RESPONSE RETRIEVAL IN INFORMATION-SEEKING CONVERSATIONS

A Dissertation Presented

by

LIU YANG

Submitted to the Graduate School of the
University of Massachusetts Amherst in partial fulfillment
of the requirements for the degree of

DOCTOR OF PHILOSOPHY

September 2019

College of Information and Computer Sciences

RESPONSE RETRIEVAL IN INFORMATION-SEEKING CONVERSATIONS

A Dissertation Presented

by

LIU YANG

Approved as to style and content by:

W. Bruce Croft, Chair

James Allan, Member

Jiafeng Guo, Member

David Jensen, Member

Rajesh Bhatt, Member

James Allan, Department Head
College of Information and Computer Sciences

To Kaixi, mom and dad

ACKNOWLEDGMENTS

I would like to thank many people for helping me and supporting me throughout last five years in this program. This dissertation would not have been possible without them. First and foremost, I would like to thank my advisor, W. Bruce Croft, for his tireless effort in guiding and supporting me on research. Bruce taught me valuable lessons on picking important research topics with lasting impact, insisting on high standard to do solid research and staying focused on what really matters in the future. I also thank Bruce on giving me lots of freedom on developing research projects. His deep insights in the field and rigorous academic training shaped and sharpened me, personally and professionally, helping me move a huge step towards a great researcher.

I would like to thank James Allan for all his valuable feedback and comments on my thesis. I learned a lot and gained a systematic view of information retrieval from his IR course. I thank Jiafeng Guo for the great mentoring and collaborations over these years. His pointers and suggestions are very important for many papers on neural ranking models for QA and conversations, which were included in this dissertation. I also thank him for providing the opportunity to visit Chinese Academy of Sciences, which was really a wonderful experience. I would like to also thank my other PhD committee members, David Jensen and Rajesh Bhatt. Their insightful comments and encouragement made this thesis better in many ways.

I would like to thank all other mentors and collaborators who have helped me in my past research career. I especially thank Jing Jiang for introducing me to high quality academic research when I visited Singapore Management University. I also collaborated with Feida Zhu, Minghui Qiu, Swapna Gottipati, Yongfeng Zhang, Chen Qu and Qingyao Ai on several research papers before and during my PhD

study. I especially thank Minghui Qiu who I met in Singapore, along with Chen Qu and Yongfeng Zhang, for long-term collaborations during these years, which also initiated many exciting joint projects between UMass CIIR and Alibaba Group on information-seeking conversations. I did three internships in Microsoft Research and Bing during my PhD study. I would like to thank all my mentors in Microsoft: Susan Dumais, Paul Bennett, Ahmed Hassan Awadallah, Jianfeng Gao, Yelong Shen, Xiaodong Liu, Jingjing Liu, Kieran McDonald, Qi Guo, Yang Song, Sha Meng, Milad Shokouhi, for their tremendous support and mentoring during my internships. I especially thank Susan Dumais and Jianfeng Gao for their sharp visions and tireless pursuit of academic excellence, which inspired me a lot to keep trying my best to create real research impact in the community.

I thank all staffs of our lab, Kate Morruzzi, Jean Joyce, Dan Parker, and Glenn Stowell for their incredible support, which makes it possible for me to enjoy research and not worry about others. I also thank our Graduate Program Manager, Leeanne Leclerc and Eileen Hamel, for their help throughout this PhD program.

I thank all my friends and colleagues in our lab for sharing various ideas about research and life. In alphabetical order: Qingyao Ai, Elif Aktolga, Michael Bendersky, Keping Bi, Hamed Bonab, Ethem Can, Daniel Cohen, Jeffery Dalton, Laura Dietz, Shiri Dori-Hacohen, David Fisher, John Foley, Stephen Harding, Helia Hashemi, Myung-ha Jang, Jiepu Jiang, Mostafa Keikha, Youngwoo Kim, Jin Young Kim, Chia-Jung Lee, Yen-Chieh Lien, Ali Montazeralghaem, Venkatesh Narasimha Murthy, Nada Naji, Shahrzad Naseri, Jae Hyun Park, Chen Qu, Manmeet Singh, Sheikh Muhammad Sarwar, Lakshmi Vikraman, David Wemhoener, Hamed Zamani, Yongfeng Zhang, Michael Zarozinski. I wish them all the best for their future endeavors.

Finally, I would like to thank the most important people in my life: my family. I thank Kaixi, my wife and love of my life. This thesis would not have been possible without Kaixi's love, support, optimism, care, and advice. I am forever grateful to

Kaixi for her everlasting love and support during the last five years. We can now move to the next new chapter of our life together. I also thank my parents for their care, encouragement and unconditional love. Although we are on different hemispheres of the earth, my heart is with you forever.

This work was supported in part by the Center for Intelligent Information Retrieval and in part by NSF grant #IIS-1419693 and NSF grant IIS-1715095. Any opinions, findings and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect those of the sponsor.

ABSTRACT

RESPONSE RETRIEVAL IN INFORMATION-SEEKING CONVERSATIONS

SEPTEMBER 2019

LIU YANG

B.Eng., NORTHEASTERN UNIVERSITY

M.Sc., PEKING UNIVERSITY

M.Sc., UNIVERSITY OF MASSACHUSETTS AMHERST

Ph.D., UNIVERSITY OF MASSACHUSETTS AMHERST

Directed by: Professor W. Bruce Croft

The increasing popularity of mobile Internet has led to several crucial changes in the way that people use search engines compared with traditional Web search on desktops. On one hand, there is limited output bandwidth with the small screen sizes of most mobile devices. Mobile Internet users prefer direct answers on the search engine result page (SERP)(Li et al., 2009). On the other hand, voice-based / text-based conversational interfaces are becoming increasing popular as shown in the wide adoption of intelligent assistant services and devices such as Amazon Echo, Microsoft Cortana and Google Assistant around the world. These important changes have triggered several new challenges that search engines have had to adapt to in order to better satisfy the information needs of mobile Internet users. In this dissertation, we

investigate several aspects of *single-turn answer retrieval* and *multi-turn information-seeking conversations* to handle the new challenges of search on the mobile Internet.

We start from the research on single-turn answer retrieval and analyze the weaknesses of existing deep learning architectures for answer ranking. Then we propose an attention based neural matching model with a value-shared weighting scheme and attention mechanism to improve existing deep neural answer ranking models. Our proposed model achieves state-of-the-art performance for answer sentence retrieval compared with both feature engineering based methods and other neural models.

Then we move on to study response retrieval in multi-turn information-seeking conversations beyond single-turn interactions. Much research on response selection in conversation systems is modeling the matching patterns between user input message (either with context or not) and response candidates, which ignores external knowledge beyond the dialog utterances. We propose a learning framework on top of deep neural matching networks that leverages external knowledge with pseudo-relevance feedback and QA correspondence knowledge distillation for response retrieval. We also study how to integrate user intent modeling into neural ranking models to improve response retrieval performance. Finally, hybrid models of response retrieval and generation are investigated in order to combine the merits of these two different paradigms of conversation models.

Our goal is to develop effective learning models for answer retrieval and information-seeking conversations, in order to improve the effectiveness and user experience when accessing information with a touch screen interface or a conversational interface, as commonly adopted by millions of mobile Internet devices.

CONTENTS

Page

ACKNOWLEDGMENTS . . . . . **v**

ABSTRACT . . . . . **viii**

LIST OF TABLES . . . . . **xv**

LIST OF FIGURES . . . . . **xx**

CHAPTER

1. INTRODUCTION **1**

1.1 Single-Turn Answer Retrieval 2
1.2 Multi-Turn Information-seeking Conversations 4

1.2.1 Classification of Different Types of Conversations 4
1.2.2 Incorporating External Knowledge into Response Retrieval 6
1.2.3 Modeling User Intent for Response Retrieval 8
1.2.4 Fusing Conversation Response Retrieval with Generation 9

1.3 Contributions 10
1.4 Outline 13

2. BACKGROUND AND RELATED WORK **14**

2.1 Answer Passage Retrieval 14
2.2 Factoid Question Answering 15
2.3 Non-factoid Question Answering 16
2.4 Answer Ranking in CQA 17
2.5 Neural Ranking Models 18
2.6 Conversational Search 20
2.7 Neural Conversation Models 21

2.7.1 Retrieval-based Conversation Models 22
2.7.2 Generation-based Conversation Models 23

2.8 Utterance Intent Modeling 24

3. SINGLE-TURN ANSWER RETRIEVAL 26

3.1 Introduction 26
3.2 Attention-based Neural Matching Model 29

3.2.1 Terminology 29
3.2.2 Model Overview 30
3.2.3 Value-shared Weighting 31
3.2.4 Question Attention Network 32
3.2.5 Model Training 34
3.2.6 Extension to Deep Neural Networks with Multiple Sets of Value-shared Weights 35

3.2.6.1 Forward Propagation Prediction 35
3.2.6.2 Back Propagation for Model Training 36

3.3 Experiments 38

3.3.1 Data Set and Experiment Settings 38
3.3.2 Evaluation and Metrics 39
3.3.3 Model Learning Results 40

3.3.3.1 Value-shared Weight 40
3.3.3.2 Question Term Importance 41

3.3.4 Experimental Results for Ranking Answers 43

3.3.4.1 Learning without Combining Additional Features 43
3.3.4.2 Learning with Combining Additional Features 45
3.3.4.3 Results Summary 46

3.3.5 Parameter Sensitivity Analysis 47
3.3.6 Experimental Results on Microsoft Research WikiQA Data 48

3.3.6.1 WikiQA Data 48
3.3.6.2 Results on WikiQA Data 50

3.4 Summary 51

4. MULTI-TURN INFORMATION-SEEKING CONVERSATIONS 52

4.1 Introduction 52
4.2 Deep Matching Networks with External Knowledge 56

4.2.1 Problem Formulation 56
4.2.2 Method Overview 57
4.2.3 Deep Matching Networks with Pseudo-Relevance Feedback 59
4.2.3.1 Relevant QA Posts Retrieval 59
4.2.3.2 Candidate Response Expansion 59
4.2.3.3 Interaction Matching Matrix 60
4.2.3.4 Convolution and Pooling Layers 62
4.2.3.5 BiGRU Layer and MLP 62
4.2.3.6 Model Training 63
4.2.4 Deep Matching Networks with QA Correspondence Knowledge Distillation 64
4.3 Experiments 65
4.3.1 Data Set Description 65
4.3.1.1 Ubuntu Dialog Corpus 65
4.3.1.2 MSDialog 66
4.3.1.3 AliMe Data 67
4.3.2 Experimental Setup 68
4.3.2.1 Baselines 68
4.3.2.2 Evaluation Methodology 70
4.3.2.3 Parameter Settings 70
4.3.3 Evaluation Results 71
4.3.3.1 Performance Comparison on UDC and MSDialog 71
4.3.3.2 Performance Comparison on AliMe Data 72
4.3.3.3 Performance Comparison over Different Response Types 73
4.3.4 Model Ablation Analysis 74
4.3.5 Impact of Conversation Context Length 75
4.3.6 Case Study 75
4.4 Summary 77
5. USER INTENT IN INFORMATION-SEEKING CONVERSATIONS 78
5.1 Introduction 78
5.2 Intent-aware Response Ranking 83

5.2.1 Problem Formulation 83
5.2.2 Method Overview 84
5.2.3 User Intent Taxonomy 85
5.2.4 Utterance/ Response Input Representations 86
5.2.4.1 User Intent Representation 86
5.2.4.2 Utterance/ Response Encoding and Matching with Transformers 88
5.2.5 Intent-aware Attention Mechanism 91
5.2.6 Loss and Model Training 93
5.3 Experiments 93
5.3.1 Data Set Description 93
5.3.2 Experimental Setup 94
5.3.2.1 Baselines 94
5.3.2.2 Evaluation Methodology 95
5.3.2.3 Parameter Settings and Implementation Details 96
5.3.3 Evaluation Results 96
5.3.3.1 Performance Comparison on UDC and MSDialog 96
5.3.3.2 Performance Comparison on AliMe Data 98
5.3.4 Impact of Different Context Utterance Number and Utterance Length 98
5.3.5 Case Study and User Intent Visualization 99
5.4 Summary 103
6. HYBRID RETRIEVAL-GENERATION NEURAL CONVERSATION MODELS 104
6.1 Introduction 104
6.2 Hybrid Neural Conversation Models 108
6.2.1 Problem Formulation 108
6.2.2 Method Overview 109
6.2.3 Generation Module 111
6.2.3.1 Context Encoder 111
6.2.3.2 Facts Encoder 111
6.2.3.3 Response Decoder 112
6.2.3.4 Train and Decode 113

6.2.4 Retrieval Module . . . 113
6.2.5 Hybrid Ranking Module . . . 114
6.2.5.1 Interaction Matching Matrix . . . 114
6.2.5.2 CNN Layers and MLP . . . 115
6.2.5.3 Distant Supervision for Model Training . . . 115
6.3 Experiments . . . 117
6.3.1 Data Set Description . . . 117
6.3.2 Experimental Setup . . . 117
6.3.2.1 Competing Methods . . . 117
6.3.2.2 Evaluation Methodology . . . 119
6.3.2.3 Parameter Settings . . . 119
6.3.3 Evaluation Results . . . 120
6.3.3.1 Automatic Evaluation . . . 120
6.3.3.2 Human Evaluation . . . 121
6.3.4 Analysis of Top Responses Selected by Re-ranker . . . 125
6.3.5 Impact of Distant Supervision Signals . . . 127
6.3.6 Impact of Ratios of Positive Samples . . . 127
6.3.7 Examples and Case Study . . . 128
6.4 Summary . . . 129
7. CONCLUSIONS AND FUTURE WORK . . . 130
7.1 Closing Remarks . . . 130
7.2 Future Work . . . 133
7.2.1 On Single-Turn Answer Retrieval . . . 134
7.2.2 On Response Retrieval with External Knowledge . . . 134
7.2.3 On User Intent Modeling for Response Retrieval . . . 135
7.2.4 On Hybrid Models of Response Retrieval and Generation . . . 136
BIBLIOGRAPHY . . . 137

LIST OF TABLES

Table **Page**

3.1 The statistics of the TREC QA data set. 38

3.2 Examples of learned question term importance by aNMM-1. 42

3.3 The comparision of aNMM-1/aNMM-2 with aNMM-IDF which is a degenerate version of our model where we use IDF to directly replace the output of question attention network. 42

3.4 Results of TREC QA on TRAIN and TRAIN-ALL without combining additional features (Compare with deep learning methods). 44

3.5 Results of TREC QA on TRAIN-ALL without combining additional features (Compare with methods using feature engineering). 44

3.6 Results of TREC QA on TRAIN and TRAIN-ALL with combining additional features. 46

3.7 Overview of previously published systems on the QA answer ranking task. All reported results are from the best setting of each model trained on TRAIN-ALL data. 47

3.8 The statistics of the WikiQA data set. Note that "CandidateAS", "CorrectAS", "AvgLenOfQ", "AvgLenOfCanAS", "'QWithNoCAS' denote "candidate answer sentence", "correct answer sentence", "average length of question", "average length of candidate answer sentence", "question with no correct answer sentence" respectively. 49

3.9 The experimental results on WikiQA data set. Note that although the performances of our method aNMM are close to the baseline CNN-Count, our method does not need to be combined with additional features like overlapped word count features. 49

4.1 Sample utterance and response from the conversations in the Microsoft Answers community. This figure could be more readable with color print. Note that the purpose of this figure is to illustrate examples and differences among these three types of matches instead of exhaustively labeling all three types of matches between the two texts. 55

4.2 A summary of key notations in this work. Note that all vectors are denoted with bold cases. 58

4.3 Statistics of external collections for QA pairs retrieval and knowledge extraction. Note that "#QWithAcceptedA" means "number of questions with an accepted answer". The other names use similar abbreviations. 60

4.4 The statistics of data sets used in experiments. 68

4.5 Comparison of different models over Ubuntu Dialog Corpus (UDC) and MSDialog data sets. Numbers in bold font mean the result is better compared with the best baseline. ‡ means statistically significant difference over the best baseline with $p < 0.05$ measured by the Student's paired t-test. 71

4.6 Comparison of different models over the AliMe data. Numbers in bold font mean the result is better compared with the best baseline. ‡ means statistically significant difference over the best baseline with $p < 0.01$ measured by the Student's paired t-test. 72

4.7 Evaluation results of model ablation. "TB4.5" means the setting is the same with the results in Table 4.5. For DMN-KD, the model is the same with DMN if we remove M3. Numbers in bold font mean the result is better compared with other settings. 75

4.8 Examples of Top-1 ranked responses by different methods. y_i^k means the label of a response candidate. 76

5.1 An example dialog to illustrate user intent transition patterns in information-seeking conversations from the Microsoft Answers Community. We define different user intent types following previous research (Qu et al., 2018, 2019). We show a conversation context with 4 utterances and two response candidates where there is one correct candidate and one wrong candidate. The user intent of utterances and response candidates are labeled. "OQ", "IR", "PA", "FQ", "FD", "GG" denote "Original Question", "Information Request", "Potential Answer", "Follow-up Question", "Further Details", "Greetings/ Gratitude" respectively. We also highlight some lexical match between utterances and response candidates using colorful underlines. This table can be more readable with color print. 80

