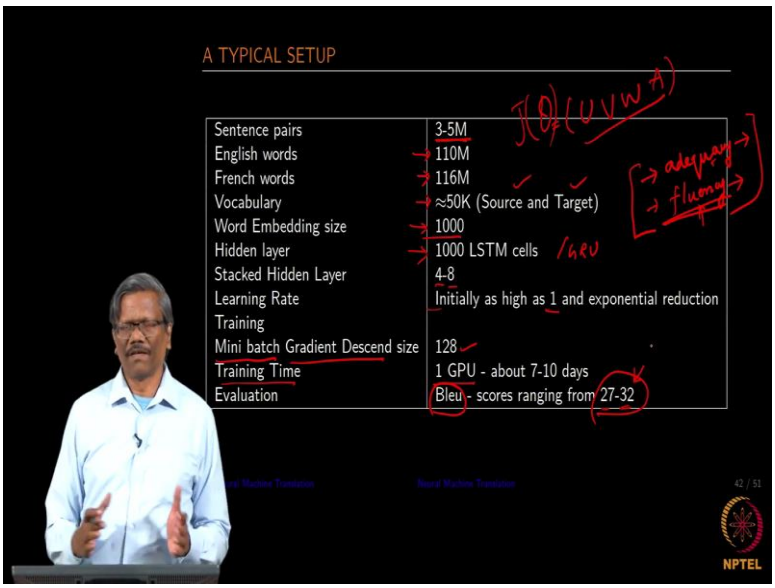
Applied Natural Language Processing
Prof. Ramaseshan Ramachandran
Department of Computer Science and Engineering
Chennai Mathematical Institute, Madras

Lecture - 80
Typical NMT architecture and models for multi-language translation

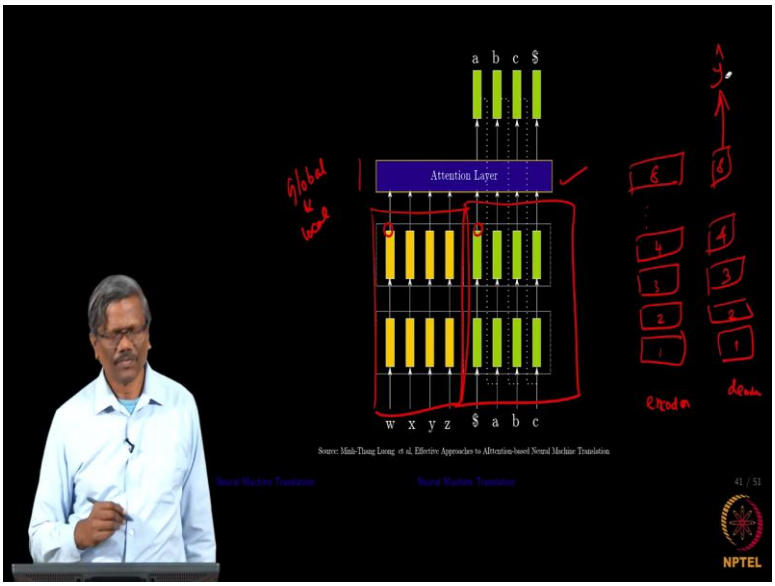
(Refer Slide Time: 00:15)



So, how do we do this training? They have been talking about at a very high level the architecture and so on, but we are not talking about; what they really contain? What we feel as input and so on. So, in this I am going to be talking about a typical set up in the translation site, where we are going to be having 3 to 5 million pairs of sentences in the corpora right. And then a typical corpus parallel corpora contains about 110 million English words, 116 million French words, in this case, it is going to be English and French.

And then the vocabulary size is pretty close to 50K both source and target ok. And then as I mentioned earlier the embedded words are fed into the encoder and then the hidden layer could contain 1000 LSTM cells or GRU cells and then we can have hidden layers from 4 to 8. So, we just saw one or two layers right in the earlier cases. So, we can have more of this.

(Refer Slide Time: 01:35)



So, we can have 2, 4 in the same fashion, this is your encoder the same way you stack them upright. So, it is the number of cells that are going to be huge there right the number of weights is going to increase thereto. There are multiple ways of providing learning rates. So, some will keep it fix it, some will start with a high value to have the learning faster and then it exponentially reduces the value so, that is one way of doing.

