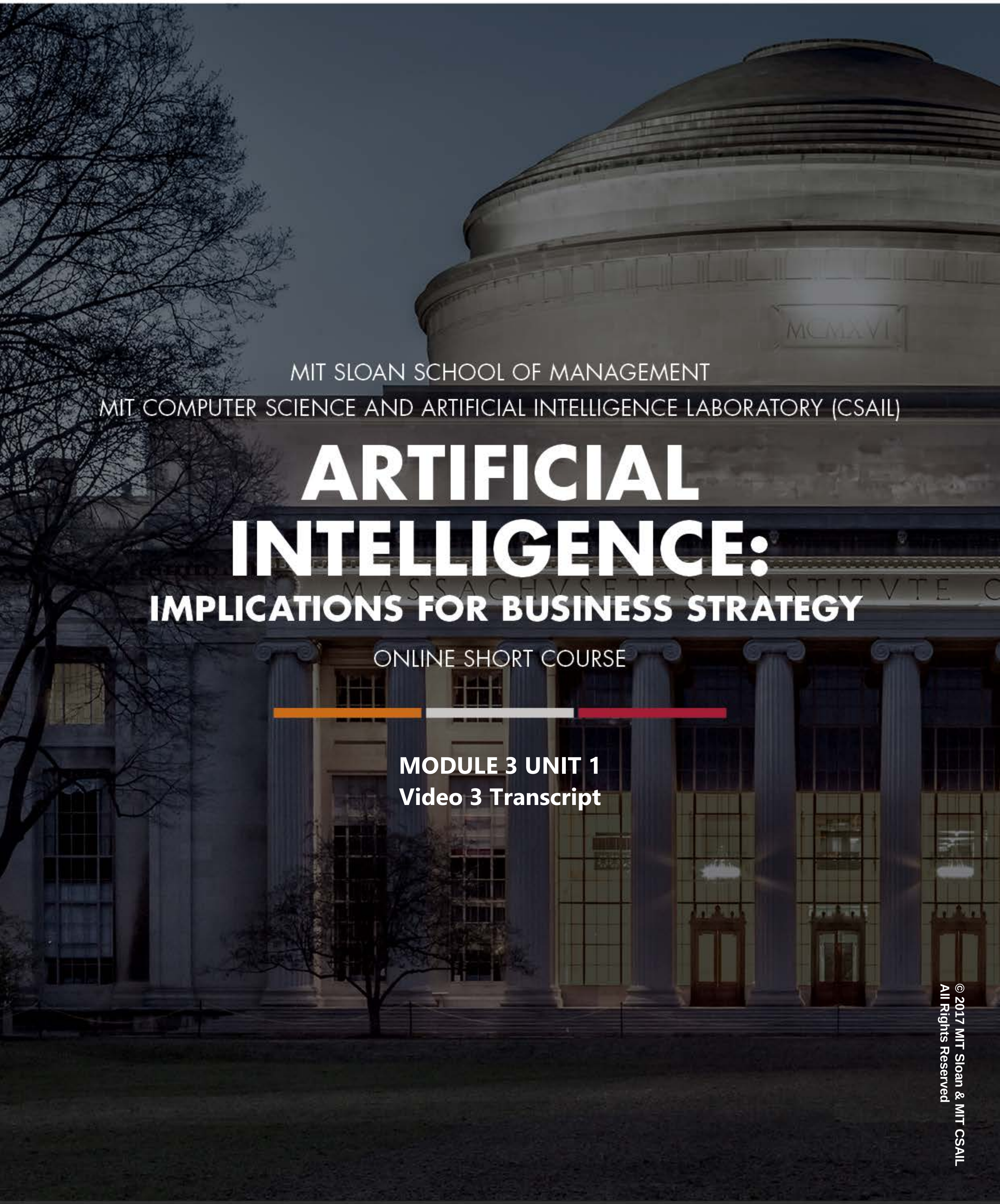

MIT SLOAN SCHOOL OF MANAGEMENT

MIT COMPUTER SCIENCE AND ARTIFICIAL INTELLIGENCE LABORATORY (CSAIL)

ARTIFICIAL INTELLIGENCE:

IMPLICATIONS FOR BUSINESS STRATEGY

ONLINE SHORT COURSE

MODULE 3 UNIT 1
Video 3 Transcript

DELIVERED IN COLLABORATION WITH getsmarter

Module 3 Unit 1 Video 3 Transcript

REGINA BARZILAY: So how can we make machines which can understand language? So first of all, the machines have to understand the grammar in the language itself, and it also has to understand something about the world to which it tries to interpret the language. And there are two possible approaches that we can think about. One approach is what, you know, we will do with a child. You see the person, or the computer really has to understand all this information, so we need to take all our knowledge and encode it in the machine, and then it can try to understand it. And another one is a statistical approach where we would say we are going to give to the computer lots and lots of language samples and try and use machine to learn and to find the pattern.