5.2 A summary of key notations in this chapter. Note that all vectors are denoted with bold cases. 84

5.3 Descriptions of user intent taxonomy. 86

5.4 Evaluation results of different user intent classification models on MSDialog from Qu et al. (2018). The significance test can only be performed on accuracy. In a multi-label classification setting, accuracy gives a score for each individual sample, while other metrics evaluate the performance over all samples. ‡ means statistically significant difference over the best baseline with $p < 10^{-4}$ measured by the Student's paired t-test. 88

5.5 Comparison of different models over Ubuntu Dialog Corpus (UDC) and MSDialog. Numbers in bold font mean the result is better compared with the best baseline DAM. † and ‡ means statistically significant difference over the best baseline DAM with $p < 0.1$ and $p < 0.05$ measured by the Student's paired t-test respectively. 97

5.6 Comparison of different models over the AliMe data. Numbers in bold font mean the result is better compared with the best baseline DAM. † and ‡ means statistically significant difference over the best baseline DAM with $p < 0.1$ and $p < 0.05$ measured by the Student's paired t-test respectively. 98

5.7 A case study and examples of Top-1 ranked responses by different methods. y_i^k means the label of a response candidate. The predicted user intent and user roles are highlighted by bold font. 101

6.1 A comparison of retrieval-based methods and generation-based methods for data driven conversation models. 105

6.2 A summary of key notations in this chapter. Note that all vectors or matrices are denoted with bold cases. 110

6.3 Statistics of experimental data used in this paper. 117

6.4 The hyper-parameter settings in the generation-based baselines and the generation module in the proposed hybrid neural conversation model. 119

6.5 Comparison of different models over the Twitter/ Foursquare data. Numbers in bold font mean the result is the best under the metric corresponding to the column. ‡ means that the improvement from the model on that metric is statistically significant over all baseline methods with $p < 0.05$ measured by the Student's paired t-test. Note that we can only do significance test for ROUGE-L since the other metrics are corpus-level metrics. 121

6.6 Comparison of different models with human evaluation on appropriateness. ‡ means that the improvement from the model on that metric is statistically significant over all baseline methods with $p < 0.05$ measured by the Student's paired t-test. The agreement score is evaluated by Fleiss' kappa (Fleiss et al., 1971) which is a statistical measure of inter-rater consistency. Agreement scores are comparable to previous results (0.2-0.5) as reported in (Shang et al., 2015; Song et al., 2018). Higher scores indicate higher agreement degree. 122

6.7 Comparison of different models with human evaluation on informativeness. ‡ means that the improvement from the model on that metric is statistically significant over all baseline methods with $p < 0.05$ measured by the Student's paired t-test. The agreement score is evaluated by Fleiss' kappa (Fleiss et al., 1971) which is a statistical measure of inter-rater consistency. Agreement scores are comparable to previous results (0.2-0.5) as reported in (Shang et al., 2015; Song et al., 2018). Higher scores indicate higher agreement degree. 122

6.8 Side-by-side human evaluation results. Win/Tie/Loss are the percentages of conversation contexts a method improves, does not change, or hurts, compared with the method after "v.s." on human evaluation scores. HNCM denotes HybridNCM. Seq2Seq-F denotes Seq2Seq-Facts. 123

6.9 The number and percentage of top responses selected by the hybrid ranking module from retrieved/ generated response candidates. #PickedGenRes is the number of selected responses from generated response candidates. #PickedRetRes is the number of selected responses from retrieved response candidates. #PickedTop1BM25 is the number of selected responses which is also ranked as top 1 responses by BM25. 125

6.10 The response generation performance when we vary the ratios of positive samples in distant supervision. 125

6.11 The response generation performance when we vary different distant supervision signals. This table shows the results for the setting "k'=3", where there are 3 positive response candidates for each conversation context. "SentBLEU" denotes using sentence-level BLEU scores as distant supervision signals. 126

6.12 Examples of output responses by different methods. r means the response is retrieved. g means the response is generated. Entities marked with [ENTITY] have been anonymized to avoid potentially negative publicity. "HNCM" denotes "HybridNCM". 128

LIST OF FIGURES

Figure **Page**

1.1 An example of showing direct answers in search results. 2

3.1 The proposed architecture of attention-based neural matching model (aNMM-2) for ranking answers. 29

3.2 The comparison of position-shared weight in CNN and value-shared weight in aNMM. In CNN, the weight associated with a node only depends on its position or relative location as specified by the filters. In aNMM, the weight associated with a node depends on its value. 31

3.3 Visualization of learned value-shared weights of aNMM-1. The x-axis is index of bin ranges and the y-axis is the value-shared weights corresponding to each bin range. The range of match signals is [-1,1] from the left to the right. 41

3.4 Visualization of learned question term importance by aNMM-1. 42

3.5 Tune hyper-parameters on validation data. 48

4.1 The architecture of DMN-PRF model for conversation response ranking. 59

4.2 The left figure shows the architecture of DMN-KD model for conversation response ranking. The input channel $\mathbf{M}_3$ denoted as blue matrices capture the correspondence matching patterns of utterance terms and response terms in relevant external QA pairs retrieved from $\mathcal{E}$. Note that we omit the details for CNN layers here to save spaces as they have been visualized in Figure 4.1. The right figure shows the detailed pipeline of external relevant QA pairs retrieval and QA correspondence matching knowledge distillation in DMN-KD model. 63

4.3 Performance comparison over different response types on MSDialog data. 73

4.4 Performance of DMN-KD and DMN-PRF with different choices of context length over UDC and MSDialog data. 76

5.1 The architecture of IART model for intent-aware conversation response ranking. 87

5.2 Performance of IART with different choices of maximum context utterance number and maximum utterance length over the validation partition of UDC and MSDialog data. 100

5.3 Visualization of learned user intent representation of context utterances and returned top-1 ranked response by DAM and IART from the case study in Table 5.7. U-0 to U-4 denotes the 0-th turn to the 4-th utterance turn in the context. R-DAM and R-IART denotes the top-1 ranked response returned by DAM and IART respectively. Darker spots mean higher predicted probabilities. 102

6.1 An example of the conversational response generation task. The factual information from external knowledge is denoted as blue color. 109

6.2 The architecture of the Hybrid Neural Conversation Model (HybridNCM). 109

CHAPTER 1

INTRODUCTION

The increasing popularity of intelligent mobile devices has seen a rapid growth in mobile Internet users. In 2019, the global unique mobile Internet users is 3.9 billion[1]. The average time spent per adult per day on mobile devices is 3.3 hours in 2017 compared to less than 1 hour as in 2011 (Meeker, 2018). As for Web search, more than 50% of search queries globally now come from mobile devices[2]. This trend has led to important changes in the way that people use search engines compared with traditional Web search on desktops. For instance, there is only limited output bandwidth such as small screen sizes of most mobile devices. Mobile Internet users prefer direct answers on the search engine result page (SERP)(Li et al., 2009). Another change is that voice-based / text-based conversational interfaces are becoming increasing popular as shown in the wide adoption of intelligent assistant services and devices such as Amazon Echo, Microsoft Cortana and Google Now[3]. These important changes have triggered several new challenges that search engines have had to adapt to in order to better satisfy the information needs of mobile Internet users.

---

[1]Global digital population as of January 2019, `https://www.statista.com/statistics/617136/digital-population-worldwide/` (as of March 29th, 2019)

[2]Building for the next moment, `https://adwords.googleblog.com/2015/05/building-for-next-moment.html` (as of March 29th, 2019)

[3]For example, over 100M installations of Google Now (Google, `http://bit.ly/1wTckVs`); 100M sales of Amazon Alexa devices (TheVerge, `https://bit.ly/2FbnzTN`); more than 141M monthly users of Microsoft Cortana (Windowscentral, `http://bit.ly/2Dv6TVT`). All urls are as of March 29th, 2019.

In this dissertation, we explore several aspects of *single-turn answer retrieval* and *multi-turn information-seeking conversations* to handle the new challenges of search on mobile Internet. We present the motivations and our work in each aspect as follows.

1.1 Single-Turn Answer Retrieval

With smaller screens, users want more direct answers instead of 10 blue links in the search results. Search engines need to show more direct answers for various queries, especially for natural language questions, in order to save user effort in fulfilling their information needs and improve user search experiences. Figure 1.1 shows an example of presenting direct answers in search results. Such results are more likely to directly satisfy users in the SERP without making them click and browse items in the rank list as in traditional Web search on desktops.

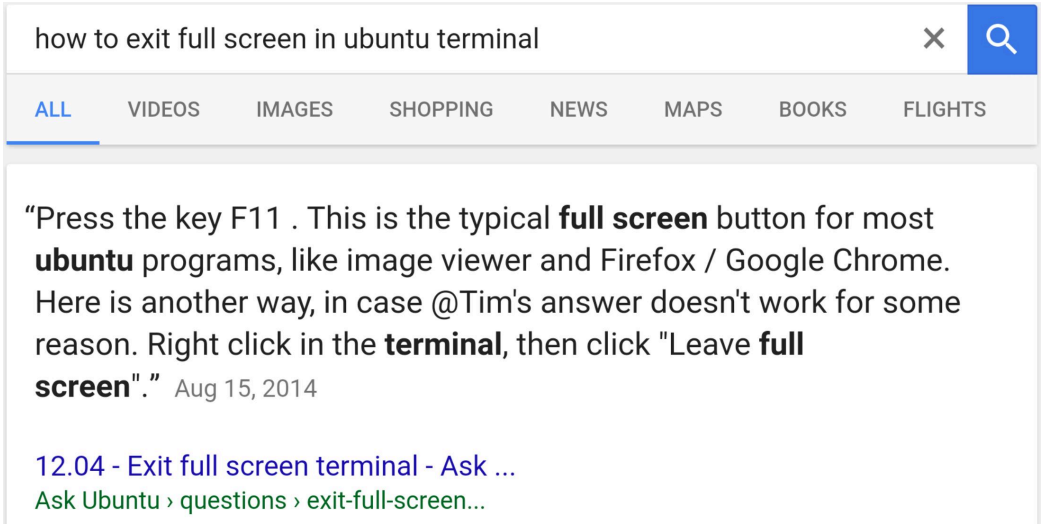


Figure 1.1: An example of showing direct answers in search results.

Question Answering (QA), which returns exact answers as either short facts or longer passages to natural language questions issued by users, plays a central role in showing direct answers in search results (Etzioni, 2011; Sun et al., 2015). Many of the

current QA systems use a learning to rank approach that encodes question/answer pairs with complex linguistic features including lexical, syntactic and semantic features (Severyn and Moschitti, 2015; Surdeanu et al., 2008; Yang et al., 2016b). For instance, Surdeanu et al. (2008, 2011) investigated a wide range of feature types including similarity features, translation features, density/frequency features and web correlation features for learning to rank answers and showed improvements in accuracy. However, such methods rely on manual feature engineering, which is often time-consuming and requires domain dependent expertise and experience. Moreover, they may need additional NLP parsers or external knowledge sources that may not be available for some languages.

In recent years, researchers have been studying deep learning (LeCun et al., 2015) approaches to automatically learn text representations and semantic matches between questions and answers. Such methods are built on top of neural network models such as convolutional neural networks (CNNs) (Yu et al., 2014; Severyn and Moschitti, 2015; Qiu and Huang, 2015) and Long Short-Term Memory Models (LSTMs) (Wang and Nyberg, 2015). The proposed models have the benefit of not requiring handcrafted linguistic features and external resources. Some of them (Severyn and Moschitti, 2015) achieved state-of-the art performance for the answer sentence selection task benchmarked by the TREC QA track. However, the weakness of the existing studies is that the proposed deep models, either based on CNNs or LSTMs, need to be combined with additional features such as word overlap features and BM25 to perform well. Without combining these additional features, their performance is significantly worse than the results obtained by the best methods based on linguistic feature engineering (Yih et al., 2013).

In our recent work (Yang et al., 2016a), we analyzed the existing deep learning architectures for answer ranking and observed two key issues: (1) Many deep learning architectures are not specifically designed for question/answer matching. For

instance, CNNs are originally designed for computer vision (CV), which uses position-shared weights with local perceptive filters, to learn spatial regularities in many CV tasks. However, such spatial regularities may not exist in semantic matching between questions and answers, since important similarity signals between question and answer terms could appear in any position due to the complex linguistic properties of natural languages. Meanwhile, models based on LSTMs view the question/answer matching problem in a sequential way. Without direct interactions between question and answer terms, the model may not be able to capture sufficiently detailed matching signals between them. (2) There is a lack of modeling question focus in the existing deep learning architectures. Understanding the focus of questions, e.g., important terms in a question, is helpful for ranking the answers correctly . For example, given a question like "*Where was the first burger king restaurant opened?*", it is critical for the answer to talk about "burger", "king", "open", etc. Most existing text matching models do not explicitly model question focus. For example, models based on CNNs treat all the question terms as equally important when matching to answer terms. Models based on LSTMs usually model question terms closer to the end to be more important. Chapter 3 describes in details the architecture and effectiveness of our proposed attention based neural match model, which is specifically designed for the answer retrieval task to address these issues.

1.2 Multi-Turn Information-seeking Conversations

1.2.1 Classification of Different Types of Conversations

Personal assistant systems, such as Apple Siri, Google Now, Amazon Alexa, and Microsoft Cortana, are becoming ever more widely used. These systems, with either text-based or voice-based conversational interfaces, are capable of voice interaction, information search, question answering and voice control of smart devices. This trend has led to an interest in developing information-seeking conversation systems,

where users would be able to ask questions to seek information with conversation interactions. Research on speech and text-based information-seeking conversation systems has recently attracted significant attention in the information retrieval (IR) community.

A personal assistant system should have the capabilities to perform several different types of conversations. These conversations can be classified into the following different categories (Gao et al., 2018):

- Question Answering: the agent needs to provide direct and correct answers to user questions based on either structured data sources like knowledge bases or unstructured data sources such as Web documents.

- Task Completion: the agent needs to accomplish tasks specified by users ranging from ordering a flight ticket to scheduling a business meeting.

- Social Chit-chat: the agent needs to interact with users seamlessly and appropriately with conversations, just like a human, in order to provide emotional support or useful recommendations to the user.

Different types of conversations requires different criteria to evaluate the performances of conversation agents. For example, answer correctness plays a key role in the evaluation of QA agents whereas user engagement optimization is more critical for a social chit-chat agent. A social chit-chat agent would like to maximize the usage time of users, which is an indicator of user engagement. On the contrary, a QA agent should satisfy the user query in the shortest time and fewest utterance turns. A task completion agent would like to maximize the task completion success rate and minimize the time and user efforts for task completion.

Information-seeking conversations are closer to question answering oriented conversations, since the goal is also answering the user's informational queries. But there

are several key differences. First the system outputs of an information-seeking conversation agent can be not only answers, but also greetings/ gratitude, clarification questions and feedback. There can be various different types of responses beyond answers. We argue that a functional conversation agent should have multiple capabilities including both question answering and social chat in real deployed systems. Second, the evaluation of an information-seeking conversation agent is more challenging as a result of the diversity of response types. For the same conversation context, there can be multiple correct responses. It is difficult to collect comprehensive reference responses given a set of conversation contexts.

1.2.2 Incorporating External Knowledge into Response Retrieval

Existing approaches to building conversational systems include generation-based methods (Ritter et al., 2011; Shang et al., 2015) and retrieval-based methods (Ji et al., 2014; Yan et al., 2016a,b, 2017). Compared with generation-based methods, retrieval-based methods have the advantages of returning fluent and informative responses. Most work on retrieval-based conversational systems studies response ranking for *single-turn conversation* (Wang et al., 2013), which only considers a current utterance for selecting responses. Recently, several researchers have been studying *multi-turn conversation* (Yan et al., 2016a; Zhou et al., 2016; Wu et al., 2017; Yan et al., 2017), which considers the previous utterances of the current message as the conversation context to select responses by jointly modeling context information, current input utterance and response candidates. However, existing studies are still suffering from the following weaknesses:

(1) Most existing studies are on open domain chit-chat conversations or task / transaction oriented conversations. Most current work (Ritter et al., 2011; Shang et al., 2015; Ji et al., 2014; Yan et al., 2016a,b, 2017) is looking at open domain chit-chat conversations as in microblog data like Twitter and Weibo. There is some

research on task oriented conversations (Young et al., 2010; Wen et al., 2017; Bordes et al., 2017), where there is a clear goal to be achieved through conversations between the human and the agent. However, the typical applications and data are related to completing transactions like ordering a restaurant or booking a flight ticket. Much less attention has been paid to *information oriented conversations*, which is referred to as *information-seeking conversations* in this thesis. Information-seeking conversations, where the agent is trying to satisfy the information needs of the user through conversation interactions, are closely related to conversational search systems. More research is needed on response selection in information-seeking conversation systems.

(2) Lack of modeling of external knowledge beyond the dialog utterances. Most research on response selection in conversation systems is purely modeling the matching patterns between user input message (either with context or not) and response candidates, which ignores external knowledge beyond the dialog utterances. Similar to Web search, information-seeking conversations could be associated with massive external data collections that contain rich knowledge that could be useful for response selection. This is especially critical for information-seeking conversations, since there may be not enough signals in the current dialog context and candidate responses to discriminate a good response from a bad one due to the wide range of topics for user information needs. An obvious research question is how to utilize external knowledge effectively for response ranking. This question has not been well studied, despite the potential benefits for the development of information-seeking conversation systems.

Chapter 4 presents our research on deep matching networks with external knowledge for response ranking in information-seeking conversations. We proposed two effective methods based on pseudo-relevance feedback and QA correspondence knowledge distillation.

