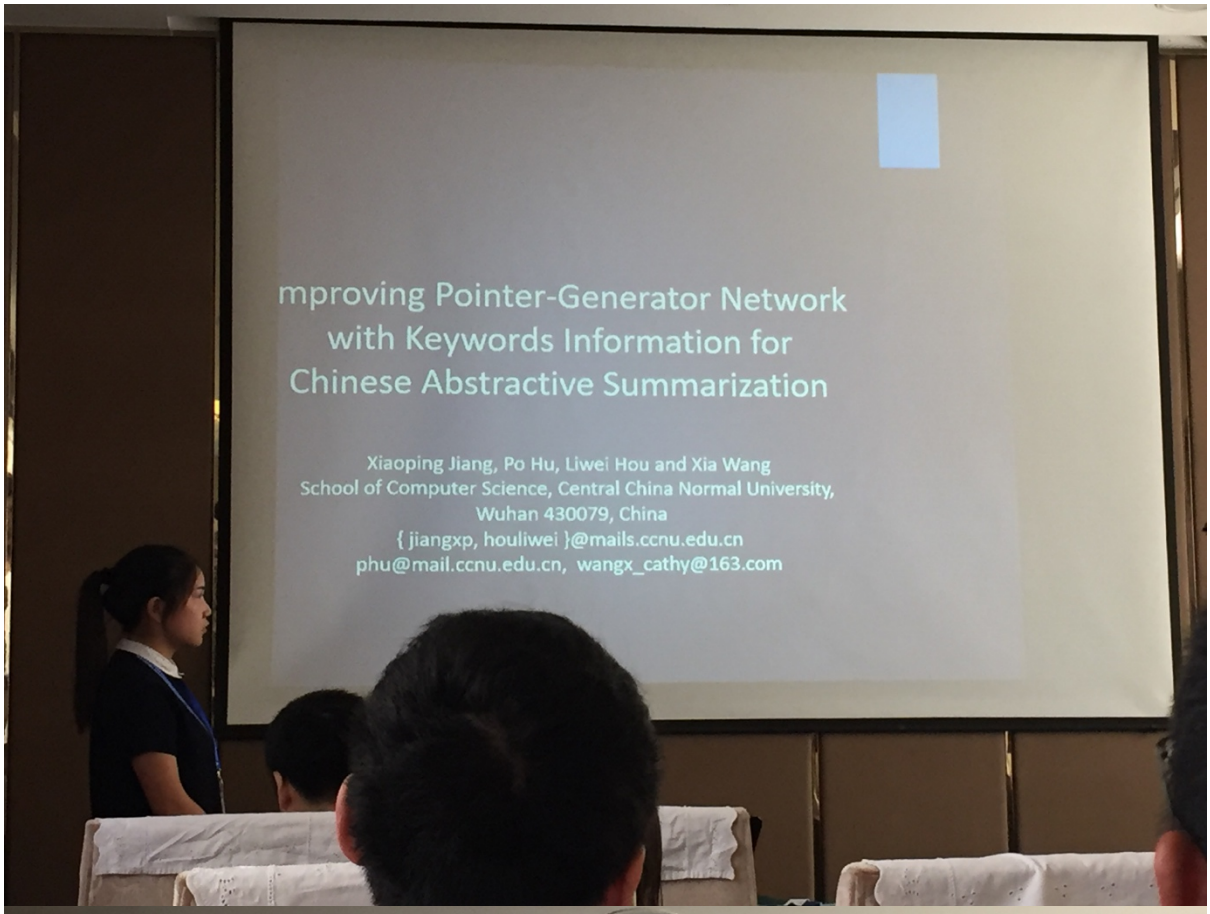
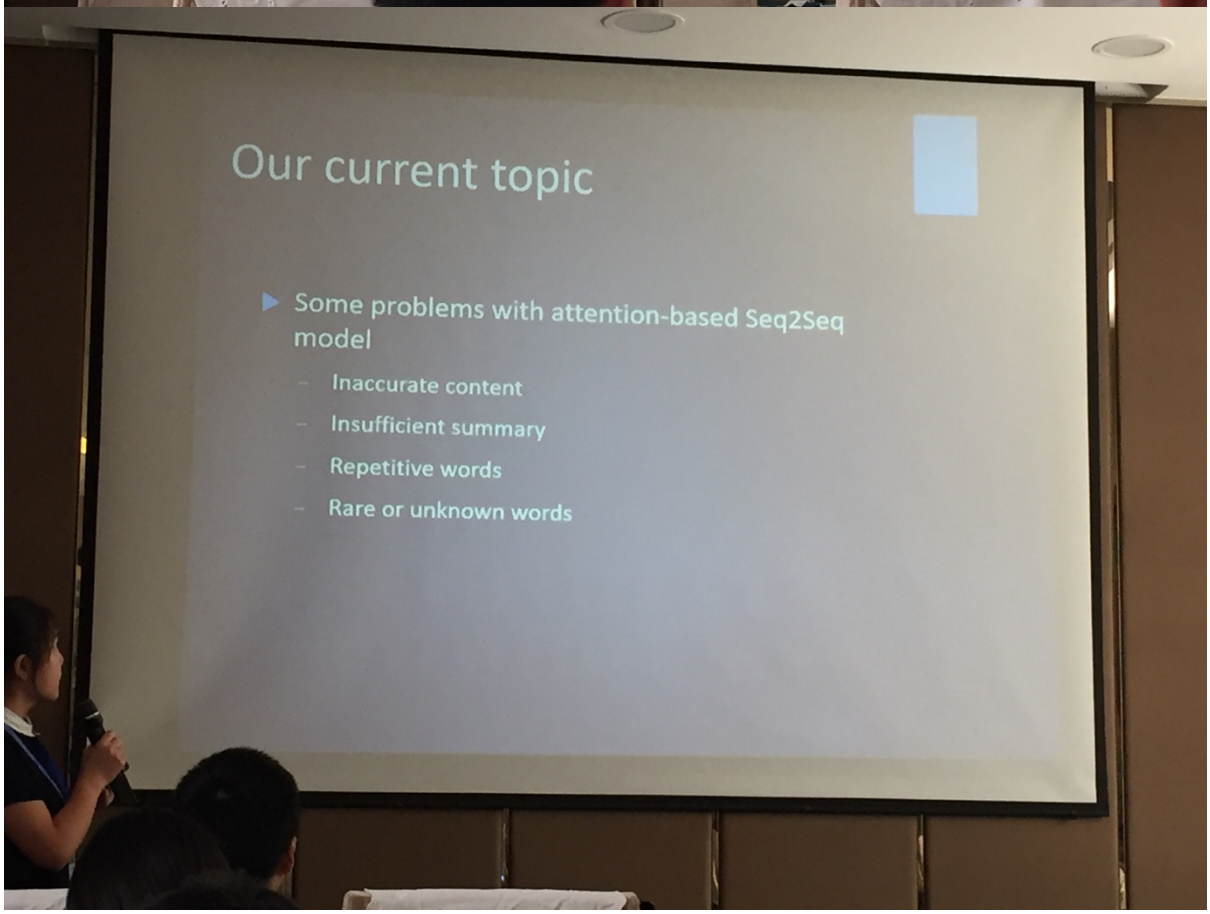
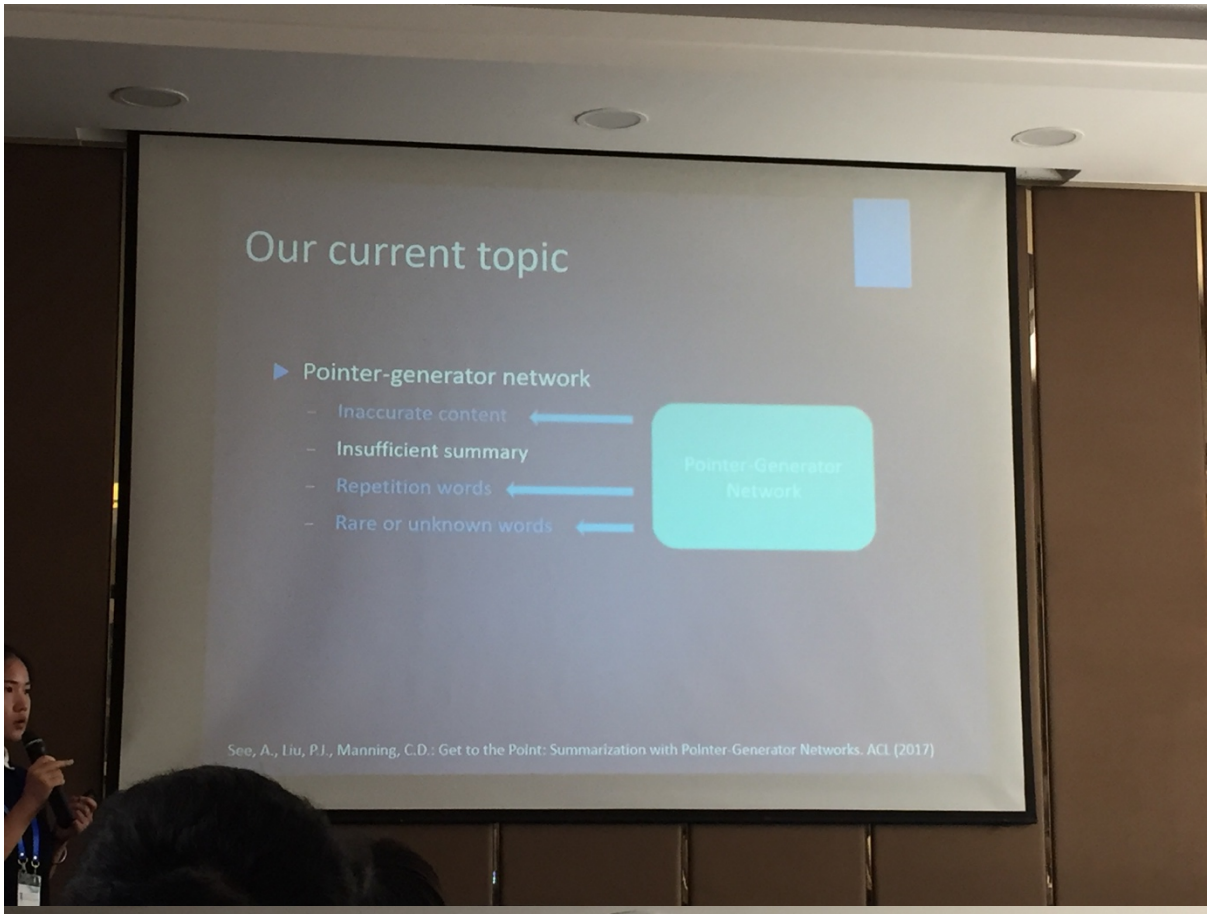
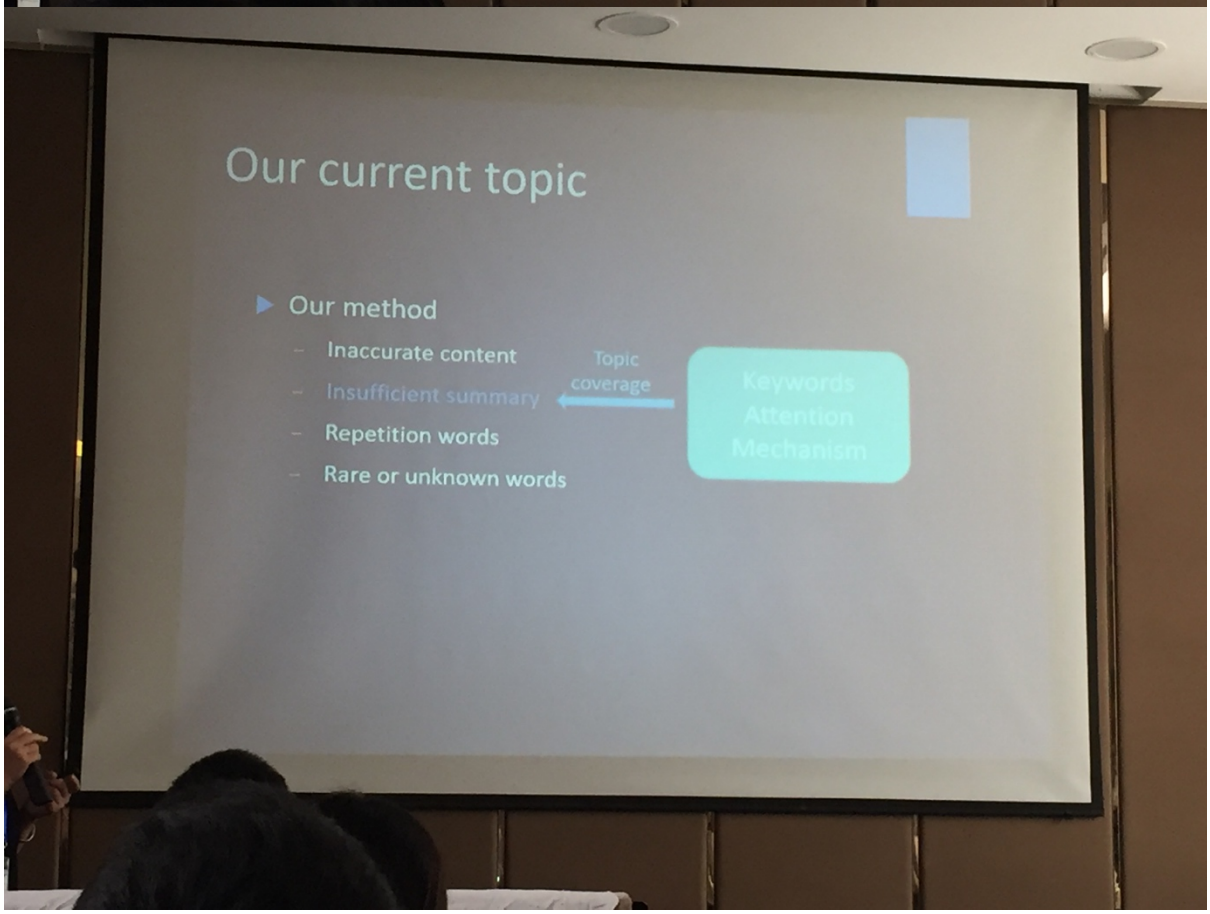

mproving Pointer-Generator Network
with Keywords Information for
Chinese Abstractive Summarization
Xiaoping Jiang, Po Hu, Liwei Hou and Xia Wang
School of Computer Science, Central China Normal University,
Wuhan 430079, China
{ jiangxp, houliwei }@mails.ccnu.edu.cn
phu@mail.ccnu.edu.cn, wangx_cathy@163.com


