Applied Natural Language Processing
Prof. Ramaseshan Ramachandran
Department of Computer Science and Engineering
Chennai Mathematical Institute, Madras

Lecture - 75
Encoder-Decoder Model for Neural Machine Translation

Given the background of what we have done so far right, I am going to be talking about the Neural Machine Translation in this particular session. So, earlier we had seen how sentences can be translated from one language to the other using this statistical model right. Especially, we saw models in terms of IBM models, we actually spoke about two models. One is IBM model 1, the second one is IBM model 2 and then both had given a very interesting perspective with respect to translation; model 2 really provided us in terms of alignments right.

So, how do you really align words from once and one foreign sentence to the native sentence that you have in mind right? So, it provides some alignment models and then we used a noisy channel model to really break that into two distribution. One is the language model, the second one is the translation model. In the translation model we actually added more nuances in both models 1 and 2, models 3, 4 and 5 actually added more nuances to that.

And, then later we also spoke about the phrase-based models where we try to align phrases and not just words. The reason for the better translation is we started attaching the context, you know we not just looking at only one word, one-word translation; we are now looking at a context. So, that context really provided some meaningful translations ok. So, until 2014 many systems used phrase-based models. And, then when they evaluate the system they used a blue meteor, NIS based model, WR WER all those evaluation matrices to find out how close their translation is with the human translation ok.

So, now we jump into the next phase of translation, where neural machines are coming into play right. Why do you think neural machines would do a better job? You remember again in the case of a word embedding model given three inputs or four inputs in the (Refer Time: 02:47) model right.

The neural net is able to really relate them together because the context is available. So, it is able to really relate one single word with multiple words in the output or multiple words in the input, one single word in the output layer. So, it is all about identifying the patterns in those and then how do we really align those patterns. So, that when a new input is given which is close to the pattern that it has found, it reasonably provides a good output.

So, can we extend the same thing to the neural machine translation? I remember even in the RNN model we provided a sequence of input right. The sequence of input we provided and then there is an outcome that comes out of that which is nothing, but the encoding of that sequence right. For example, I can encode a variable in the sentence into one vector at the outcome. I can use the same sequence to train the network to recognize the words that are coming as next or I can simply put it, I can call it as a language model.

So, it helps you in terms of training a language model or you can also use the same sequence mechanism to find the spellings. Supposing, if you had given a large corpus and then make the system learn the large corpus with respect to the English word and in that, you are only giving one character at a time. So, when you provide the characters in characters to the RNN has a sequence of letters or sequences of events, it learned the spellings as well based on the corpus that you have provided. So now, we want to extend this further, again you should understand that in the progression that we have been seeing so far.

We started with understanding the frequency of words and starting at the beginning ok. We have that term frequency and then the ideas that we have inverse document frequency and then we combined them and then later we have used an LSA model to find out the association of the words that are coming in in the term-document matrix.

And, then later we move from the words alone into the language elements of that, we try to create a language model by using the corpus or the statistics that are available as part of the corpus right. And, then from the words and then the group of words we started to understand the context by giving more input words to the models, either a trigram language model.

When you go to the neuron that side of it, you had given sea bow where you give more words, and then one word is obtained at the output or in the script graham you give one word and then it identifies the context word surrounding that and so on. So, we are able to go from the word to some extent to identify the context of the words and also to some extent or to identify the meanings of those words that occur in the context right.

So, next we extended that in the RNN model, where we were able to encode a sentence correct. So, the next stage is after you encoded a sentence, is it possible for me to use the encoded sentence to do the translation. So, this is the progression that we have been making correct, the translation is the next level of intelligence in Bloom's taxonomy. I am not sure whether you still remember Bloom's taxonomy part that we studied earlier, where we started with the knowledge recall and then application analysis, and then later we said the creation part of that right.

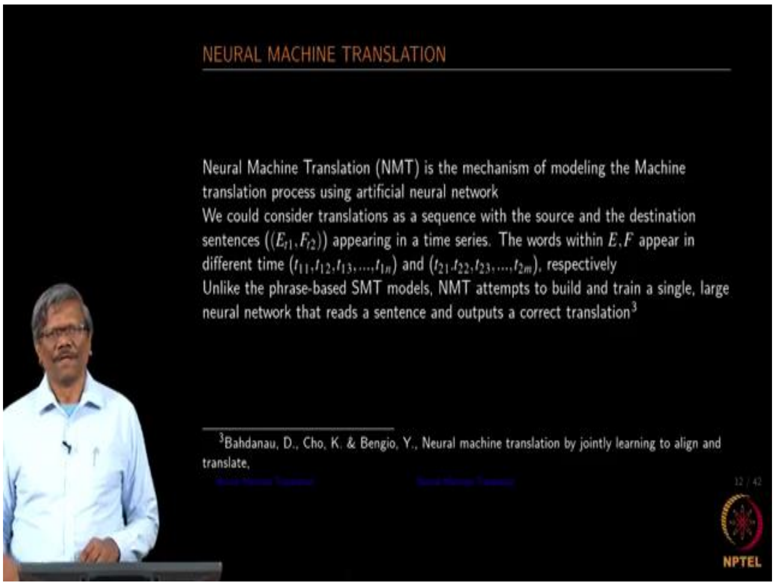
So, slowly and steadily we are trying to provide some intelligence to the machines so, that it moves up in the pyramid part of Bloom's taxonomy as well. So, when you look at the translation part, it is considered to be a human intelligence task correct. So, we want to see whether the machine is able to really create something to evaluate that creation and then say that what I have created is good enough for you to take it forward.