1.2.3 Modeling User Intent for Response Retrieval

User intent modeling plays a key role in understanding user information needs in information-seeking conversations. In our recent work (Qu et al., 2018, 2019), we created information-seeking conversation data crawled from the Microsoft Answers Community, which is a customer support QA forum where users could ask questions relevant to Microsoft products. Agents like Microsoft employees or other experienced expert users will reply to these questions. There can be multi-turn conversation interactions between users and agents. We define a taxonomy of user intent and perform data analysis to characterize user intent in information-seeking conversations. (Qu et al., 2018, 2019). We observed that there are diverse user intents like "original question", "information request", "potential answers", "follow-up questions", "further details", etc. in an information-seeking conversation. Moreover, several transition patterns can happen between different user intents. For example, given a question from the user, an agent can provide a potential answer directly or ask some information as clarification questions before providing answers. Users will provide further details regarding the information requests from agents. At the start of a conversation, the agent would like to greet customers or express gratitude to users before they move on to next steps. Near the end of a conversation, the user may provide a positive or negative feedback towards answers and services from agents, or ask a follow-up question to continue the conversation interactions.

Such user intent transition patterns can be useful for conversation models to select good responses given conversation contexts. More research needs to be done to understand the role of user intent in response retrieval and to develop effective methods for intent-aware response ranking in information-seeking conversations. In Chapter 5, we analyze user intent in information-seeking conversations and propose neural ranking models with the integration of user intent modeling.

1.2.4 Fusing Conversation Response Retrieval with Generation

All the previous presented research only considered retrieval-based methods to find relevant existing response candidates to satisfy users' information needs. However, there are two different paradigms to produce responses given conversation inputs from users: generation-based methods (Ritter et al., 2011; Shang et al., 2015; Sordoni et al., 2015; Vinyals and Le, 2015; Li et al., 2016b; Bordes et al., 2017) and retrieval-based methods (Ji et al., 2014; Yan et al., 2016a,b, 2017; Yang et al., 2018).

Given some conversation context, retrieval-based methods try to find the most relevant context-response pairs in a pre-constructed conversational history repository. Some of these methods achieve this in two steps: 1) retrieve a candidate response set with basic retrieval models such as BM25 (Robertson and Walker, 1994) or QL (Ponte and Croft, 1998); and 2) re-rank the candidate response set with neural ranking models to find the best matching response (Yan et al., 2016a,b, 2017; Wu et al., 2017; Yang et al., 2018). These methods can return natural human utterances in the conversational history repository, which is controllable and explainable. Retrieved responses often come with better diversity and richer information compared to generated responses (Song et al., 2018). However, the performance of retrieval-based methods is limited by the size of the conversational history repository, especially for long tail contexts that are not covered in the history. Retrieval-based methods lack the flexibility of generation-based models, since the set of responses of a retrieval system is fixed once the historical context/response repository is constructed.

On the other hand, generation-based methods could generate highly coherent new responses given the conversation context. Much previous research along this line was based on the Seq2Seq model (Shang et al., 2015; Sordoni et al., 2015; Vinyals and Le, 2015), where there is an encoder to learn the representation of conversation context as a contextual vector, and a decoder to generate a response sequence conditioning on the contextual vector as well as the generated part of the sequence. The encoder/

decoder could be implemented by an RNN with long short term memory (LSTM) (Hochreiter and Schmidhuber, 1997) or gated recurrent units (GRU) (Chung et al., 2014) hidden units. Although generation-based models can generate new responses for a conversation context, a common problem with generation-based methods is that they are likely to generate very general or universal responses with insufficient information such as "I don't know", "I have no idea", "Me too", "Yes please". The generated responses may also contain grammar errors. Ghazvininejad et al. (2018) proposed a knowledge-grounded neural conversation model in order to infuse the generated responses with more factual information relevant to the conversation context without slot filling. Although they show that the generated responses from the knowledge-grounded neural conversation model are more informative compared with responses from the vanilla Seq2Seq model, their model is still generation-based, and it is not clear how well this model will perform compared to retrieval-based methods. Clearly these two types of methods have their own advantages and disadvantages, it is thus necessary to study how to integrate the merits of these two methods.

In Chapter 6, we study the integration of retrieval-based and generation-based conversation models in an unified framework. We propose a hybrid neural conversational model with a generation module, a retrieval module and a hybrid ranking module to fuse both response retrieval and response generation.

1.3 Contributions

In this thesis, we study several aspects about single-turn answer retrieval and multi-turn information-seeking conversations in order to better satisfy the information needs of mobile Internet users. In the following, we highlight the major contributions of this thesis.

- Presenting an attention based neural matching model (aNMM) for answer retrieval. We introduce a novel value-shared weighting scheme in deep neural net-

works as a counterpart of the position-shared weighting scheme in CNNs, based on the idea that semantic matching between a question and answer is mainly about the (semantic similarity) value regularities rather than spatial regularities. Furthermore, we incorporate the attention mechanism over the question terms using a gating function, so that we can explicitly discriminate the question term importance. Experimental results with TREC QA data (Wang et al., 2007) show that our model can achieve better performance than a state-of-art method using linguistic feature engineering and comparable performance with previous deep learning models with combined additional features. If we combine our model with a simple additional feature like QL, our method can achieve state-of-the-art performance, with much less feature engineering costs.

- Presenting a learning framework on top of deep neural matching networks with external knowledge for response ranking in information-seeking conversations. We study two different methods of integrating external knowledge into deep neural matching networks with pseudo-relevance feedback and QA correspondence knowledge distillation. Inspired by the key idea of PRF (Lavrenko and Croft, 2001; Lv and Zhai, 2009; Zamani et al., 2016; Zhai and Lafferty, 2001; Rocchio, 1971; Cao et al., 2008; Diaz and Metzler, 2006), we propose using the candidate response as a query to run a retrieval round on a large external collection. Then we extract useful information from the (pseudo) relevant feedback documents to enrich the original candidate response representation. We also propose to extract the "correspondence" regularities between question and answer terms from retrieved external QA pairs and incorporate them into deep matching networks as external knowledge to help response selection. Experimental results on MSDialog data (Qu et al., 2018), Ubuntu Dialog Corpus (UDC) (Lowe et al., 2015), and another commercial customer service data from

Alibaba show that our proposed methods outperform all baseline methods using a variety of metrics for response ranking in information-seeking conversations.

- Incorporating user intent modeling for response retrieval in information-seeking conversations. We analyze and characterize different user intent in information-seeking conversations. We propose an intent-aware response ranking model with Transformers (Vaswani et al., 2017): IART. IART derives the importance weighting scheme of utterances in conversation context with user intent signals towards better conversation history modeling. Experimental results with three different information-seeking conversation data sets show that our methods outperform various baselines including the state-of-the-art method. We also perform visualization on learned user intent and ranking examples to provide insights.

- Presenting a hybrid neural conversational model to combine conversation response retrieval with generation. In order to let retrieval-based conversation models and generation-based conversation models complement each other, we propose a hybrid neural conversational model with a generation module, a retrieval module and a hybrid ranking module. The generation module generates a response candidate given a conversation context, using a Seq2Seq model consisting of a conversation context encoder, a facts encoder and a response decoder. The retrieval module adopts a "context-context match" approach to recall a set of response candidates from the historical context/ response repository. The hybrid ranking module is built on the top of neural ranking models to select the best response candidate among retrieved/ generated response candidates. To construct the training data of the neural ranker for response selection, we propose a distant supervision approach to automatically infer labels for retrieved/ generated response candidates. Experimental results show that the proposed

model can outperform both retrieval-based models and generation-based models for both automatic evaluation and human evaluation. We also perform qualitative analysis on top responses selected by the neural re-ranker and response generation examples to provide insights.

1.4 Outline

The remainder of this thesis is organized as follows. In Chapter 2, we provide background information and literature survey related to this thesis. In Chapter 3, we present an attention based neural matching model for answer retrieval. In Chapter 4, we present our work on a learning framework on top of deep neural matching networks that leverage external knowledge for response ranking in information-seeking conversation systems. In Chapter 5 we present intent-aware neural ranking models for response retrieval in order to integrate user intent modeling into conversation response ranking. In Chapter 6 we present a hybrid neural conversational model to combine conversation response retrieval with generation. Finally, in Chapter 7, we summarize the contributions made in this thesis and discuss potential future directions for more research in this area.

CHAPTER 2

BACKGROUND AND RELATED WORK

This dissertation is related to several research areas, including answer passage retrieval, factoid question answering, non-factoid question answering, answer ranking in CQA, neural ranking models, conversational search, neural conversation models and utterance intent modeling.

2.1 Answer Passage Retrieval

Our work is related to previous research on answer passage retrieval. Tymoshenko and Moschitti (2015) studied the use of syntactic and semantic structures obtained with shallow and deeper syntactic parsers in the answer passage re-ranking task. Corrada-Emmanuel and Croft (2004) extended the techniques of language modeling to create answer models for answer passage retrieval and demonstrate their effectiveness on the TREC 2002 QA Corpus. Tellex et al. (2003) conducted a thorough quantitative component evaluation for passage retrieval algorithms employed by state-of-the-art QA systems. Cui et al. (2005) proposed a novel fuzzy relation matching method which examines grammatical dependency relations between question terms to improve passage retrieval techniques for question answering. Ageev et al. (2013) studied how to incorporate searcher examination data, such as mouse cursor movements and scrolling, to infer the parts of the document the searcher found interesting, and then incorporate this signal into passage retrieval for QA. Keikha et al. (2014a,b) developed an annotated data set for non-factoid answer finding using TREC GOV2 collections and topics. They annotated passage-level answers, revisited several pas-

sage retrieval models with this data, and came to the conclusion that the current methods are not effective for this task. We explored representation learning with deep neural networks for answer retrieval. Unlike learning to rank approaches with feature engineering, representation learning methods can achieve good performance for ranking answers without preprocessing of NLP parsers and external resources like knowledge bases.

Some previous research on answer passage retrieval has been based on statistical translation models for answer finding in FAQ data (Riezler et al., 2007; Berger et al., 2000). Riezler et al. (2007) presented an approach to query expansion in answer retrieval that uses machine translation techniques to bridge the lexical gap between questions and answers. Berger et al. (2000) studied multiple statistical methods such as query expansion, statistical translation, and latent variable models for answer finding in FAQ data.

2.2 Factoid Question Answering

There have been many previous studies on factoid question answering, most of which use the benchmark data from TREC QA track (Yih et al., 2013; Wang and Nyberg, 2015; Yao et al., 2013; Yu et al., 2014; Severyn and Moschitti, 2015) or knowledge bases (Yin et al., 2016). Collins-Thompson et al. (2004) examined the relationship between the quality of document retrieval and the overall accuracy of QA systems. Lin (2007) examined the underlying assumptions and principles behind redundancy-based techniques for mining answers to factoid questions. Bilotti et al. (2010) proposed a general rank-learning framework for passage ranking within QA systems using linguistic and semantic features. The framework enables query-time checking of complex linguistic and semantic constraints over keywords. Yih et al. (2013) formulated answer sentence selection as a semantic matching problem with a latent word-alignment structure and conducted a series of experimental studies on

leveraging proposed lexical semantic models. Iyyer et al. (2014) introduced a recursive neural network (RNN) model that can reason over text that contains very few individual words by modeling textual compositionality. Yu et al. (2014) proposed an approach for answer sentence selection via distributed representations, and learned to match questions with answers by considering their semantic encoding. They combined the learning results of their model with word overlap features by training a logistic regression classifier. Wang and Nyberg (2015) proposed a method which uses a stacked bidirectional Long-Short Term Memory (BLSTM) network to sequentially read words from question and answer sentences, and then output their relevance scores. Their system needs to combine the stacked BLSTM relevance model with a BM25 score to achieve good performance. Severyn and Moschitti (2015) presented a convolutional neural network architecture for re-ranking pairs of short texts, where they learned the optimal representation of text pairs and a similarity function to relate them in a supervised way from the available training data. They also need to combine additional features into their model to outperform previous methods. Unlike the previous research, the proposed attention based neural matching model (aNMM) can outperform previous methods using feature engineering without combining any additional features.

2.3 Non-factoid Question Answering

Our research is also relevant to previous works on non-factoid question answering (Surdeanu et al., 2008, 2011; Chaturvedi et al., 2014; Tymoshenko et al., 2016). Non-factoid question answering, unlike many previous research on factoid QA, aims to find longer answers which could be sentences or passages for questions with complex information needs including definition, manner, reason, description, etc. Surdeanu et al. (2008, 2011) investigated a wide range of feature types including similarity features, translation features, density/frequency features and web correlation features

for ranking answers to non-factoid questions. Soricut and Brill (2006) built a QA system around a noisy-channel architecture which exploits both a language model for answers and a transformation model for answer/question terms, trained on a corpus of 1 million question/answer pairs collected from the Web to find answers for a large variety of complex, non-factoid questions. Recent years, there are some research on building deep neural models for non-factoid question answering (Tan et al., 2015; Cohen and Croft, 2016; Rücklé and Gurevych, 2017). Tan et al. (2015) built the embeddings of questions and answers based on bidirectional long short-term memory models, and measure their closeness by cosine similarity for non-factoid answer selection. Cohen and Croft (2016) showed that end to end training with a Bidirectional Long Short Term Memory network with a rank sensitive loss function results in significant performance improvements for non-factoid QA. Since answers towards non-factoid questions could be very long or come from multiple documents, some researchers studied answer summarization for non-factoid questions (Song et al., 2017; Chen et al., 2015; Yulianti et al., 2016). Song et al. (2017) proposed a sparse coding-based summarization strategy that includes short document expansion, sentence vectorization, and a sparse-coding optimization framework for answer summarization of non-factoid questions. Yulianti et al. (2016) investigated the effectiveness of using semantic and context features for extracting document summaries that are designed to contain answers for non-factoid queries.

2.4 Answer Ranking in CQA

There is also previous research on ranking answers from community question answering (CQA) sites. Bian et al. (2008) proposed a ranking framework to take advantage of user interaction information to retrieve answers that are relevant, factual, and of high quality in CQA sites. Jansen et al. (2014) presented an answer re-ranking model for non-factoid questions that integrate lexical semantics with discourse in-

formation driven by two representations of discourse. Xue et al. (2008) proposed a retrieval model that combines a translation-based language model for the question part with a query likelihood approach for the answer part. Yang et al. (2013) proposed Topic Expertise Model (TEM), a probabilistic generative model with GMM hybrid, to jointly model topics and expertise by integrating textual content model and link structure analysis. The learning results of TEM model is used to measure user interests and expertise score under different topics to rank answers given questions in CQA.

2.5 Neural Ranking Models

Recently a number of deep neural models have been proposed for text matching and ranking. Such neural models include DSSM (Huang et al., 2013; Gao et al., 2014; Shen et al., 2014), ARC-I/ARC-II(Hu et al., 2014), DCNN (Kalchbrenner et al., 2014), DeepMatch (Lu and Li, 2013), MultiGranCNN (Yin and Schütze, 2015), MatchPyramid (Pang et al., 2016), DRMM (Guo et al., 2016), Match-Tensor (Jaech et al., 2017) etc. These approaches can be generally divided into two groups: *representation-focused* and *interaction-focused* models (Guo et al., 2016). Representation-focused models independently learn the representations of queries and documents separately and then calculate the similarity score of the learned representations with functions such as cosine, dot, bilinear or tensor layers. A typical example is the DSSM (Huang et al., 2013) model, which is a feed forward neural network with a word hashing phase as the first layer to predict the click probability given a query string and a document title. ARC-I (Hu et al., 2014) firstly finds the representation of each sentence and then compares the representations of the two sentences with a multi-layer perceptron (MLP). The drawback of ARC-I is that it defers the interaction between two sentences until their individual representation matures in the convolution model, and therefore has the risk of losing details, which could be important for the text matching task.

The second category is the *interaction-focused* models, which build a query-document interaction matrix to capture the exact matching and semantic matching information between the query-document pairs. The interaction matrix is further fed into deep neural networks which could be a CNN (Hu et al., 2014; Pang et al., 2016; Yu et al., 2018), term gating network with histogram mechanism (Guo et al., 2016) to generate the final ranking score. These models have an opportunity to capture the interactions between query and document, while representation-focused models look at the inputs in isolation. For instance, DeepMatch (Lu and Li, 2013) is an interaction-focused model that construct the interactions between two texts with topic models, and then makes different levels of abstractions with a deep architecture to model the relationships between topics. ARC-II (Hu et al., 2014) is built directly on the interaction space between two sentences. Thus ARC-II makes two sentences interact before their own high-level representations mature, while still retaining the space for individual development of abstraction of each sentence. Our proposed aNMM architecture adopts a similar design with ARC-II in the QA matching matrix where we built neural networks directly on the interaction of QA sentence term pairs. However, we adopted value-shared weights instead of position-shared weights as in the CNN used by ARC-II. We also added an attention scheme to learn question term importance.

In the end, neural ranking models in the third category combine the ideas of the *representation-focused* models and *interaction-focused* models to joint learn the lexical matching and semantic matching between queries and documents (Mitra et al., 2017; Yu et al., 2018). For example, Mitra et al. (2017) proposed the Duet model in order to simultaneously learn local and distributional representations to capture both exact term matching and semantic term matching information for document ranking, which is a combination of representation-focused models and interaction-focused models.

All the aforementioned models are trained based on either explicit relevance judgments or clickthrough data. More recently, Dehghani et al. (2017) proposed to train neural ranking models when no supervision signal is available. They used an existing retrieval model, e.g., BM25 or query likelihood, to generate large amount of training data automatically and proposed to use these generated data to train neural ranking models with weak supervision.

