisun An: But but the way that it trains are the all the same. So now I mean, you can kind of imagine that each of these are the neural units, and you design a framework that, depending on how you kind of add up all the layers, or this this units. You can kind of design this kind of audit and models.

272

01:03:16.516 --> 01:03:31.493

Jisun An: The one of the biggest issue of this Rnn. Was the vanishing gradient. And this was something that we also talked before. And the re. So the problem is, basically, if this layer is getting deeper.

273

01:03:31.960 --> 01:03:47.910

Jisun An: or, in other words, if you are, your gradients are in like the multi layers, then these values are getting smaller and smaller. So at some point the gradients itself will be very, very small, so it will not going to update the parameter at all.

274

01:03:48.873 --> 01:03:57.039

Jisun An: So this also happens by this kind of Rnn, or like really deep network. So So

275

01:03:57.360 --> 01:04:08.820

Jisun An: to solve this vanishing gradient problem, the long term long, short term memory at Stm, probably also the terms you've heard somewhere. This was the solution for the vanishing gradients.

276

01:04:08.940 --> 01:04:10.000

Jisun An: And

277

01:04:10.840 --> 01:04:21.460

Jisun An: interestingly, what they did was simply they just made an editive connections between the time steps so that they can. They will just do not. Don't lose all these informations.

278

01:04:23.060 --> 01:04:44.330

Jisun An: and as a I mean, it's a characteristic of the derivatives, so the addition does not modify the gradients, and there will be no vanishing. And so also there are additional thing which are the gates to control the information flow. So that's the like. Stm, so I once again, I don't want you to understand everything here. But

279

01:04:44.330 --> 01:05:04.429

Jisun An: so this is also the one unit of the the model. So if you recall back to the Rnn. Model that the On Road, Rnn. There was one unit that named as Rnn, right? So that was very simple, that input was as a input x 1. And they also take the previous hidden layers outputs.

280

01:05:04.430 --> 01:05:14.829

Jisun An: But instead of since that. So that was very simple network. But instead of that one unit, the scm has this as our one unit, and the the rest of the structure will be the same

281

01:05:14.830 --> 01:05:18.979

Jisun An: in this unit. They have this memory cell, which is a C,

282

01:05:19.040 --> 01:05:42.151

Jisun An: and this one is simply they don't do much. They, they just just simply add edit up like across the time. So they will. They will kind of prevent the Spanish ingredient problem. And also there are 3 different gate, which is the uni and the All. I mean, not the UN, the gates that are controlling for

283

01:05:42.680 --> 01:05:55.769

Jisun An: in between you and in O, which are they will control. How much information you want to take from the previously the layer, or how much information you want to take from the your input. And how much output?

284

01:05:56.120 --> 01:06:11.069

Jisun An: How much weight do you want to also give for the output for the next hidden state? So these were kind of like the some ideas that we're putting into the Stm. That were improved a lot from the Rnn model.

285

01:06:12.930 --> 01:06:18.589

Jisun An: I just just really giving you love some high level ideas here. But but any questions from up to here.

286

01:06:21.500 --> 01:06:22.305

Jisun An: right?

287

01:06:23.590 --> 01:06:34.654

Jisun An: and then the another types of sequence model was the convolution. And these were using like the local context. So I mean, it's a very simple neural network where you're just using

288

01:06:35.090 --> 01:07:02.800

Jisun An: your like neighbors, information. So once again, the equation here would be easy to understand. So instead of simply waiting based on your one, input, they take like X minus T minus one xt and xt plus one. And I mean there were some attempts to use this convolution network for the language modeling or the Nlp. But to be honest, convolution is far more used in the image processing and in the image.

289

01:07:02.800 --> 01:07:26.470

Jisun An: Looking at your local context would be more important. Right? Because I mean, if you look at like, assume the meat as an image, then like here will be dark and dark and dark, right? So I mean looking at surrounding would be really important in the imaging processing. So they have been using the convolution natural model far more and and to certain extent it was also working for the language, but it was not as effective as the other models.

290

01:07:27.706 --> 01:07:31.679

Jisun An: And also they are considering x plus one.

291

01:07:31.730 --> 01:07:59.440

Jisun An: In other words, the goal of the language modeling is to predict the next token. But if you need information from something that happens in the future, it just doesn't make sense as a language modeling. So the convolution network was used and explored. But once again it was not very successful. But you can change the convolution network to autoregressive model by only considering your historical information as an input to the neural net. But then.

292

01:08:00.900 --> 01:08:04.449

Jisun An: do you think this is similar to something that you've seen before?