So, that is where the higher levels of intelligence coming into play correctly. So, we want to see whether the translation models are really getting into the higher levels of inclusion, which is actually used to really figure out how these students are learning the subjects ok. I am also sure that we will know that many of the question papers that you had seen so far, in the exams have followed some patterns of that, there will be some recall right.

So, you just have to recall the answer, in some cases, you have to use your knowledge and some ideas and then apply it to solve certain problems. And, then you are asked to create or synthesize something that is where the creation starts right and then finally, you innovate a few things. For example, the (Refer Time: 08:40) writing part in English you know you are really doing some innovative writing there, after looking at the phrases or the paragraphs given to you. So, the progression is very interesting even in the case of natural language processing; especially when you start with the data modeler.

I am sure you have noticed that I have not gone into the syntax elements of the language. We have been only talking about the data part of that all the time and we moved from the probability to the neural elements because of a lot of interesting pattern matching that is taking place in the neural net world. So, coming back to this part of neural translation we are going to be using neural networks to really find out whether it can do a good job in the translation or not. So, it all started in 2014, we are not very far from the current world of translation alright.

(Refer Slide Time: 09:51)



NEURAL MACHINE TRANSLATION

Neural Machine Translation (NMT) is the mechanism of modeling the Machine translation process using artificial neural network

We could consider translations as a sequence with the source and the destination sentences $((E_{t1}, F_{t2}))$ appearing in a time series. The words within E, F appear in different time $(t_{11}, t_{12}, t_{13}, ..., t_{1n})$ and $(t_{21}, t_{22}, t_{23}, ..., t_{2m})$, respectively

Unlike the phrase-based SMT models, NMT attempts to build and train a single, large neural network that reads a sentence and outputs a correct translation[3]

[3] Bahdanau, D., Cho, K. & Bengio, Y., Neural machine translation by jointly learning to align and translate,

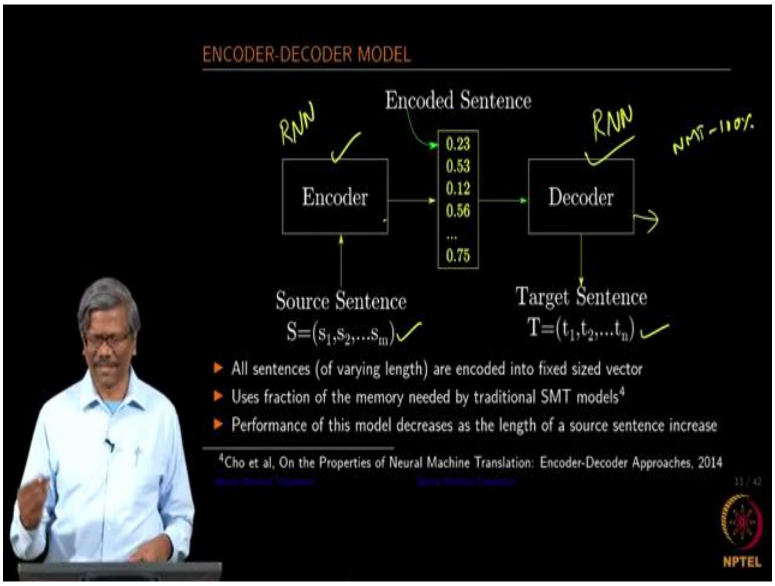
12 / 42

So, given the introduction let me jump in, I think I gave some ideas about this. So, we are going to be using two sequences: one is the native language sequence and then there is a translated sequence on the other side. So, you can in this case it can be interchanged ok. So, I can give English as one sentence and we have a box which we call it the neural machine translation black box and it will output the foreign sentence or you feed in the foreign sentence, it outputs the English sentence or any other language a sentence ok.

So, in this case there are going to be two sequences: one is the English sequence, we will use the convention and the second one is the foreign sequence. And, we are going to be using the same RNN model for the translation, we know that RNN is meant for a sequence modeling correct or sequence learning. So, in this case we are going to be feeding two sequences at different times and then when this is fed ok, we expect in a different time slice and output of a different language.

