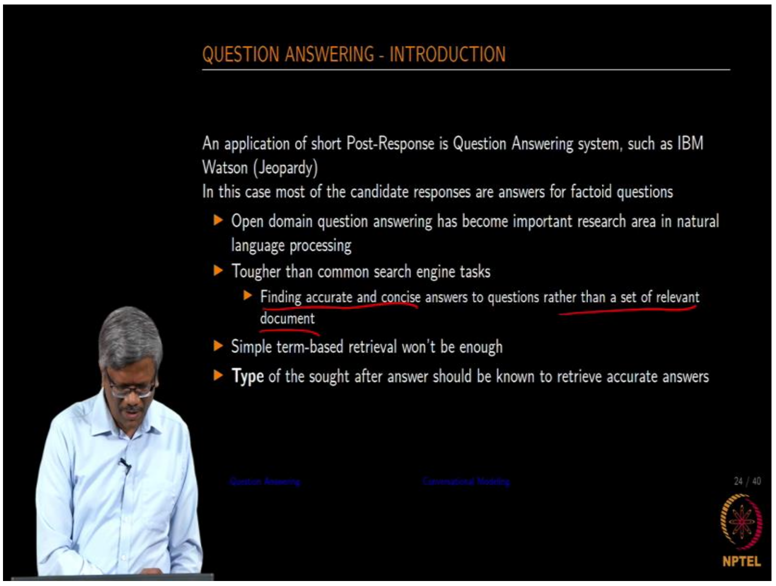
Applied Natural Language Processing
Prof. Ramaseshan Ramachandran
Department of Computer Science and Engineering
Chennai Mathematical Institute, Madras

Lecture - 88
Discussion of some ideas in Question Answering

(Refer Slide Time: 00:15)

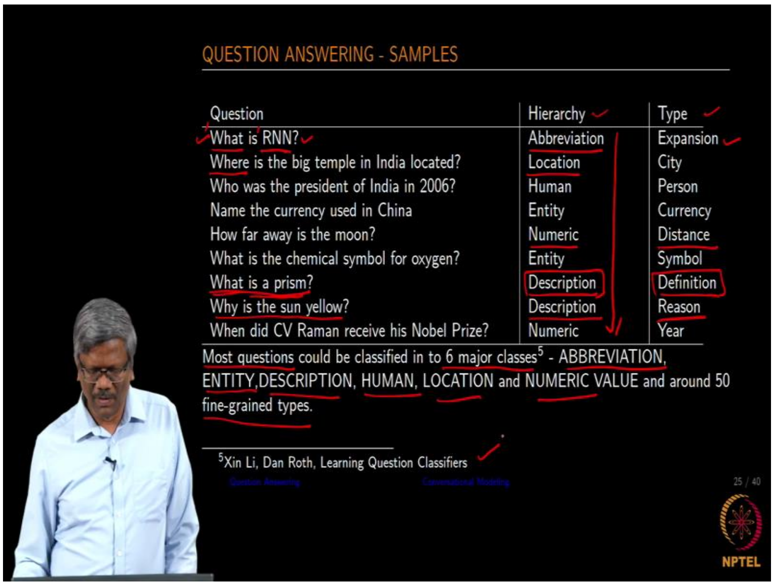


Let us look at what can be done, you know in order to improve this model. So, we are looking at you know the bits and pieces of conversation modeling, we are not looking at every aspect of that. So, initially you know we showed that it is possible to do some kind of conversational modeling using the information retrieval system right. And, now we see that it is not enough to give you good retrieval.

So, what should be added as part of this? So, as we mentioned earlier it finding the accurate answer is important rather than giving a set of relevant documents. So, that is not what IR based models are providing the right. So, I need a span of tokens. So, there is no way I can go and then say that from the 10th to 15th word pick up the answer and then give me the response.

So, this is we do not have a mechanism that is evolved as part of the IR model. So, we need to bring in additional mechanisms that should be able to go and then pick up those a span of text and then respond positively to the user queries.

(Refer Slide Time: 01:44)



So, let us look at some samples and then see what can be done? In terms of breaking the query into various pieces in terms of hierarchy and type and see if that could help us in terms of achieving the precise answers for the query. Now, let us take a look at some of the questions right. So, what is RNN? So, we know that there is an abbreviation here.

So, we can keep creating a hierarchy called abbreviation. And then when what is asked when what is the part of the question we are expecting an expansion for that right when there is an abbreviation we want the expansion for that. Where is the big temple in India located? So, where gives you the hierarchy called location ok. Let us pick up what is a prism? So, we know this is a question and it is expecting a definition.

