

MITOCW | Lec-15

PROFESSOR PATRICK WINSTON: You know, some of you who for instance-- I don't know, Sonya, Krishna, Shoshana-- some of you I can count on being here every time.

Some of you show up once in a while.

The ones of you who show up once in a while happen to be very lucky if you picked today, because what we're going to do today is I'm going to tell you stuff that might make a big difference in your whole life.

Because I'm going to tell you how you can make yourself smarter.

No kidding.

And I'm also going to tell you how you can package your ideas so you'll be the one that's picked instead of some other slug.

So that's what we're going to do today.

It's the most important lecture of the semester.

The sleep lecture is only the second most important.

This is the most important.

Now the vehicle that's going to get us there is a discussion about how it's possible to learn in a way that is a little reminiscent of what we talked about last time.

Because last time we learned something very definite from a small number of examples.

This takes it one step further and shows how it's possible to learn in a human-like way from a single example in one shot.

So it's extremely different, very different from everything you've seen before.

Everything that involves learning from thousands of trials and gazillions of examples and only learning a little tiny bit, if anything, from each of them.

This is going to learn something definite from every example.

So here's the classroom example.

What's this?

It's an arch.

I know the architects are complaining that it's not an arch in architecture land.

It's a post and lintel construction.

But for us today it's going to be an arch.

Now if you were from Mars and didn't know what an arch was, I might present this to you and you'd get a general idea of some things that might be factors, but you'd have no idea what's really important.

So then I would say, that's not an arch.

And you would learn something very definite from that.

And then I would shove these together and put this back on, and I would say, that's not an arch either.

And you'd learn something very definite from that.

And then I could paint the top one blue, and you'd learn something very different from that.

And how can that happen is the question?

How can that happen in detail, and what might it mean for human learning and how you can make yourself smarter?

And that's where we're going to go.

All right?

So how can we make a program that's as smart as a martian about learning things like that?

Well, if you were writing that program, surely the first thing you would do is you'd try to get off the picture as quickly as possible and into symbol land where things are clearer about what the important parts are.

So you'd be presented with an initial example that might look like this.

We'll call that an example.

And it's more than just an example.

It's the initial model.

That's the starting point.

And now we're going to couple that with something that's not actually an arch but looks a whole lot like one, at least on the descriptive level to which we're about to go.

So here's something that's not an arch, but its description doesn't differ from that of an arch very much.

In fact, if we were to draw this out in a kind of network, we would have a description that looks like this, and these relations would be support relations.

And this would be drawn out like so.

And the only difference would be-- the only difference would be that those support relations that we had in the initial model-- the example-- have disappeared down out here in this configuration.

But since it's not very different from the model, we're going to call this a near miss.

And now, you see, we've abstracted away from all the details that don't matter to us.

Last time we talked about a good representation having certain qualities-- qualities like making the right things explicit.

Well, this makes the structure explicit, and it suppresses information about blemishes on the surface.

We don't care much about how tall the objects are.

We don't think it matters what they're made of.

So this is a representation that satisfies the first of the criteria from last time.

It makes the right things explicit.

And by making the right things explicit, it's exposing some constraint here with respect to what it takes to be an arch.

And we see that if those support relations are missing, it's not an arch.

So we ought to be able to learn something from that.

What we're going to do is we're going to put these two things together.

We're going to describe the difference between the two.

And we're going to reach the conclusion that since there's only one difference-- one kind of difference with two manifestations to disappearing support relations, we're going to conclude that those support relations are important.

And we're going to turn them red because they're so important.

And we're going to change the name from "support" to "must support." So this is our new model.

This is an evolving model that now is decorated with information about what's important.

So if you're going to match something against this model, it must be the case that those support relations are there.

If it's not there-- if they're not there, it's not an arch.

All right?

So we've learned something definite from a single example.

This is not 10,000 trials.

This is a teacher presenting something to the student and the student learning something immediately in one step about what's important in an arch.

So let's do it again.

That was so much fun.

Let's do this one.