So, in this case we are not going to be doing any complex mechanism that we had done in the phrase base model or in the standard statistical machine translation models. We going to build a very simple single large neural network that reads one sentence and outputs the correct translation ok; so, this is the idea ok.

(Refer Slide Time: 11:45)



ENCODER-DECODER MODEL

$S=(s_1,s_2,\ldots s_m)$ ✓ $T=(t_1,t_2,\ldots t_n)$ ✓

- All sentences (of varying length) are encoded into fixed sized vector
- Uses fraction of the memory needed by traditional SMT models[4]
- Performance of this model decreases as the length of a source sentence increase

[4] Cho et al, On the Properties of Neural Machine Translation: Encoder-Decoder Approaches, 2014

11 / 42

Let us jump in into the next assuming in part right. So, we have seen from the very top level what neural machine translation is given a sentence in a foreign language it is going to output an English sentence ok. So, the next part is we are going to have two things here, one is an encoder another one is a decoder. It is very similar to what we had seen in the noisy channel also, there are encoder and decoder. So, in this case we do not need to really compare that there, the encoder is an RNN our decoder also is an RNN because, both say take a sequence of sentences right ok.

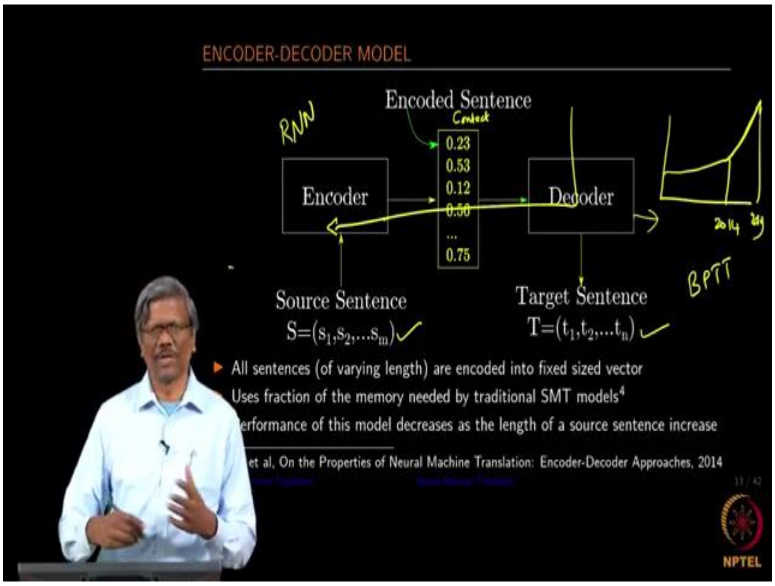
So, we have a target source sentence target sentence, advantage of this again is you can give variable, the sentences of variable length ok. So, that is the advantage and then again these sentence lengths when you have in this source right, need not be the same as what is coming out in the decoder as well. It would be smaller, shorter maybe the encoder will have a 5-word sentence, the decoder may provide you about 7 words or 8 words or 3-word sentence and so on.

It depends on how we provide this sentence parts during the training and it uses a fraction of memory needed. You remember when we spoke about the phrase base model and then I also showed a table the size of the phrases, that are required correct.

It runs a few GB’s and it is not really very efficient in terms of using the machine. So, RNN’s use a very small memory footprint ok. We have to say that the model is not going to be doing very well when the sentence is long. Why? It is the same reason that we mentioned earlier with respect to vanishing gradients right. So, when the sentence is long, it is not able to remember everything very clearly ok. So, the performance of this system would degrade soon ok.

So, it is not that we have cracked the entire translation problem, we are making progress and doing better than what we have done earlier. So, that is what I should tell you at this point in time. It is not that every translation system, the neural net side provides you 100 percent translation. It is reasonably better than what we had seen in the phrase-based models of SMT ok. It is still in progress, the research is still in progress; we still have not succeeded in every area of translation.

(Refer Slide Time: 15:04)