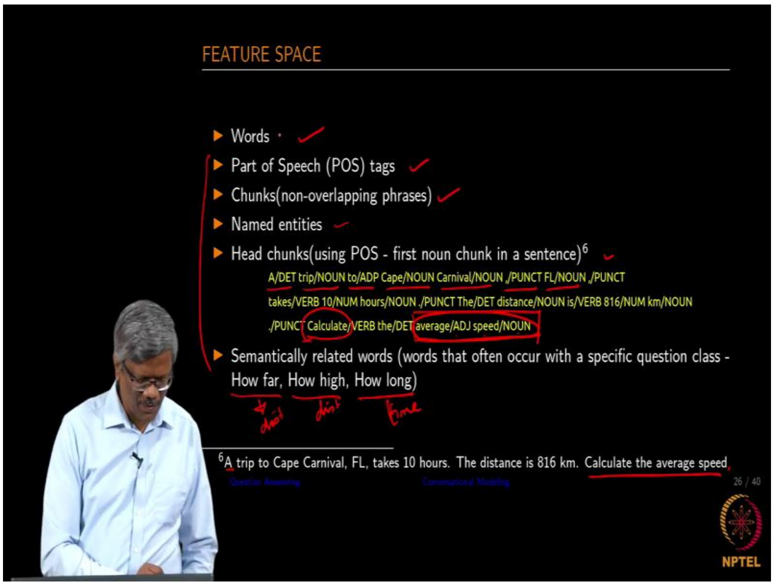
So, it is expecting a description from you and the description is in the form of a definition. Why is the sun yellow? Again, it is a description at a very high level since why is there. So, we need to find the reason why the sun is yellow ok. And, then there is one numeric part how far away is the moon ok. It is numeric and we need to look at or rather classify the type as a distance.

So, we can create a set of hierarchy based on the question. And, then use the hierarchy to fine-tune or create finer details for each of the question. So, now, if you look at RNN it is very well known that it is a recurrent neural network, it is an abbreviation for the recurrent neural network and we just have to provide the expansion. And, if you go to this question what is a prism is expected at the very high level, at the very high level this is a hierarchy and we need to provide the definition.

So, you can classify most questions in 6 major classes such as this ok, like an abbreviation, entity, description, human location, a numeric value and so on. And, then all these 6 major classes again can be classified into fine-grained types ok. So, how do we learn this? So, that is the next question right. So, now, based on the information retrieval model, we have only worried about the keywords part now, we are able to create some kind of a hierarchy and fine-grained types can this help in terms of understanding the query better, it is what we want to find out.

So, maybe you may want to look at this paper on learning question classifiers, where the details of all this hierarchy and types are well documented.

(Refer Slide Time: 05:47)



Now, look at the feature space we have words that is what we use in the IR model. So, we can get the part of speech for each of the words, we can get chunks or 2 or more words joined together and the named entities, we can have head chunks using the POS. For example, for the problem that stated here a trip to cape carnival Florida takes 10 hours the distance is 800 and 16 kilometers, calculate the average speed ok.

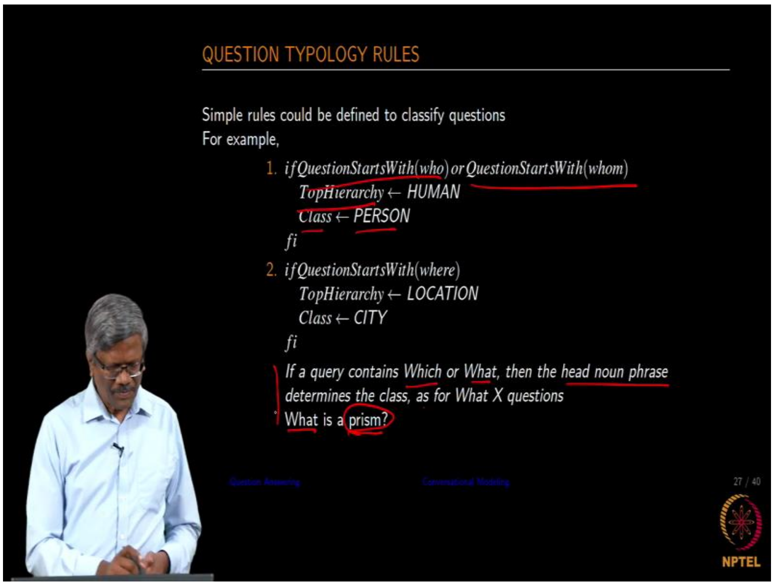
So, now we need to find the answer to this question. So, when you make it through the POS you will see that there is a determiner and then the trip is a noun. And, then there is a preposition here that is coded as ADP, we have a noun and then there is a punctuation noun and so on right. So, we are able to get the part of speech for all of this in this fashion. And, then you have a question at the end.

So, based on the keywords that you have created you know. In this case, it is not possible to identify a question if you do not have a set of keywords defined for it. In the case of kinematic there are about 8 or 10 different verbs that you can use to distinguish the sentence from the normal statement from a question ok. So, in this case you will see that calculate is part of the question or tagged as part of that and then we have average speed. So, this is what I am talking about as a chunk here.

So, in this case the chunk is coming from the yes adjective and the noun here. So, this is your chunk. And, then if you look at that chunk you know that and this is what you want to find out calculate is the keyword in the question and then this chunk will tell you what you want to find out ok. And, this is what we are talking about here. And, then semantically related words, words that often occur with a specific question class how far, how high, how long? Right.