Same as before except that now when we describe this thing, there are some additional relations-- these relations, and those are touch relations.

So now when we compare that-- is that an arch?

No.

It's a near miss.

When we compare that near miss with our evolving model, we see immediately that once again there's exactly one difference, two manifestations, the touch relations.

So we can immediately conclude that these touch relations are interfering with our belief that this could be an arch.

So what do we do with that?

We put those together again and we build ourselves a new model.

It's much like the old model.

It still has the imperatives up here.

We have to have the support relations.

But now down here-- and we draw not signs through there-- these are must not touch relations.

So now you can't match against that model if those two side supports are touching each other.

So in just two steps, we've learned two important things about what has to be in place in order for this thing to be construed to be an arch.

So our martian is making great progress.

But our martian isn't through, because there's some more things we might want it to know about the nature of arches.

For example, we might present it with this one.

Well, that looks just like our initial example.

It's an example just like our initial example.

But this time the top has been painted red.

And I'm still saying that that's an arch.

So once again, there's only one difference and that difference is that in the description of this object, we have the additional information that the color of the top is red.

And we've been carrying along without saying so, that the color of the top in the evolving model is white.

So now we know that the top doesn't have to be white.

It can be either red or white.

So we'll put those two together and we'll get a new model.

And that new model this time once again will have three parts.

It will have the relations, an imperative form that we've been carrying along now, the must support and the must not touch, but now we're going to turn that color relation itself into an imperative.

And we're going to say that the top has to be either red or white.

So now, once again, in one step we've learned something definite about archness.

Two more steps.

Suppose now we present it with this example.

It's an example.

And this time there's going to be a little paint added here as well.

This time we're going to have the top painted blue like so.

So the description will be like so.

And now we have to somehow put that together with our evolving model to make a new model.

And there's some choices here.

And our choice depends somewhat on the nature of the world that we're working in.

So suppose we're working in flag world.

There are only three colors-- red, white, and blue.

Now we've seen them all.

If we've seen them all, then what we're going to do is we're going to say that the evolving model now is adjusted yet again like so.

Oh-- but those are imperatives still.

Let me carry that along.

At this time, this guy-- the color relation-- goes out here to anything at all.

So we could have just not drawn it at all, but then we would have lost track of the fact that we've actually learned that anything can be there.

So we're going to retain the relation but have it point to the "anything goes" marker.

Well, we're making great progress and I said there's just one more thing to go.

So let me compress that into this area here.

What I'm going to add this time is I'm going to say that the example is like everything you've seen before except that the top is now one of those kinds of child's bricks.

So you have a choice actually about whether this is an arch or not.

But if I say, yeah, it's still an arch, then we'd add a little something to its description.

So this description would look like this.

Same things that we've seen before in terms of support, but now we'd have a relation that says that this top is a wedge.

And over here-- something we've been carrying along but not writing down-- this top is a block.

A brick, I guess in the language of the day.

So if we say that it can be either a wedge or a brick on top, what do we do with that?

Once again, it depends on the nature of representation, but if we say that we have a representation, that has a hierarchy of parts.

So bricks and wedges are both children's blocks and children's box or toys.

Then we can think of drawing in a little bit of that hierarchy right here and saying well, let's see.

Immediately above that we've got the brick or wedge.

And a little bit above that we've got block.

And a little bit above that we've got toy.

And a little bit above that we eventually get to any physical object.

So what does it do in response to that kind of situation?

You have the choice.

But what the program I'm speaking of actually did was to make a conservative generalization up here just to say that it's one of those guys.

So once again it's learned something definite.

Let me see.

Let me count the steps.

One, two, three, four, five.

And I just learned four things.

So the generalization of a color, it took two steps to get all the way up to "don't care." So note how it contrasts with anything you've seen in a neural net.

Or anything you will see downstream in some of the other learning techniques that we'll be talking about that involve using thousands of samples to learn what it is-- to learn whatever it is that is intended to be learned.

Let me show you another example of how these heuristics can be put to work.