Neural ranking models used in our research in this thesis belong to the interaction-focused models due to their better performance on a variety of text matching and ranking tasks compared with representation-focused models (Hu et al., 2014; Pang et al., 2016; Guo et al., 2016; Yang et al., 2016a; Wu et al., 2017; Xiong et al., 2017).

2.6 Conversational Search

Conversational search has received significant attention with the emerging of conversational devices in the recent years. Radlinski and Craswell described a theoretic framework of conversational search systems (Radlinski and Craswell, 2017). Based on state-of-the-art advances on machine reading, Kenter and de Rijke (2017) adopted a conversational search approach to question answering, and Vakulenko et al. (2017) adopted interactive storytelling as a tool to enable exploratory search within a conversational interface. Except for conversational search models, researchers have also studied the medium of conversational search. Arguello et al. (2018) studied how the medium (e.g., voice interaction) affects user requests in conversational search. Spina et al. (2017) studied the ways of presenting search results over speech-only channels to support conversational search.

To facilitate research on conversational search, we need open accessible benchmark datasets to develop and evaluate different methods. However, currently there is a lack of large scale conversational search data sets with high quality. Current related data sets are either built with chit-chat microblogs (Shang et al., 2015; Sordoni et al., 2015)

or simulation data from user studies (Thomas et al., 2017). Thomas et al. (2017) introduced the Microsoft Information-Seeking Conversation data (MISC), which is a set of recordings of information-seeking conversations between human "seekers" and "intermediaries". Although this data records real multi-turn QA interactions between users and search assistants, it is only generated by 44 people working together on solving 5 information-seeking tasks, which is not a sufficiently large scale dataset that could be used to train machine learning models like neural models. Many companies own large scale multi-turn QA chat logs between users and customer service staffs over phones or online chatting. But these data sets are usually private and not accessible for the broad research community. To address this problem, we crawled technical support conversation data MSDialog from the Microsoft Answer community[1], which is a QA forum on topics about a variety of Microsoft products. With MSDialog data, we performed a variety of research including user intent characterization and prediction (Qu et al., 2018, 2019), response ranking with external knowledge (Yang et al., 2018) and response ranking with intent modeling (see Chapter 5).

2.7 Neural Conversation Models

Recent years there are growing interests on research about conversation response generation and ranking with deep learning and reinforcement learning (Shang et al., 2015; Yan et al., 2016a; Li et al., 2016a,b; Sordoni et al., 2015; Bordes et al., 2017). Existing work includes retrieval-based methods (Wu et al., 2017; Zhou et al., 2016; Yan et al., 2016a, 2017, 2016b; Ji et al., 2014; Lowe et al., 2015) and generation-based methods (Shang et al., 2015; Tian et al., 2017; Ritter et al., 2011; Sordoni et al., 2015; Vinyals and Le, 2015; Li et al., 2016b; Bordes et al., 2017). We briefly review them as follows.

[1] answers.microsoft.com

2.7.1 Retrieval-based Conversation Models

There have been several recent studies on retrieval based-conversation models (Wu et al., 2017; Zhou et al., 2016; Yan et al., 2016a,b, 2017; Ji et al., 2014; Lowe et al., 2015; Yang et al., 2017). Yan et al. (2016a) proposed a retrieval-based conversation system with the deep learning-to-respond schema by concatenating context utterances with the input message as reformulated queries. Zhou et al. (2016) proposed a multi-view response selection model that integrates information from two different views including word sequence view and utterance sequence view with deep neural networks. Wu et al. (2017) proposed a sequential matching network that matches a response with each utterance in the context on multiple levels of granularity to distill important matching information. Our proposed models in Chapter 4 are retrieval-based models. The difference between our work with previous research is that we considered external knowledge beyond dialog context for multi-turn response selection. We showed that incorporating external knowledge with pseudo-relevance feedback and QA correspondence knowledge distillation are important and effective for response selection in information-seeking conversations.

Although retrieval-based methods can return fluent responses with great diversity, these approaches lack the flexibility of generation based methods since the set of responses of a retrieval system is fixed once the historical context/ response repository is constructed in advance. Thus retrieval systems may fail to return any appropriate responses for those unseen conversation context inputs (Gao et al., 2018). In Chapter 6, we also studied the integration of retrieval-based methods and generation-based methods for conversation response generation to combine the merits of these two types of methods.

2.7.2 Generation-based Conversation Models

There has also been a number of recent studies on conversation response generation with deep learning and reinforcement learning (Ritter et al., 2011; Shang et al., 2015; Sordoni et al., 2015; Vinyals and Le, 2015; Li et al., 2016b,a; Tian et al., 2017; Bordes et al., 2017; Dhingra et al., 2017; Qiu et al., 2017; Zhang et al., 2018b; Pandey et al., 2018; Wu et al., 2018; Zhang et al., 2018a). Early generation-based conversation models were inspired by statistical machine translation (SMT) (Ritter et al., 2011), which applied a phrase-based translation approach (Koehn et al., 2003) to conversation response generation. In order to utilize longer conversation context, Sordoni et al. (2015) proposed a neural network architecture for response generation that is both context-sensitive and data-driven utilizing the Recurrent Neural Network Language Model architecture. Shang et al. (2015) proposed the Neural Responding Machine (NRM), which is an RNN encoder-decoder framework for short text conversations and showed that it outperformed retrieved-based methods and SMT-based methods for a single round conversation. Bordes et al. (2017) proposed a testbed to break down the strengths and shortcomings of end-to-end dialog systems in goal-oriented applications based on Memory Networks (Weston et al., 2014; Sukhbaatar et al., 2015). Li et al. (2016b) applied deep reinforcement learning to model future reward in chatbot dialogs towards building a neural conversational model based on the long-term success of dialogs. In order to mitigate the blandness problem of universal responses generated by Seq2Seq models, Li et al. (2015) proposed the Maximum Mutual Information (MMI) objective function for conversation response generation. The approach first generates N-best lists and rescores them with MMI during decoding process. Zhang et al. (2018a) proposed a model which introduces an additional variable modeled using a Gaussian kernel layer to control the level of specificity of the response.

Some previous work augmented the context encoder to not only represent the conversation history, but also some additional input from external knowledge. Ghazvininejad et al. (2018) proposed a knowledge-grounded neural conversation model which infuses factual content that is relevant to the conversation context. Our research in Chapter 6 shared a similar motivation with this work, but we did not adopt a pure generation-based approach. Instead, we looked at a hybrid approach of retrieval-based models and generation-based models. Similar hybrid approaches were also used in some popular personal intelligent assistant systems including the "Core Chat" component of Microsoft XiaoIce (Zhou et al., 2018a). Our proposed model is distinguished from prior work using the boosted tree ranker (Zhou et al., 2018a; Song et al., 2018) by using a neural ranking model which holds the advantage of reducing feature engineering efforts for the conversation context/ response candidates pairs during the hybrid re-ranking process.

2.8 Utterance Intent Modeling

Some previous research studied utterance intent modeling in conversation systems. Stolcke et al. (2000) performed dialog acts classification with a statistical approach on the SwitchBoard corpus (Godfrey and Holliman, 1997), which consists of human-human chit chats conversations. Olney et al. (2003) classified students' utterances in an intelligent tutoring system with cascaded finite state transducers. Surendran and Levow (2006) conducted dialog acts tagging on the HCRC MapTask corpus (Thompson et al., 1993) with a combined method with SVM and Hidden Markov Model. Madan and Joshi (2017) proposed an approach to find frequent user utterances which serve as examples for intents, and corresponding agent responses by extending standard K-means algorithm to simultaneously cluster user utterances and agent utterances. Shiga et al. (2017) studied how people express a broad range of information needs in conversations and analyzed a range of features such as semantic features,

dialogue features and temporal features that are useful for detecting utterances that contain conversational information needs. Bhatia et al. (2012, 2014) focused on forum post classification for applications in information extraction and summarizing. Recent advances in deep learning have made it possible to use neural networks for text classification, which can also be applied for utterance intent classification. Related research is conducted on both word level (Kingma and Ba, 2014; Lai et al., 2015) and character-level (Zhang et al., 2015; Schwenk et al., 2017). Specifically, such methods are applied to intent modeling in medical dialog systems (Datta et al., 2016). Chapter 5 is related to utterance intent modeling. We explored how to combine utterance intent modeling with response ranking in conversations, so that the learned user intent of context utterances can help select better responses in information-seeking conversations.

CHAPTER 3

SINGLE-TURN ANSWER RETRIEVAL

3.1 Introduction

Question answering (QA), which returns exact answers as either short facts or long passages to natural language questions issued by users, is a challenging task and plays a central role in the next generation of advanced web search (Etzioni, 2011; Sun et al., 2015). Many current QA systems use a learning to rank approach that encodes question/answer pairs with complex linguistic features including lexical, syntactic and semantic features (Severyn and Moschitti, 2015; Surdeanu et al., 2008; Yang et al., 2016b). For instance, Surdeanu et al. (2008, 2011) investigated a wide range of feature types including similarity features, translation features, density/frequency features and web correlation features for learning to rank answers and show improvements in accuracy. However, such methods rely on manual feature engineering, which is often time-consuming and requires domain dependent expertise and experience. Moreover, they may need additional NLP parsers or external knowledge sources that may not be available for some languages.

Recently, researchers have been studying deep learning approaches to automatically learn semantic match between questions and answers. Such methods are built on the top of neural network models such as convolutional neural networks (CNNs) (Yu et al., 2014; Severyn and Moschitti, 2015; Qiu and Huang, 2015) and Long Short-Term Memory Models (LSTMs) (Wang and Nyberg, 2015). The proposed models have the benefit of not requiring hand-crafted linguistic features and external resources. Some of them (Severyn and Moschitti, 2015) achieve state-of-the art performance for the

answer sentence selection task benchmarked by the TREC QA track. However, the weakness of the existing studies is that the proposed deep models, either based on CNNs or LSTMs, need to be combined with additional features such as word overlap features and BM25 to perform well. Without combining these additional features, their performance is significantly worse than the results obtained by the state-of-the-art methods based on linguistic feature engineering (Yih et al., 2013). This led us to propose the following research questions:

RQ1 *Without combining additional features, could we build deep learning models that can achieve comparable or even better performance than methods using feature engineering?*

RQ2 *By combining additional features, could our model outperform state-of-the-art models for question answering?*

To address these research questions, we analyze the existing deep learning architectures for answer ranking and make the following two key observations:

1. Architectures not specifically designed for question/answer matching: Some methods employ CNNs for question/answer matching. However, CNNs are originally designed for computer vision (CV), which uses position-shared weights with local perceptive filters, to learn spatial regularities in many CV tasks. However, such spatial regularities may not exist in semantic matching between questions and answers, since important similarity signals between question and answer terms could appear in any position due to the complex linguistic property of natural languages. Meanwhile, models based on LSTMs view the question/answer matching problem in a sequential way. Without direct interactions between question and answer terms, the model may not be able to capture sufficiently detailed matching signals between them.

2. Lack of modeling question focus: Understanding the focus of questions, e.g., important terms in a question, is helpful for ranking the answers correctly.

For example, given a question like "*Where was the first burger king restaurant opened?*", it is critical for the answer to talk about "burger", "king", "open", etc. Most existing text matching models do not explicitly model question focus. For example, models based on CNNs treat all the question terms as equally important when matching to answer terms. Models based on LSTMs usually model question terms closer to the end to be more important.

To handle these issues in the existing deep learning architectures for ranking answers, we propose an attention based neural matching model (**aNMM**). The novel properties of the proposed model and our contributions can be summarized as follows:

1. Deep neural network with value-shared weights: We introduce a novel value-shared weighting scheme in deep neural networks as a counterpart of the position-shared weighting scheme in CNNs, based on the idea that semantic matching between a question and answer is mainly about the (semantic similarity) value regularities rather than spatial regularities.

2. Incorporate attention scheme over question terms: We incorporate the attention scheme over the question terms using a gating function, so that we can explicitly discriminate the question term importance.

3. Extensive experimental evaluation and promising results. We perform a thorough experimental study based on the TREC QA dataset from TREC QA tracks 8-13, which appears to be one of the most widely used benchmarks for answer re-ranking. Unlike previous methods using CNNs (Yu et al., 2014; Severyn and Moschitti, 2015) and LSTMs (Wang and Nyberg, 2015), which showed inferior results without combining additional features, our model can achieve better performance than a state-of-art method using linguistic feature engineering and comparable performance with previous deep learning models with combined additional features. If we combine our model with a simple additional feature like

QL, our method can achieve the state-of-the-art performance among current existing methods for ranking answers under multiple metrics.

3.2 Attention-based Neural Matching Model

In this section we present the proposed model referred as **aNMM** (attention-based Neural Matching Model), which is shown in Figure 3.1. Before we introduce our model, we firstly define some terminologies.

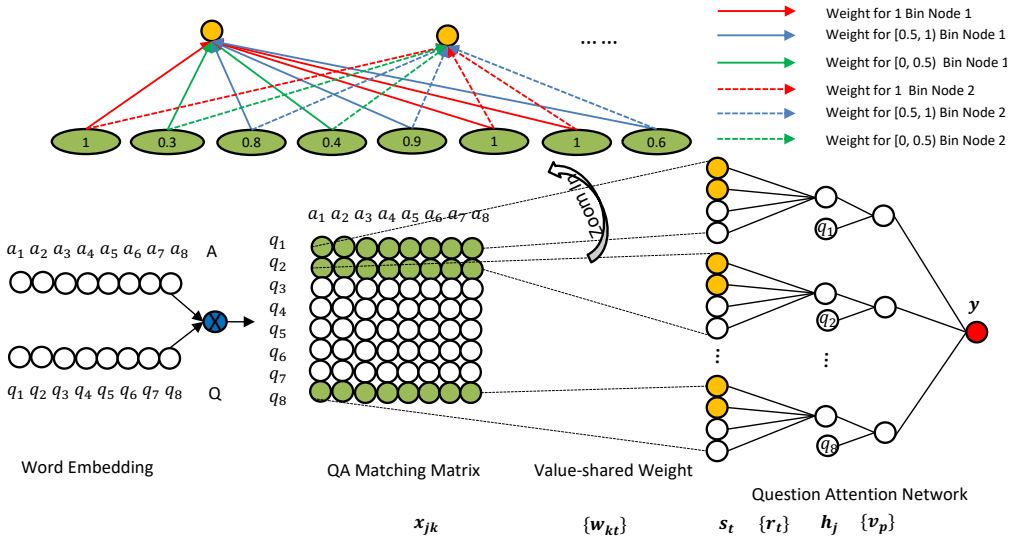


Figure 3.1: The proposed architecture of attention-based neural matching model (aNMM-2) for ranking answers.

3.2.1 Terminology

- Short Answer Text: we use *Short Answer Text* to refer to a short fact, answer sentences or answer passages that can address users' information needs in the issued questions. This is the ranking object in this paper and includes answers in various lengths. In the experiments of this paper, we mainly focus on ranking answer sentences that contain correct answer facts as in TREC QA data.

- QA Matching Matrix: we use *QA Matching Matrix* to refer to a matrix which represents the semantic matching information of term pairs from a question and

answer pair. Given a question $\mathbf{q}$ with length M and an answer $\mathbf{a}$ with length N, a QA matching matrix is an M by N matrix $\mathbf{P}$, where $\mathbf{P}_{j,i}$ denotes the semantic similarity between term $\mathbf{q}_j$ and term $\mathbf{a}_i$ measured by the cosine similarity of the corresponding word embeddings of terms. If $\mathbf{q}_j$ and $\mathbf{a}_i$ are the same term, we assign $\mathbf{P}_{j,i}$ as 1.

- QA Matching Vector: we use *QA Matching Vector* to refer to a row in the QA matching matrix. As presented before, the j-th row of the QA matching matrix $\mathbf{P}$ contains the semantic similarity between $\mathbf{q}_j$ and all terms in answer $\mathbf{a}$. We can make a similar observation to find the association between the j-th column in $\mathbf{P}$ with the j-th term in answer $\mathbf{a}$.

3.2.2 Model Overview

Our method contains three steps as follows:

1. We construct QA matching matrix for each question and answer pair with pre-trained word embeddings.

2. We then employ a deep neural network with value-shared weighting scheme in the first layer, and fully connected layers in the rest to learn hierarchical abstraction of the semantic matching between question and answer terms.

3. Finally, we employ a question attention network to learn question term importance and produce the final ranking score.

We propose two neural matching model architectures and compare the effectivenesses of them. We firstly describe a basic version of the architecture, which is referred to as **aNMM-1**.

In the following sections, we will explain in detail the two major designs of aNMM-1, i.e., value-shared weights and question attention network.

