CS520 - INTRO TO ARTIF INTEL

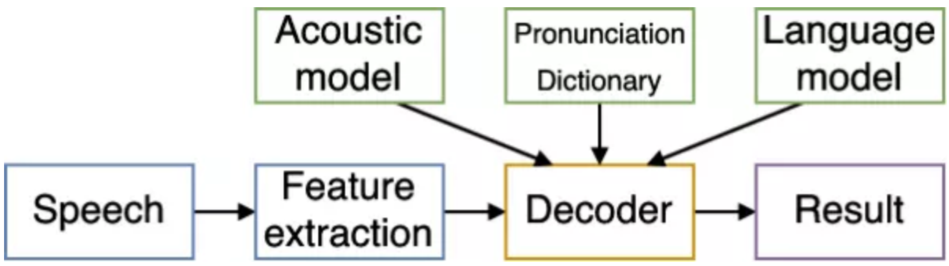
Final: Question 5 - The Big Picture

Name:
Xinyu Lyu(xl422)
182007396

Pick one of the following. Identify some of the properties and abilities it would need. Identify at least five issues that would be faced in trying to implement or build such a thing, and possible algorithmic solutions for each based on the material covered in class. Credit will be given for thoroughness and detail.
• ArtForgerBot
• CharadesBot
• LieDetectorBot
• SheepdogBot
• PulitzerPrizeWinningBot
• NobelPrizeInChemistyOrBiologyBot
• PharmacistBot
• PsychotherapistBot
• TerrorThreatDetectionBot
• FiremanBot
• ArtForgeryDetectorBot
• DrugTreatmentDesignerBot
• MovieAdvertisingAndPromotionBot
• TakeThisTestForMeBot (that could actually pass)
• HumanConsciousnessBot

LieDetectorBot
Human-computer interaction is a fundamental field for AI. During the communication to the human, if the LieDetectorBot wants to detect if the person is lying, there are some abilities it needs. First, it needs to know what the person is saying, so it should be able to recognize the speech made by the human. Secondly, as we know, during the conversation the sentiment will reveal the mental activity from the person, which will help us to interpret if the person is lying or not. So, it should be able to understand the emotion by person's words and facial expressions. Thirdly, when a person is lying, his logic is very confusing. So, after reading the texture record of the person's words, we can find if the person is lying or not. Therefore, our robot should be able to see the logical error from the texture record of a person's word. Last but not least, as a robot, I think it should be able to move, for I can't carry it anywhere because it is too heavy !! Therefore, the LieDetectorBot should be able to recognize what the person saying, and the ability to interpret the emotion and facial

1.Speech recognization



Speech recognition is a process of encoding and decoding first. Signal processing and feature extraction are the processes of encryption, that is, Feature Extraction in the figure, feature extraction, and the speech vector is obtained from the original speech. The latter is the decoding of the speech vector, and the Acoustic Model and Language Model required for decoding are the acoustic models and language models mentioned above. Acoustic Models This article will focus on the language model is the N-gram part of the "Basic on Natural Language Processing."

The problem of acoustic model processing mainly lies in the variable length of the feature vector sequence and the rich variability of the audio signal. Because the speech length is uncertain, the length of the feature vector sequence is also unknown. We generally adopt dynamic time warping method and implicit Markov model to deal with. The rich variability of audio signals means that the speaker's gender will cause the diversity of audio signals, health, tension, speech style and environmental noise, surrounding vocals, channel distortions, dialect differences, non-native accents, etc. cause.

In the past, signal processing and feature extraction were generally performed using Mel cepstral coefficients or relative spectral transform-perceptual linear prediction.
As a feature vector, then a mixed Gaussian model-hidden Markov model (GMM-HMM) is used as the acoustic model, and then trained by maximum likelihood (ML), followed by sequence discriminative training algorithms, such as minimum Guidelines such as classification error (MCE) and minimum phoneme error (MPE) was proposed.

HMM has three main problems. The first is the possibility; it uses the Forward algorithm. The second is decoding, which uses the Viterbi algorithm. The third is training, which uses the EM (forward and backward) algorithm.

1.1 Problem 1 : Possibility

<u>The Forward algorithm</u> `

The probability of an HMM generating a series of observation sequences x

Initialization: $a_0(s_i) = 1,\ a_0(s_j) = 0, where\ i \neq j$

Recursion: $a_t(s_j) = \sum_{i=1}^{N} a_{t-1}(s_i) a_{ij} b_j(x_t)$

Termination: $P(X|r) = a_T(S_E) = \sum_{i=1}^{N} a_T(S_i) a_i(S_E)$

1.2 Problem 2: Decoding

Instead of calculating the probability of each possible sequence of states in the actual calculation, a Viterbi approximation is used: from time 1:t, only the state and probability with the highest transition probability are recorded.