Our current topic
Some problems with attention-based Seq2Seq model
Inaccurate content
Insufficient summary
Repetitive words
Rare or unknown words


Our current topic
Pointer-generator network
Inaccurate content
Insufficient summary
Repetition words
Rare or unknown words
Pointer-Generator Network
See, A., Liu, P.J., Manning, C.D.: Get to the Point: Summarization with Pointer-Generator Networks. ACL (2017)


Our current topic
Our method
Inaccurate content
Insufficient summary
Repetition words
Rare or unknown words
Topic coverage
Keywords Attention Mechanism

Pointer-generator network

- Attention distribution

$$e_i^t = v^T tanh(W_h h_i + W_s s_t + W_c c_i^t + b_{attn})$$

$$a^t = softmax(e^t)$$

- Generation probability

$$P_{gen} = \sigma(w_h^T h_t^* + w_s^T s_t + w_x^T x_t + b_{ptr})$$

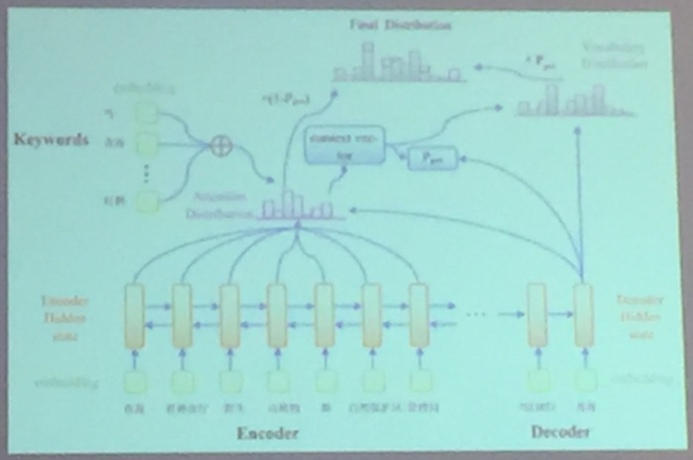
$$P(w) = p_{gen} P_{vocab}(w) + (1 - p_{gen}) \sum_{i:w_i} a_i^t$$