And then we will also have different training models; one is the mini-batch you will find a lot of mini-batch models in the machine translation ok. So, we use a gradient descent using mini-batch we will talk about that in the water slide. The training time if you use 1 GPU it is about 7 to 10 days for this set of English and French powers of sentences and then the evaluation score is obtained using blow, we learned about it earlier and the scores are usually in the range of 27 to 32 ok.

So, this is what you will find in most of the papers, the range is in the values range from 27 to 32; 32 is really good you know you remember we had something called adequacy right. So, what is adequacy? Adequacy is that you know this is just sufficient or two for the translation ok. If you see a sentence and it is just translating the meaning of the input sentence you know in some way that is good enough for me and later we also look we were also looking at the fluency of the sentences right.

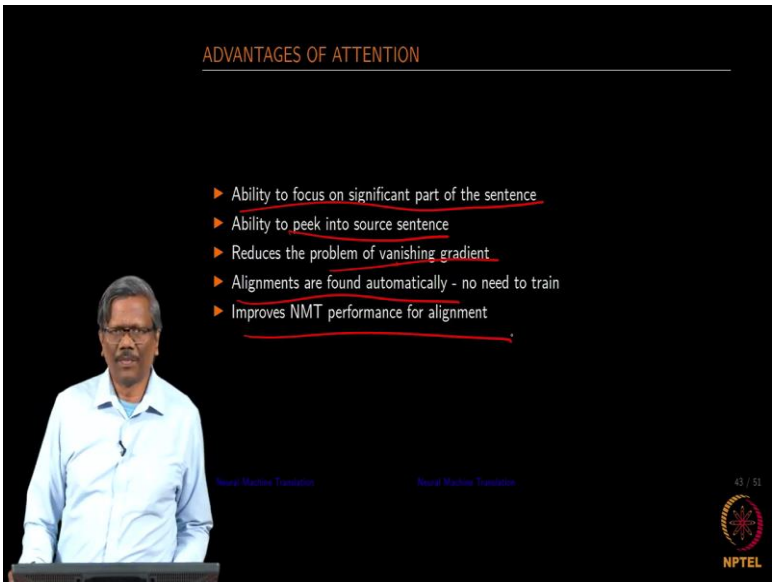
So, this one could be a broken English type of translation, this n could have some really syntactically formed sentences coming in from the translation as well. So, both are

important if we get to this level, we already have crossed adequacy part right. So, we want to make sure that, we are in this space when the translation happens and most of the translation in this range has the value of 32 or sometimes 30 sometimes 29 and so on all right.

So, this is a setup that we have and then there are certain parameters that we call as hyperparameters like in terms of what is the mini-batch size, the learning parameter size and so on and there are parameters that are learned in the training process like you know the theta parameters that contains our U V and W and also our alpha now right.

So, these are the parameters that are learned during the training process and we use a mini-batch or a batch or a stochastic gradient descent algorithm for this and we require really good machines to do the translation ok. So, we have a good set of GPUs in your machine that is very good to start the translation process using neural networks.

(Refer Slide Time: 06:17)



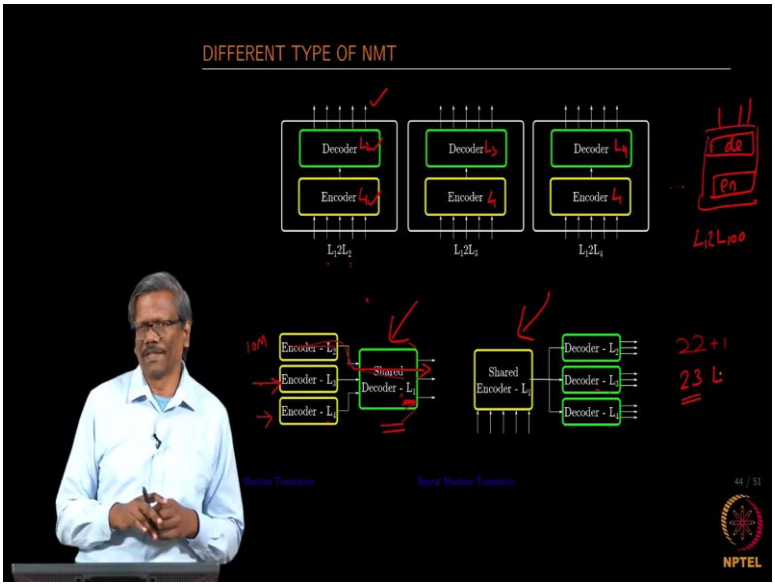
So, what are the advantages? I think we saw all of this. The ability to focus on the significant part of the sentence, the ability to peek into the source sentence, or to really align the target words or the phrases into these source sentences for all the neural networks when you use LSTM or GRU you do not have this issue ok.