$V_t(s_j)$ is the maximum probability of state j from all states at time t-1 to time t.

$b_{t_t}(s_j)$ is the represent that,from which state at time t-1 transitions to the state t at time t,the probability is the maximum.

```
The Viterbi algorithm                                                    `
The probability of an HMM generating a series of observation sequences x
          Initialization: $a_0(s_i) = 1,\ a_0(s_j) = 0, b_{t_0}(s_j) = 0, if\ S_j \neq S_i$
          Recursion:      $V_t(s_j) = max V_{t-1}(s_i) a_{ij} b_j(x_t)$
                          $b_{t_t}(s_j) = argmax V_{t-1}(s_i) a_{ij} b_j(x_t)$
          Termination:    $P* = V_T(S_E) = max V_T(S_i) a_{iE}$
                          $S^* = b_{t_T}(q_E) = agrmax(S_i) V_T(S_i) a_{iE}$
```

1.3 Problem 3: HMM Training has been explained in the notes. So, I won't tell it in details for this part.

In recent years, hierarchical authentication models such as DNN have become feasible, such as context-dependent deep neural network-context-dependent DNN-HMM (CD-DNN-HMM) than traditional GMM-HMM The performance is much better. This article will focus on CD-DNN-HMM.

2. Sentiment recognization

Actually, this is my current research topic in this semester. Basically, I worked on Multimodal Emotion Recognition. It means I use a fusion of acoustic, texture and video with word-level alignment data to predict the sentiment during the conversations. Actually, teaching machines to speculate human emotions from humans conversations is by no means an easy thing. And there are three problems in this question show following:
1. For us, it is still hard to precisely figure out the principles for emotion analysis between humans. 2. It is difficult to teach computers how to speculate human emotions from humans conversations. Because it is difficult to extract the proper features revealing the actual human state of mind. 3. We, humans, make emotion analysis based on a fusion of facial expressions, gestures linguistic contents, and vocal signals. So, it is necessary for us to find out how to integrate multiple resources to make proper emotion analysis.

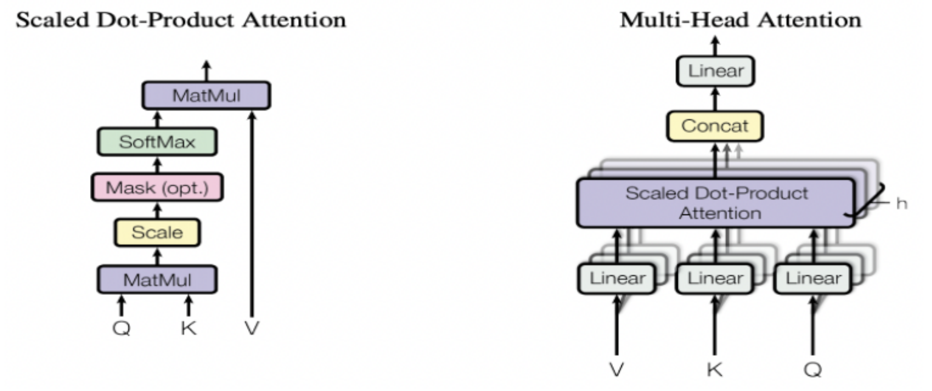
There is a variety of data from the different source, like linguistic content and vocal signal, facial and body movements of from the conversation. And for our project, we choose to use linguistic content and the vocal signal of the communication to build the baseline model for multimodal data fusion. That is because these two branches of data are the most potent source of information in human conversations.

2.1 Problem: Word-level Aligment

Recent work has done too much on extracting the features based on the sentence level from audio data and text data. And it doesn't make sense to me, because the basic unit for us to understand the meaning of the conversations is the word-level. For example, when learning English, it is common for us to find the corresponding meaning from our native spoken language. So, we make a word-level alignment between text data and the audio data before inputting them into our model.

2.2 Problem: Extract the associated features from audio data and text data

Another challenge is that we need to extract the associated features from the audio data and text data. Recent works use traditional methods to obtain the handcrafted features from audio and text data with the help of LSTM and CNN. However, with the inspired from the 'Attention is all you need' published by google research, we want to introduce an architecture only with the multihead self-attention strategy to extract the features individually from each branch.



Here, I only introduce how to use self-attention in our texture model. As described in the figure above, the self-attention can be representing as below:

$$MultiHead(Q,K,V) = Concat(head_1, \ldots, head_h)W^O$$

Where $head_i = Attention(QW_i^Q, KW_i^K, VW_i^V)$, the Q is the query vector, K is the key vector, V is the value vector for each word in the texture data. More specifically, for self-attention, the inputs of Q, K, V are all the embedding word-level features we extracted by word2vec. And we can represent the text attention extracted from the self-attention as follows.

$$A_i^t = \tanh(W_t[em_t, em_t.em_t] + b_t), i \in [1, \mathrm{N}]$$

Where W_t, b_t are learnable parameters from self-attention. And em_t denotes the word2vec features after position embedding operation.