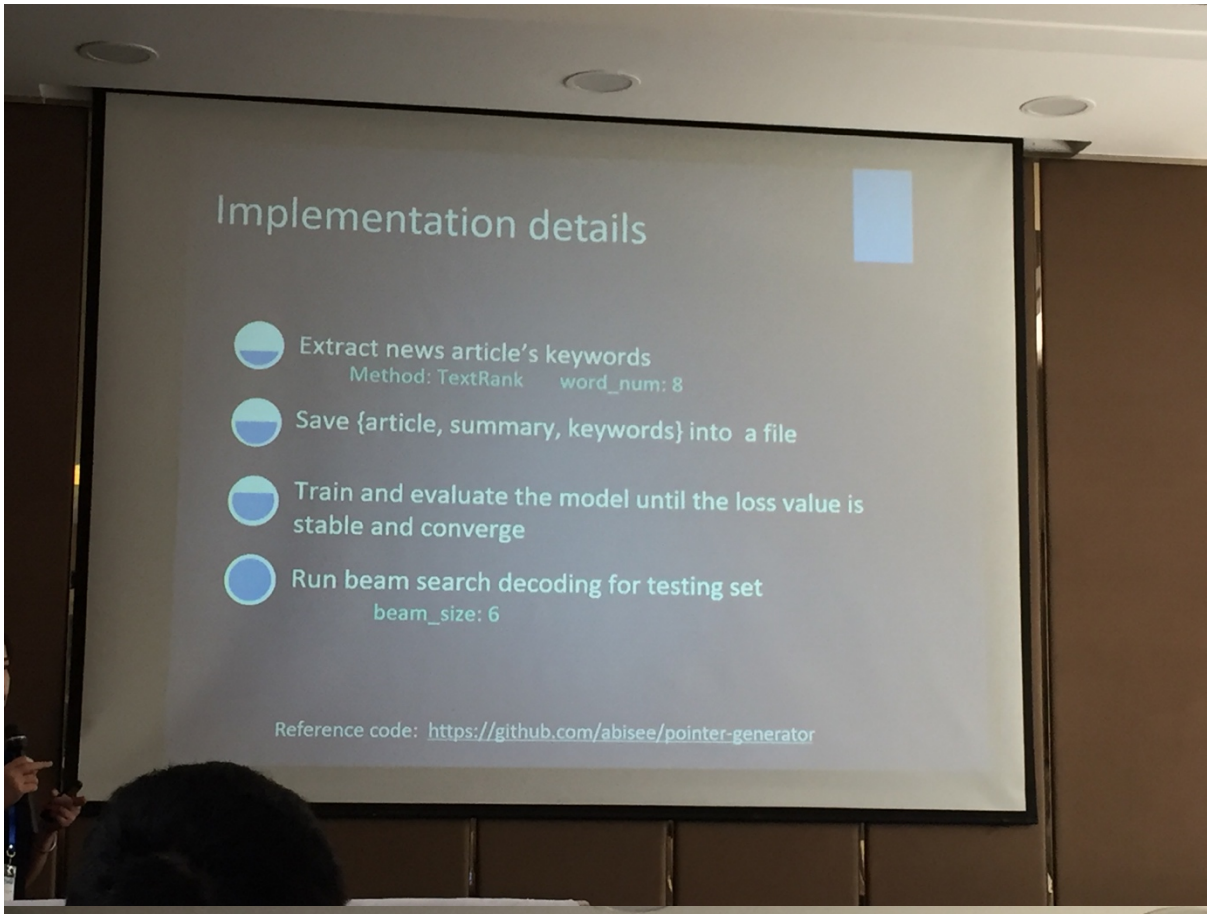
Keywords attention mechanism

- Attention distribution

$$e_i^t = v^T tanh(W_h h_i + W_s s_t + W_c c_i^t + W_t t + b_{attn})$$

$$t = \sum_{i=1}^{d} k_i$$

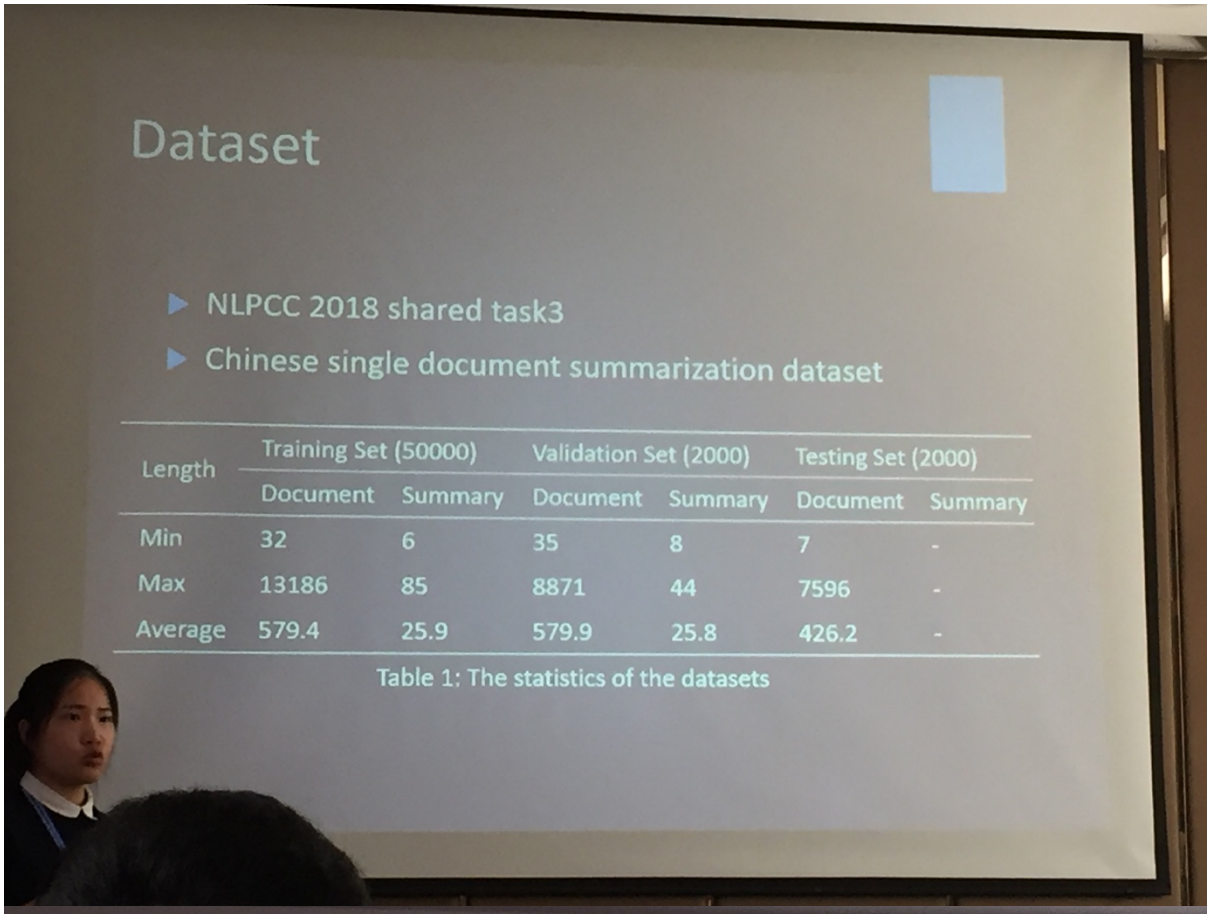


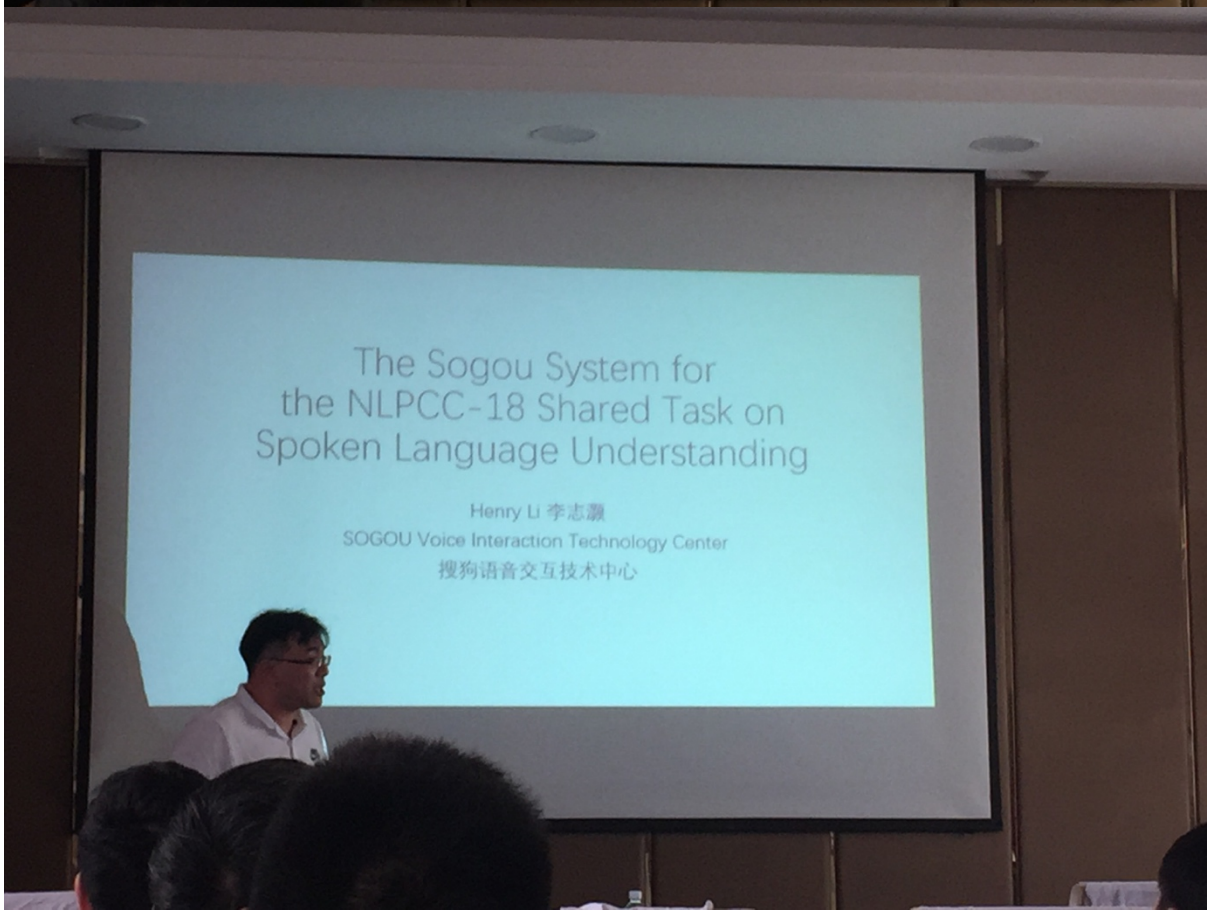


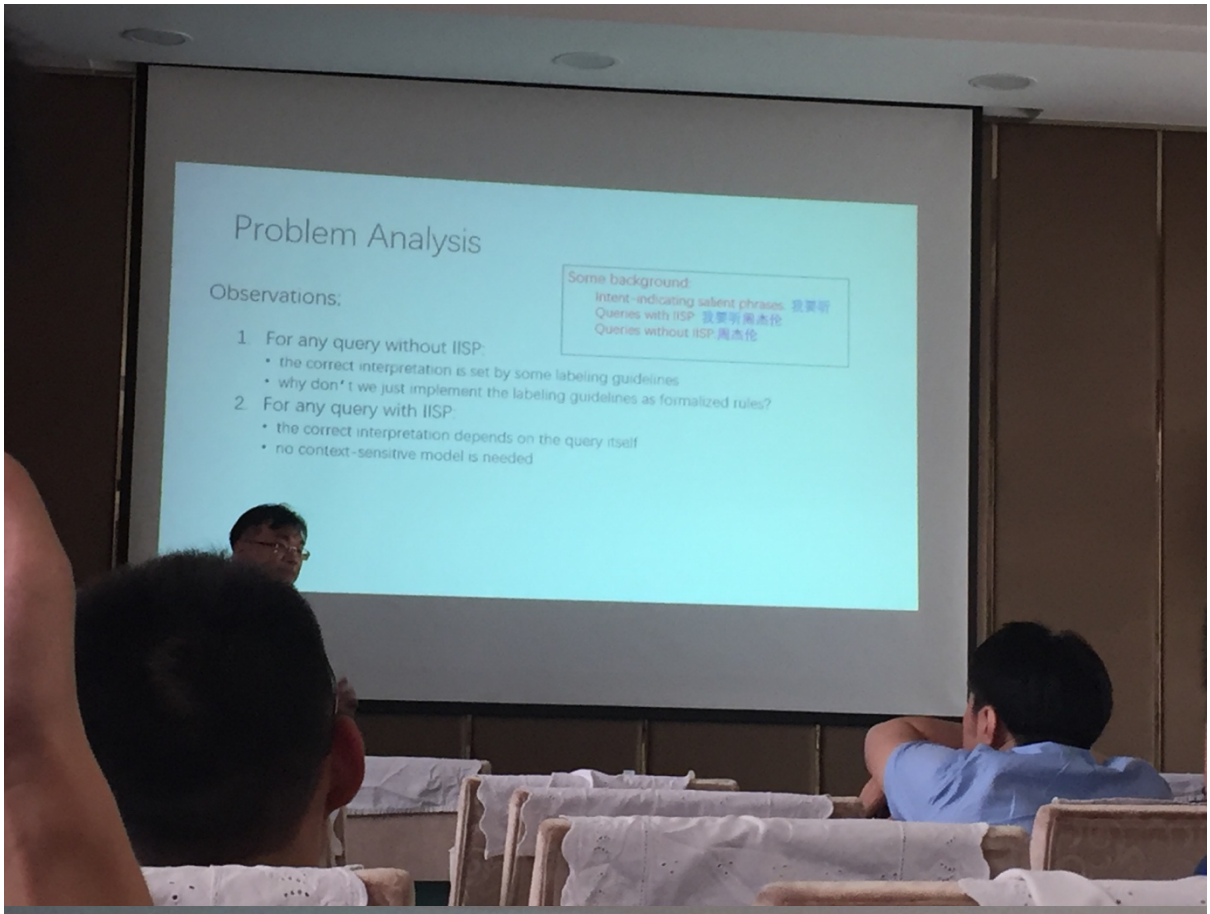
Experimental Results

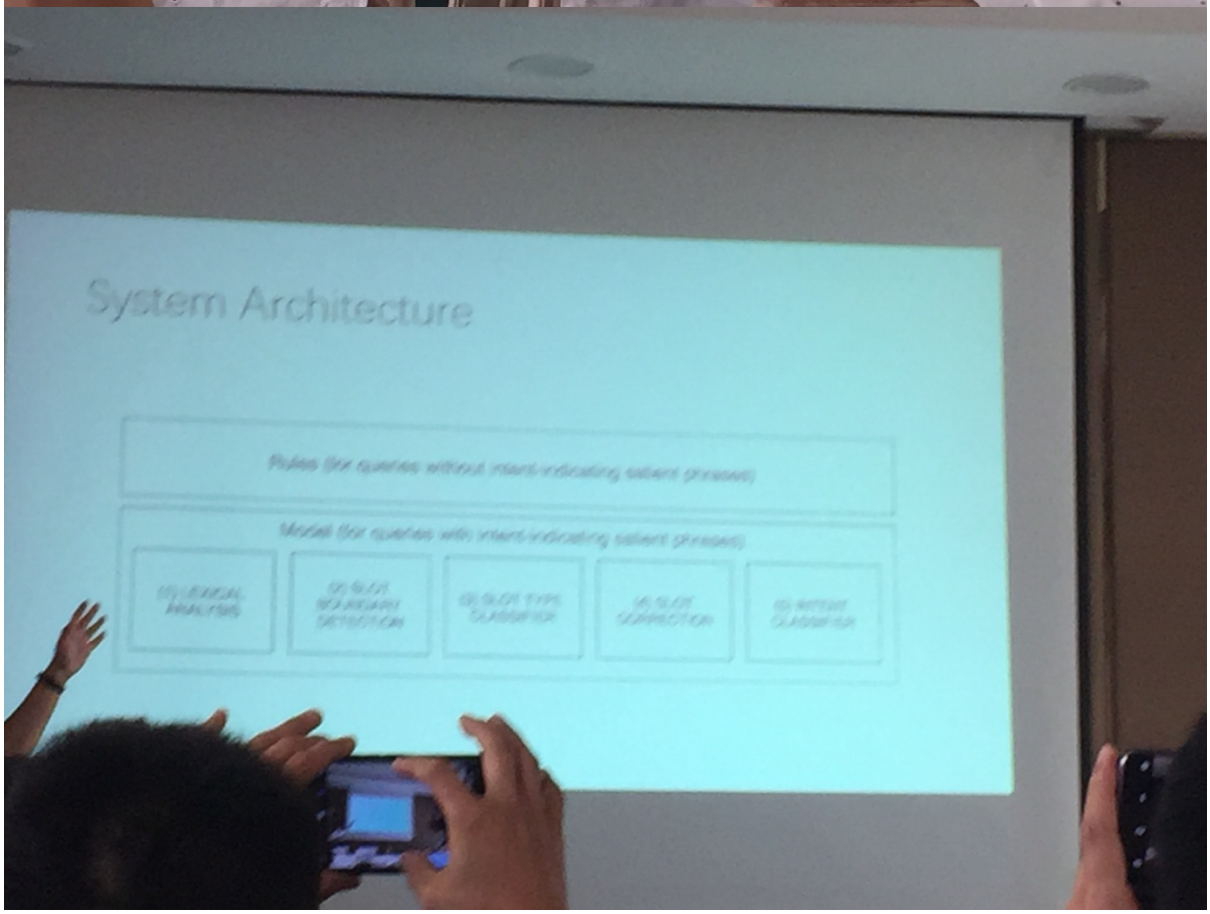
Models	R-1	R-2	R-L
TextRank	35.78	23.58	29.60
Pointer-Generator	43.44	29.46	37.66
Pointer-Generator (character)	38.01	24.43	32.32
TextRank + Pointer-Generator	42.79	28.93	37.33
Our Model	**44.60**	**30.46**	**38.83**

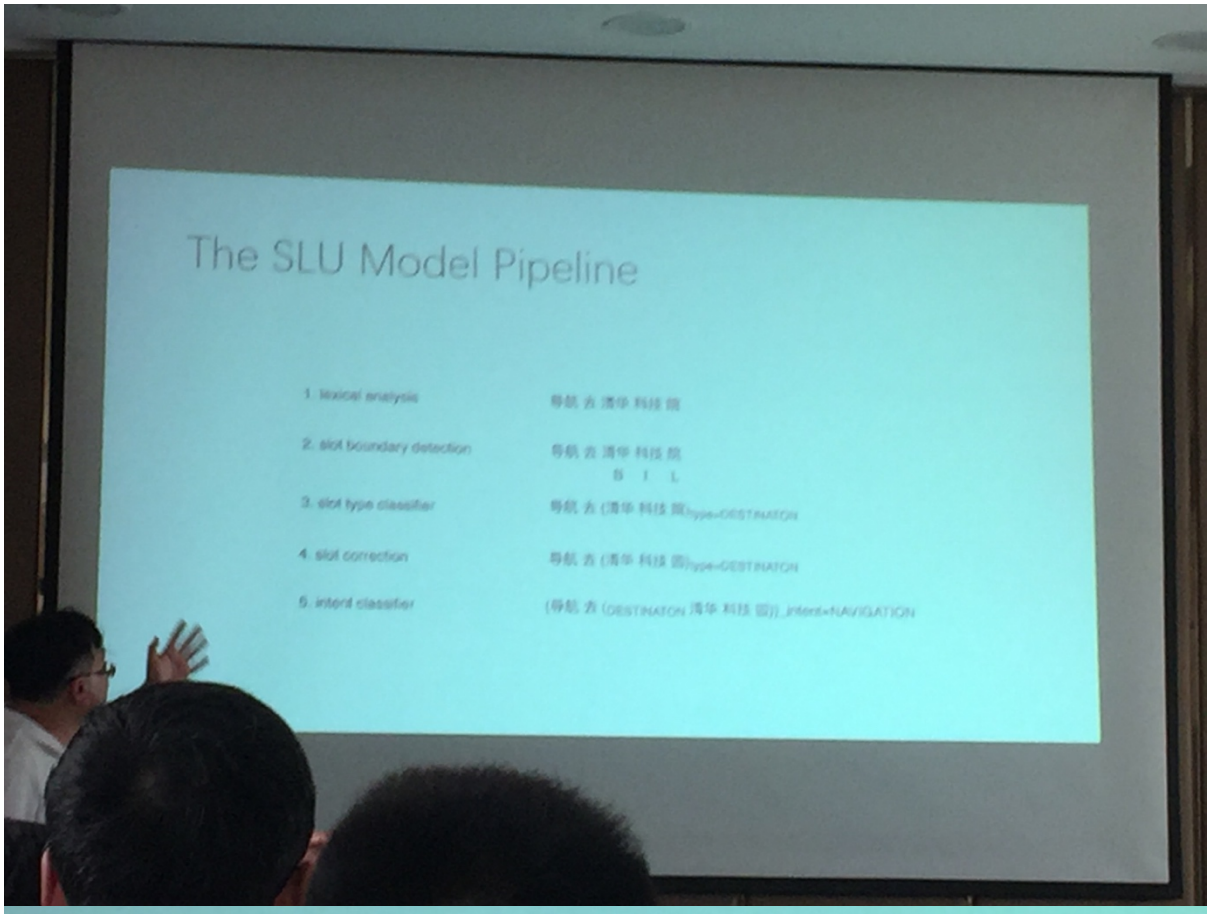
Table 2: Comparison results on the NLPCC 2018 validation dataset

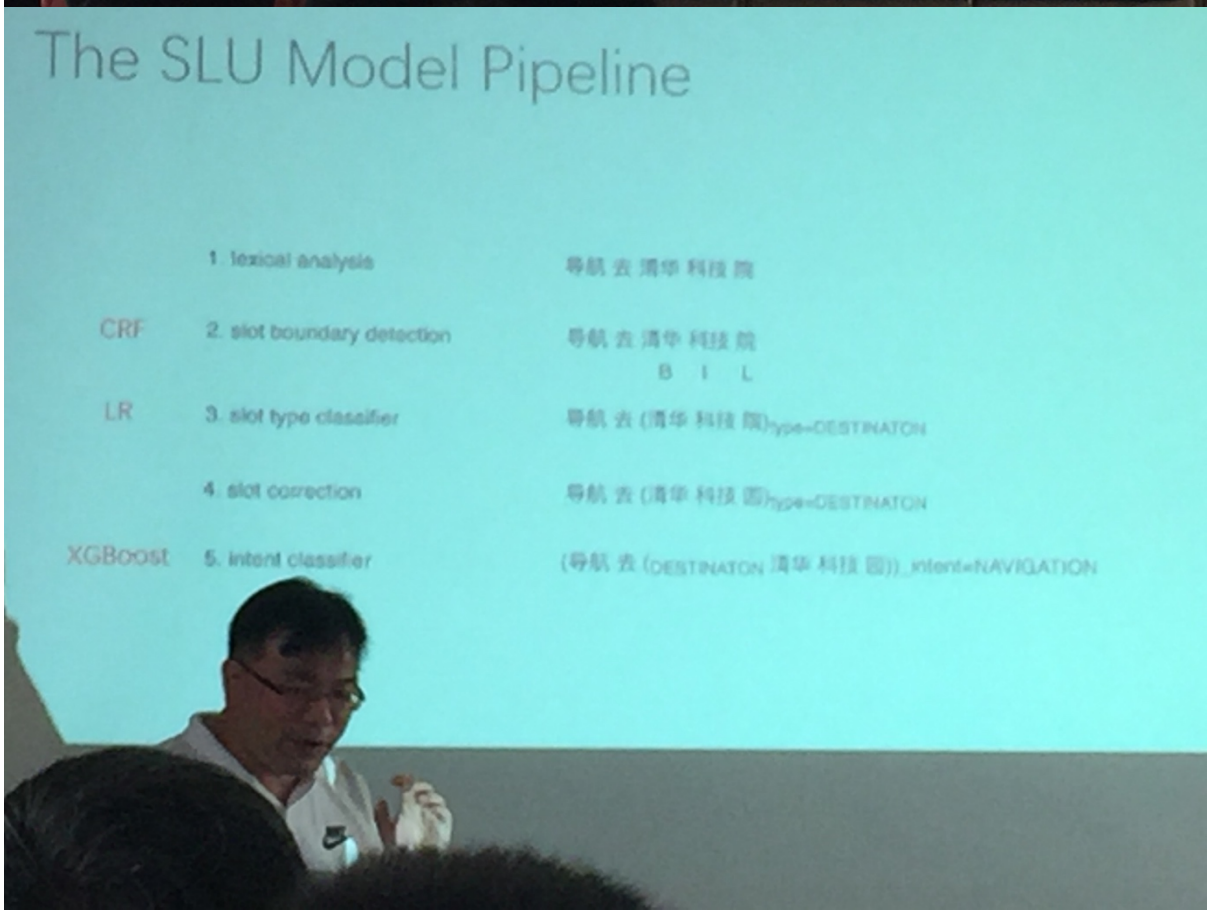

Dataset
NLPCC 2018 shared task3
Chinese single document summarization dataset
Length
Training Set (50000)
Validation Set (2000)
Testing Set (2000)
Document
Summary
Document
Summary
Document
Summary
Min 32 6 35 8 7 -
Max 13186 85 8871 44 7596 -
Average 579.4 25.9 579.9 25.8 426.2 -
Table 1: The statistics of the datasets