So, in this case you know that it is talking about this is talking about distance, again this is distance and this is time in most of the cases in the kinematics problems ok. So, now, we can expand our space feature space with all the other new features, it is not going to be just about words alone.

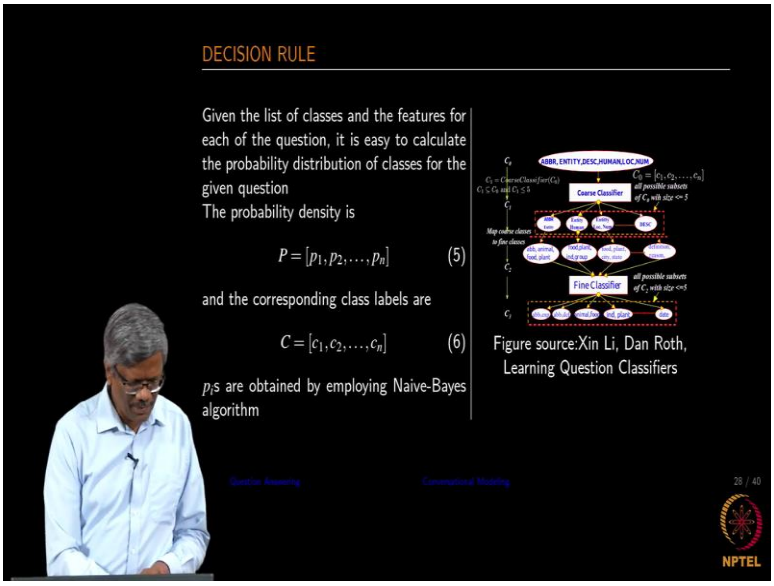
(Refer Slide Time: 08:56)



And, then in the question typology rules, we can start creating simple rules such as this if the question starts with who or the question starts with whom, then the top hierarchy is human and the class is the person. So, in this fashion, you can create or rather you can classify the questions in this fashion, what is going to be the top hierarchy, what is going to be the type? If, a query contains which or what, then the head noun phrase determines the class as for what X questions.

So, in this case what and then here you have the noun phrase right. So, this determines what is to be extracted ok, that what is being said in this ok.

(Refer Slide Time: 10:02)

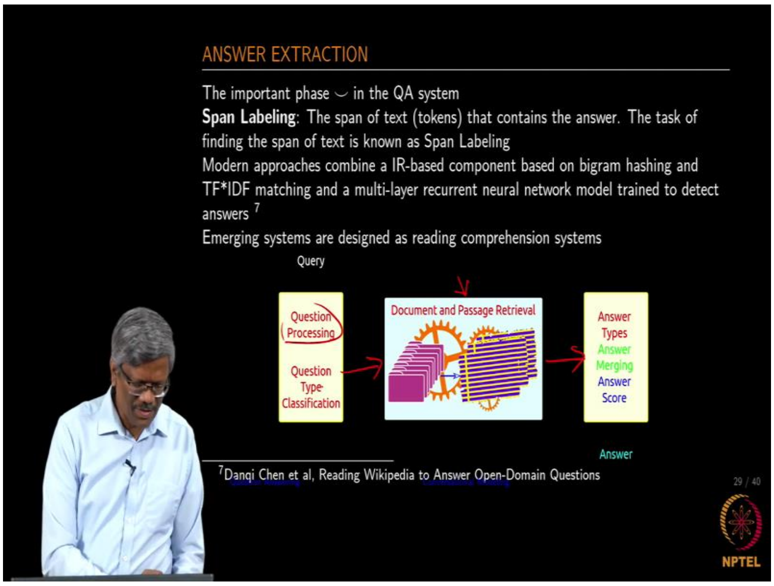


Again you can use the decision rules such as these and then for all the questions that you have, you can find the course class or the hierarchy and the fine types. So, the idea here is to find the top hierarchy and the fine-grained types ok. So, if you read through the papers you will be able to understand the algorithm described on the right side very well. So, here what they are trying to do is instead of creating only one hierarchy value for each of the words, they try to get more than 1.

So, they have the size of 5 you know that could be the wrong classification you know if you just pick up one of the values for the hierarchy value. So, they try to provide 5 different hierarchy values based on this course that they compute. And, then based on the query and the response that you are looking at they use all of those to find out which one really matches really well with the post or response.

So, that is why they are keeping the hierarchy values to 5 I mean there will be for each word there will be 5 hierarchy values that are available. So, in the same fashion even the fine-grained values will be chosen me for each one of the words there will be at least fine-grained types that you will find ok.