And it's very interesting from the scientific viewpoint how this process developed. Noam Chomsky, who is a professor of linguistics, and one of the fathers of this field at MIT, in '57 wrote this famous passage about, "Colorless green ideas sleep furiously", versus "Furiously sleep ideas green colorless". Both of the sentences don't make any sense, but one of the sentences is grammatical, and the other one is not. And then he was saying any statistical approach will be unable to distinguish between the two because it has never seen those sentences. So, this was a concern that he brought up.

I would say that his very strong opposition to the statistical method managed to stop all the empirical approaches for decades in natural language processing. Then there was something very interesting that happened in IBM, when, in the early '90s, they actually demonstrated that using a statistical technique, you would be able to do the first robust speech recognition system. And the person who did, who led this development, Fred Jelinek, who is a graduate of our own ECS department at MIT, wrote this famous phrase, "Whenever I fire a linguist our system performance improves".

So, there was this huge tension between these two different techniques, but what happened is, as I said, for a very long time, due to Noam Chomsky and many others, believed that statistical approaches would not work. And in the '70s and '80s, what people tried to do is to take small domains and to build a system which has full understanding of those domains. And one of these very famous systems called SHRDLU was developed at MIT. And what they saw – it was really '70s, and think about it, at the time the computers were less powerful than your watches, by far less powerful than your watches. They tried to encode everything about the cube, so that the task was to give some cubes – and you can see an example on the slide – and to manipulate, to move them around. So, you would encode where exactly they are located, what is their color, shape, and so on and so forth. And their hope was by doing that, you would put all the knowledge to the machine, and the machine would be really able to make interesting assertions. And what they found out, that even for the tiny domain, you need to encode so much that it really cannot be expanded to anything besides these few blocks.

Then what happened was, you know, the field was kind of going into decline, people couldn't really generate anything exciting, and then in the '90s, the big revolution happened. DARPA, which is an agency that funds science, generated a tree bank, where they took sentences, and for every sentence they had a syntactic tree associated with it, and there were, like, millions of these sentences. And then they said, I don't want grammar, I'll just teach the machine to learn the mapping from the sentence to the tree. And this was the first time that people were able to generate very robust parses that can actually do something. This was very exciting, and people were like, I don't need to do linguistics, I don't need any knowledge, I'm just going to give a lot of statistics and a lot of corpora.

And let me just demonstrate to you, in a very simple example, how this approach can be working. So, let's go with a task which is difficult for me as a non-native speaker, which is determiner placing, you need to decide where to put “a” and “the”. So here, on the slide, you can see a text where I removed all the determiners. And what we would ask the machine to do is to place, for all the eliminated “a” and “the” and so on, the correct answers. So, you can say, one way to do it, if we are believers in the symbolic approach, is let's go to the dictionary and try to extract, you know, the rules, and look at the grammar books. And you would say some of them make sense, like type of noun, whether it's countable or not countable. What is a number? Is it single or plural? You can write some of those rules.

But very soon you will discover in grammar books that there are lots and lots of obscure rules, and one of them I'll show you here. The definite article is used with newspaper titles, like *The Times*, but zero article in names of magazines and journals, like *Time*. So, you can really see that in order to incorporate these kind of rules you need to have a lot, a lot of information, and it will take us forever to build it.

So, let me propose to you a different approach. You have seen supervised classifiers. So, we will treat this problem as a classification problem. Let's say, I will make it simple and say you just want to determine whether the noun phrase needs an article or doesn't need an article. Okay, so, say we have just two outcomes, those are our predictions, and then you can put whatever questions you desire to put as features for this classifier. So, for instance, you can ask, is this noun plural? Does it appear for the first time in the text? What is the head token of it? So now you will take every noun phrase and translate it into a feature vector. So, what is a feature vector? As you've seen, it's just an array of questions. So, for instance, for the phrase, “lazy monkeys”, it happened to be plural, that's why we have in the first place, one, and whether it is the first appearance in text, yes, we put one, and so on. So, you can really translate every occurrence into a feature vector, and then you know the label, minus one or plus one. And again, to remind you what you've already seen in the previous machine learning lecture, we can now think that every noun phrase is actually a point in, some high dimensional space, and your goal is to find a separator which divides it into two parts. And now if somebody gives you a new occurrence that you haven't seen before, you would use this classifier to decide whether it, requires a determiner or not. And as you can see, that actually, models that do it today do it very well, with a very high accuracy. Better than humans – humans like myself who are non-native speakers, obviously.