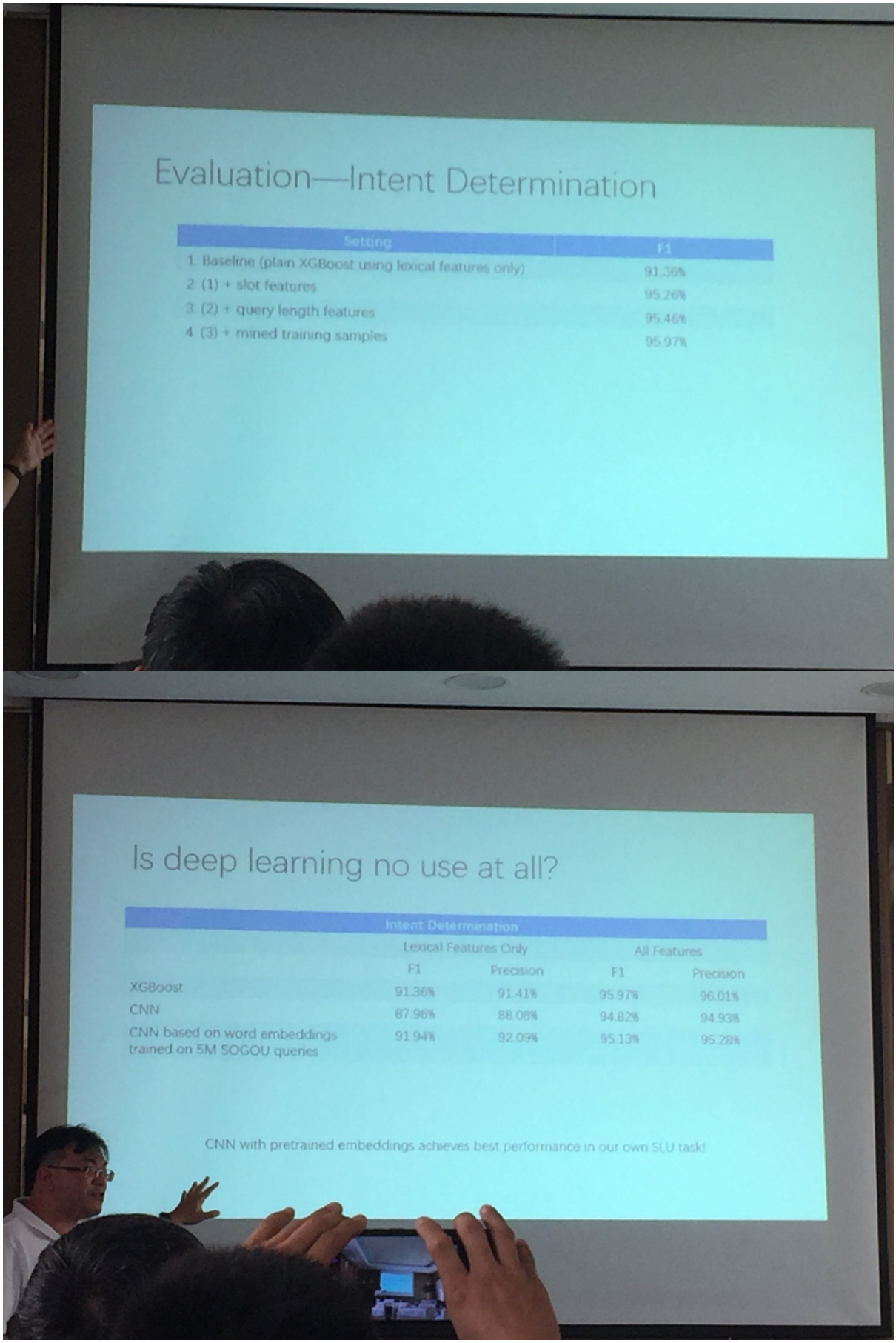

The Sogou System for
the NLPCC-18 Shared Task on
Spoken Language Understanding
Henry Li 李志灏
SOGOU Voice Interaction Technology Center
搜狗语音交互技术中心

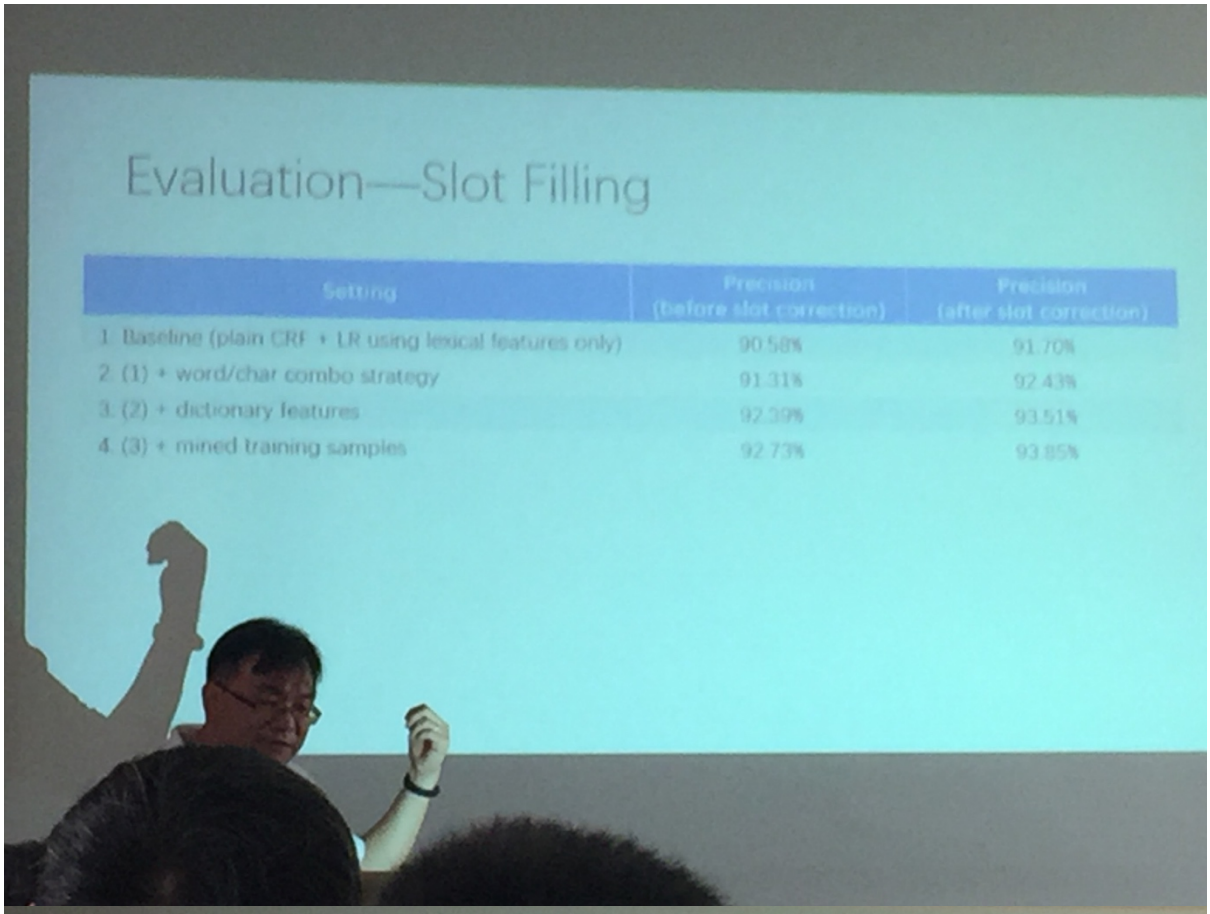

Problem Analysis
Observations:
1. For any query without IISP:
• the correct interpretation is set by some labeling guidelines
• why don't we just implement the labeling guidelines as formalized rules?
2. For any query with IISP:
• the correct interpretation depends on the query itself
• no context-sensitive model is needed
Some background:
Intent-indicating salient phrases: 我要听
Queries with IISP: 我要听周杰伦
Queries without IISP:周杰伦

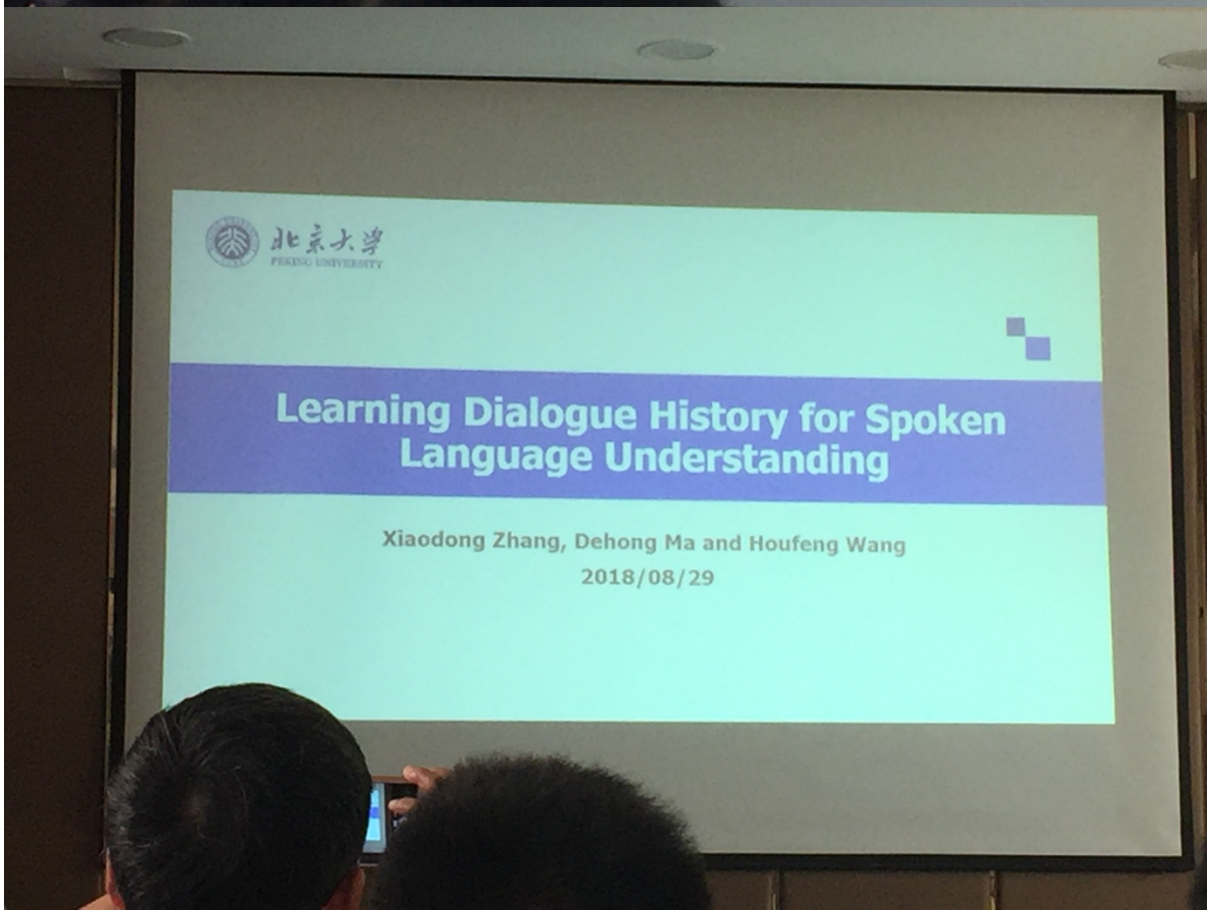

System Architecture


The SLU Model Pipeline
1. lexical analysis
导航 去 清华 科技 园
2. slot boundary detection
导航 去 清华 科技 园
B I L
3. slot type classifier
导航 去 (清华 科技 园)type=DESTINATON
4. slot correction
导航 去 (清华 科技 园)type=DESTINATON
5. intent classifier
(导航 去 (DESTINATON 清华 科技 园)) intent=NAVIGATION


The SLU Model Pipeline
1. lexical analysis
导航 去 清华 科技 园
CRF
2. slot boundary detection
导航 去 清华 科技 园
B I L
LR
3. slot type classifier
导航 去 (清华 科技 园)type=DESTINATON
4. slot correction
导航 去 (清华 科技 园)type=DESTINATON
XGBoost
5. intent classifier
(导航 去 (DESTINATON 清华 科技 园)) intent=NAVIGATION


Evaluation—Intent Determination
Setting
F1
1. Baseline (plain XGBoost using lexical features only)
91.36%
2. (1) + slot features
95.26%
3. (2) + query length features
95.46%
4. (3) + mined training samples
95.97%
Is deep learning no use at all?
Intent Determination
Lexical Features Only
All Features
F1
Precision
F1
Precision
XGBoost
91.36%
91.41%
95.97%
96.01%
CNN
87.96%
88.08%
94.82%
94.93%
CNN based on word embeddings trained on 5M SOGOU queries
91.94%
92.09%
95.13%
95.28%
CNN with pretrained embeddings achieves best performance in our own SLU task!