So there are two sets of drawings.

We have the upper set and the lower set.

And your task, you smart humans working in vast parallelism, your task is to give me a description of the top trains that distinguishes and separates them from the trains on the bottom.

You got it?

Nobody's got it?

Well, let me try one on you.

The top trains all have a short car with a closed top.

So how is it possible that a computer could have figured that out?

It turns out that it figured it out with much the same apparatus that I've shown you here in connection with the arches, just deployed in a somewhat different manner.

In this particular case, the examples are presented one at a time by a teacher who's eager for the student to learn.

In this case, the examples are presented all at once and the machine is expected to figure out a description that separates the two groups.

And here's how it works.

What you do is you start with one of them.

But you have a lot of them.

You have some examples-- we'll call the examples on top the "plus examples" and the examples on the bottom the "negative examples." So the first thing that you do is you pick one of the positive examples to work with.

Anybody got any good guesses about what we're going to call that?

Yeah, you do.

We're going to call that the seed.

It's just highly reminiscent of what we did last time when we were doing [? phonology ?] but now at a much different level.

We're going to pick one of those guys to be the seed, and then we're going to take these heuristics and we're going to search for one that loosens this description so that it covers more of the positives.

You see, if you have a seed that is exactly a description of a particular thing and you insist that everything be just like that, then nothing will match except itself.

But you can use these heuristics to expand the coverage of the description, to loosen it so that it covers more of the positives.

So in your first step you might cover, for example, that group of objects.

Too bad for your side, you've also in that particular case included a negative example in your description, but perhaps in this next step beyond that you'll get to the point where you've eliminated all of those negative examples and zeroed in on all the positive examples.

So how might a program be constructed that would do that sort of thing?

Well, think about the choices.

The first choice that you have it is to pick a positive example to be the seed.

And once you've picked a particular example to be the seed, then you can apply heuristics, all of them that you have, to make a new description that may cover the data better.

It may have more of the positives and fewer of the negatives than in your previous step.

But this, if you have a lot of heuristics, and these are a lot of heuristics because there's a lot of description in that set of trains, there are lots of possible things that you could do with those heuristics because you could apply them anywhere.

So this tree is extremely large.

So what do you do to keep it under control?

Well, now you have answers to questions like that by knee-jerk, right?

The branching factor is too big.

You want to keep a few solutions going.

You have some way of measuring how well you're doing so you can use a beam search.

This piece here was originally worked out by a friend of mine, now, alas, deceased, [? Rashad ?] [? Malkowski ?] when he was at the University of Illinois.

And of course, he wasn't interested in toy trains, he was just interested in soybean diseases.

And so this exact program was used to build descriptions of soybean diseases.

It turned out to be better than the plant pathology books.

We now have two ways of deploying the same heuristics.

But my vocabulary is in need of enrichment, because I'm talking about "those" heuristics.

And one of the nice things that [? Malkowski ?] did for me a long time ago is give each of them a name.

So here are the names that were developed by [? Malkowski. ?] What's happening here?

You're going from an original model to an understanding-- some things are essential.

So he called this the "require link" heuristic.

And here in the next step, we're forbidding some things from being there.

So [? Malkowski ?] called that heuristic the "forbid link" heuristic.

And in the next step, we're saying it can be either red or white.

So we have a set of colors and we're extending it.

And over here in this heuristic, going from red or white to anything goes, that's essentially forgetting about color altogether, so we're going to call that "drop link" even though for reasons of keeping track, we don't actually get rid of it.

We just have it pointing to the "anything" marker.

And finally, in this last step, what we're doing with this tree of categories is we're climbing up it one step.

So he called that the "climb tree" heuristic.

So now we have a vocabulary of things we can do in the learning process, and having that vocabulary gives us power over it, right?

Because those are names.

We can now say, well, what you need here is the "drop link" heuristic.

And what you need over there is the "extend set" heuristic.

So now I want to back up yet another time and say, well, let's see.

When we were working with that phonology stuff, all I did was generalize.