ENCODER-DECODER MODEL

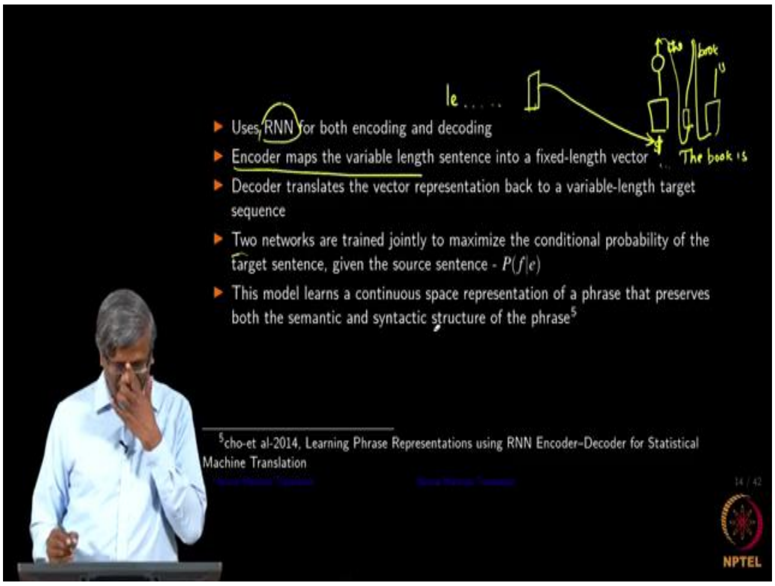
- All sentences (of varying length) are encoded into fixed sized vector
- Uses fraction of the memory needed by traditional SMT models[4]
- erformance of this model decreases as the length of a source sentence increase

et al, On the Properties of Neural Machine Translation: Encoder-Decoder Approaches, 2014

11 / 42

But these are the fundamental elements that are now taking the translations from where we are actually you know the translation was going in the speed and then suddenly it moved up from 2014. So, this speed with respect to the research that is going on in their machine translation world is fascinating and a lot of excellent results are coming out as well in this alright.

(Refer Slide Time: 15:35)



- Uses RNN for both encoding and decoding
- Encoder maps the variable length sentence into a fixed-length vector
- Decoder translates the vector representation back to a variable-length target sequence
- Two networks are trained jointly to maximize the conditional probability of the target sentence, given the source sentence - $P(f|e)$
- This model learns a continuous space representation of a phrase that preserves both the semantic and syntactic structure of the phrase[5]

[5] cho-et al-2014, Learning Phrase Representations using RNN Encoder–Decoder for Statistical Machine Translation

14 / 42

So, as I mentioned earlier there are going to be two RNN that we would use: one for encoding another one for decoding. So, let me go back to the previous slide, I think I have not spoken about this part. So, what we have here is there is an encoder that encodes the source sentence and then provides you this is I think you know right, it gives you a vector.

So, we call it a context vector. So, it encodes all these words in terms of a vector, in terms of some numbers and then this is fed into the decoder. And, then decoder uses this input along with its initial state and then start decoding word my word and the training happens in this fashion; let me erase some of this.

So, earlier in the case of RNN when you have right when you start reading the sentences; it happened within this encoder part right in one single box. So now, the training has to happen because we have added the decoder part and the entire chain has to be trained; that means, we are going to be having a lot of parameters to update alright during the training process. Again you can use the same backpropagation through time to do the job alright. So, remember this is a very simple set of several matrix manipulations and vector manipulations alright; nothing more, nothing less. This encoder RNN could be your LSDM, GRU or whatnot, the cell could change as well.

There are so, many combinations that you can bring in because of that alright; continuing on this part. Encoder maps the variable-length sentence into a fixed-length vector, as I mentioned that the output is going to be a fixed-sized vector. Decoder translates the vector representation back to a variable-length target sequence. Again when the decoder receives the context of vector value, it inputs that as part of the first value along with the hidden values that it has and then it starts providing the first word. For example, this is the starting of this sentence and I have the RNN, after softmax it provides some value right.

And, let us say the is the or the is the first word that needs to be output from the decoder ok, encoder has a French sentence. Let us say le is the first word and then after encoding all the words or we have a context vector and then this context vector is fed in here as part of this. And, then since this sentence is known to us ok, let us say the book is on the table is this and we know that this is the book; should be output and then this is going to be outputting is and so on; you know how the progression goes right.

Sorry about the messy stuff, but I think you understand this. So, from the encoder after incorporating the context vector, the output of the first decoder should be correct. So, it knows that there has to be the first one, and then it outputs that and then makes the changes, you can have a small backpropagation in this to adjust this or you can do it till the end. So, there are mechanisms as I mentioned you can have several learning mechanisms combined into this as well ok. These two networks are trained jointly.

So, as I mentioned so, it goes from the decoder to encoder and then encoder to the decoder and the process goes on until the network stabilizes right. So, what essentially; that means, it is again the same right, the conditional probability that we talked about. Given the English sentence, supposing if you are inputting into the encoder English sentence we are expecting f as the outcome. So, the conditional probability is what we are estimating here ok, given e we are providing the probability of this particular sentence in French right. It says that is what we are trying to estimate at the end of this.

So, it learns a continuous space representation of a phrase that preserves both semantics and syntactic structure. Since we are inputting these sequence rights, we are not losing anything here, we are inputting sequences of words. So, we are not providing only a bigram or trigram, we are providing the entire sequence. So that means, the entire semantics also is passed along with this and the network learns in both cases for both in the encoder box as well as on the decoder box ok. So, it learns continuously those phrases in the continuous phase.