Evaluation—Slot Filling
Setting
Precision (before slot correction)
Precision (after slot correction)
1. Baseline (plain CRF + LR using lexical features only)
90.58%
91.70%
2. (1) + word/char combo strategy
91.31%
92.43%
3. (2) + dictionary features
92.39%
93.51%
4. (3) + mined training samples
92.73%
93.85%


北京大学
PEKING UNIVERSITY
Learning Dialogue History for Spoken Language Understanding
Xiaodong Zhang, Dehong Ma and Houfeng Wang
2018/08/29

Introduction

- Spoken language understanding
  - Converting users' queries expressed by natural language to structured representations.
- Two subtasks:
  - Intent Identification (classification)
  - Slot filling (sequence labeling)

Utterance	Go	to	Peking	University	from	the	Forbidden	City
Slots	O	O	B-dest	I-dest	O	B-orig	I-orig	I-orig
Intent	Navigation							

2

Motivation

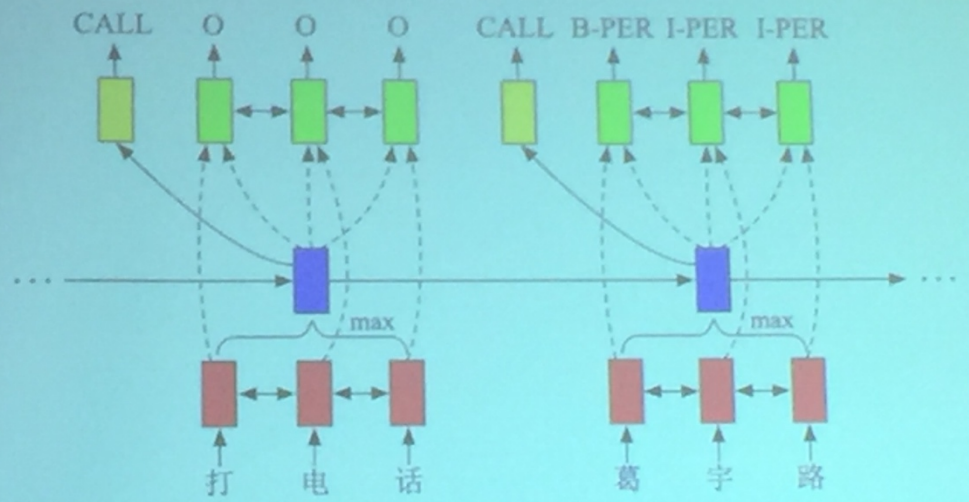
- Dialogue history is important to the understanding of the current utterance.

ID	Dialogue
1	A: Go to Peking University. B: Planning route to Peking University. A: Cancel.
2	A: Call Tom. B: Calling Tom for you. A: Cancel.
3	A: Where are you going? B: Washington.
4	A: Who would you like to call? B: Washington.

- How to use dialogue history for the two subtasks?

3

HLSTM-SLU



4

Input

- HLSTM-SLU processes Chinese text at character level.
- The input contains two parts:
  - Characters
  - Additional features: part-of-speech and domain lexicons
- An example:

		B-A	I-A	B-N	I-N	...	B-song	I-song	...
小	...	1	0	0	0	...	1	0	...
苹	...	0	0	1	0	...	0	1	...
果	...	0	0	0	1	...	0	1	...

- The sparse feature vector is convert to a low-dimensional dense vector via a linear transformation.

5

Model Details

- A character-level LSTM to learn representation of a turn:

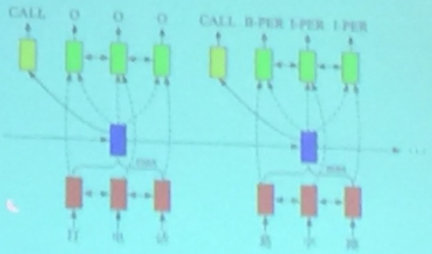
$$h^c_{i,j} = BLSTM(x_{i,j}, h^c_{i,j-1}, h^c_{i,j+1})$$

- A fixed-length representation via max-pooling:

$$r^T_i = \max_{k=1}^{n} h^c_{i,k}$$

- A turn-level LSTM for a dialog:

$$r^D_i = LSTM(r^T_i, r^D_{i-1})$$



Intent Tagger

- A fully-connected layer for calculating scores at each turn:

$$s^T_i = W^T_s r^D_i + b^T_s$$

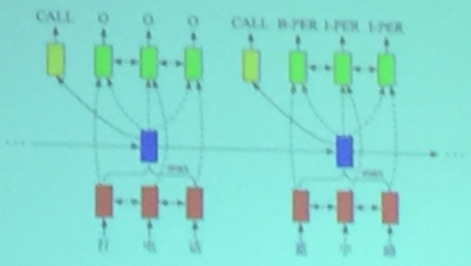
- A CRF layer for modeling label dependency between turns:

$$S(y^T) = \sum_{i=1}^{m} s^T_{i,y^T_i} + \sum_{i=1}^{m-1} A_{y^T_i, y^T_{i+1}}$$

- Training and testing:

$$p(y^T) = \frac{e^{S(y^T)}}{\sum_{\tilde{y}^T \in Y^T} e^{S(\tilde{y}^T)}}$$

$$\hat{y}^T = \underset{\tilde{y}^T \in Y^T}{\operatorname{argmax}} S(\tilde{y}^T)$$



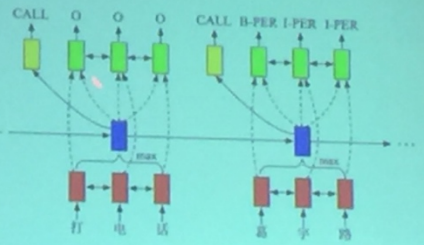
Slot Tagger

- For slot filling, the model rereads a turn with the contextual turn representation:

$$h^d_{i,j} = BLSTM(x^d_{i,j}, h^d_{i,j-1}, h^d_{i,j+1})$$
$$x^d_{i,j} = [h^e_{i,j}; r^D_i]$$

- A CRF layer is also used for predict slot tags for each turn.
- The loss function of is the sum of loss for intent and slot:

$$L = -\frac{1}{m}\left(\log(p(y^T)) + \sum_{i=1}^{m}\log(p(y^e_i))\right)$$



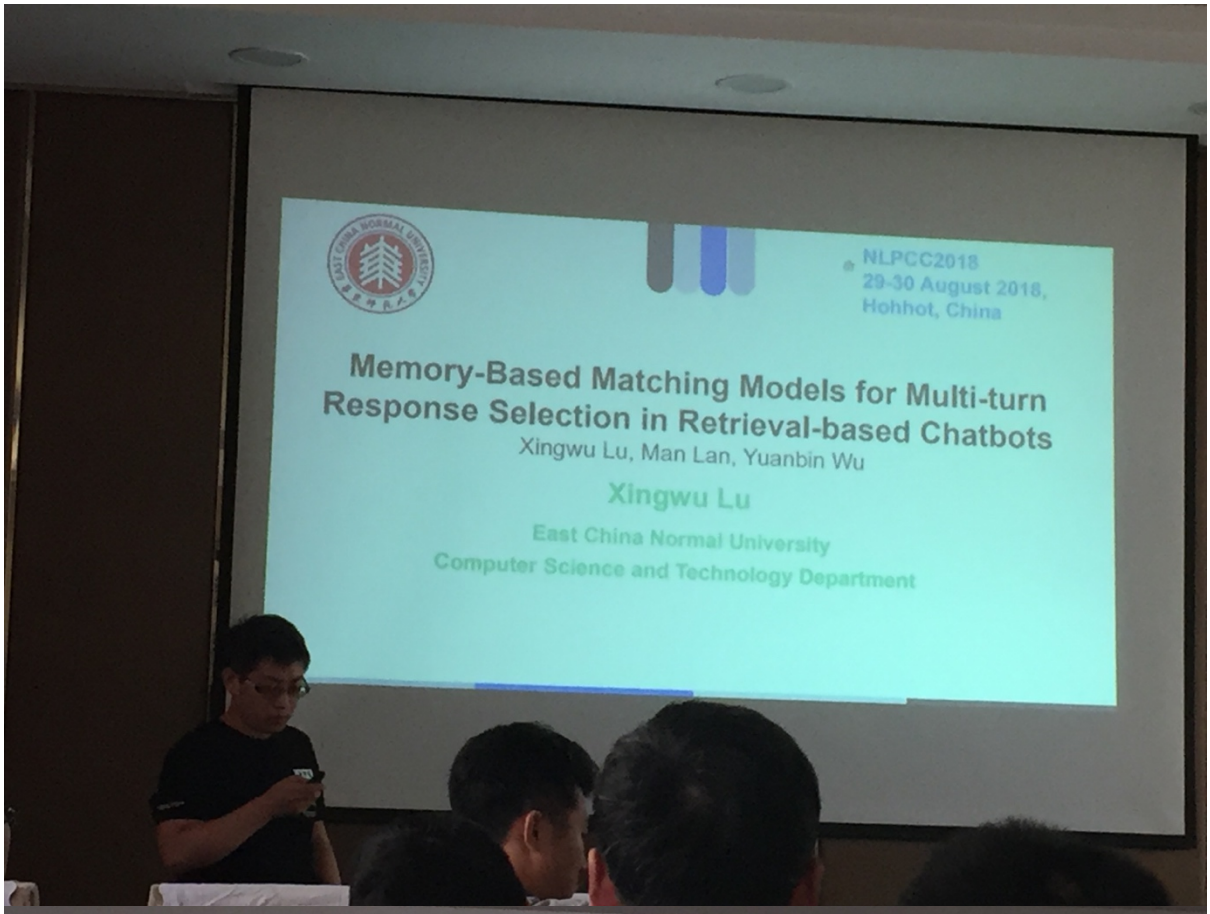
8

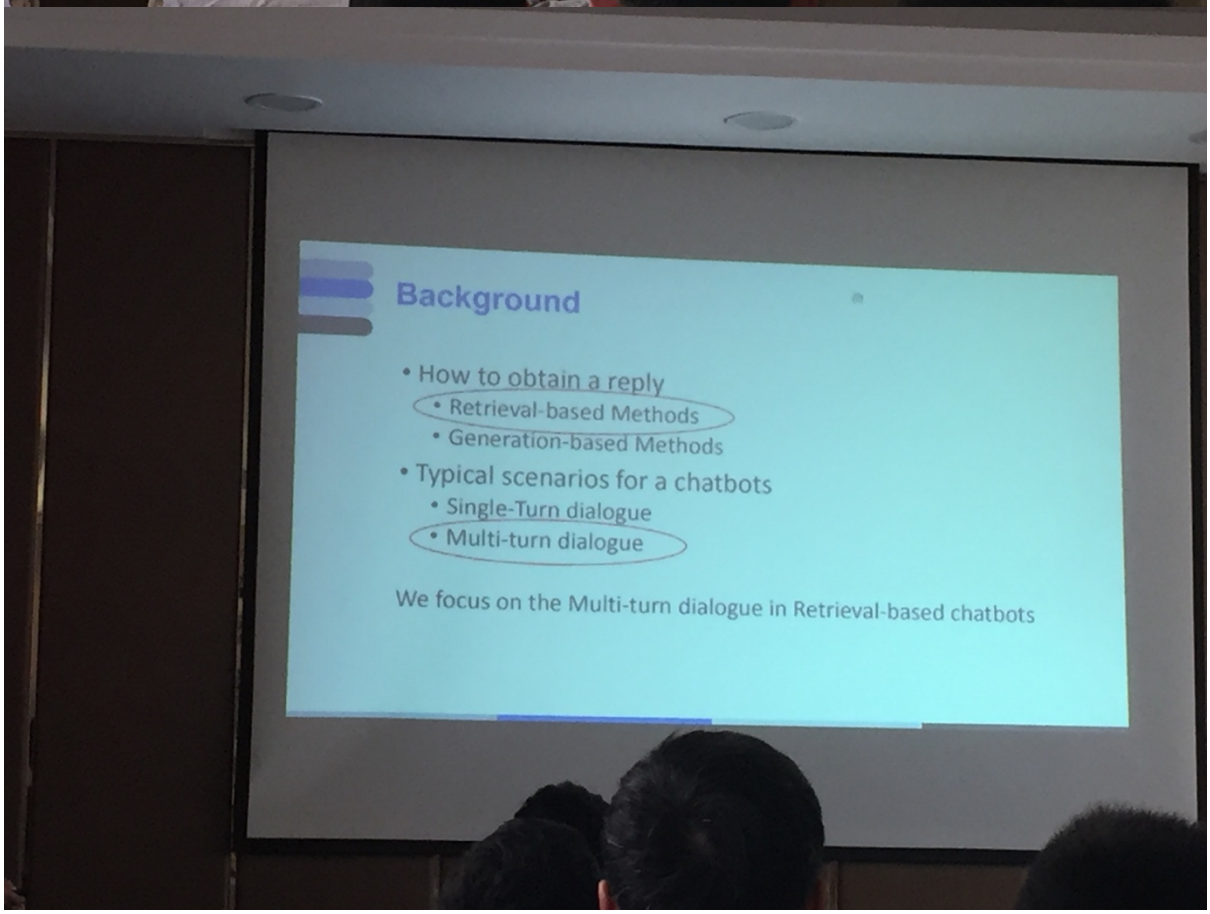
Experiments

- NLPCC 2018 Shared Task 4 dataset:
  - 4705 dialogues with 21352 turns in the training set and 1177 dialogues with 5350 turns in the testing set.
  - We do not use additional data.