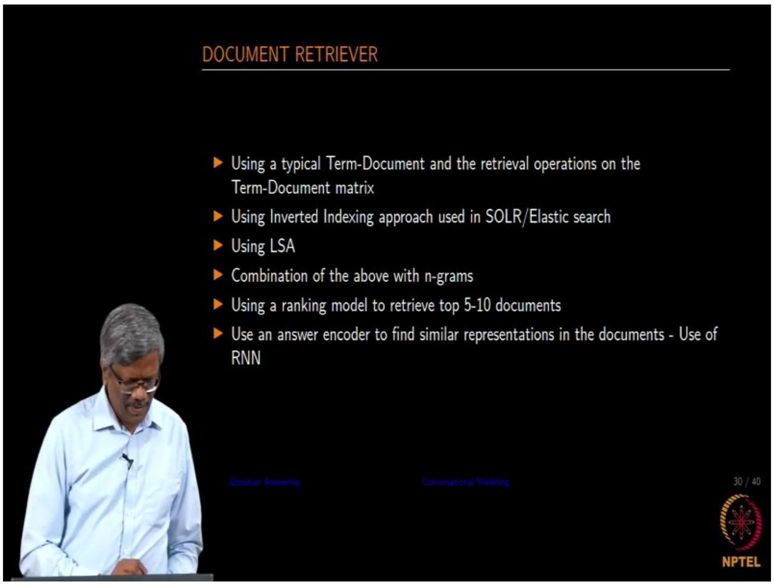
(Refer Slide Time: 11:52)



So, in this case as we mentioned earlier in the IR based model, you also have or the same retrieval as part of this. What additional thing, that we have added is we have added the question processing mechanism where, they type the hierarchy and the fine-grained types are found.

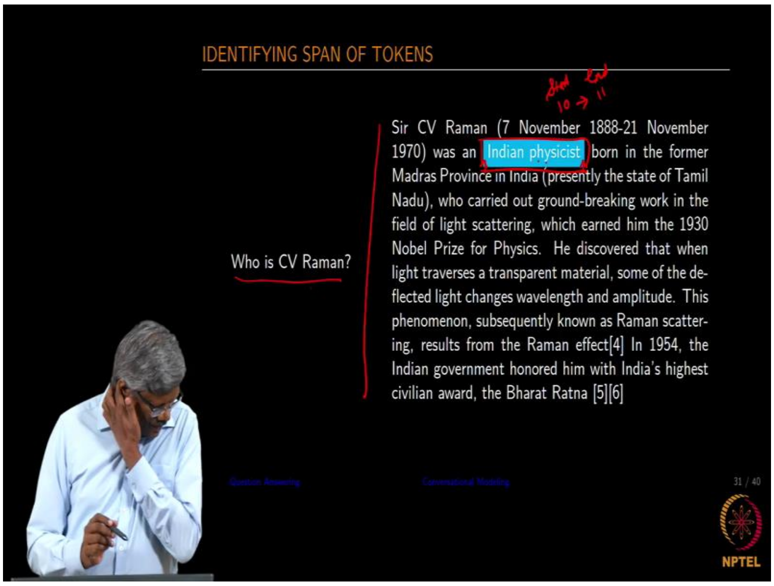
So, from the documents we retrieve various passages and then using the newly found feature set. So, using the newly found feature space, we should be in a position to extract answers, which is better than IR based models. So, that is the idea here ok.

(Refer Slide Time: 12:48)



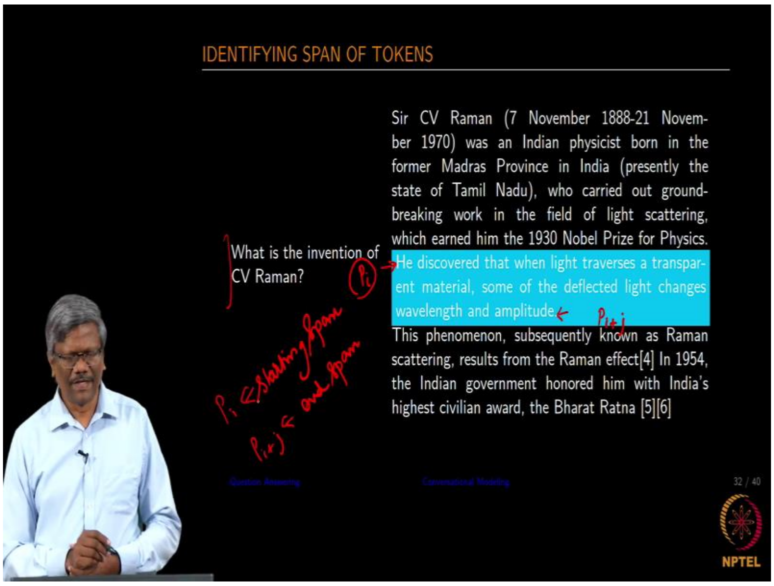
So, again the document retriever as I mentioned earlier it could be a solar, elastic search or let us say that we had used earlier. And, answer encoder could be based on the RNN.

(Refer Slide Time: 13:10)



Let us look at one of the examples of how this could be achieved using neural nets? So, before getting onto that let us find out what is a span I think I mentioned about this right. So, we already spoke about identifying the span of tokens for each question correct.