Are we just generalizing here?

No.

We're both generalizing and specializing.

So when I say that the links over here that are developed in our first step are essential, this is a specialization step.

And when I say they can't be-- they cannot be touch relations, that's a specialization step.

Because we're able to match fewer and fewer things when we say you can't have touch relations.

But over here, when I go here and say, well, it doesn't have to be white.

It can also be red.

That's a generalization.

Now we can match more things.

And when I drop the link altogether, that's a generalization.

And when I climb the tree, that's a generalization.

And that's why when I do this notional picture of what happens when [? Malkowski ?] program does a tree search to find a solution to the train problem, they're both specialization steps which draw in the number of things that can be matched, and generalization steps that make it broader.

So, let's see.

We've also got the notion of near miss.

And we've got the notion of example-- some of these things are examples, some are near misses.

We've got generalization specialization.

Does one go with one or the other, or are they all mixed up in their relationship to each other?

Can you generalize and specialize with near misses?

What do you think?

You think-- you don't think so, [INAUDIBLE]?

What do you think?

STUDENT: [INAUDIBLE] specialization.

PROFESSOR PATRICK WINSTON: [INAUDIBLE] lead to specialization.

Let's see if that's right.

So we've got specialization here, and that's a near miss.

We've got specialization here, and that's a near miss.

We've got generalization here, and that's an example.

And we've got generalization here, and that's an example.

And we've got generalization here, and that's an example.

So [INAUDIBLE] has got that one nailed.

The examples always generalize, and the near misses always specialize.

So we've got apparatuses in place that allow us to both expand what we could match and shrink what we could match.

So what has this got to do anything?

Well, which one of these methods is better, by the way?

This one-- this one requires a teacher to organize everything up.

This one can handle it in batch mode.

This one is the sort of thing you would need to do with a human because we don't have much memory.

That one is the sort of thing that a computer's good at because it has lots of memory.

So which one's better?

Well, it depends on what you're trying to do.

If you're trying to build a machine that analyzes the stock market, you might want to go that way.

Or soybean diseases, or any one of a variety of practical problems.

If you're trying to model people, then maybe this is a way that deserves additional merit.

How do you get all that sorted out?

Well, one way to get it all sorted out is to talk in terms of what are sometimes called "felicity conditions." So when I talk about felicity conditions, I'm talking about a teacher and a student and covenants that hold between them.

So here's the teacher.

That's me.

And here's the student.

That's you.

And the objective of interaction is to transform an initial state of knowledge into a new state of knowledge so that the student is smarter and able to make use of that new knowledge to do things that couldn't be done before by the student.

So the student over here has a learner.

And he has something that uses what is learned.

And the teacher over here has a style.

So if any learning is to take place, one side has to know something about the other side.

For example, it's helpful if the teacher understands the initial state of the student.

And here's one way of thinking about that.

You can think of what you know as forming a kind of network.

So initially, you don't know anything.

But as you learn, you start developing quanta of knowledge.

And these quanta of knowledge are all linked together by prerequisite relationships that might indicate how you get from one quantum to another.

So maybe you have generalization links, maybe you have specialization links, maybe you have combination links, but you can think of what you know as forming this kind of network.

Now your state of knowledge at any particular time can then be viewed as a kind of wavefront in that space.

So if I, the teacher, know where your wavefront is, can I do a better job of teaching you stuff?

Sure, for this reason.

Suppose you make a mistake, m_1 , that depends on q_1 .

Way, way behind your wavefront.

What do I do if I know that you made a mistake of that kind?

Oh, I just say, oh, you forgot you need a semicolon after that kind of statement.

I just remind you of something that you certainly know, you just overlooked.

Right?

On the other hand, suppose you make a mistake that depends on a piece of knowledge way out here.

That kind of mistake, m_2 .

What do I say to you then?

What do you think, Patrick?

What do you think I would say if you made that kind of mistake?

STUDENT: [INAUDIBLE].

PROFESSOR PATRICK WINSTON: No.