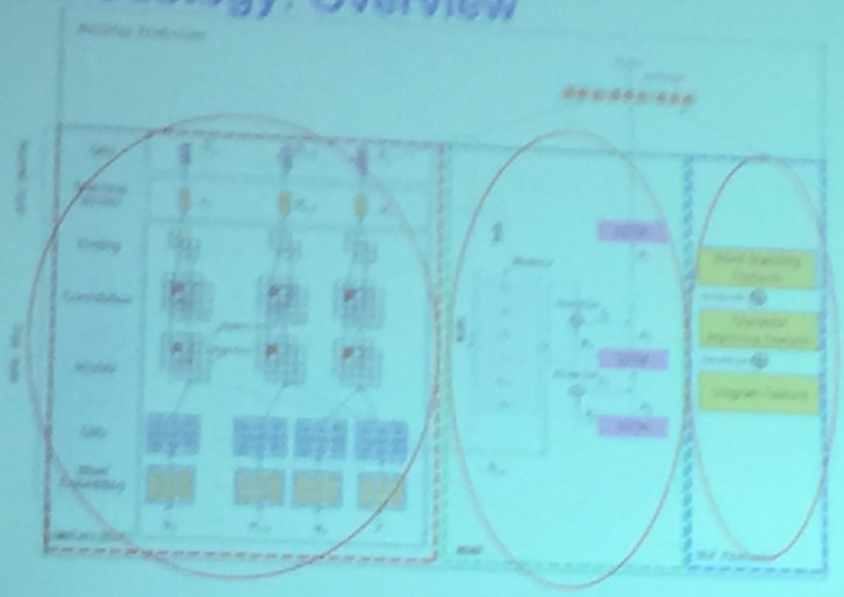
Method	F1 for Intent	Precision for Intent & Slot
BLSTM	88.56	83.38
BLSTM-FT	89.24	86.95
BLSTM-CRF	89.31	88.62
HLSTM-Intent	93.61	90.21
HLSTM-SLU	94.19	90.84

9

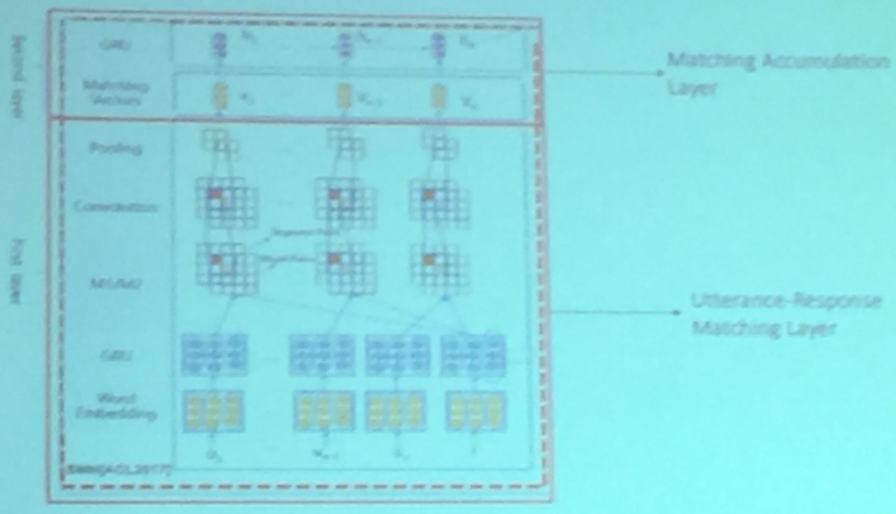

NLPCC2018
29-30 August 2018,
Hohhot, China
Memory-Based Matching Models for Multi-turn Response Selection in Retrieval-based Chatbots
Xingwu Lu, Man Lan, Yuanbin Wu
Xingwu Lu
East China Normal University
Computer Science and Technology Department


Background
• How to obtain a reply
• Retrieval-based Methods
• Generation-based Methods
• Typical scenarios for a chatbots
• Single-Turn dialogue
• Multi-turn dialogue
We focus on the Multi-turn dialogue in Retrieval-based chatbots

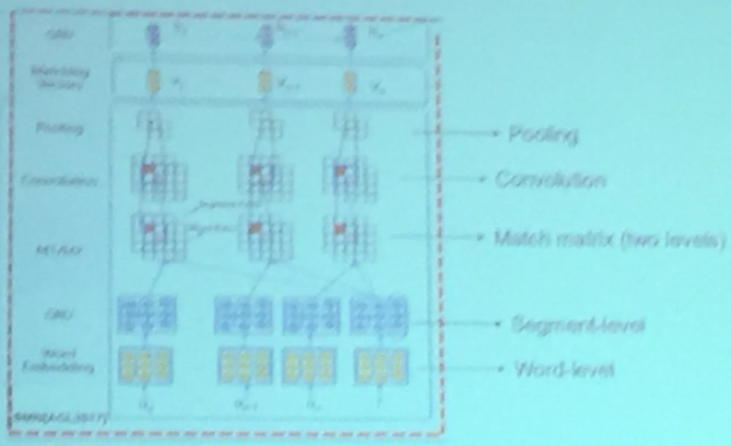
Methodology: Overview



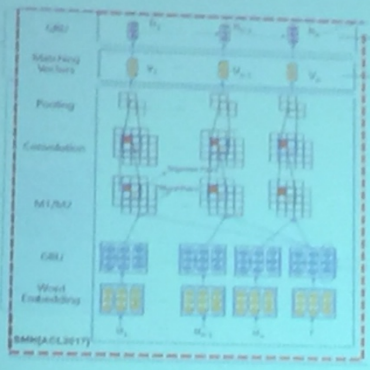
Sequential Matching Network (SMN)



SMN Module: First Layer



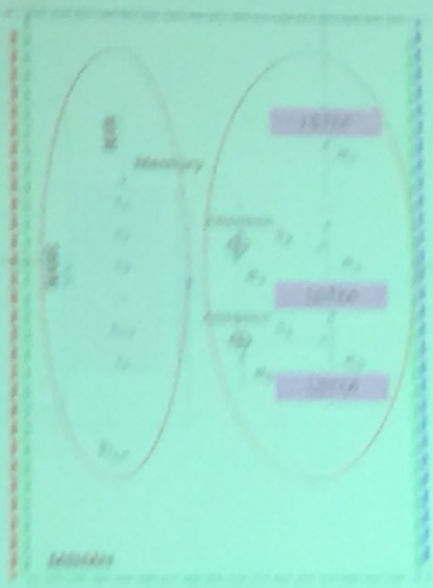
SMN Module: Second Layer



Memory-Based Matching Network (MBMN)

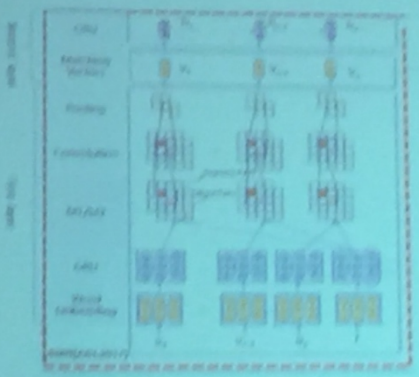
two steps

Memory Building



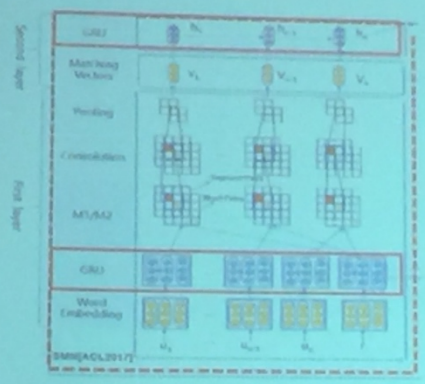
Apply Multiple Attentions on the Memory

MBMN Module: Memory Building



Matching Vectors Memory (MVM)

MBMN Module: Memory Building

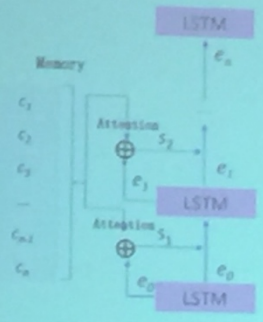


Sequential Matching Vectors Memory (SMVM)

$$c_i = tanh(W_{1,1}h_{u_i,n_u} + W_{1,2}h_t + b_t)$$

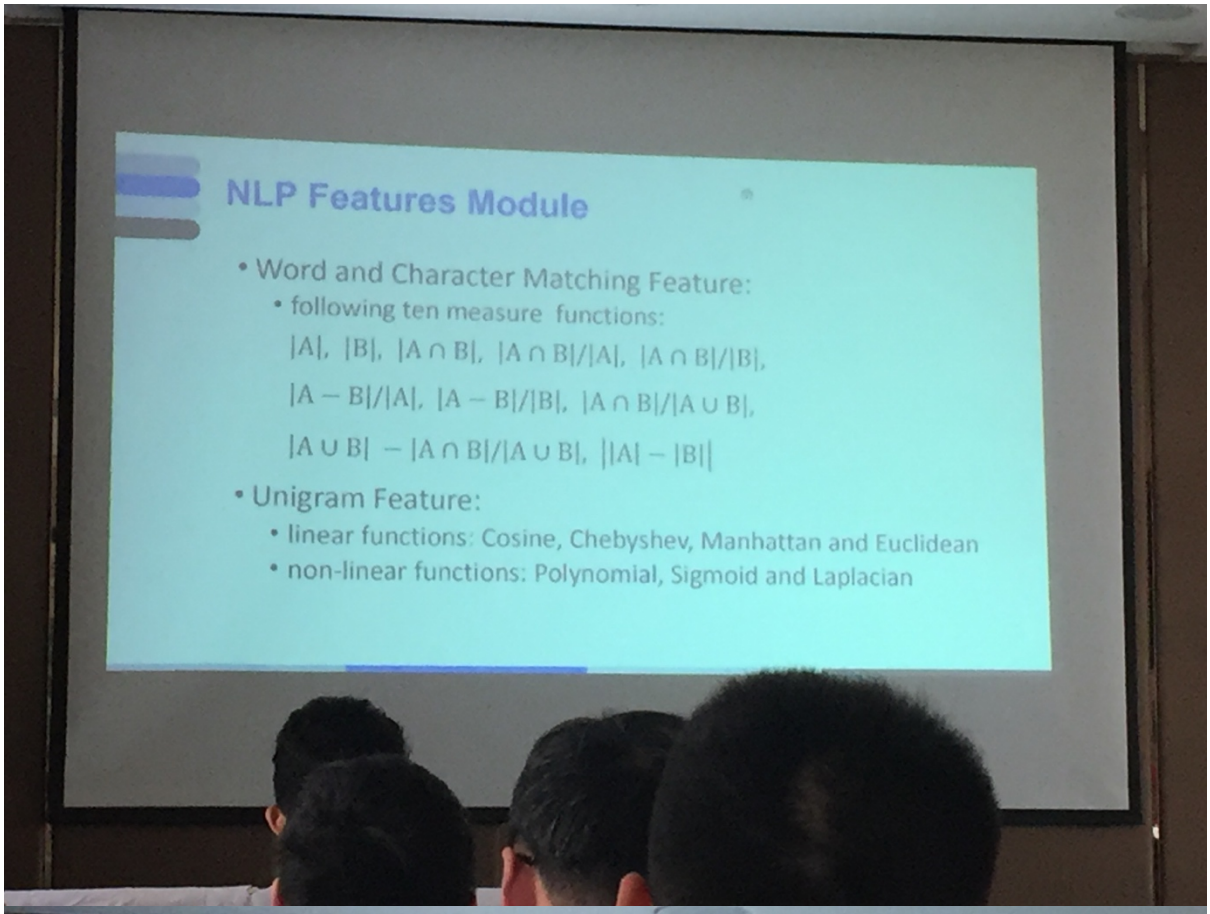
MBMN Module: Multiple Attentions on Memory