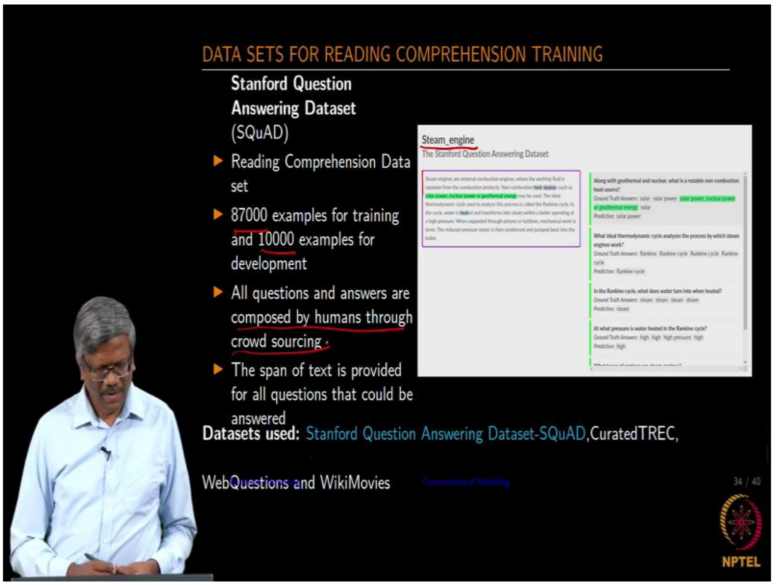
(Refer Slide Time: 13:35)



So, for another I have questioned this span of tokens is he discovered what is the invention of CV Raman? Given a set of paragraphs you find out the span of token that matches the question you have here. So, again the span of token starts from letting us say it is starting from Pi to Pi plus j ok.

So, this is what you want to achieve. So, in this case there is one more thing that we are introducing we need to find out what is the value of Pi is the starting span that is what I have mentioned as Pi here Pi plus j. So, this is something that we need to find out not just the paragraph, but we need to find the span of text.

(Refer Slide Time: 14:41)



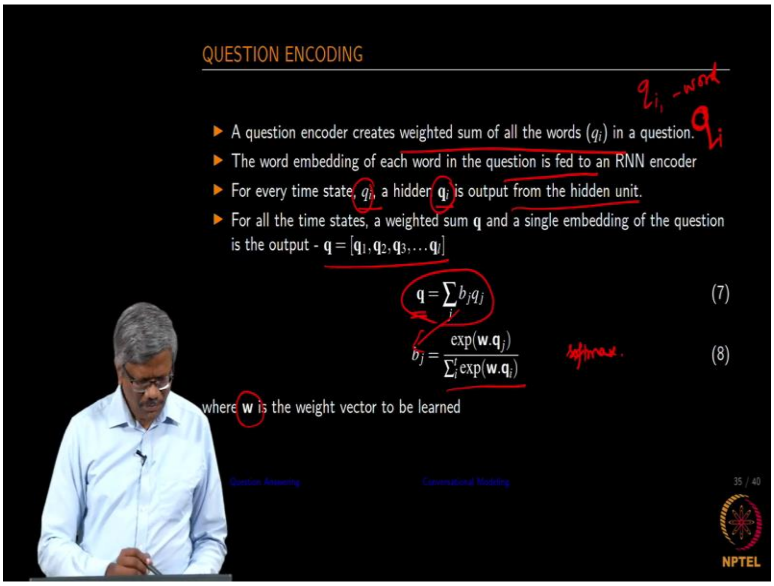
So, how do we do that? So, for that again you know, if you want to use any models using a neural network you definitely require a corpus. So, Stanford question answering data set provides you an interesting corpus, where it helps you in terms of finding out for a given paragraph, what are all possible questions that you can come up with. So, they used humans to really codify the questions, when they provided a lot of paragraphs from Wikipedia.

So, in this case, there is for a given paragraph, what are all the possible questions that you can expect for the given paragraphs. So, in this case I have taken the keyword steam engine and it provided a paragraph and then gave a set of question and the ground truth answers are also available as part of that. So, the question would be along with geothermal and nuclear, what is a notable non combustion heat source? The answer is available as part of this a paragraph and the ground truth answers are provided by the human.

They form the question then also provide the answer for that is what is used as the ground truth answer. They also provide the span of the text so, where it starts and where it ends for each question. So, if you go to the website here you search for squad a Stanford you will get this link, you will be able to go and then browse through the squad data set which is really used for reading comprehension for the machine. It has 87, 000 examples for training and 10, 000 examples for development.

All questions and answers are composed of humans through crowdsourcing. And, then they also have provided the span of text for all the questions that could be answered ok.

(Refer Slide Time: 16:54)



Having obtained additional feature vectors. Now, we should be able to do a better job I guess right with respect to training the network with respect to matching the question with your answers and so on. So, in this case I am not going to be doing the short text conversation model as we had done earlier. I am just skipping that part of the short text conversation model, getting into some portions of question answering.