And also it depends on how well you start the input process right in terms of providing the input values to the hidden units and so on. Alignments are found automatically there

is no need for you to train this. And the performance of the NMT increases because we are including the alignment process, alignment of phrases into this as part of the training as well.

So, if you use the vanilla training model where we did not have any of the alignment problems. The performance is a lot lower than some of the statistical machine translation based on phrase-based models. So, now, the attention-based models are comparable with the statistical model as well in terms of alignment ok.

(Refer Slide Time: 07:51)



So, now we look at different types of NMTs. So, here this is what we saw right. So, there is an encoder, there is a decoder for a pair of language phrase the language 1 and language 2 you are going to be translating from language 1 to language 2 so, L 1 L 2 right so, L 1, L 3, L 1, L 4. Suppose if you have about 100 languages and you have this standard encoder, you will have 100 of this right. So, we need to have so, many of them.

So, what had happened is. So, when you have the same target language, you do not need to have a separate training process for all the language pairs right. So, you can just take one encoder language 2, language 3 which has the same one because the target is the same right, the encoding part only is different. And then you can also have a shared encoder; suppose you want to translate from.

We can also have a shared encoder if you want to translate from one language to multiple languages or you can have several decoders ok, but you can have only one encoder right. So, how do we train this process? So, when you saw the one assumption is, you can start feeding this, and then this could be one batch.

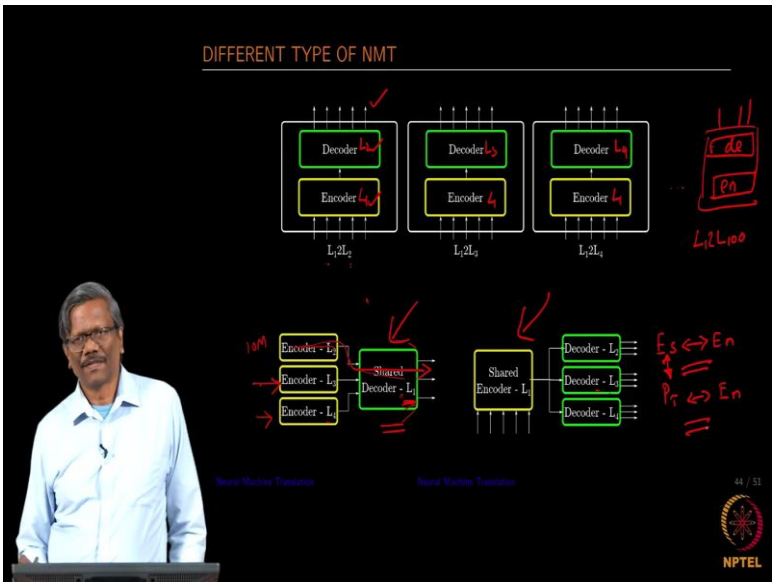
For example, if you have about 10 million pairs, you start doing this and then do the mini-batch or batch or stochastic gradient descent algorithm and train this. And then do not change anything here and then use this one start training this use this and then start training this and then keep doing it until these states are in equilibrium and then you can start feeding in the input from the decoder side, you know for you to translate I am sorry.

So, you can feed the input from the encoder and it would translate for you. So, you need to have some certain kind of mechanisms from which language you want to translate into or from which language you want to translate it into L 1. So, for example, if you have Tamil, Telugu, Malayalam, and so on you want to translate from Tamil to English, Malayalam English or from Telugu to English.

So, we need to specify or mark certain token so, that the encoders know of which one you are going to be using ok. So, this is one approach where you can at least reduce the number of RNNs right. So, what next? So, is it possible to have so, many of them for example, there are hundred recognized languages that are available in the world.