Let me show you another example, which is more practical, where you can use the same approach to solve a problem which has a lot of practical significance related to understanding clinical records. So, one of the big issues in utilizing clinical records for big data is how to take data which is hidden in free text and build a database from it. And today in most hospitals in the United States, this task is done by people at different levels of medical education, starting from just entry people to the clinical fellows, who sit and really enter this information by hand. And the question is, can we automate it?

So, let me show you a concrete example from a system that we built for Massachusetts General Hospital. So, the question that we had to address is to take a pathology report, which people write after a biopsy was taken, and extract all this different information. For instance, to say if it has cancer, and what kind of cancer. So, you can see that in pathology reports, they are so systematic that you can just write a few rules and you don't need to do any machine learning. But it turns out – and this is actually a slide from a doctor who tried to do it by hand – that even one technical notion of, like, globular carcinoma in situ, it can be stated in at least 22 different ways, and you can negate it in various fashions. So, the point is, when you start writing these rules, it never converges. However, we can take exactly the same approach as we've taken with determiner placement, take every document represented again, as a feature vector, and here's how we can do it. We can decide, let's say we are looking at all the words that appear in medical vocabulary, and we will ask, does this word appear in the document or not? So, for instance, for all the documents where the word "density" appears, the first coordinate of the vector would be "1". And we would ask whether the word "finding" appears, and if it didn't appear we are going to put "0". So now we have a process which takes a document, translates it into a vector, and the machine knows whether it has a particular diagnosis or it doesn't. And then the machine can learn the pattern of how to translate these vectors into plus or minus decisions, and again, we are going to be using our classifier to make this decision for us. And in this case, using this very robust technology, we were able to create a database of 50,000 breast cancer patients for Massachusetts General Hospital and other hospitals in the area, with the accuracy of 96%. So, you can see the case where very simple, robust NLP technology really delivered very good performance.

The last part of my talk will be dedicated to the very modern and strong techniques that are currently dominating natural language processing related to deep learning. So, before we move there, let me just give you a motivation of why it can make a huge difference. So, if you look at the representations that they currently describe, you know, every word has its location, so we are counting how many times a certain word appears. And in order to get a representative sample, you need to have lots and lots of sentences. And here we can see an example of the graphs where we demonstrate how many unseen events you observe, given the number of sentences. And clearly the more sentences we have, the more things we see in these sentences. And the problem of this technology is that when we are mapping every word as an independent event, the machine doesn't see that there is a correlation between synonyms. So, for instance, the words "pear" and "apple" are really distinct from each other, and the similarity between "pear" and "apple" will be exactly the same as "pear" and "dog", which really forces the machine to see enough examples with pears and enough examples with apples.

So instead, what we ideally would want to do is to take these very sparse vectors that represent words, and to map them into some very dense, low-dimensional representation, in such a way that the words which have similar meaning are really close to each other as vectors. And let me show you the automatically built representation of words in 2D projection, where we kind of take a very high dimensional space and look at their projection in 2D. So, what you can see here, is that the words which are semantically similar, like "academic", "faculty", "graduate", "university", are very close to each other. The ability to generate this kind of vector representation is what deep learning delivered for us.

And you don't just have to stop on individual words. You can say, I want to have, as a vector, a representation of the whole sentence so that I can reason about the meaning of the sentences. There are many possibilities here, you can take just individual words and sum them together, by their vectors. You can concatenate them. You can try to learn hierarchical representation – this is currently an area of very active research in natural language processing. And what's interesting about it, is that this technique totally enables us to rethink traditional approaches. So, for instance, the way, today, we do machine translation, we take a sentence, translate it into a vector, and then from that vector we generate a sentence in a different language. So, we are thinking about this process as sequence-to-sequence mapping. And as you can see, this is my slide which shows the performance across different tasks, with and without deep learning. You can see there is significant improvement across the board, across all the tasks. And I would like to conclude by saying, this technique really opened the door for making a revolution in natural language processing, and I believe many of these tasks that Turing was talking about years ago will be something we can achieve in the very near future.

THOMAS MALONE: Did you understand all the concepts covered in this video? If you'd like to go over any of the sections again, please click on the relevant button.