And between the self-attention layers, we apply Batch Normalization layers to solve "Internal Covariate Shift" problem to make the model easier to train and converges more quickly. For Mini-Batch SGD, there are m training instances in one training process. The specific Batch Normalization operation is to transform the activation value of each neuron in the hidden layer as follows:

$$x^{(k)} = \frac{x^{(k)} - E[x^{(k)}]}{\sqrt{Var[x^{(k)}]}}$$

Then we use GlobalAveragePooling1D to reduce the dimension of the feature matrix to one dimension which will be convenient in the word-level fusion model. In GlobalAveragePooling1D, the input word vector sequences are added to the averaging and integrated into a vector.

With the text self-attention, we apply softmax activation function to calculate the distribution of the text attention.

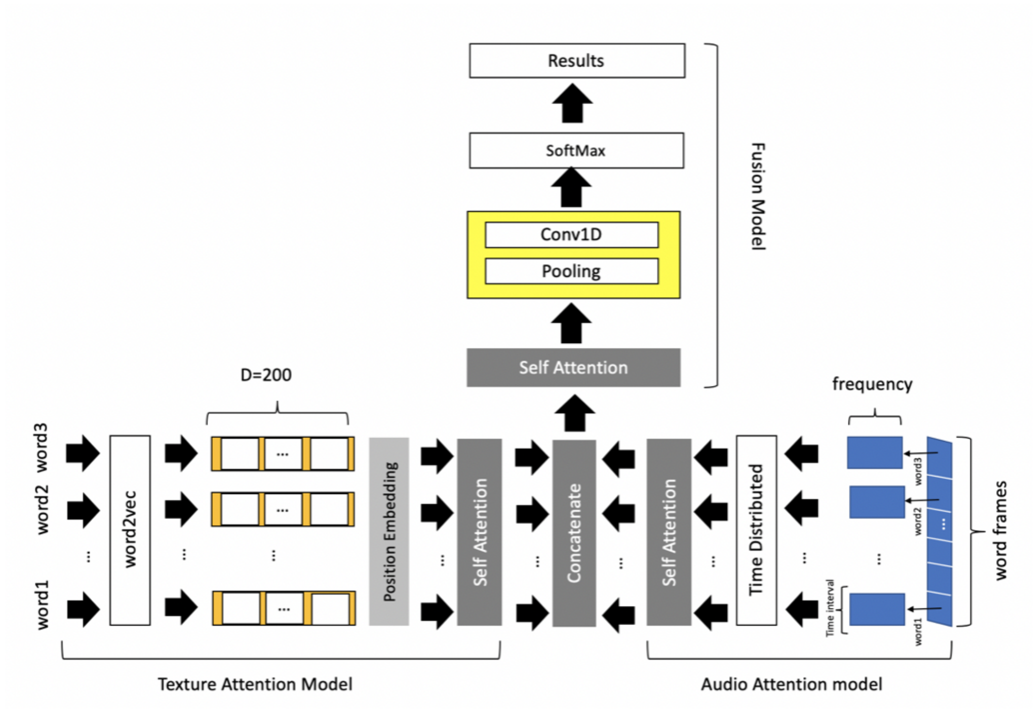
$$a_i^t = \frac{\exp(A_i^t)}{\sum_{k=1}^{N}(A_k^t)}$$

Where a_i^t represents the score of each attention.

Finally, we use fully connected layers to help us make the prediction of the result in the audio part.

2.3 Problem: How can we make a fusion on it to decide on emotion classification

The last challenge is that since we have extracted the associated features individually from the audio data and text data, how can we make a fusion on it to decide on emotion classification. In this part, we introduce a weighted fusion model with self-attention and CNN's. That makes the model learn the weights of audio data and text data on the final decision of emotion classification. And below is the whole model architecture for the fusion model.



3. Search algorithm

Before the robot can move, he should be able to find out the path between the starting point and ending point. In order to make it more efficiency, it should always find the shortest path just like me. I'm a little lazy, so for the most occasions, I always want to find the shortest path towards the target. The A*(A-star) algorithm is the most efficient direct search algorithm for solving shortest paths in static networks. The main application in video games is to find the best route between two points on the map. In the field of robotics, the A* algorithm is often used for mobile robot path planning. Since, we have learned a lot about A* during our class, I won't tell it in details in this part.

Bonus:

i) What did my dog dress as for Halloween?

Superman costume !!!

ii) Draw a picture of my dog in her Halloween costume.