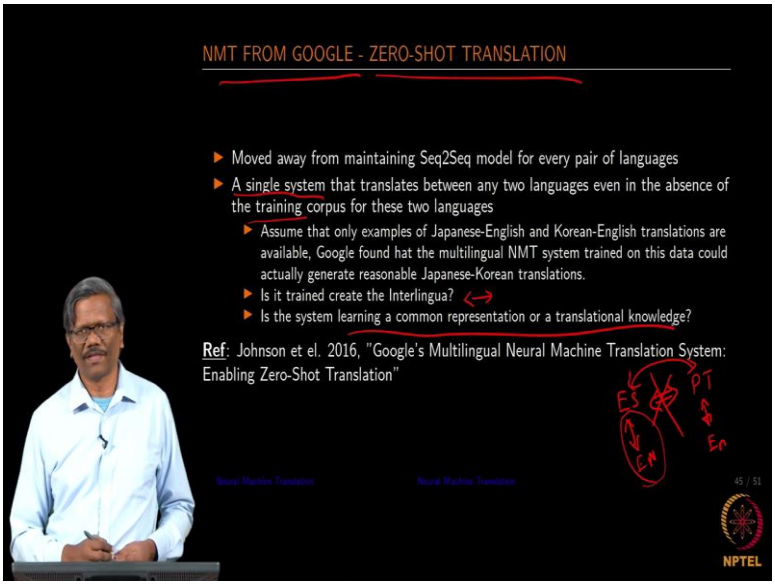
And if you look at the Indian subcontinent, we have 23 languages alone right 23 radio languages that are available in the Indian constitution-, but there are about a hundred languages across the world. So, we cannot be having. So, many translation modules in this fashion are it possible to have one single module that does the whole job, or is it possible for me to do this.

(Refer Slide Time: 12:19)



So, let me write this here, for example, I have the translation from English to French with French to English and I have let me write Spanish to English and then Portuguese to English let us take this for example, these two points. So, we have learned these two models, is it possible to do the translation from Spanish to Portuguese without making any changes? That is why some of the researchers what and then especially from Google, created something called a zero-shot translation.

(Refer Slide Time: 13:06)

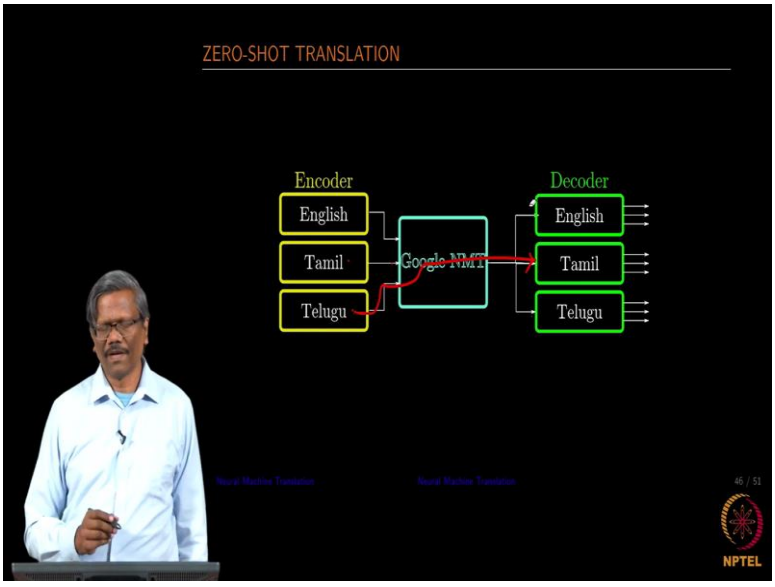


What they have done is, they moved away from the sequence to sequence translations instead of having those paths of translation modules, or is it possible to have one that helped me in terms of translating into various languages ok? So, they are trying to establish a single system that translates between two languages even in the absence of the training corpus for these two languages like what I had shown right. So, if I do not have I do not have these pairs of sentences, but I have these.

So, using this knowledge can I translate from Spanish to Portuguese is the idea ok. So, what happens when we create? You remember we spoke about the interlingua earlier at the beginning of the machine translation, where we want to capture the knowledge related to the source language as an independent knowledge source and then used it to translate into multiple languages.

So, this is exactly what they are trying to attempt, even though they claim that it works very well, but the results are still not at the mark in many cases, but in a sense their attempt is to create the internal lingua. So, they are also trying to see if there is a common representation or a translation knowledge that is possible to create.