293

01:08:07.230 --> 01:08:15.019

Jisun An: Oh, yes, exactly so, I mean, not exactly the Rnm. But this would be the bit forward network.

294

01:08:15.520 --> 01:08:18.510

Jisun An: because oh, I mean, just just as it is.

295

01:08:19.149 --> 01:08:22.922

Jisun An: So now you to predict something

296

01:08:24.439 --> 01:08:36.770

Jisun An: to predict something. You are looking at your previous context. So you are getting basically 3 previous awards to predict the next one. So these are exactly the same as a fit for the newer model that we've seen before.

297

01:08:37.197 --> 01:08:46.729

Jisun An: This may not be, really like the Rnm. Because it doesn't share the between the his and state. They don't share departments. so that would be that one.

298

01:08:50.120 --> 01:09:04.510

Jisun An: And yeah, that's something sorry that I was really in a rush for the last couple of slides, but that's the the key kind of aspects that that lead it to all the

299

01:09:06.012 --> 01:09:08.969

Jisun An: the modern model, which is a transformer.

300

01:09:09.290 --> 01:09:13.350

Jisun An: So before wrapping up so

301

01:09:13.600 --> 01:09:38.359

Jisun An: for the last, the sequence model part, the most important part will be the vanishing gradients that will be really critical for the neural network models and the Lstm. Was introduced. This additive structure, and this will also appear back to the transformer model. So I will. I will go back there again. But I think that's the

302

01:09:38.689 --> 01:09:40.279

Jisun An: the most important part.

303

01:09:40.580 --> 01:09:46.019

Jisun An: Yeah. So I mean, we still have a few minutes. So any any questions for today? Yes.

304

01:09:47.950 --> 01:09:49.529

Jisun An: here's my zoom.

305

01:09:50.270 --> 01:09:54.579

Jisun An: Oh, here's strength. So I mean, I mean, it's just it's it's just a term. So

306

01:09:55.242 --> 01:10:00.470

Jisun An: the strength of the neural models where they are is just shared. So

307

01:10:00.930 --> 01:10:07.290

Jisun An: it just i it. It's just the yeah, basically, the.

308

01:10:09.580 --> 01:10:39.110

Jisun An: So before the neural models like Engram models, they are not able to share what they learned to predict something else. But in the neural network. Now they can they can share. If the if the words are similar, then they will have the similar vectors. Even for the output words, they will, if they are also similar, they will also share, have the similar vector so that's the so they will share the the actual vector characteristics.

309

01:10:41.480 --> 01:10:55.280

Jisun An: Yeah. So if you are computing based on like Engram model, the probability will be just totally different. Right?

310

01:10:55.450 --> 01:11:04.669

Jisun An: But then, in the neural natural model, those sentence phrase will now have more similar probability, because they know that these 2 words are dissimilar as well.

311

01:11:05.630 --> 01:11:06.349

Jisun An: Okay, thank you.

312

01:11:21.810 --> 01:11:23.420

Jisun An: Any other questions.

313

01:11:28.000 --> 01:11:38.079

Jisun An: Okay, so next week we will start to group format the groups. And so I decided to go with

314

01:11:39.163 --> 01:11:53.726

Jisun An: let you to form the group, so it'll be nice like to say hello to each other, and like asking around what they are interested in. So I will put excel where you just put your group and

315

01:11:55.090 --> 01:12:00.220

Jisun An: I. And also I will.

316

01:12:01.120 --> 01:12:22.020

Jisun An: I will leave now totally to you how many of people you want to have in your group? But ideally like 3 to 5. But for the Phd students you can also do independent study if you want to incorporate your own work to the class. So yeah, any number from one to 5.

317

01:12:23.469 --> 01:12:42.610

Jisun An: And let's see how many groups we will have if there are too many. I I may do some adjustments, because we only have 2 classes to the presentation and etc. So I don't want to make them to too many groups, but so so that we're gonna happen next week? I think we are almost finalizing the class. Yeah. So

318

01:12:43.640 --> 01:12:59.609

Jisun An: and and if you need like random selection, you can, you can just leave maybe notes on the excel that you need, like some some group. Then there could be some other people who also want some random group. Then they will can become a group together. So so let's do the group formation next week.

319

01:13:00.480 --> 01:13:09.949

Jisun An: Alright thank you so much, and I will see you on Thursday. Once again, it'll be a lab on the text classification. Bring your laptop

320

01:13:10.510 --> 01:13:14.969

Jisun An: with the collab. Okay, thank you. See you. On Thursday.