That's not what I would say [INAUDIBLE].

STUDENT: You'd tell us that we don't know that yet.

PROFESSOR PATRICK WINSTON: I would say something like that.

What [INAUDIBLE] suggested I would say.

Oh, don't worry about that.

We'll get to it.

We're not ready for it yet.

So in this case, I remind somebody of something they already know.

In this case, I tell them they'll learn about it later.

So what do I do with mistake number three?

That's the learning moment.

That's where I can push the wavefront out.

Because everything's in place to learn the stuff at the next radius.

So if I know that the student has made a mistake on that wavefront, that's when I say, this is the teaching moment.

This is when I explain something.

So that's why it's important for the teacher to have a good model of where the student is in the initial state of knowledge.

Next thing that's important for the teacher to know is the way that the student learns.

Because if the student is a computer, they can handle the stuff in batch.

That's one thing.

If the student is a third grader who has a limited capacity to store stuff, then that makes a difference in how you teach it.

You might teach it that way to the third grader, and that way, buried underneath this board, to a computer.

So you need to understand the way that the learner-- the computational capacity of the learner.

And there's also a need to understand the computational capacity of the user box down there, because sometimes you can be taught stuff that you can't actually use.

So by now, most of you have attempted to read that sentence up there, right?

And it seems screwy, right?

It seems unintelligible, perhaps?

It's a garden path sentence.

It makes perfectly good English, but the way you generally read it, it doesn't, because you have a limited buffer in your language processor.

What does this mean?

You're expecting this to be "to." Question.

But it's actually a command.

Here's the deal.

Somebody's got to give the students their grades.

Well, we can have their parents do it.

Have the grades given to their students by their parents, then.

So it's a command.

And you garden path on it, because you have limited buffer space in your language processor.

So with parentheses you can understand it.

You can learn about it.

You can see that it's good English, but you can't generally process that kind of sentence without going back and starting over.

And what about going the other way?

Are there covenants that we have to have here that involve the student understanding some things about the teacher?

Well, first thing there is is trust.

The student has to presume that the teacher is teaching the student correct information, not lying to student.

Ratified that you're all here because presumably you all think that I'm not trying to screw you by telling you stuff that's a lie.

There's also this sort of thing down here.

Understanding of the teacher's style.

So you might say, well, professor x, all he does is read slides to us in class, so why go?

You wouldn't be entirely misadvised.

That's an understanding of one kind of style.

Or you can say, well, old Winston, he tries to tell us something definite and convey a family of powerful ideas in every class.

So maybe it's worth dragging yourself out of bed at 10 o'clock in the morning.

Those are style issues, and those are things that the student uses to determine how to match the student's style against that of the instructor.

So that helps us to interpret or think about differences in style so that we can appreciate whether we ought to be learning that way, where that way is the way that's underneath down here, the way you would teach a computer, the way [? Malkowski ?] taught a computer about soybean diseases.

We can do it that way, or we can do it this way with a teacher who deliberately organizes and shapes the learning sequence for the benefit of a student who has a limited processing capability.

Now you're humans, right?

So think about what the machine has to do here.

The machine-- in order to learn anything definite in each of those steps, the machine has to build a description.

So it has to describe the examples to itself.

That's unquestioned, right?

Because what it's doing is looking at the differences.

So it can't look at the differences unless it's got descriptions of things.

So if you're like the machine, then you can't learn anything unless you build descriptions.

Unless you talk to yourself.

And if you talk to yourself, you're building the kind of descriptions that make it possible for you to do the learning.

And you say to me, I'm an MIT student.

I want to see the numbers.

So let me show you the numbers.

And when I'm going to show numbers-- the numbers that I'm going to show you show you the virtues of talking to yourself.

So here's the experiment.

The experiment was done by a friend of mine, Michelene Chi.

Always seems to go by the name Mickey Chi.

There he is.

So here's the deal.

The students that she worked with were expected to learn about elementary physics.

801 type stuff.

And she took eight subjects, and she had them-- she took them through a bunch of examples and then she gave them an examination.