(Refer Slide Time: 15:10)



So, this is their architecture. So, in their encoder, they have English I am taking three two Indian languages, Tamil and Telugu. I have English to Telugu I have English to Tamil translation and then Tamil to English and Tamil to or a Telugu to English. So, this is the box that we are going to be training in. So, during the training process what you do is,

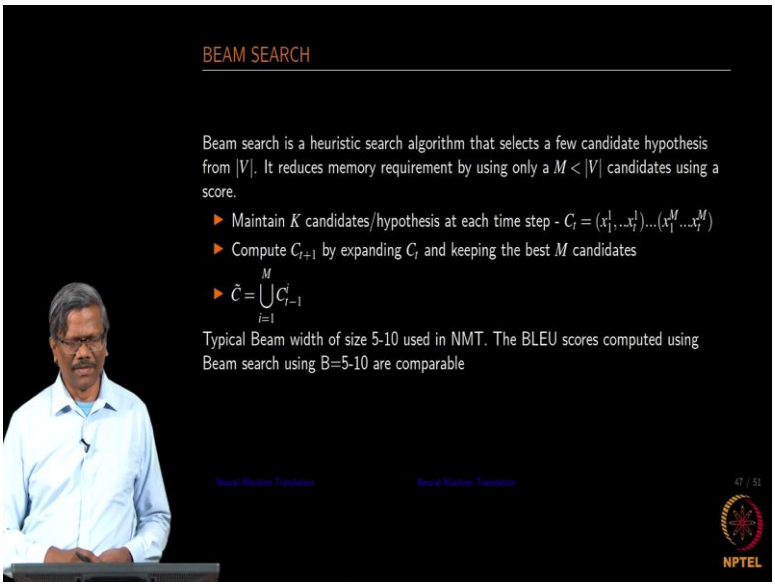
you just let us say this is going be doing the training process for Tamil and then we do this for Telugu right I think it is a little messy let me erase first.

So, from Tamil to English and then English to Tamil we train the model and then take the other Pairs Telugu to English and English to Telugu. So, without changing any of the model parameters, you know once the training is completed in these pairs without changing any of the models start reading the next one.

For example, when you are finished with English to Tamil and then Tamil to English we have certain states in the neural network or without changing any of these values in this state start training again. So, the pairs are trained we know instead of training only in batches of or in pairs of languages. So, you finish one pair, second pair, third pair fourth pair and do the cyclic part of them ok.

So, use the mini-batch or whatever model that you want to use. So, they claim that once this training is completed, they are able to translate from Telugu to Tamil effortlessly. So, this is the attempt that they have made and they claim that it is working very well in many cases.

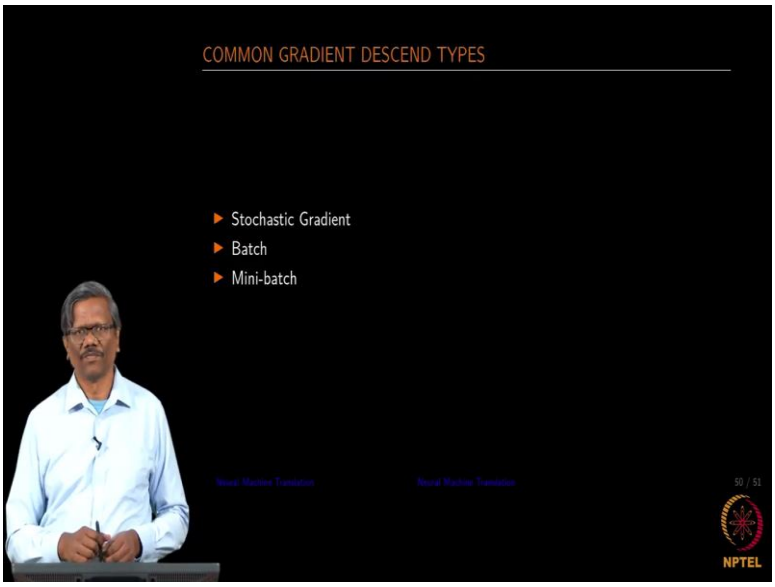
(Refer Slide Time: 17:44)



So, with this I complete this translation part both in the SMT as well as in the NMT site and this is the current state of the art right now ok. And most of the Google translation mechanism I guess is using this zero-shot model.

So, in the next session what I will be doing is, I will be talking about the beam search is we remember we have been talking about this at the end of every machine translation ok. So, here we never spoke about how this is done ok. So, we will talk about the beam search and then later we will talk about the common gradient descent types.

(Refer Slide Time: 18:37)



So, we will cover these two in the next sessions.