3.2.3 Value-shared Weighting

We first train word embeddings with the Word2Vec tool by Mikolov et al. (2013) with the English Wikipedia dump to construct QA matching matrices. Given a question sentence and an answer sentence, we compute the dot product of the normalized word embeddings of all term pairs to construct the QA matching matrix $\mathbf{P}$ as defined in Section 3.2.1. A major problem with the QA matching matrix is the variable size due to the different lengths of answers for a given question. To solve this problem, one can use CNN with pooling strategy to handle the variable size. However, as we have mentioned before, CNNs basically use a position-shared weighting scheme which may not fit semantic matching between questions and answers. Important question terms and semantically similar answer words could appear anywhere in questions/answers due to the complex linguistic property of natural languages. Thus we adopt the following method to handle the various length problem:

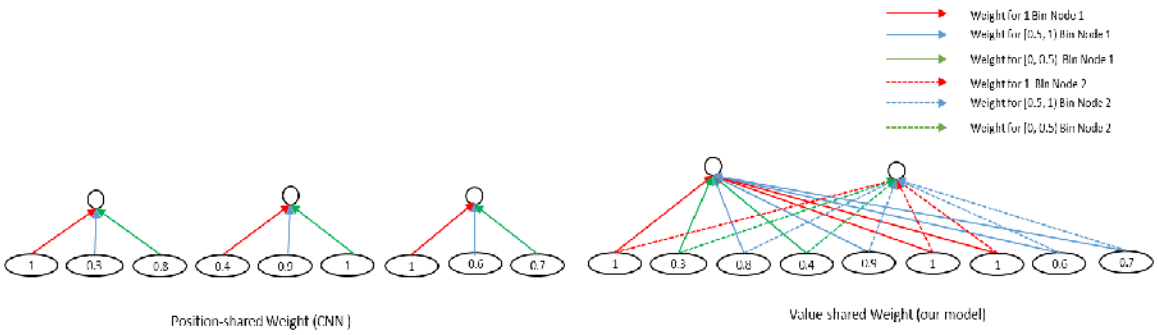


Figure 3.2: The comparison of position-shared weight in CNN and value-shared weight in aNMM. In CNN, the weight associated with a node only depends on its position or relative location as specified by the filters. In aNMM, the weight associated with a node depends on its value.

Value-shared Weights: For this method, the assumption is that matching signals in different ranges play different roles in deciding the final ranking score. Thus we introduce the value-shared weighting scheme to learn the importance of different levels of matching signals. The comparison between the position-shared weight and value-shared weight is shown in Figure 3.2. We can see that for position-shared weights,

the weight associated with a node only depends on its position or relative location as specified by the filters in CNN. However in our model, the weight associated with a node depends on its value. The value of a node denotes the strength of the matching signal between term pairs of questions and answers from the QA matching matrix, as explained in Section 3.2.1. Such a setting enables us to use the learned weights to encode how to combine different levels of matching signals. After this step, the size of the hidden representation becomes fixed and we can use normal fully connected layers to learn higher level representations. We use the term *bin* to denote a specific range of matching signals. since $\mathbf{P}_{j,i} \in [-1, 1]$, if we set the size of bins as 0.1, then we have 21 bins where there is a separate bin for $\mathbf{P}_{j,i} = 1$ to denote exact match of terms.

Specifically, value-shared weights are adopted in the forward propagation prediction process from the input layer to the hidden layer over each question term in aNMM-1 as follows: let $\mathbf{w}$ denote a $K + 1$ dimensional model parameter from input layer to hidden layer. x_{jk} denotes the sum of all matching signals within the k-th value range or bin. For each QA matching vector of a given query $\mathbf{q}$, the combined score after the activation function of the j-th node in hidden layer is defined as

$$h_j = \delta(\sum_{k=0}^{K} w_k \cdot x_{jk}) \tag{3.1}$$

where j is the index of question words in $\mathbf{q}$. We use the sigmoid function as the activation function, which is commonly adopted in many neural network architectures.

3.2.4 Question Attention Network

In addition to value-shared weighting, another model component of aNMM-1 is the question attention network. In a committee of neural networks which consists of multiple networks, we need to combine the output of these networks to output a

final decision vector. The question attention network uses the gating function (Su and Basu, 2001) to control the output of each network in this process. Specifically, in aNMM-1 we use the softmax gate function to combine the output of multiple networks where each network corresponds to a question term as shown in Figure 3.1. We feed the dot product of query word embedding and model parameter to the softmax function to represent the query term importance. In this setting, we can directly compare the relative term importance of query words within the same query with softmax function. We also tried sigmoid gate function, but this did not perform as well as softmax gate function.

Softmax gate function is used in the forward propagation process from the hidden layer to the output layer as follows: from the hidden layer to the output layer, we add a softmax gate function to learn question attention. Let $\mathbf{v}$ denote a P dimensional vector which is a model parameter. We feed the dot product of query word embedding $\mathbf{q}_j$ and $\mathbf{v}$ to the softmax function to represent the query term importance as shown in Equation 3.2. Note that we normalize the query word embedding before computing the dot product.

$$y = \sum_{j=1}^{M} g_j \cdot h_j = \sum_{j=1}^{M} \frac{\exp(\mathbf{v} \cdot \mathbf{q}_j)}{\sum_{l=1}^{L} \exp(\mathbf{v} \cdot \mathbf{q}_l)} \cdot \delta(\sum_{k=0}^{K} w_k \cdot x_{jk}) \tag{3.2}$$

Unlike previous models like CNNs (Severyn and Moschitti, 2015) and BLSTM (Wang and Nyberg, 2015), which learn the semantic match score between questions and answers through representation learning from matching matrix or question / answer pair sequences, aNMM achieves this by combining semantic matching signals of term pairs in questions and answers weighted by the output of question attention network, where softmax gate functions help discriminate the term importance or attention on different question terms.

3.2.5 Model Training

For aNMM-1, the model parameters contain two sets: 1) The value-shared weights $\mathbf{w_k}$ for combining matching signals from the input layer to the hidden layer. 2) The parameters $\mathbf{v_p}$ in the gating function from the hidden layer to the output layer.

To learn the model parameters from the training data, we adopt a pair-wise learning strategy with a large margin objective. Firstly we construct triples $(\mathbf{q}, \mathbf{a}^+, \mathbf{a}^-)$ from the training data, with $\mathbf{q}$ matched with $\mathbf{a}^+$ better than with $\mathbf{a}^-$. We have the ranking-based loss as the objective function as following:

$$e(\mathbf{q}, \mathbf{a}^+, \mathbf{a}^-; \mathbf{w}, \mathbf{v}) = \max(0, 1 - S(\mathbf{q}, \mathbf{a}^+) + S(\mathbf{q}, \mathbf{a}^-)) \tag{3.3}$$

where $S(\mathbf{q}, \mathbf{a})$ denote the predicted matching score for QA pair $(\mathbf{q}, \mathbf{a})$. During training stage, we will scan all the triples in training data. Given a triple $(\mathbf{q}, \mathbf{a}^+, \mathbf{a}^-)$, we will compute $\Delta S = 1 - S(\mathbf{q}, \mathbf{a}^+) + S(\mathbf{q}, \mathbf{a}^-)$. If $\Delta S \leq 0$, we will skip this triple. Otherwise, we need to update model parameters with back propagation algorithm to minimize the objective function.

Under softmax gate function setting, the gradients of e w.r.t. $\mathbf{v}$ from hidden layer to the output layer is shown in Equation 3.4

$$\frac{\partial e}{\partial v_p} = \sum_{j=1}^{M} \frac{\partial g_j}{\partial v_p} \cdot (-\delta(u^+) + \delta(u^-)) \tag{3.4}$$

where

$$u^+ = \sum_{k=0}^{K} w_k \cdot x_{jk}^+, u^- = \sum_{k=0}^{K} w_k \cdot x_{jk}^-$$

$\frac{\partial g_j}{\partial v_p}$ can be derived as

$$\frac{\exp(\mathbf{v}\cdot\mathbf{q}_j)\cdot q_{jp}\sum_{l=1}^{M}\exp(\mathbf{v}\cdot\mathbf{q}_l)-\exp(\mathbf{v}\cdot\mathbf{q}_j)\sum_{l=1}^{M}\exp(\mathbf{v}\cdot\mathbf{q}_l)\cdot q_{lp}}{(\sum_{l=1}^{M}\exp(\mathbf{v}\cdot\mathbf{q}_l))^2}$$

The gradient of e w.r.t. $\mathbf{w}$ from input layer to hidden layer is shown in Equation 3.5.

$$\begin{aligned}\frac{\partial e}{w_k} &= \sum_{j=1}^{M}\frac{\exp(\mathbf{v}\cdot\mathbf{q}_j)}{\sum_{l=1}^{L}\exp(\mathbf{v}\cdot\mathbf{q}_l)}\cdot(-\delta(u^+)(1-\delta(u^+))x_{jk}^+ \\ &\quad +\delta(u^-)(1-\delta(u^-))x_{jk}^-) \end{aligned} \tag{3.5}$$

With the formulas of gradients, we can perform stochastic gradient descent to learn model parameters. We use mini-batch gradient descent to achieve more robust performance on the ranking task. For the learning rate, we adopt adaptive learning rate: $\eta = \eta_0(1-\epsilon)$, where ϵ will approach 1 with more iterations. Such a setting has better guarantee for convergence.

3.2.6 Extension to Deep Neural Networks with Multiple Sets of Value-shared Weights

In aNMM-1, we can only use one set of value-shared weights for each QA matching vector. We further propose a more flexible neural network architecture which could enable us to use multiple sets of value-shared weights for each QA matching vector, leading to multiple intermediate nodes in the first hidden layer, as shown in Figure 3.1 by the yellow color. We refer to this extended model as **aNMM-2**. The model architecture shown in Figure 3.1 is corresponding to aNMM-2.

3.2.6.1 Forward Propagation Prediction

For aNMM-2, we add a hidden layer in the neural network where we learn multiple combined scores from the input layer. With this hidden layer, we define multiple

weight vectors as $\mathbf{w}$. Thus $\mathbf{w}$ becomes a two dimensional matrix. The formula for the forward propagation prediction is as follows:

$$y = \sum_{j=1}^{M} \tau(\mathbf{v} \cdot \mathbf{q}_j) \cdot \delta(\sum_{t=0}^{T} r_t \cdot \delta(\sum_{k=0}^{K} w_{kt} x_{jk})) \tag{3.6}$$

where $\tau(\mathbf{v} \cdot \mathbf{q}_j) = \frac{\exp(\mathbf{v} \cdot \mathbf{q}_j)}{\sum_{l=1}^{L} \exp(\mathbf{v} \cdot \mathbf{q}_l)}$ and τ denote the softmax gate function. T is the number of nodes in hidden layer 1. r_t is the model parameter from hidden layer 1 to hidden layer 2, where we feed the linear combination of outputs of nodes in hidden layer 1 to an extra activation function comparing with Equation 3.2. Then from hidden layer 2 to output layer, we sum over all outputs of nodes in hidden layer 2 weighted by the outputs of softmax gate functions, which also form the question attention network.

3.2.6.2 Back Propagation for Model Training

For aNMM-2, we have three sets of model parameters: 1) w_{kt} from input layer to hidden layer 1; 2) r_t from hidden layer 1 to hidden layer 2; 3) v_p from hidden layer 2 to output layer. All three sets of parameters are updated through back propagation. The definition of the objective function is the same as Equation 3.3. The back propagation process for model parameter learning is described as follows:

From hidden layer 2 to output layer: the gradients of the objective function w.r.t. $\mathbf{v}$ is as following:

$$\frac{\partial e}{\partial v_p} = \sum_{j=1}^{M} \frac{\partial g_j}{\partial v_p} \cdot (-h_j^+ + h_j^-) \tag{3.7}$$

Where

$h_j^+ = \delta(\sum_{t=0}^{T} r_t \cdot \delta(\sum_{k=0}^{K} w_{kt} x_{jk}^+))$

$h_j^- = \delta(\sum_{t=0}^{T} r_t \cdot \delta(\sum_{k=0}^{K} w_{kt} x_{jk}^-))$

From hidden layer 1 to hidden layer 2: the gradients of the objective function w.r.t. $\mathbf{r}$ is as following:

$$\frac{\partial e}{\partial r_t} = \sum_{j=1}^{M} \tau(\mathbf{v} \cdot \mathbf{q}_j)(-h_j^+)(1-h_j^+)s_t^+ + h_j^-(1-h_j^-)s_t^-)$$

Where

$s_t^+ = \delta(\sum_{k=0}^{K} w_{kt} x_{jk}^+)$

$s_t^- = \delta(\sum_{k=0}^{K} w_{kt} x_{jk}^-)$.

From input layer to hidden layer 1: the gradients of the objective function w.r.t. $\mathbf{w}$ is as following:

$$\begin{aligned}\frac{\partial e}{\partial w_{kt}} = & \sum_{j=1}^{M} \tau(\mathbf{v} \cdot \mathbf{q}_j)(-\frac{\partial y^+}{u_j'^+} \cdot r_t \cdot \delta(u_t^+)(1-\delta(u_t^+)) \cdot x_{jk}^+ \\ & + \frac{\partial y^-}{u_j'^-} \cdot r_t \cdot \delta(u_t^-)(1-\delta(u_t^-)) \cdot x_{jk}^-) \end{aligned} \tag{3.8}$$

Where

$u_j'^+ = \sum_{t=0}^{T} r_t \cdot \delta(\sum_{k=0}^{K} w_{kt} x_{jk}^+)$

$u_j'^- = \sum_{t=0}^{T} r_t \cdot \delta(\sum_{k=0}^{K} w_{kt} x_{jk}^-)$

Initially we will randomly give the values of model parameters. Then we will use back propagation to update the model parameters. When the learning process converge, we use the learned model parameters for prediction to rank short answer texts.

Table 3.1: The statistics of the TREC QA data set.

Data	#Questions	#QA pairs	%Correct	#Answers/Q
TRAIN-ALL	1,229	53,417	12.00%	43.46
TRAIN	94	4,718	7.40%	50.19
DEV	82	1,148	19.30%	14.00
TEST	100	1,517	18.70%	15.17

3.3 Experiments

3.3.1 Data Set and Experiment Settings

We use the TREC QA data set [1] created by Wang et al. (2007) from TREC QA track 8-13 data, with candidate answers automatically selected from each question's document pool using a combination of overlapping non-stop word counts and pattern matching. This data set is one of the most widely used benchmarks for answer re-ranking. Table 3.1 shows the statistics of this data set. The dataset contains a set of factoid questions with candidate answers which are limited to a single sentence. There are two training data sets: TRAIN and TRAIN-ALL. Answers in TRAIN have manual judgments for the answer correctness. The manual judgment of candidate answer sentences is provided for the entire TREC 13 set and for a part of questions from TREC 8-12. TRAIN-ALL is another training set with much larger number of questions. The correctness of candidate answer sentences in TRAIN-ALL is identified by matching answer sentences with answer pattern regular expressions provided by TREC. This data set is more noisy, however it provides many more QA pairs for model training. There is a DEV set for hyper-parameter optimization and TEST set for model testing. We use the same train/dev/test partition in our experiments to directly compare our results with previous works. For data preprocess, we perform tokenization without stemming. We maintain stop words during the model training stage.

[1] `https://github.com/aseveryn/deep-qa`

Word Embeddings: we obtain pre-trained word embeddings with the Word2Vec tool by Mikolov et al. (2013) with the English Wikipedia dump. We use the skip-gram model with window size 5 and filter words with frequency less than 5 following the common practice in many neural embedding models. For the word vector dimension, we tune it as a hyper-parameter on the validation data starting from 200 to 1000. Embeddings for words not present are randomly initialized with sampled numbers from uniform distribution U[-0.25,0.25], which follows the same setting as (Severyn and Moschitti, 2015).

Model Hyper-parameters: for the setting of hyper-parameters, we set the number of bins as 600, word embedding dimension as 700 for aNNM-1, the number of bins as 200, word embedding dimension as 700 for aNNM-2 after we tune hyper-parameters on the provided DEV set of TREC QA data.

3.3.2 Evaluation and Metrics

For evaluation, we rank answer sentences with the predicted score of each method and compare the rank list with the ground truth to compute metrics. We choose Mean Average Precision (MAP) and Mean Reciprocal Rank (MRR), which are commonly used in information retrieval and question answering, as the metric to evaluate our model.

The definition of MRR is as follows:

$MRR = \frac{1}{|Q|} \sum_{q \in Q} \frac{1}{rank(fa)}$

where $rank(fa)$ is the position of the first correct answer in the rank list for the question q. Thus MRR is only based on the rank of the first correct answer. It is more suitable for the cases where the rank of the first correct answer is emphasized or each question only have one correct answer. On the other hand, MAP looks at the ranks of all correct answers. It is computed as following:

$MAP = \frac{1}{|Q|} \sum_{q \in Q} AP(q)$

where $AP(q)$ is the average precision for each query $q \in \mathbf{Q}$. Thus MAP is the average performance on all correct answers. We use the official $trec\_eval$[2] scripts for computing these metrics.

3.3.3 Model Learning Results

In this section, we give some qualitative analysis and visualization of our model learning results. Specifically, we analyze the learned value-shard weights and question term importance by aNMM.

3.3.3.1 Value-shared Weight

We take the learned value-shared weights of aNMM-1 as the example. Figure 3.3 shows the learned value-shared weights by aNMM-1. In aNMM-1, for each QA matching vector, there is only one bin node. Thus the learned value-shared weights for aNMM-1 is a one dimension vector. For aNMM-1, we set the number of bins to 600 as presented in Section 3.3.1. Note that the x-axis is the index of bin range and the y-axis is the value-shared weights corresponding to each bin range. The range of match signals is [-1,1] from the left to the right. We make the following observations: (1) The exact match signal which is corresponding to 1 in the last bin is tied with a very large weight, which shows that exact match information is very important. (2) For positive matching score from $(0, 1)$, which is corresponding to bin index $(300, 600)$, the learned value-shared weights are different for matching score range $(0.5, 1)$ (bin index $(450, 600)$) and matching score range $(0, 0.5)$ (bin index $(300, 450)$) . We can observe many positive value-shared weights for matching score range$(0.5, 1)$ and negative value-shared weights for matching score range$(0, 0.5)$. This makes sense since high semantic matching scores are positive indicators on answer correctness, whereas low semantic matching scores indicate that the candidate answer

[2] `http://trec.nist.gov/trec_eval/`

sentences contain irrelevant terms. (3) For negative matching scores from $(-1, 0)$, we can see there is not a lot of differences between value-shared weights for different ranges. A major reason is that most similarity scores based on word embeddings are positive. Therefore, we can remove bins corresponding to negative matching scores to reduce the dimension of value-shared weight vectors, which can help improve the efficiency of the model training process. We will show more quantitative results on the comparison between value-shared weights and position-shared weights in CNN in Section 3.3.4.