Using what we had seen earlier you know in terms of classifying the questions into a various hierarchy and different types and we can use the part of speech we can use chunks and so on. We can provide better input to the system, we also have word vectors assuming that we have good quality word vectors coming in from some of these a word embedding models, we should be able to provide good input to the question answering system.

So, again when you look at the question answering system, it is very similar to what we saw earlier in the conversation and modeling, where there is a query and the query is related to the post and the response. In this case we have a query question and we have a corresponding answer to that question.

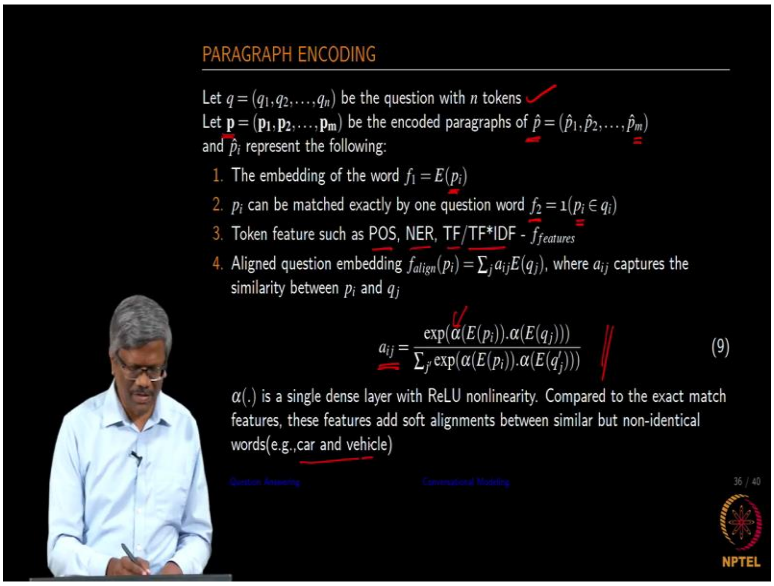
So, again, in this case, we should be using the information retrieval engine in order to get the document, extract the paragraphs relevant to the question from the documents and spot the span of text within the paragraph that we have retrieved and then a trained the models as we had done earlier. So, we can do the training differently in this case, here we train the question module separately we train the answer module separately.

But, connect the two modules using the question encoding that we create. So, and that is what we are going to be looking at right. Now, the question encoding is a separate entity by itself, where we create the weighted sum of all the words in the question ok. So, we have 100 1000 questions I use an RNN to train the question module or question encoding. So, the idea is to find out if a similar question of that comes in after the training, we would retrieve an equivalent question encoding which we could use in the q_i

So, here we have the weighted sum of all the words is what we are going to be encoding. The word embedding for each word is fed, every time state hidden q_i is output see note the as a type of the notation here. So, this is the question word $_i$ is the question word, q is the output from the hidden unit ok. For all the time states awaited some q and a single embedding of the question is output.

So, at the end of the hidden layer this is what you get ok. And, then every q vector is obtained using this relationship ok. Where b_j is nothing, but the exponential of rather the ratio of the exponential of the weight and the hidden value of the j th element to the sum of all of those, it is very similar to what we saw in this softmax site right. So, this is a weighted sum of all the hidden values that you find from the hidden unit and this is what we are going to be learning ok.

(Refer Slide Time: 21:47)



So, in the same fashion, we have to have a certain notation for the paragraph as well. We have the question with n tokens and then we have a capital P again, this is the encoded paragraphs of P hat which contains m, elements what is that P hat or we will define now? So, we for every paragraph that we have retrieved right using the retrieval model or if you are you can be using the squad or you have the paragraphs made available, we can find the embedding for each of the words in the paragraph ok.

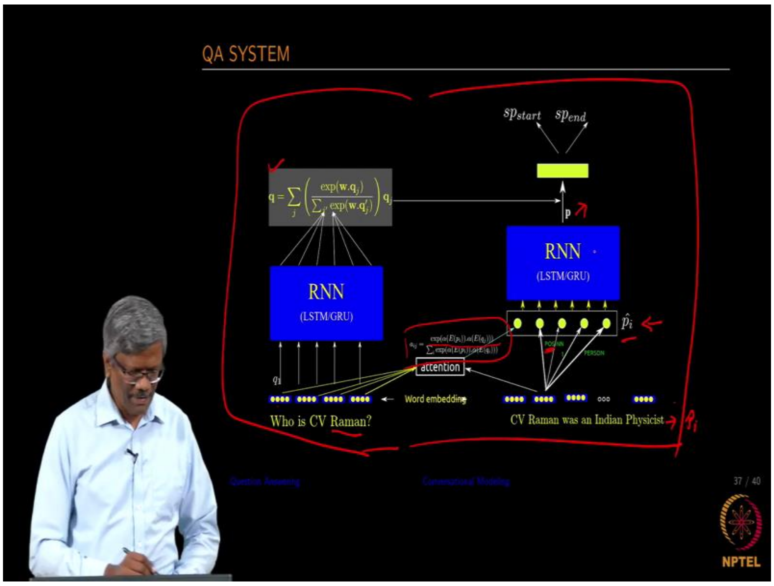
So, the small pi pays attention to the notation here without any bold. So, you have a paragraph and every word is represented in that paragraph by this rotation. Pa can be matched exactly by one question word ok. Suppose, if we have the word which is also available as part of the question, then we have one feature called f_2 =1. And, then we can use the token features such as POS NER TF or TFIDF as another feature set.