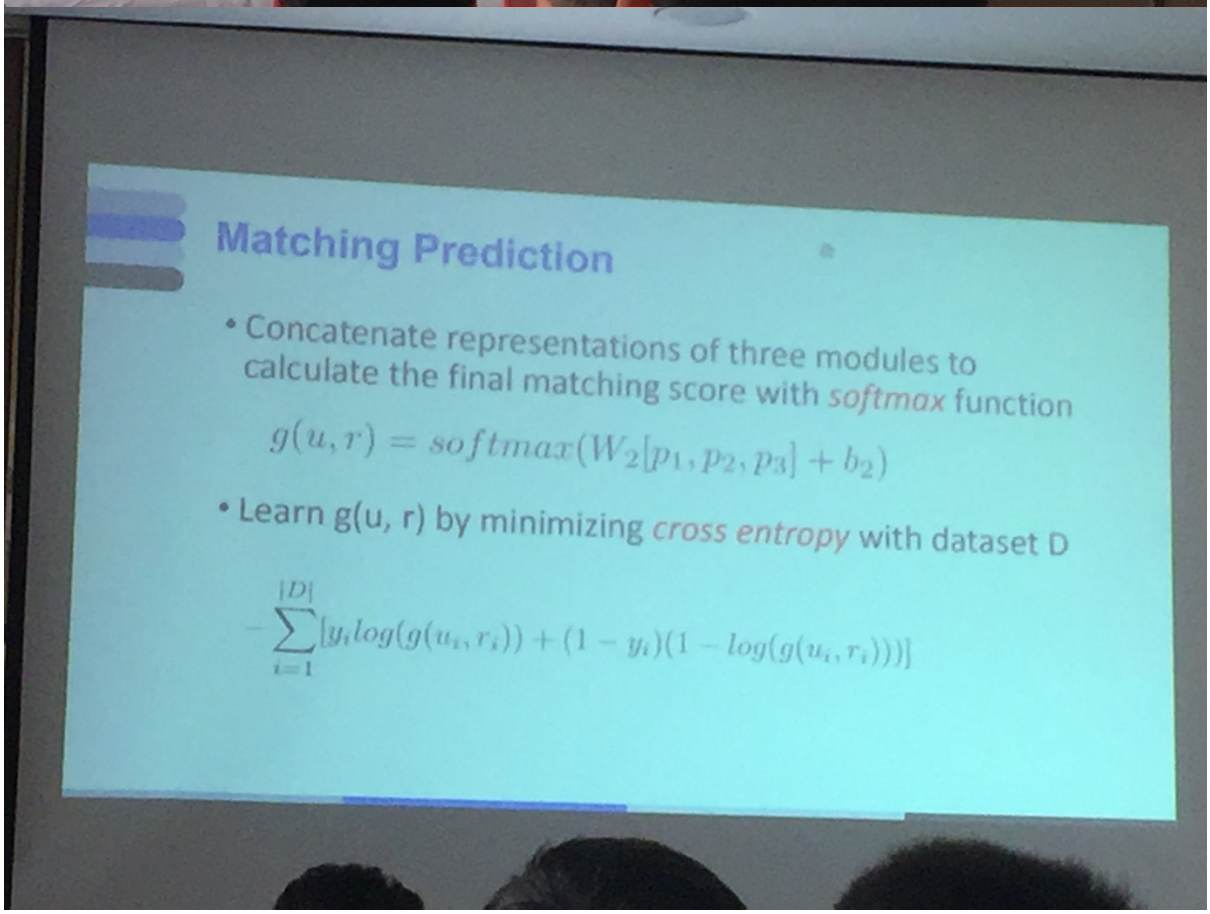
$$g_i^t = v^T tanh(W_c c_i + W_e e_{t-1} + W_r h_{r,n} + b_{attn})$$

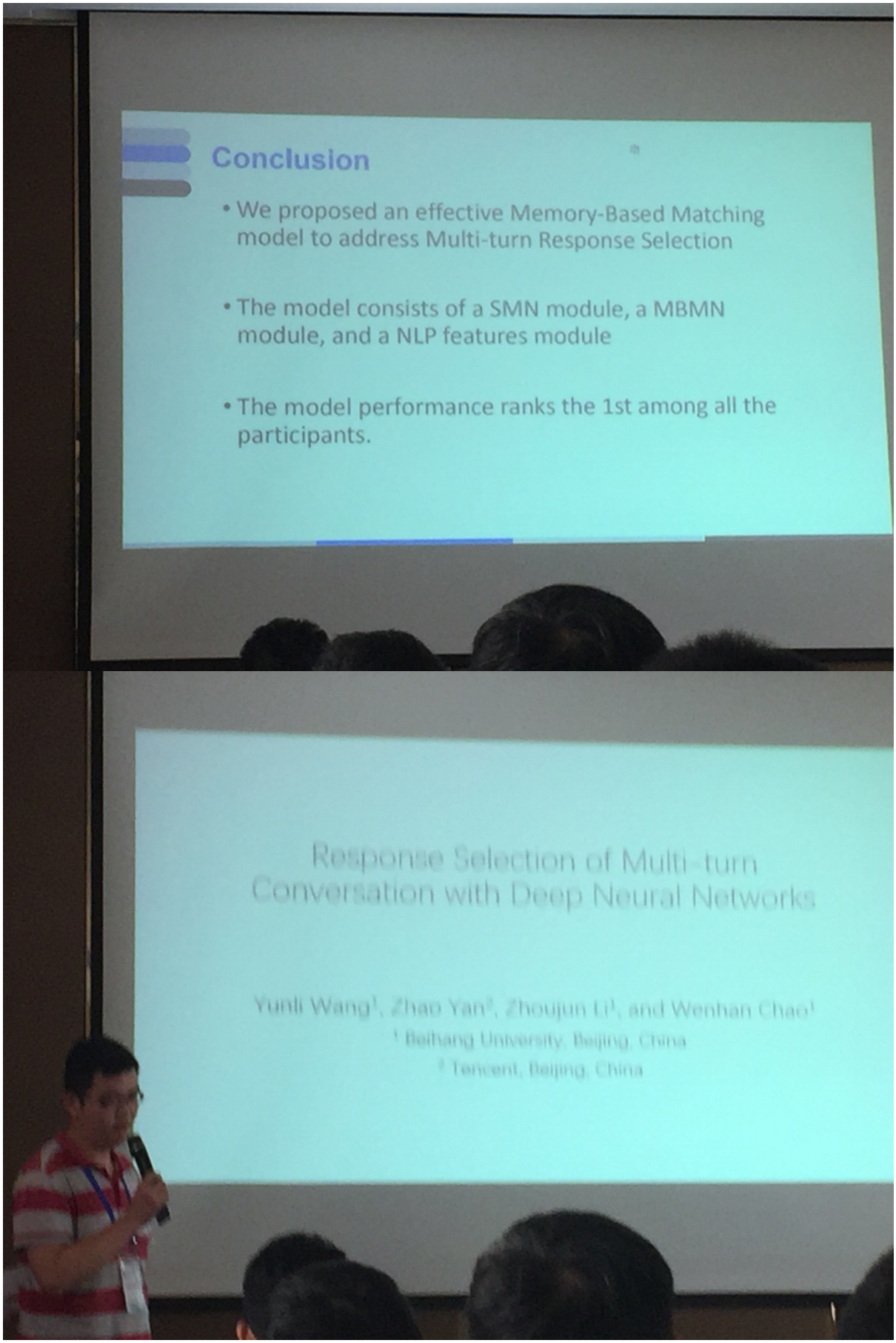


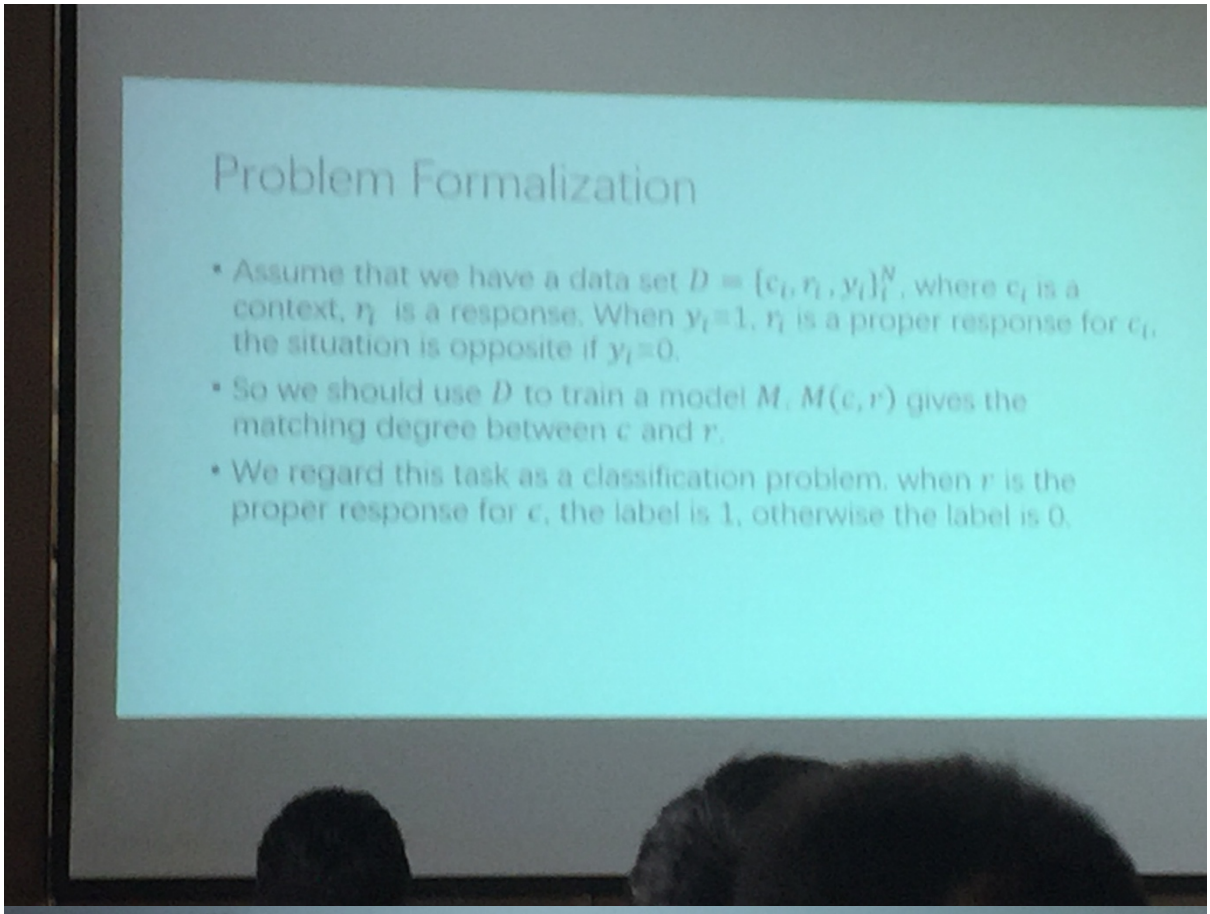
$$\alpha_{t_i} = \frac{exp(g_i^t)}{\sum_{j=1}^{I} exp(g_j^t)}$$

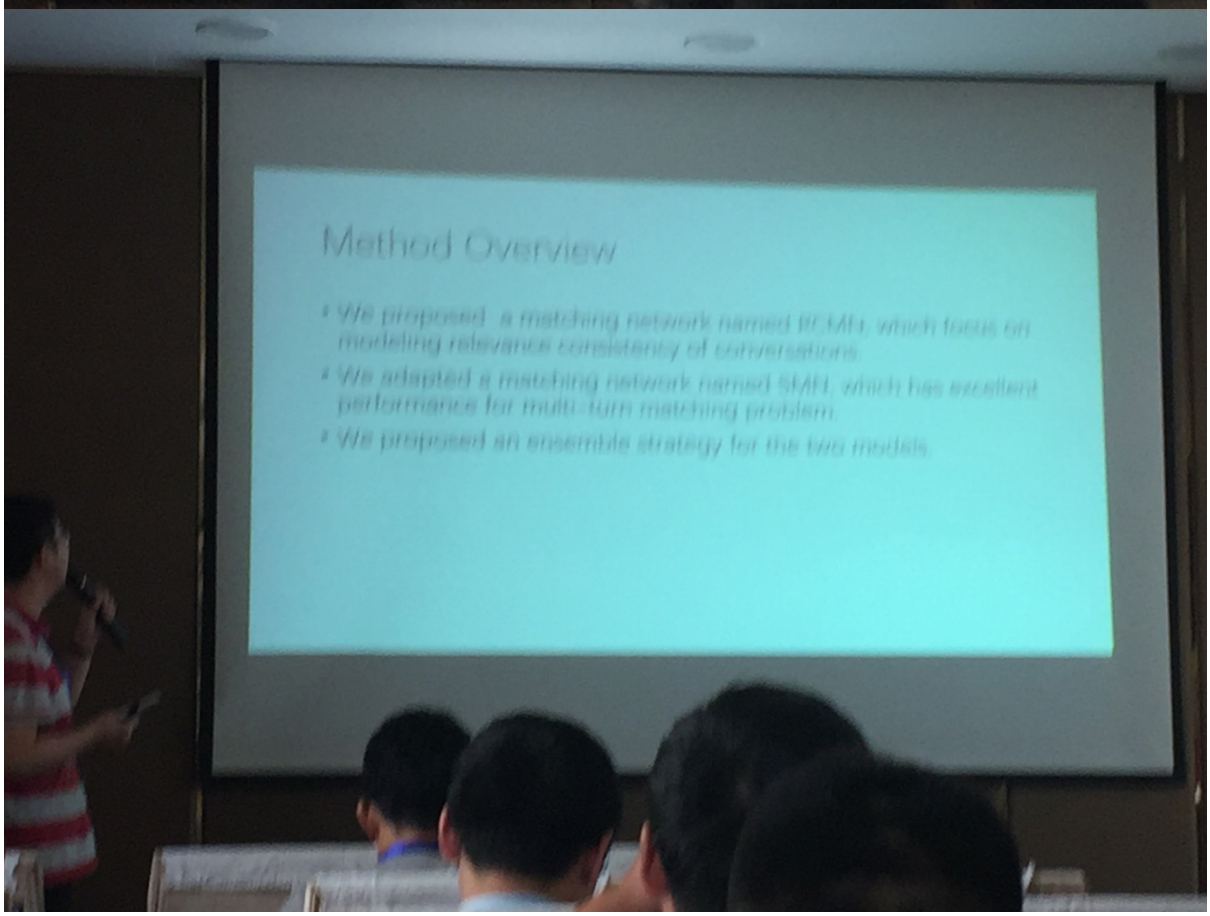
$$s_t = \sum_{i=1}^{N} \alpha_{t_i} c_i$$


NLP Features Module
• Word and Character Matching Feature:
• following ten measure functions:
|A|, |B|, |A ∩ B|, |A ∩ B|/|A|, |A ∩ B|/|B|,
|A − B|/|A|, |A − B|/|B|, |A ∩ B|/|A ∪ B|,
|A ∪ B| − |A ∩ B|/|A ∪ B|, ||A| − |B||
• Unigram Feature:
• linear functions: Cosine, Chebyshev, Manhattan and Euclidean
• non-linear functions: Polynomial, Sigmoid and Laplacian


Matching Prediction
• Concatenate representations of three modules to calculate the final matching score with softmax function
$g(u, r) = softmax(W_2[p_1, p_2, p_3] + b_2)$
• Learn g(u, r) by minimizing cross entropy with dataset D
$-\sum_{i=1}^{|D|}[y_i log(g(u_i, r_i)) + (1 - y_i)(1 - log(g(u_i, r_i)))]$


Conclusion
• We proposed an effective Memory-Based Matching model to address Multi-turn Response Selection
• The model consists of a SMN module, a MBMN module, and a NLP features module
• The model performance ranks the 1st among all the participants.
Response Selection of Multi-turn Conversation with Deep Neural Networks
Yunli Wang[1], Zhao Yan[2], Zhoujun Li[1], and Wenhan Chao[1]
[1] Beihang University, Beijing, China
[2] Tencent, Beijing, China


Problem Formalization
• Assume that we have a data set $D = \{c_i, r_i, y_i\}_i^N$, where c_i is a context, r_i is a response. When $y_i=1$, r_i is a proper response for c_i, the situation is opposite if $y_i=0$.
• So we should use D to train a model M. $M(c, r)$ gives the matching degree between c and r.
• We regard this task as a classification problem, when r is the proper response for c, the label is 1, otherwise the label is 0.