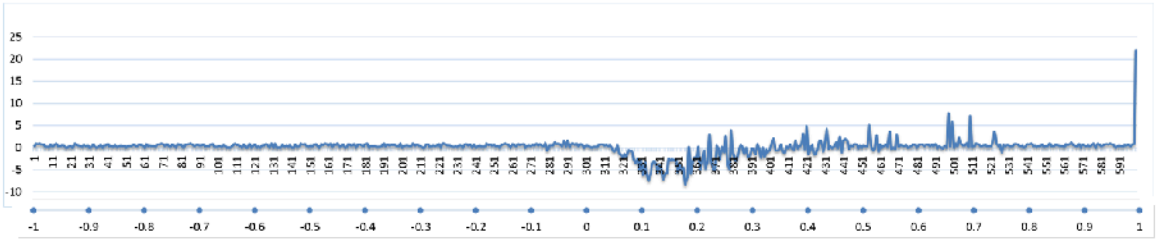


Figure 3.3: Visualization of learned value-shared weights of aNMM-1. The x-axis is index of bin ranges and the y-axis is the value-shared weights corresponding to each bin range. The range of match signals is [-1,1] from the left to the right.

3.3.3.2 Question Term Importance

Next we analyze the learned question term importance of our model. We also use the learned question term importance of aNMM-1 as an example. Table 3.2 shows the examples of learned question term importance by aNMM-1. We also visualize the question term importance in Figure 3.4. Based on the results in the table and the figure, we can clearly see that aNMM-1 learns reasonable term importance. For instance, with the question attention network, aNMM-1 discovers important terms like "khmer", "rouge", "power" as for the question "*When did the khmer rouge come into power?*". Terms like "age", "rossinin", "stop", "writing","opera" are highlighted for the question "*At what age did rossini stop writing opera?*". For the question "*Where*

Table 3.2: Examples of learned question term importance by aNMM-1.

test_14	when	did	the	khmer	rouge	come	into	power
Term Importance	4.91E-03	7.18E-04	8.97E-04	5.67E-01	2.13E-01	1.81E-02	6.59E-03	1.89E-01
test_66	where	was	the	first	burger	king	restaurant	opened
Term Importance	2.16E-04	5.67E-04	1.96E-04	2.57E-03	3.43E-01	4.39E-01	5.35E-03	2.08E-01
train_84	at	what	age	did	rossini	stop	writing	opera
Term Importance	5.06E-02	2.54E-03	6.17E-02	2.68E-03	3.89E-01	4.28E-01	9.29E-03	5.64E-02

Table 3.3: The comparision of aNMM-1/aNMM-2 with aNMM-IDF which is a degenerate version of our model where we use IDF to directly replace the output of question attention network.

Training Data	TRAIN		TRAIN-ALL	
Method	MAP	MRR	MAP	MRR
aNMM-IDF	0.6624	0.7376	0.7225	0.7873
aNMM-2	0.7191	0.7974	**0.7407**	0.7969
aNMM-1	**0.7334**	**0.8020**	0.7385	**0.7995**

was the first burger king restaurant opened?" mentioned in Section 3.1, "burger", "king", "opened" are treated as important question terms.

Figure 3.4: Visualization of learned question term importance by aNMM-1.

An interesting question is how the learned term importance compare with traditional IR term weighting methods such as IDF. We design an experiment to compare aNMM-1/aNMM-2 with aNMM-IDF, which is a degenerate version of our model where we use IDF to directly replace the output of question attention network. In this case, $\tau(\mathbf{v} \cdot \mathbf{q}_j)$ in Equation 3.6 is replaced by the IDF of the j-th question term. Table 3.3 shows the results. We find that if we replace the output of question attention network of aNMM with IDF, it will decrease the answer ranking performance,

especially on TRAIN data. Thus, we can see that with the optimization process in the back propagation training process, aNMM can learn better question term weighting score than heuristic term weighting functions like IDF.

3.3.4 Experimental Results for Ranking Answers

3.3.4.1 Learning without Combining Additional Features

Our first experimental setting is ranking answer sentences directly by the predicted score from aNMM without combining any additional features. This will enable us to answer RQ1 proposed in Section 3.1. Table 3.4 shows the results of TREC QA on TRAIN and TRAIN-ALL without combining additional features. In this table, we compare the results of aNMM with other previous deep learning methods including CNN (Yu et al., 2014; Severyn and Moschitti, 2015) and LSTM (Wang and Nyberg, 2015). We summarize our observations as follows: (1) Both aNMM-1 and aNMM-2 show significant improvements for MAP and MRR on TRAIN and TRAIN-ALL data sets comparing with previous deep learning methods. Specifically, if we compare the results of aNMM-1 with the strongest deep learning baseline method by Severyn et al. (Severyn and Moschitti, 2015) based on CNN, we can see aNMM-1 outperform CNN for 14.67% in MAP on TRAIN, 9.15% in MAP on TRAIN-ALL. For MRR, we can also observe similar significant improvements of aNMM-1. These results show that with the value-shared weight scheme instead of the position-shared weight scheme in CNN and term importance learning with question attention network, aNMM can predict ranking scores with much higher accuracy comparing with previous deep learning models for ranking answers. (2) If we compare the results of aNMM-1 and aNMM-2, we can see their results are very close. aNMM-1 has slightly better performance than aNMM-2. This result indicates that adding one more hidden layer to incorporate multiple bin nodes does not necessarily increase the performance for answer ranking in TREC QA data. From the perspective of model efficiency, aNMM-1 could be

Table 3.4: Results of TREC QA on TRAIN and TRAIN-ALL without combining additional features (Compare with deep learning methods).

Training Data	TRAIN		TRAIN-ALL	
Method	MAP	MRR	MAP	MRR
Yu et al. (2014) (Yu et al., 2014)	0.5476	0.6437	0.5693	0.6613
BLSTM(2015) (Wang and Nyberg, 2015)	/	/	0.5928	0.6721
CDNN (2015) (Severyn and Moschitti, 2015)	0.6258	0.6591	0.6709	0.7280
aNMM-2	0.7191	0.7974	**0.7407**	0.7969
aNMM-1	**0.7334**	**0.8020**	0.7385	**0.7995**

Table 3.5: Results of TREC QA on TRAIN-ALL without combining additional features (Compare with methods using feature engineering).

Method	MAP	MRR
Wang et al. (2007)	0.6029	0.6852
Heilman and Smith (2010)	0.6091	0.6917
Wang and Manning (2010)	0.5951	0.6951
Yao et al. (2013)	0.6307	0.7477
Severyn and Moschitti (2013)	0.6781	0.7358
Yih et al. (2013)	0.7092	0.7700
aNMM-2	**0.7407**	0.7969
aNMM-1	0.7385	**0.7995**

a better choice since it can be trained much faster with good prediction accuracy. However, for larger training data sets than TREC QA data, aNMM-2 could have better performance since it has more model parameters and is suitable for fitting larger training data set. We leave the study of impact of the number of hidden layers in aNMM to future work.

Table 3.5 shows the comparison between aNMM with previous methods using feature engineering on TRAIN-ALL without combining additional features. We find that both aNMM-1 and aNMM-2 achieve better performance comparing with other methods using feature engineering. Specifically, comparing the results of aNMM-1 with the strongest baseline by Yih et al. (Yih et al., 2013) based on enhanced lexical semantic models, aNMM-1 achieves 4.13% gain for MAP and 3.83% gain for MRR. These results show that it is possible to build a uniform deep learning model such that it can achieve better performance than methods using feature engineering. To

the best of our knowledge, aNMM is the first deep learning model that can achieve good performance comparing with previous methods either based on deep learning models or feature engineering for ranking answers without any additional features, syntactic parsers and external resources except for pre-trained word embeddings.

3.3.4.2 Learning with Combining Additional Features

Our second experimental setting is to address RQ2 proposed in Section 3.1, where we ask whether our model can outperform the state-of-the-art performance achieved by CNN (Yu et al., 2014; Severyn and Moschitti, 2015) and LSTM (Wang and Nyberg, 2015) for answer ranking when combining additional features. We combine the predicted score from aNMM-1 and aNMM-2 with the Query Likelihood (QL) (Croft et al., 2009) score using LambdaMART (Wu et al., 2010) following a similar approach to (Wang and Nyberg, 2015). We use the implementation of LambdaMART in jforests.[3] We compare the results with previous deep learning models with additional features. Table 3.6 shows the results on TRAIN and TRAIN-ALL when combining additional features. We can see that with combined features, both aNMM-1 and aNMM-2 have better performance. aNMM-1 also outperforms CNN by Severyn et al. (Severyn and Moschitti, 2015) which is the current state-of-the-art method for ranking answers in terms of both MAP and MRR on TRAIN and TRAIN-ALL.

We also tried to combine aNMM score with other additional features such as word overlap features, IDF weighted word overlap features and BM25 as in previous research (Yu et al., 2014; Severyn and Moschitti, 2015; Wang and Nyberg, 2015). The results were either similar or worse than combining aNMM score with QL. For aNMM, the gains after combining additional features are not as large as neural network models like CNN in (Severyn and Moschitti, 2015) and LSTM in (Wang and Nyberg, 2015). We think the reasons for this are two-fold: (1) The QA matching ma-

[3] `https://github.com/yasserg/jforests` (Ganjisaffar et al., 2011).

Table 3.6: Results of TREC QA on TRAIN and TRAIN-ALL with combining additional features.

Training Data	TRAIN		TRAIN-ALL	
Method	MAP	MRR	MAP	MRR
Yu et al. (2014) (Yu et al., 2014)	0.7058	0.7800	0.7113	0.7846
BLSTM (2015) (Wang and Nyberg, 2015)	/	/	0.7134	0.7913
CDNN (2015) (Severyn and Moschitti, 2015)	0.7329	0.7962	0.7459	0.8078
aNMM-2	0.7306	0.7968	0.7484	0.8013
aNMM-1	**0.7417**	**0.8102**	**0.7495**	**0.8109**

trix in aNMM model can capture exact match information by assigning 1 to matrix elements if the corresponding answer term and question term are the same. This exact match information includes match between numbers and proper nouns, which are highlighted in previous research work (Severyn and Moschitti, 2015) as especially important for factoid questions answering, where most of the questions are of type *what*, *when* , *who* that are looking for answers containing numbers or proper nouns. Within aNMM architecture, this problem has already been handled with QA matching matrix. Thus incorporating word overlap features will not help much for improving the performance of aNMM. (2) In addition to exact match information, aNMM could also learn question term importance like IDF information through question attention network. Instead of empirically designing heuristic functions like IDF, aNMM can get learning based question term importance score. As analyzed in Section 3.3.3.2, with the optimization process in the back propagation training process, aNMM can learn similar or even better term weighting score than IDF. Thus combining aNMM score with features like IDF weighted word overlap features and BM25 may not increase the performance of aNMM by a large margin as the case in related research works (Yu et al., 2014; Severyn and Moschitti, 2015; Wang and Nyberg, 2015).

3.3.4.3 Results Summary

Finally we summarize the results of previously published systems on the QA answer ranking task in Table 3.7. We can see aNMM trained with TRAIN-ALL set

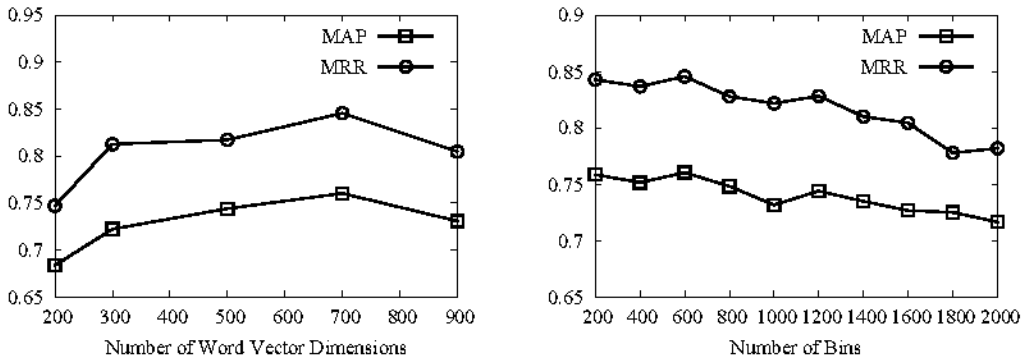
Table 3.7: Overview of previously published systems on the QA answer ranking task. All reported results are from the best setting of each model trained on TRAIN-ALL data.

Method	MAP	MRR
Wang et al. (2007)	0.6029	0.6852
Heilman and Smith (2010)	0.6091	0.6917
Wang and Manning (2010)	0.5951	0.6951
Yao et al. (2013)	0.6307	0.7477
Severyn and Moschitti (2013)	0.6781	0.7358
Yih et al. (2013)	0.7092	0.7700
Yu et al. (2014)	0.7113	0.7846
Wang and Nyberg (2015)	0.7134	0.7913
Severyn and Moschitti (2015)	0.7459	0.8078
aNMM	**0.7495**	**0.8109**

beats all the previous state-of-the art systems including both methods using feature engineering and deep learning models. These results are very promising since aNMM requires no manual feature engineering, no expensive processing by various NLP parsers and no external results like large scale knowledge base except for pre-trained word embeddings. Furthermore, even without combining additional features, aNMM still performs well for answer ranking, showing significant improvements over previous deep learning model with no additional features and linguistic feature engineering methods.

3.3.5 Parameter Sensitivity Analysis

We perform parameter sensitivity analysis of our proposed model aNMM. We focus on aNMM-1 as the example due to the space limitation. For aNMM-1, we fix the number of bins as 600 and change the dimension of word vectors. Similarly, we fix the dimension of word vectors as 700 and vary the number of bins. Figure 3.5 shows the change of MAP and MRR on the validation data as we vary the hyper-parameters. We summarize our observations as follows: (1) For word vector dimension, the range $(300, 700)$ is a good choice as much lower or higher word vector dimensions will hurt the performance. The choice of word vector dimension also depends on the size of



(a) Tune vector dimension.

(b) Tune bin number.

Figure 3.5: Tune hyper-parameters on validation data.

training corpus. Larger corpus requires higher dimension of word vectors to embed terms in vocabulary. (2) For the number of bins, we can see that MAP and MRR will decrease as the bin number increase. Too many bins will increase the model complexity, which leads aNMM to be more likely to overfit the training data. Thus choosing suitable number of bins by optimizing hyper-parameter on validation data can help improve the performance of aNMM.

3.3.6 Experimental Results on Microsoft Research WikiQA Data

3.3.6.1 WikiQA Data

We further show the experiment results of aNMM-1 and aNMM-2 on the Microsoft Research WikiQA Data (Yang et al., 2015). The WikiQA corpus is a publicly available set of question and sentence pairs, collected and annotated for research on open-domain question answering. In order to reflect the true information need of search users, Yang et al. (Yang et al., 2015) used Bing query logs as the question source. Each question is linked to a Wikipedia page that potentially has the answer. They used sentences in the summary section of a Wiki page as the candidate answers because the summary section could provide the basic and most important information about the topic. With the help of crowdsourcing, the data set included 3,047 questions

Table 3.8: The statistics of the WikiQA data set. Note that "CandidateAS", "CorrectAS", "AvgLenOfQ", "AvgLenOfCanAS", "'QWithNoCAS' denote "candidate answer sentence", "correct answer sentence", "average length of question", "average length of candidate answer sentence", "question with no correct answer sentence" respectively.

Data	Train	Dev	Test	Total
#Questions	2,118	296	633	3,047
#CandidateAS	20,360	2,733	6,165	29,258
#CorrectAS	1,040	140	293	1,473
AvgLenOfQ	7.16	7.23	7.26	7.18
AvgLenOfCanAS	25.29	24.59	24.95	25.15
#QWithNoCAS	1,245	170	390	1,805

Table 3.9: The experimental results on WikiQA data set. Note that although the performances of our method aNMM are close to the baseline CNN-Count, our method does not need to be combined with additional features like overlapped word count features.

Method	MAP	MRR
WordCount	0.4891	0.4924
WeightedWordCount	0.5099	0.5132
LCLR	0.5993	0.6086
PV	0.5110	0.5160
CNN	0.6190	0.6281
PV-Count	0.5976	0.6058
CNN-Count	0.6520	0.6652
aNMM-2	0.6455	0.6527
aNMM-1	**0.6562**	**0.6687**

and $29,258$ sentences in the dataset, where $1,473$ sentences were labeled as answer sentences to their corresponding questions. The statistics of this data set is shown in Table 3.8.

Note that the advantage of this data set is that all the questions are from the real search logs. Thus those questions are more likely search queries issued by users in real web search engines. However, one weakness with this data is that nearly two-thirds of questions contain no correct answers in in the candidate answer sentences. These questions could be useful in training data since they provide some negative training instances. But we need to filter such questions in the Dev/Test data to evaluate the performance of the answer sentence ranking task.