And then align question embedding is also captured ok. So, we also want to find out along with the question that we have, we want to see how these words are aligned with the paragraph that we have formed. So, you remember in the translation model, we try to find out the phrase alignments or word alignments.

In the same fashion we want to find out the alignment of the words that are available as part of the question as well as in the paragraph using this relationship ok. Here this could be a ReLU and so on. I think this is the first time we are looking at ReLU in the equation correct.

So, why do we do this the idea of bringing in this is to find out if you know if the question if you have a car and then in the paragraph if you have the vehicle we should be able to match that you know there should not be considered as 2 different words. So, that the reason why we want to bring in the soft alignment through this model ok.

(Refer Slide Time: 24:31)



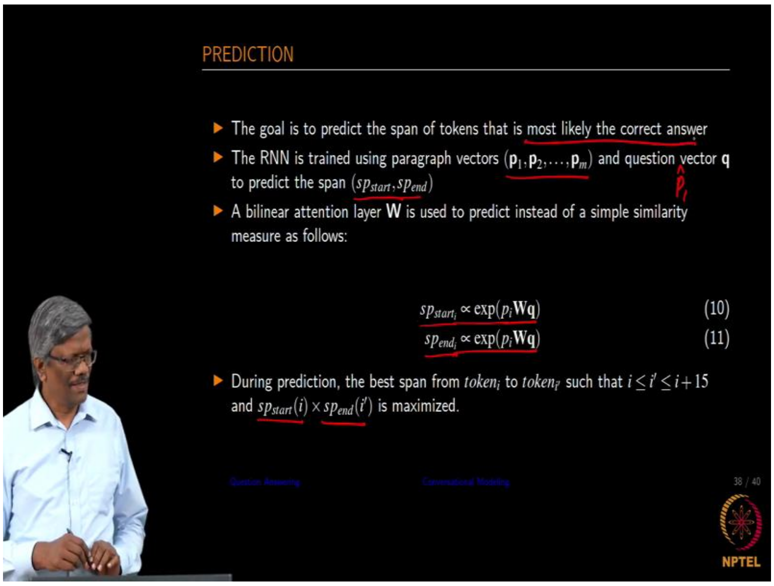
So, if you look at the QA system I am not going to go into the details of this again, this is available as part of the research paper, I want you to go through the paper to understand this in detail. So, we have a question module and we have an answer module here ok. And, each can be independently trained. So, what is that we get from this as we mentioned earlier, the question is who is CV Raman?

And then for each word we have the embedding and you know well in the RNN we through the time slice we input one word at a time. And, then at the end, we create an embedding for the question, which is the weighted sum. And, that is what we are going to be using it in the answer module as well ok.

So, let me remove this now. And, then in the answer module for the question that we have who is CV Raman? CV Raman was an Indian physicist ok. So, that is the answer. And, this span is available as part of the paragraph ok. So, we are going to be feeding in the entire paragraph one word at a time and then we are going to be trying to get the attention using this a j model as mentioned in the previous slide, that would be input as part of the vector, which is defined by this $\hat{p}_i$

So, here it is p_i each one is a so, this is let me p_i here. And, then once you provide other features into this it is transformed the input is transformed into $\hat{p}_i$ and that is what we feed as input to the RNN. So, now we have a paragraph encoding available as part of that we have the question encoding available that is coming in from there. And then using this we want to estimate the starting and the ending position of the span and this is the goal of this. So, the goal of this is to train the model to find the start and end of this span ok.

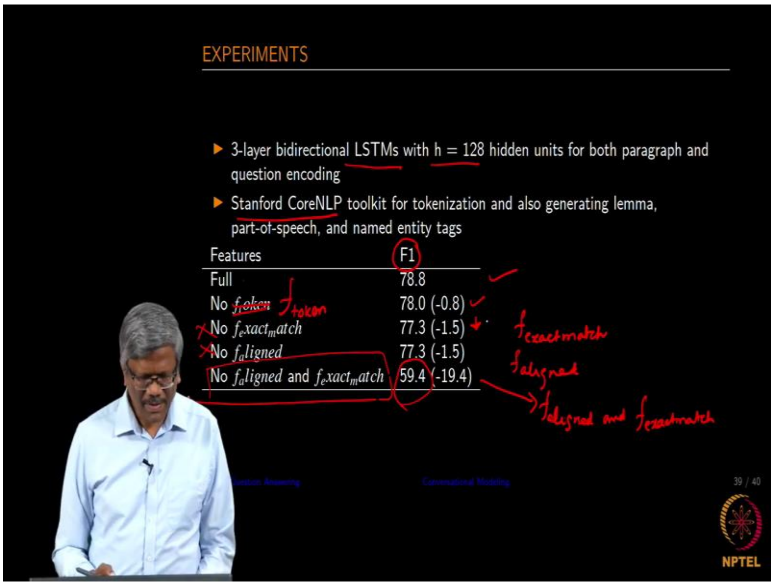
(Refer Slide Time: 27:08)