So eight subjects, and so they divide into two groups.

The bottom half and the top half.

The ones who did better than average and the ones who did worse than average.

So then you can say, well, OK, what did that mean?

You can say, how much did they talk to themselves?

Well, that was measured by having them talk out loud as they solved the problems on an examination.

So we could ask how much self explanation was done by the smart ones versus the less smart ones?

And here are the results.

The worst ones-- the worst four said about 10 things to themselves.

The best four said about 35 things to themselves.

That's a pretty dramatic difference.

Here's the data in a more straightforward form.

This, by the way, points out that the smart ones scored twice as high as the less smart ones.

And when we look at the number of explanations they gave themselves in two categories, smart ones said three times as much stuff to themselves as the less smart ones.

So, as you can see, the explanations break down into two groups.

Some have to do with monitoring and not with physics at all.

They're things like, oh hell, I'm stuck.

Or, I don't know what to do.

And the others have to do with physics.

Things like, well, maybe I should draw a force diagram.

Or let me write down f equals ma , or something like that, as physics knowledge.

I think it's interesting that this average score is different by a factor of two, and the average talking to oneself differed by a factor of three.

Now this isn't quite there, because what's not clear is if you encourage somebody to talk to themselves, and they talk to themselves more than they would have ordinarily, does that make them score better?

All we know is that the ones who talk to themselves more do score better.

But anecdotally, talking to some veterans of 6.034, they've started talking to themselves more when they solve

problems, and they think that it makes them smarter.

Now I would caution you not to do this too much in public.

Because people can get the wrong idea if you talk to yourself too much.

But it does seem-- it does, in fact, seem to help.

Now what I did last time is I told you how to be a good scientist.

What I'm telling you now is how to make yourself smarter.

And I want to conclude this hour by telling you about how you can package your ideas so that they have greater impact.

So I guess I could have said, how to make yourself more famous, but I've limited myself to saying how to package your ideas better.

And the reason you want to package your ideas better is because if you package your ideas better than the next slug, then you're going to get the faculty position and they're not.

If you say to me, I'm going to be an entrepreneur, same thing.

You're going to get the venture capitalist money and the next slug won't if you package your ideas better.

So this little piece of work on the arch business got a whole lot more famous than I ever expected.

I did it when I was young and stupid, and didn't have any idea what qualities might emerge from a piece of work that would make it well known.

I only figured it out much later.

But in retrospect, it has five qualities that you can think about when you're deciding whether your packaging of your idea is in a form that will lead to that idea becoming well known.

And since there are five of them, it's convenient to put them all on the points of a star like so.

So quality number one.

I've made these all into s-words just to make them easier to remember.

Quality number one is that there's some kind of symbol associated with a work.

Some kind of visual handle that people will use to remember your idea.

So what's the visual symbol here?

Well, that's astonishingly easy to figure out, right?

That's the arch.

For years without my intending it, this was called arch learning.

So you need a symbol.

Then you also need a slogan.

That's a kind of verbal handle.

It doesn't explain the idea, but it's enough of a handle to, as Minsky would say, put you back in the mental state you were in when you understood the idea in the first place.

So what is the slogan for this work?

Anybody have any ideas?

Pretty obvious.

What's essential to this process working?

The ability to present an example is very similar [INAUDIBLE], that constitutes a model but isn't one of those.

STUDENT: [INAUDIBLE].

PROFESSOR PATRICK WINSTON: So it's a near miss.

The next thing you need if your work is going to become well known is a surprise.

What's the surprise with this stuff?

Well, the surprise-- everything that had been done in artificial intelligence having to do with learning before this time was precursors to neural nets.

Thousands of examples to learn anything.

So the big surprise was that it was possible for a machine to learn something definite from each of the examples.

So that now goes by the name of one shot learning.

That was the surprise, that a computer could learn something definite from a single example.

So let's see.

We've almost completed our star.

But there are more points on it.

So this point is the salient.

What's a salient-- what's a salient idea?