Method Overview
• We proposed a matching network named IECMN, which focus on modeling relevance consistency of conversations.
• We adapted a matching network named SMN, which has excellent performance for multi-turn matching problem.
• We proposed an ensemble strategy for the two models.

Relevance Consistency Matching Network

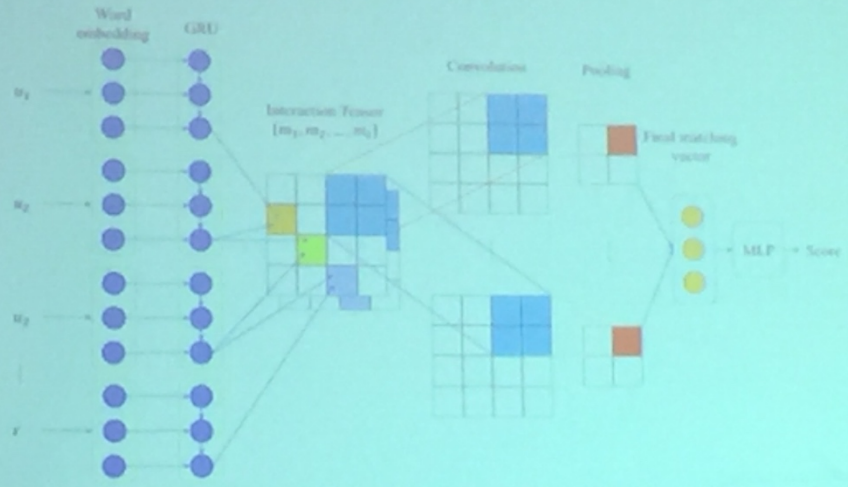


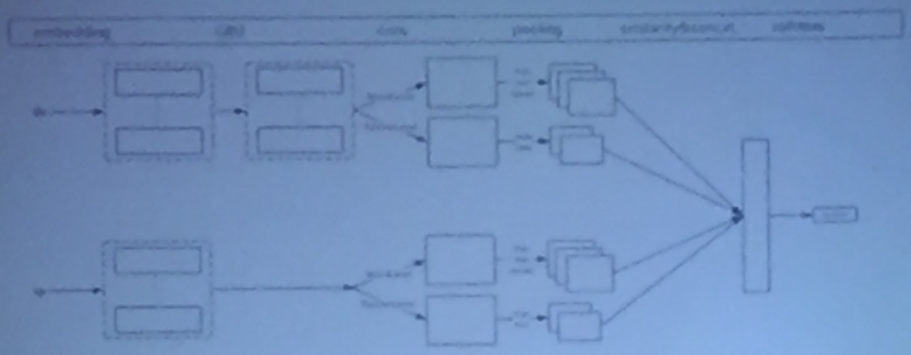
Fig. 1. The architecture of RCMN.

Semantic Matching

- Problem Formalization
  - Identify the Q_iA_i from n candidate QA pairs $\{(Q_1,A_1),\ldots,(Q_n,A_n)\}$ of Q-A KB that best matches Qu.
- Algorithm
  - TFIDF
  - WDM
  - SMN
  - MPCNN
    - Product Normalization
    - Matching

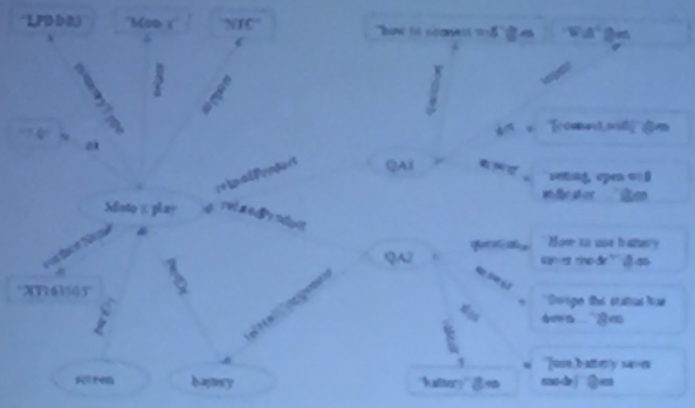
Lenovo

Semantic Matching

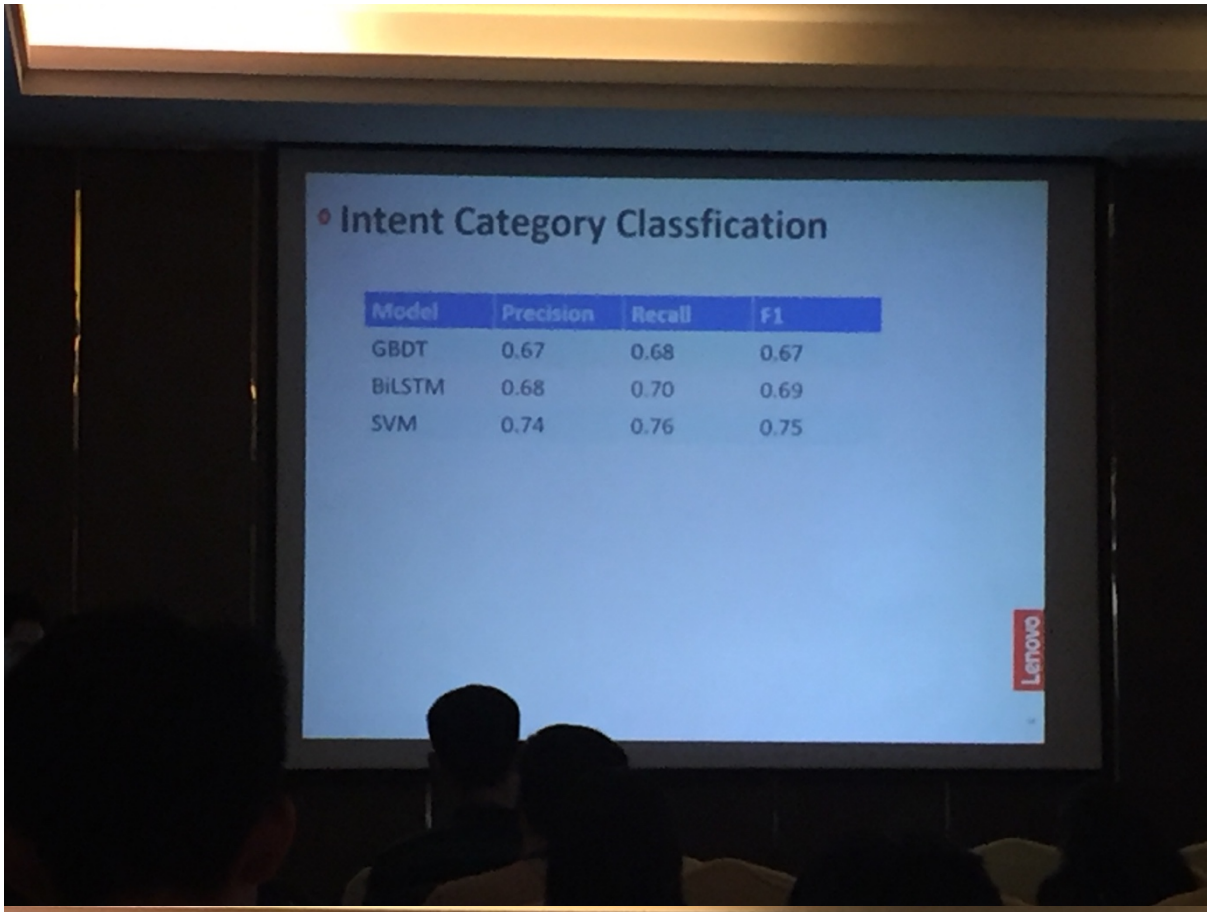


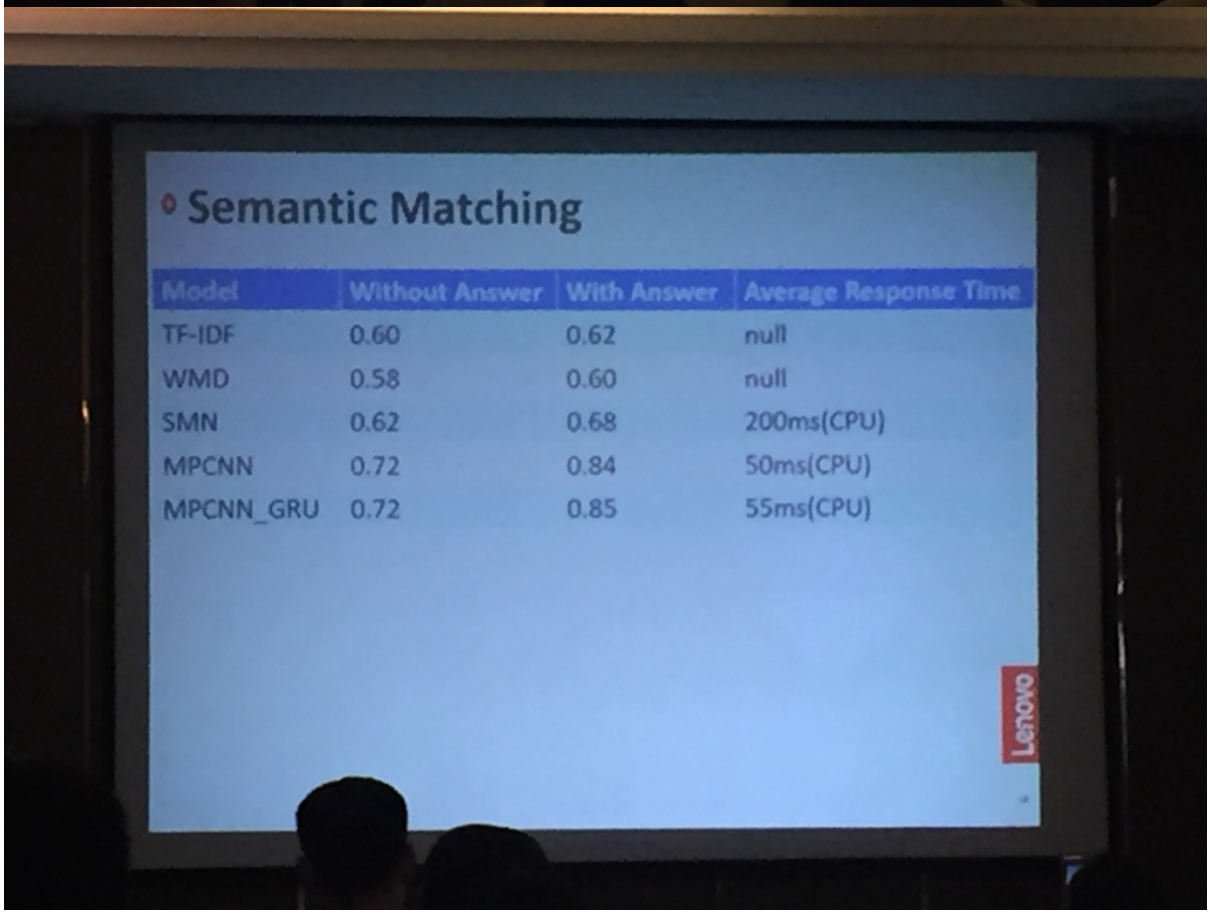
Lenovo

Data Set,QA-KB and Product-KB



Lenovo


Intent Category Classfication
Model Precision Recall F1
GBDT 0.67 0.68 0.67
BiLSTM 0.68 0.70 0.69
SVM 0.74 0.76 0.75
Lenovo


Semantic Matching
Model Without Answer With Answer Average Response Time
TF-IDF 0.60 0.62 null
WMD 0.58 0.60 null
SMN 0.62 0.68 200ms(CPU)
MPCNN 0.72 0.84 50ms(CPU)
MPCNN_GRU 0.72 0.85 55ms(CPU)
Lenovo