So, the goal is to predict the span of token that most likely the correct answer. RNN is trained using the paragraph vectors, that are obtained using the $\hat{p}_i$ to predict this by an sp start and spend. So, here the sp start and spend can be found using this relationship here. And then finally, the sp start and the product of those two should be maximized to get the probable starting and ending position for that.

The goal is to find the sp start and spend and then the prediction of the span of the token is going to be found using the neural net model and that would give you the most likely answer for the question ok.

(Refer Slide Time: 28:25)



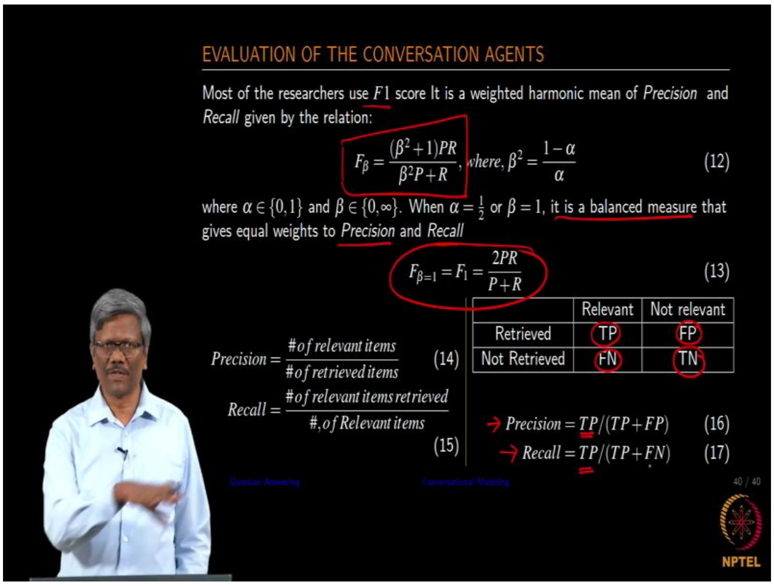
So, in the experiment what they have done is they have used 3 layers bidirectional LSTM with the hidden 100 and 28 hidden units for both paragraphs and question encoding. They used a Stanford CoreNLP toolkit for tokenization generating lemma part of speech and so on ok. And, then at the end what also they have done is they try to find out a score F and score ok, which is used as the benchmark to find out how good that model is ok?

So, with all the features you know using all the features that I have mentioned earlier. The f 1 score is about 78.8 and then this should be f token in the same fashion it is and then here it is f, then the last one is. So, remember we spoke about those features as input to the system. If you look at this answer part of the RNN so, we have a part of speech that we are feeding in see for example; in this case Raman is being fed so, it is a noun form. And then it is available as part of the question so, we have this one here.

And then they type is a person that is what we are feeding in? So, all the combinations of those feature vectors are formed here rather than just feeding only the embedding. So, we have a new combination of the feature vector and that is what is being fed? And, that is what we are talking about in this case. So, when you use all the feature vectors the F 1 score is about 78.8, when you use all the features the F 1 score is 78.8, when you do not use f token the score is 78 there is not much of change ok.

When you do not use f token feature there is not much of a change, when you do not use an exact match see the reduction in the F_1 s score. So, when the score is high; that means, we have a good system ok. When you do not use the aligned again it is similar to what you find, when you did not use f exact much, but when you discarded these 2 features the system performs poorly; that means, this feature is very important for identifying the span of tokens. So, that is what this experiment says.

(Refer Slide Time: 32:00)



F_1 score is obtained using this relationship with you. So, this is a very important relationship in information retrieval. And if the β =1 or α equal to half what we get is the F_1 score, F_1 score when the β =1 it gives a balanced measure, where the precision-recall is given equal weights. So, when the β =1, it is a balance to measure that gives equal weights to precision and recall ok. And, this is the formula you get when you have β =1.

These are part of the information retrieval and do not want to get into the details of all of these and then what is P here? P is nothing, but the precision. Precision is given by the ratio of the number of relevant items to the total number of retrieved items and recall is the number of relevant items retrieved to the total number of relevant items. So, you can also draw this in this table as given here.

So, you have true positives, false negatives, false-positive and true negatives, you know as it is part of the precision if you have the ratio of the true positive to the sum of the true positive plus F positive. Then, the score what we score is the precision and then for the retrieval for the recall the ratio of the true positive to the sum of the true false-negative and this is what we use as part of the evaluation of the conversation model ok.

So, with this I conclude this, I urge you to go through this paper to understand the concept very well.