Jose, do you know what a salient idea is?

He's too shy to tell me.

What's a salient idea?

Ah, who said important?

Wrong answer, but very good.

You're not shy.

So what does it really mean?

Yes.

STUDENT: Relative to what somebody's already thinking about?

PROFESSOR PATRICK WINSTON: Relative to what somebody's thinking about.

Not quite.

If you have a-- if you're an expert in-- yes?

STUDENT: [INAUDIBLE].

PROFESSOR PATRICK WINSTON: Really close.

We're getting closer.

[INAUDIBLE].

Yes?

STUDENT: Maybe an idea that wasn't obviously apparent, but becomes apparent gradually as somebody starts to understand?

PROFESSOR PATRICK WINSTON: We're zeroing-- we're circling the wagons here and zeroing in on it.

Yes?

STUDENT: If I'm preempting what you're about to say, it has sort of a doorway of how you can understand the idea.

PROFESSOR PATRICK WINSTON: It's what?

Sorry.

STUDENT: It's sort of like a doorway of how you can grasp the idea.

PROFESSOR PATRICK WINSTON: That's sort if it, too, but if you study military history, what's the salient on a fort?

Well, this is a good word to have in your vocabulary because it sort of means all of those things, but what it really means is something that sticks out.

So on a fort, if this were a fort, these would all be salients because they stick out.

So the salient idea is usually important because it sticks out.

But it's not-- the meaning is not "important," the meaning is "stick out." So a piece of work becomes more famous if it has something that sticks out.

It's interesting.

There are theses that have been written at MIT that have too many good ideas.

And how can have too many good ideas?

Well, you can have too many good ideas if no one idea rises above and becomes the idea that people think about

when they think about you.

We have people on the faculty who would have been more famous if their theses had fewer ideas.

It's amazing.

So this piece of work did have a salient.

And the salient idea was that you could get one shot learning via the use of near misses.

That was the salient idea.

The fifth thing, ah.

Talk more about this in my "How to Speak" lecture in January.

The fifth thing I like people to try to incorporate into their presentations is a story.

Because we humans somehow love stories.

We love people to tell us stories.

We love things to be packaged in stories.

And believe me, I think all of education is essentially about storytelling and story understanding.

So if you want your idea to be sold to the venture capitalist, if you want to get the faculty job, if you want to get your book sold to a publisher, if you want to sell something to a customer, ask yourself if your presentation has these qualities in it.

And if it has all of those things, it's a lot more likely to be effective than it doesn't.

And you'll end up being famous.

Now you say to me, well, being famous-- that sounds like the Sloan School type of concept.

Isn't it immoral to want to be famous?

Maybe that's a decision you can make.

But whenever I think about the question, I somehow think of the idea that your ideas are like your children.

You want to be sure that they have the best life possible.

So if they're not packaged well, they won't.

I'm also reminded of an evening I spent at a soiree with Julia Child.

Julia, and there's me.

And I have no idea how come I got to sit next to Julia Child.

I think they thought I was one of the rich Winstons.

The Winston flowers, or the Harry Winston diamonds or something like that.

There I was, sitting next to Julia Child.

And the interesting thing-- by the way, did you notice I'm now telling a story?

The interesting thing about this experience was that there was a constant flow of people-- happened to be all women-- people going past Ms. Child saying how wonderful she was to have made such an enormous change in their life.

Must have been 10 of them.

It was amazing.

Just steady flow.

So eventually I leaned over to her and I said, Ms. Child, is it fun to be famous?

And she thought about it a second and said, you get used to it.

And that had a profound effect on me, because you always say, well, what's the opposite like?

Is it fun to be ignored?

And the answer is, no, it's not much fun to be ignored.

So yeah, it's something you can get used to, but you can never get used to having your stuff ignored, especially if it's good stuff.

So that's why I commend to you this business about packaging ideas.

And now you see that 6034 is not just about AI.

It's about how to do good science.

It's how to make yourself smarter, and how to make yourself more famous.