3.3.6.2 Results on WikiQA Data

The experimental results on WikiQA data set are shown in table 3.9. The results of the baselines shown in this table are cited from the paper by Yang et. al.(Yang et al., 2015). These baselines include the following methods:

- Word matching methods. Word Count counts the number of non-stop words in the question that also occur in the answer sentence. Weighted Word Count weights these counts by the IDF values of question terms.

- LCLR. This is a method based on rich lexical semantic features including word/lemma matching, WordNet and vector-space lexical semantic models proposed by Yih et al. (Yih et al., 2013).

- Paragraph Vector (PV). This method uses the cosine similarity score between the question vector and the candidate answer sentence vector learned by the Paragraph Vector model (Le and Mikolov, 2014).

- Convolutional Neural Networks (CNN). This is the bigram CNN model proposed by Yu et al. (Yu et al., 2014).

- Methods combining deep learning methods with word count features. PV-Count and CNN-Count are two methods which combine the score from deep learning models with the two word matching features by training a logistics regression model. CNN-Count is also proposed by Yu et al. (Yu et al., 2014).

Comparing the results of our proposed aNMM-1 and aNMM-2 model with these previous proposed methods, we make the following observations: 1) Both the performances of aNMM-1 and aNMM-2 are significantly better than word matching methods including Word Count, Weighted Word Count and previous deep learning methods like PV and CNN. Note that we don't combine any additional hand crafted features into the learning score of aNMM-1 and aNMM-2 for the experiments on

WikiQA. This double confirms the advantages of the attention based neural matching models for ranking answer sentences. 2) We can see that the performance of aNMM-1 is slightly better than PV-Count and CNN-Count. Thus aNMM-1 can achieve better results than previous methods combining deep learning score with word count features. 3) If we compare the results of aNMM-1 and aNMM-2, they are pretty close to each other. However, aNMM-1 is still trained with much higher efficiency on WikiQA data. Thus aNMM-1 is a better model in terms of both effectiveness and efficiency. Overall, the experimental results on WikiQA data is quite consistent with the results on TREC QA data, which double confirms the effectiveness of our proposed model.

3.4 Summary

In this chapter, we propose an attention based neural matching model as a representation learning method for ranking short answer text. We adopt value-shared weighting scheme instead of position-shared weighting scheme for combining different matching signals and incorporate question term importance learning using a question attention network. We perform a thorough experimental study with the TREC QA dataset from TREC QA tracks 8-13 and show promising results. Unlike previous methods including CNN as in (Yu et al., 2014; Severyn and Moschitti, 2015) and LSTM as in (Wang and Nyberg, 2015), which only show inferior results without combining additional features, our model can achieve better performance than the state-of-art method using linguistic feature engineering without additional features. With a simple additional feature, our method can achieve the new state-of-the-art performance among current existing methods.

CHAPTER 4

MULTI-TURN INFORMATION-SEEKING CONVERSATIONS

4.1 Introduction

Personal assistant systems, such as Apple Siri, Google Now, Amazon Alexa, and Microsoft Cortana, are becoming ever more widely used. These systems, with either text-based or voice-based conversational interfaces, are capable of voice interaction, information search, question answering and voice control of smart devices. This trend has led to an interest in developing conversational search systems, where users would be able to ask questions to seek information with conversation interactions. Research on speech and text-based conversational search has also recently attracted significant attention in the information retrieval (IR) community.

Existing approaches to building conversational systems include generation-based methods (Ritter et al., 2011; Shang et al., 2015) and retrieval-based methods (Ji et al., 2014; Yan et al., 2016a,b, 2017). Compared with generation-based methods, retrieval-based methods have the advantages of returning fluent and informative responses. Most work on retrieval-based conversational systems studies response ranking for *single-turn conversation* (Wang et al., 2013), which only considers a current utterance for selecting responses. Recently, several researchers have been studying *multi-turn conversation* (Yan et al., 2016a; Zhou et al., 2016; Wu et al., 2017; Yan et al., 2017), which considers the previous utterances of the current message as the conversation context to select responses by jointly modeling context information, current input utterance and response candidates. However, existing studies are still suffering from the following weaknesses:

(1) Most existing studies are on open domain chit-chat conversations or task / transaction oriented conversations. Most current work (Ritter et al., 2011; Shang et al., 2015; Ji et al., 2014; Yan et al., 2016a,b, 2017) is looking at open domain chit-chat conversations as in microblog data like Twitter and Weibo. There is some research on task oriented conversations (Young et al., 2010; Wen et al., 2017; Bordes et al., 2017), where there is a clear goal to be achieved through conversations between the human and the agent. However, the typical applications and data are related to completing transactions like ordering a restaurant or booking a flight ticket. Much less attention has been paid to *information oriented conversations*, which is referred to as *information-seeking conversations* in this chapter. Information-seeking conversations, where the agent is trying to satisfy the information needs of the user through conversation interactions, are closely related to conversational search systems. More research is needed on response selection in information-seeking conversation systems.

(2) Lack of modeling external knowledge beyond the dialog utterances. Most research on response selection in conversation systems are purely modeling the matching patterns between user input message (either with context or not) and response candidates, which ignores external knowledge beyond the dialog utterances. Similar to Web search, information-seeking conversations could be associated with massive external data collections that contain rich knowledge that could be useful for response selection. This is especially critical for information-seeking conversations, since there may be not enough signals in the current dialog context and candidate responses to discriminate a good response from a bad one due to the wide range of topics for user information needs. An obvious research question is how to utilize external knowledge effectively for response ranking. This question has not been well studied, despite the potential benefits for the development of information-seeking conversation systems.

To address these research issues, we propose a learning framework on top of deep neural matching networks that leverages external knowledge for response ranking

in information-seeking conversation systems. We study two different methods on integrating external knowledge into deep neural matching networks as follows:

(1) Incorporating external knowledge via pseudo-relevance feedback. Pseudo-relevance feedback (PRF) has been proven effective in improving the performance of many retrieval models (Lavrenko and Croft, 2001; Lv and Zhai, 2009; Zamani et al., 2016; Zhai and Lafferty, 2001; Rocchio, 1971; Cao et al., 2008; Diaz and Metzler, 2006). The motivation of PRF is to assume a certain number of top-ranked documents from the initial retrieval run to be relevant and use these feedback documents to improve the original query representation. For conversation response ranking, many candidate responses are much shorter compared with conversation context, which could have negative impacts on deep neural matching models. Inspired by the key idea of PRF, we propose using the candidate response as a query to run a retrieval round on a large external collection. Then we extract useful information from the (pseudo) relevant feedback documents to enrich the original candidate response representation.

(2) Incorporating external knowledge via QA correspondence knowledge distillation. Previous neural ranking models enhanced the performance of retrieval models such as BM25 and QL, which mainly rely on lexical match information, via modeling semantic match patterns in text (Guo et al., 2016; Huang et al., 2013; Mitra et al., 2017). For response ranking in information-seeking conversations, the match patterns between candidate responses and conversation context can be quite different from the well studied lexical and semantic matching. Consider the following sample utterance and response from the conversations in the Microsoft Answers community [1] shown in Table 4.1. A Windows user proposed a question about the windows update failure on "restart install". An expert replied with a response pointing to a potential

[1] answers.microsoft.com

Table 4.1: Sample utterance and response from the conversations in the Microsoft Answers community. This figure could be more readable with color print. Note that the purpose of this figure is to illustrate examples and differences among these three types of matches instead of exhaustively labeling all three types of matches between the two texts.

QA Dialog Title: : **Windows Update Failure** *Dialog Tags*: **Windows, Windows 10, Windows update, recovery, backup, PC** USER: I have Windows10, version 1511, OS Build 10586.1106. For the past year I have tried to upgrade from this without success. Upgrade download OK but on installing only get to 85 - 93% and then on restart install previous version of windows (the 1511 version), I have Windows update assistant installed. Any help or advice on this would be most welcome. David
Responses
AGENT: James (Microsoft MVP - Windows Client) : **Response**:There's not a doubt in my mind that those Norton "leftovers" is your troublemaker here - but now that the Norton Removal Tool has been deprecated and especially since the new-fangled Norton Remove and Reinstall tool doesn't get rid of the leftovers, a manual upgrade or a clean install of Microsoft Win10 appears to be your only possible resolution here. Feel free to give Norton/Symantec a piece of your mind!
Term Match: Magenta Semantic Match: Blue Correspondence Match: Red

cause "Norton leftovers". The match signals between the problem "restart install" and the cause "Norton leftovers" may not be captured by simple lexical and semantic matching. To derive such match patterns, we need to rely on external knowledge to distill QA correspondence information. We propose to extract the "correspondence" regularities between question and answer terms from retrieved external QA pairs. We define this type of match patterns as a "*correspondence match*", which will be incorporated into deep matching networks as external knowledge to help response selection in information-seeking conversations.

We conduct extensive experiments with three information-seeking conversation data sets: the MSDialog data which contains crawled customer service dialogs from Microsoft Answers community , a popular benchmark data Ubuntu Dialog Corpus (UDC) (Lowe et al., 2015), and another commercial customer service data from a large E-commerce company. We compare our methods with various deep text matching models and the state-of-the-art baseline on response selection in multi-turn conversations. Our methods outperform all baseline methods regrading a variety of metrics.

To sum up, our contributions can be summarized as follows:

(1) Focusing on information-seeking conversations and building a new benchmark data set. We target information-seeking conversations to push the boundaries of conversational search models. To this end, we create a new information-seeking conversation data set MSDialog on technical support dialogs of Microsoft products and released it to the research community [2].

(2) Integrating external knowledge into deep neural matching networks for response ranking. We propose a new response ranking paradigm for multi-turn conversations by incorporating external knowledge into the matching process of dialog context and candidate responses. Under this paradigm, we design two different methods with pseudo relevance feedback and QA correspondence knowledge distillation to integrate external knowledge into deep neural matching networks for response ranking.

(3) Extensive experimental evaluation on benchmark / commercial data sets and promising results. Experimental results with three different information-seeking conversation data sets show that our methods outperform various baseline methods including the state-of-the-art method on response selection in multi-turn conversations. We also perform analysis over different response types, model variations and ranking examples to provide insights.

4.2 Deep Matching Networks with External Knowledge

4.2.1 Problem Formulation

The research problem of response ranking in information-seeking conversations is defined as follows. We are given an information-seeking conversation data set $\mathcal{D} = \{(\mathcal{U}_i, \mathcal{R}_i, \mathcal{Y}_i)\}_{i=1}^{N}$, where $\mathcal{U}_i = \{u_i^1, u_i^2, \ldots, u_i^{t-1}, u_i^t\}$ in which $\{u_i^1, u_i^2, \ldots, u_i^{t-1}\}$

[2]The MSDialog dataset can be downloaded from `https://ciir.cs.umass.edu/downloads/msdialog`. We also released our source code at `https://github.com/yangliuy/NeuralResponseRanking` .

is the dialog context and u_i^t is the input utterance in the t-th turn. $\mathcal{R}_i$ and $\mathcal{Y}_i$ are a set of response candidates $\{r_i^1, r_i^2, \ldots, r_i^k\}_{k=1}^M$ and the corresponding binary labels $\{y_i^1, y_i^2, \ldots, y_i^k\}$, where $y_i^k = 1$ denotes r_i^k is a true response for $\mathcal{U}_i$. Otherwise $y_i^k = 0$. In order to integrate external knowledge, we are also given an external collection $\mathcal{E}$, which is related to the topics discussed in conversation $\mathcal{U}$. Our task is to learn a ranking model $f(\cdot)$ with $\mathcal{D}$ and $\mathcal{E}$. For any given $\mathcal{U}_i$, the model should be able to generate a ranking list for the candidate responses $\mathcal{R}_i$ with $f(\cdot)$. The external collection $\mathcal{E}$ could be any massive text corpus. In this chapter, $\mathcal{E}$ are historical QA posts in Stack Overflow data dump[3] for MSDialog, AskUbuntu data dump[4] for Ubuntu Dialog Corpus and product QA pairs for AliMe data.

4.2.2 Method Overview

In the following sections, we describe the proposed learning framework built on the top of deep matching networks and external knowledge for response ranking in information-seeking conversations. A summary of key notations in this work is presented in Table 4.2. In general, there are three modules in our learning framework:

(1) Information retrieval (IR) module: Given the information seeking conversation data $\mathcal{D}$ and external QA text collection $\mathcal{E}$, this module is to retrieve a small relevant set of QA pairs $\mathcal{P}$ from $\mathcal{E}$ with the response candidate $\mathcal{R}$ as the queries. These retrieved QA pairs $\mathcal{P}$ become the source of external knowledge.

(2) External knowledge extraction (KE) module: Given the retrieved QA pairs $\mathcal{P}$ from the IR module, this module will extract useful information as term distributions, term co-occurrence matrices or other forms as external knowledge.

(3) Deep matching network (DMN) module: This is the module to model the extracted external knowledge from $\mathcal{P}$, dialog utterances $\mathcal{U}_i$ and the response candidate

[3] https://stackoverflow.com/

[4] https://askubuntu.com/

Table 4.2: A summary of key notations in this work. Note that all vectors are denoted with bold cases.

$\mathcal{D}$	The conversation data set used for training/validation/testing
$\mathcal{E}$	The collection for the retrieval and distillation of external knowledge
u_i^t,$\mathcal{U}_i$,$\mathcal{U}$	The t-th utterance of the i-th dialog, all utterances of the i-th dialog and the set of all dialog utterances
r_i^k,$\mathcal{R}_i$,$\mathcal{R}$	The k-th response candidate for the i-th dialog, all response candidates of the i-th dialog and the set of all candidate responses
$r_i^{k'}$	The k-th expanded response candidate for the i-th dialog
y_i^k,$\mathcal{Y}$	The label for the k-th response candidate for the i-th dialog and the set of all labels
$f(\cdot)$	The ranking model learnt with $\mathcal{D}$ and $\mathcal{E}$
$f(\mathcal{U}_i, r_i^k)$	The predicted matching score between $\mathcal{U}_i$ and r_i^k
N	The total number of dialogs in $\mathcal{D}$
M	The total number of response candidates for $\mathcal{U}_i$
W	The number of expanded words in response candidates
θ	The language model constructed from the pseudo relevance feedback document set for response candidate expansion
P,$\mathcal{P}$	The number of top ranked QA posts retrieved from $\mathcal{E}$ and the top ranked QA post set
l_r, l_u	The length of a response candidate and the length of an utterance
d	The number of dimensions of word embedding vectors
$\mathbf{M}_1$,$\mathbf{M}_2$,$\mathbf{M}_3$	Interaction matrices between dialog utterance u_i^t and candidate response r_i^k or $r_i^{k'}$ for word embedding similarity, sequence hidden representation similarity and QA correspondence matching similarity
$m_{1,i,j}$	The (i,j)-th element in the interaction matrix $\mathbf{M}_1$
c	The window size for the utterances in dialog context, which is the maximal number of previous utterances modeled

r_i^k to learn the matching pattern, over which it will accumulate and predict a matching score $f(\mathcal{U}_i, r_i^k)$ for $\mathcal{U}_i$ and r_i^k.

We explore two different implementations under this learning framework as follows: 1) Incorporating external knowledge into deep matching networks via pseudo-relevance feedback (DMN-PRF). The architecture of DMN-PRF model is presented in Figure 4.1. 2) Incorporating external knowledge via QA correspondence knowledge distillation (DMN-KD). The architecture of DMN-KD model is presented in Figure 4.2. We will present the details of these two models in Section 4.2.3 and Section 4.2.4.

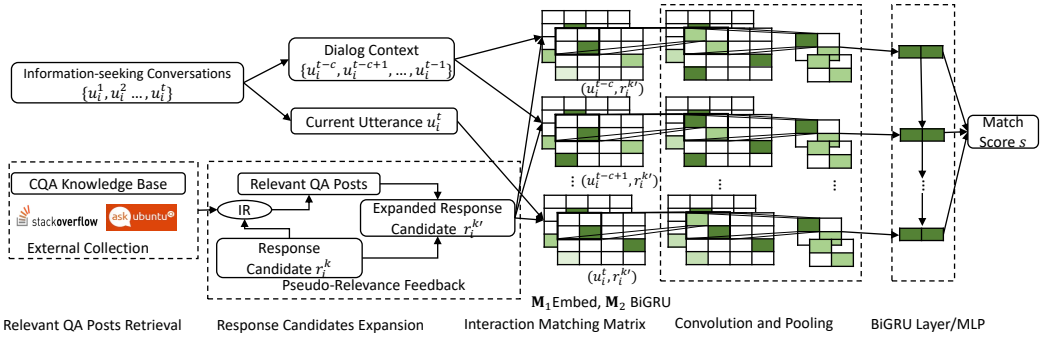


Figure 4.1: The architecture of DMN-PRF model for conversation response ranking.

4.2.3 Deep Matching Networks with Pseudo-Relevance Feedback

4.2.3.1 Relevant QA Posts Retrieval

We adopt different QA text collections for different conversation data (e.g. Stack Overflow data for MSDialog, AskUbuntu for UDC). The statistics of these external collections are shown in Table 4.3. We download the data dumps for Stack Overflow and AskUbuntu from archive.org[5]. We index the QA posts in Stack Overflow in most recent two years and all the QA posts in AskUbuntu. Then we use the response candidate r_i^k as the query to retrieve top P [6] QA posts with BM25 as the source for external knowledge.

4.2.3.2 Candidate Response Expansion

The motivation of Pseudo-Relevance Feedback (PRF) is to extract terms from the top-ranked documents in the first retrieval results to help discriminate relevant documents from irrelevant ones (Cao et al., 2008). The expansion terms are extracted either according to the term distributions (e.g. extract the most frequent terms) or extracted from the most specific terms (e.g. extract terms with the maximal IDF weights) in feedback documents. Given the retrieved top QA posts $\mathcal{P}$ from the previous step, we compute a language model $\theta = P(w|\mathcal{P})$ using $\mathcal{P}$. Then we extract

[5] `https://archive.org/download/stackexchange`

[6] In our experiments, we set $P = 10$.

Table 4.3: Statistics of external collections for QA pairs retrieval and knowledge extraction. Note that "#QWithAcceptedA" means "number of questions with an accepted answer". The other names use similar abbreviations.

Collection Name	SOTwoYears	AskUbuntu
StartDate	12/4/2015	7/28/2010
EndDate	9/1/2017	9/1/2017
#QAPosts	9,563,530	629,198
#Time	2 Years	7 years
XMLFileDiskSize	17GB	799MB
#Question	4,188,937	271,233
#QWithAcceptedA	1,751,787	92,259
#QWithAtLeastOneA	3,178,814	213,830
%QWithAcceptedA	41.82%	34.01%
%QWithAtLeastOneA	75.89%	78.84%

the most frequent W terms from θ as expansion terms for response candidate r_i^k. In our experiments, we set $W = 10$. For the query r_i^k, we perform several preprocessing steps including tokenization, punctuation removal and stop words removal. QA posts in both Stack Overflow and AskUbuntu have two fields: "Body" and "Title". We choose to search the "Body" field since we found it more effective in experiments.

4.2.3.3 Interaction Matching Matrix

The expanded response candidates and dialog contexts will be modeled by a deep neural matching network. Given an expanded response $r_i^{k'}$ and an utterance u_i^t in the context $\mathcal{U}_i$, the model first looks up a global embedding dictionary to represent $r_i^{k'}$ and u_i^t as two sequences of embedding vectors $\mathbf{E}(r_i^{k'}) = [\mathbf{e}_{r,1}, \mathbf{e}_{r,2}, \cdots, \mathbf{e}_{r,l_r}]$ and $\mathbf{E}(u_i^t) = [\mathbf{e}_{u,1}, \mathbf{e}_{u,2}, \cdots, \mathbf{e}_{u,l_u}]$, where $\mathbf{e}_{r,i} \in \mathbb{R}^d$, $\mathbf{e}_{u,i} \in \mathbb{R}^d$ are the embedding vectors of the i-th word in $r_i^{k'}$ and u_i^t respectively. Given these two word embedding sequences, there are two different methods to learn matching patterns: representation focused methods and interaction focused methods (Guo et al., 2016). Here we adopt the interaction focused methods due to their better performances over a number of text matching tasks (Hu et al., 2014; Pang et al., 2016; Wan et al., 2016; Yang et al., 2016a). Specifically, the model builds two interaction matrices with $\mathbf{E}(r_i^{k'}) \in \mathbb{R}^{d \times l_r}$

and $\mathbf{E}(u_i^t) \in \mathbb{R}^{d \times l_u}$: a word pairwise similarity matrix $\mathbf{M}_1$ and a sequence hidden representation similarity matrix $\mathbf{M}_2$. $\mathbf{M}_1$ and $\mathbf{M}_2$ will be two input channels of a convolutional neural network (CNN) to learn important matching features, which will be aggregated by the final BiGRU layer and a multi-layer perceptron (MLP) to generate a matching score.

Specifically, in the input channel one, $\forall i, j$, the element $m_{1,i,j}$ in the $\mathbf{M}_1$ is defined by $m_{1,i,j} = \mathbf{e}_{r,i}^T \cdot \mathbf{e}_{u,j}$. $\mathbf{M_1}$ models the word pairwise similarity between $r_i^{k'}$ and u_i^t via the dot product similarity between the embedding representations.

For input channel two, we firstly employ bidirectional gated recurrent units (BiGRU) (Chung et al., 2014) to encode $r_i^{k'}$ and u_i^t into two hidden representations. A BiGRU consists two GRUs that run in opposite directions on sequence $\mathbf{E}(r_i^{k'})$: a forward GRUs processing the sequence as it is ordered, and another backward GRUs processing the sequence in its reverse order. These two GRUs will generate two sequences of hidden states $(\vec{\mathbf{h}}_1, \cdots, \vec{\mathbf{h}}_{l_r})$ and $(\overleftarrow{\mathbf{h}}_1, \cdots, \overleftarrow{\mathbf{h}}_{l_r})$. BiGRU then concatenates the forward and the backward hidden states to form the final hidden vectors for $r_i^{k'}$ as $\mathbf{h}_i = [\vec{\mathbf{h}}_i, \overleftarrow{\mathbf{h}}_i]_{i=1}^{l_r}$. More specifically, $\forall i$, the hidden state vector $\vec{\mathbf{h}}_i \in \mathbb{R}^O$ is calculated by the following formulas:

$$
\begin{aligned}
\mathbf{z}_i &= \sigma(\mathbf{W}_z \mathbf{e}_{r,i} + \mathbf{U}_z \vec{\mathbf{h}}_{i-1} + \mathbf{b}_z) \\
\mathbf{r}_i &= \sigma(\mathbf{W}_r \mathbf{e}_{r,i} + \mathbf{U}_r \vec{\mathbf{h}}_{i-1} + \mathbf{b}_r) \\
\tilde{\mathbf{h}}_i &= \tanh(\mathbf{W}_h \mathbf{e}_{r,i} + \mathbf{U}_h (\mathbf{r}_i \circ \vec{\mathbf{h}}_{i-1}) + \mathbf{b}_h) \\
\vec{\mathbf{h}}_i &= (\mathbf{1} - \mathbf{z}_i) \circ \vec{\mathbf{h}}_{i-1} + \mathbf{z}_i \circ \tilde{\mathbf{h}}_i
\end{aligned}
\tag{4.1}
$$

where $\mathbf{z}_i$ and $\mathbf{r}_i$ are an update gate and a reset gate respectively. $\mathbf{e}_{r,i}, \vec{\mathbf{h}}_i$ are the input and hidden state output of the network at time step i. $\mathbf{W}_z, \mathbf{W}_r, \mathbf{W}_h, \mathbf{U}_z, \mathbf{U}_r, \mathbf{U}_h$ and $\mathbf{b}_z, \mathbf{b}_r, \mathbf{b}_h$ are parameter matrices and bias vectors to be learned. The backward hidden state $\overleftarrow{\mathbf{h}}_i \in \mathbb{R}^O$ is computed in a similar way according to Equation 4.1. The hidden vectors for the dialog utterance u_i^t can be obtained in the same procedure. Given the hidden vectors of $r_i^{k'}$ and u_i^t, we calculate element $m_{2,i,j}$ in the sequence hidden representation similarity matrix $\mathbf{M}_2$ by $m_{2,i,j} = \mathbf{h}_{r,i}^T \cdot \mathbf{h}_{u,j}$. BiGRU models

the neighbor context information around words from two directions and encode the text sequences into hidden vectors. Thus $\mathbf{M}_2$ matches $r_i^{k'}$ and u_i^t with local sequence structures such as phrases or text segments.

4.2.3.4 Convolution and Pooling Layers

The interaction matrices $\mathbf{M}_1$ and $\mathbf{M}_2$ are then fed into a CNN to learn high level matching patterns as features. CNN alternates convolution and max-pooling operations over these input channels. Let $\mathbf{z}^{(l,k)}$ denote the output feature map of the l-th layer and k-th kernel, the model will do convolution operations and max-pooling operations according to the following equations.

Convolution: let $r_w^{(l,k)} \times r_h^{(l,k)}$ denote the shape of the k-th convolution kernel in the l-th layer, the convolution operation can be defined as:

$$\mathbf{z}_{i,j}^{(l+1,k)} = \sigma(\sum_{k'=0}^{K_l-1} \sum_{s=0}^{r_w^{(l,k)}-1} \sum_{t=0}^{r_h^{(l,k)}-1} \mathbf{w}_{s,t}^{(l+1,k)} \cdot z_{i+s,j+t}^{(l,k')} + b^{(l+1,k)}) \quad \forall l = 0, 2, 4, 6, \cdots, \tag{4.2}$$

where σ is the activation function ReLU, and $\mathbf{w}_{s,t}^{(l+1,k)}$ and $b^{(l+1,k)}$ are the parameters of the k-th kernel on the $(l+1)$-th layer to be learned. K_l is the number of kernels on the l-th layer.

Max Pooling: let $p_w^{(l,k)} \times p_h^{(l,k)}$ denote the shape of the k-th pooling kernel in the l-th layer, the max pooling operation can be defined as:

$$\mathbf{z}_{i,j}^{(l+1,k)} = \max_{0 \le s < p_w^{l+1,k}} \max_{0 \le t < p_h^{l+1,k}} \mathbf{z}_{i+s,j+t}^{(l,k)} \quad \forall l = 1, 3, 5, 7, \cdots, \tag{4.3}$$

4.2.3.5 BiGRU Layer and MLP

Given the output feature representation vectors learned by CNN for utterance-response pairs $(r_i^{k'}, u_i^t)$, we add another BiGRU layer to model the dependency and temporal relationship of utterances in the conversation according to Equation 4.1 following the previous work (Wu et al., 2017). The output hidden states $\mathbf{H}_c$ =

$[\mathbf{h'}_1, \cdots, \mathbf{h'}_c]$ will be concatenated as a vector and fed into a multi-layer perceptron (MLP) to calculate the final matching score $f(\mathcal{U}_i, r_i^{k'})$ as

$$f(\mathcal{U}_i, r_i^{k'}) = \sigma_2(\mathbf{w}_2^T \cdot \sigma_1(\mathbf{w}_1^T \mathbf{H}_c + \mathbf{b}_1) + \mathbf{b}_2) \tag{4.4}$$

where $\mathbf{w}_1, \mathbf{w}_2, \mathbf{b}_1, \mathbf{b}_2$ are model parameters. σ_1 and σ_2 are tanh and softmax functions respectively.

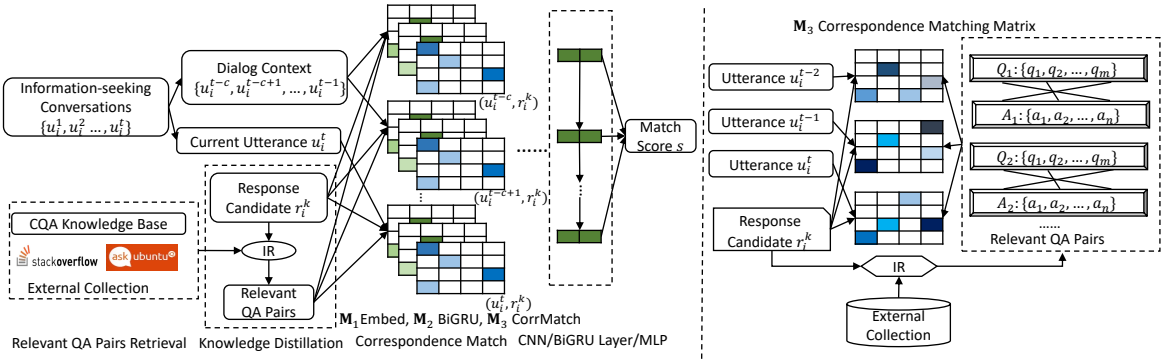


Figure 4.2: The left figure shows the architecture of DMN-KD model for conversation response ranking. The input channel $\mathbf{M}_3$ denoted as blue matrices capture the correspondence matching patterns of utterance terms and response terms in relevant external QA pairs retrieved from $\mathcal{E}$. Note that we omit the details for CNN layers here to save spaces as they have been visualized in Figure 4.1. The right figure shows the detailed pipeline of external relevant QA pairs retrieval and QA correspondence matching knowledge distillation in DMN-KD model.

4.2.3.6 Model Training

For model training, we consider a pairwise ranking learning setting. The training data consists of triples $(\mathcal{U}_i, r_i^{k+}, r_i^{k-})$ where r_i^{k+} and r_i^{k-} denote the positive and the negative response candidate for dialog utterances $\mathcal{U}_i$. Let Θ denote all the parameters of our model. The pairwise ranking-based hinge loss function is defined as:

$$\mathcal{L}(\mathcal{D}, \mathcal{E}; \Theta) = \sum_{i=1}^{I} \max(0, \epsilon - f(\mathcal{U}_i, r_i^{k+}) + f(\mathcal{U}_i, r_i^{k-})) + \lambda ||\Theta||_2^2 \tag{4.5}$$

where I is the total number of triples in the training data $\mathcal{D}$. $\lambda||\Theta||_2^2$ is the regularization term where λ denotes the regularization coefficient. ϵ denotes the

margin in the hinge loss. The parameters of the deep matching network are optimized using back-propagation with *Adam* algorithm (Kingma and Ba, 2014). For neural network regularization, we employ Dropout(Srivastava et al., 2014) in the model training process.

4.2.4 Deep Matching Networks with QA Correspondence Knowledge Distillation

In addition to the DMN-PRF model presented in Section 4.2.3, we also propose another model for incorporating external knowledge into conversation response ranking via QA correspondence knowledge distillation, which is referred to as DMN-KD model in this chapter. The architecture of DMN-KD model is presented in Figure 4.2. Compared with DMN-PRF, the main difference is that the CNN of DMN-KD will run on an additional input channel $\mathbf{M}_3$ denoted as blue matrices in Figure 4.2, which captures the correspondence matching patterns of utterance terms and response terms in relevant external QA pairs retrieved from $\mathcal{E}$. Specifically, we firstly use the response candidate r_i^k as the query to retrieve a set of relevant QA pairs[7] $\mathcal{P}$. Suppose $\mathcal{P} = \{\mathcal{Q}, \mathcal{A}\} = \{(\mathbf{Q}_1, \mathbf{A}_1), (\mathbf{Q}_2, \mathbf{A}_2), \cdots, (\mathbf{Q}_P, \mathbf{A}_P)\}$, where $(\mathbf{Q}_p, \mathbf{A}_p)$ denotes the p-th QA pair. Given a response candidate r_i^k and a dialog utterance u_i^t in dialog $\mathcal{U}_i$, the model will compute the term co-occurrence information as the *Positive Pointwise Mutual Information* (PPMI) of words of r_i^k and u_i^t in retrieved QA pair set $\{\mathcal{Q}, \mathcal{A}\}$. Let $[w_{r,1}, w_{r,2}, \cdots, w_{r,l_r}]$ and $[w_{u,1}, w_{u,2}, \cdots, w_{u,l_u}]$ denote the word sequence in r_i^k and u_i^t. We construct a QA term correspondence matching matrix $\mathbf{M}_3$ as the third input channel of CNN for r_i^k and u_i^t with the PPMI statistics from $\{\mathcal{Q}, \mathcal{A}\}$. More specifically, $\forall i, j$, the element $m_{3,i,j}$ in $\mathbf{M}_3$ is computed as

[7]Note that we want QA pairs here instead of question posts or answer posts, since we would like to extract QA term co-occurrence information with these QA pairs.

$$
\begin{aligned}
m_{3,i,j} &= PPMI(w_{r,i}, w_{u,j}|\{\mathcal{Q}, \mathcal{A}\}) \\
&= \max(0, \log \frac{\sum_{p'=1}^{P} p(w_{r,i} \in \mathbf{A}_{p'}, w_{u,j} \in \mathbf{Q}_{p'}|\mathbf{Q}_{p'}, \mathbf{A}_{p'})}{p(w_{r,i}|\mathcal{A}) \cdot p(w_{u,j}|\mathcal{Q})})
\end{aligned} \tag{4.6}
$$

where $w_{r,i}$ and $w_{u,j}$ denote the i-th word in the response candidate and j-th word in the dialog utterance. The intuition is that the PPMI between $w_{r,i}$ and $w_{u,j}$ in the top retrieved relevant QA pair set $\{\mathcal{Q}, \mathcal{A}\}$ could encode the correspondence matching patterns between $w_{r,i}$ and $w_{u,j}$ in external relevant QA pairs . Thus $\mathbf{M}_3$ is the extracted QA correspondence knowledge from the external collection $\mathcal{E}$ for r_i^k and u_i^t. These correspondence matching knowledge capture relationships such as *"(Problem Descriptions, Solutions)", "(Symptoms, Causes)", "(Information Request, Answers)"*, etc. in the top ranked relevant QA pair set $\{\mathcal{Q}, \mathcal{A}\}$. They will help the model better discriminate a good response candidate from a bad response candidate given the dialog context utterances. To compute the co-occurrence count between $w_{r,i}$ and $w_{u,j}$, we count all word co-occurrences considering $\mathbf{A}_p$ and $\mathbf{Q}_p$ as bag-of-words as we found this setting is more effective in experiments.

4.3 Experiments

4.3.1 Data Set Description

We evaluated our method with three data sets: Ubuntu Dialog Corpus (UDC), MSDialog, and AliMe data consisting of a set of customer service conversations in Chinese from Alibaba.

4.3.1.1 Ubuntu Dialog Corpus

The Ubuntu Dialog Corpus (UDC) (Lowe et al., 2015) contains multi-turn technical support conversation data collected from the chat logs of the Freenode Internet Relay Chat (IRC) network. We used the data copy shared by Xu et al. (2016), in which numbers, urls and paths are replaced by special placeholders. It is also used

in several previous related works (Wu et al., 2017)[8]. It consists of 1 million context-response pairs for training, 0.5 million pairs for validation and 0.5 million pairs for testing. The statistics of this data are shown in Table 4.4. The positive response candidates in this data come from the true responses by human and negative response candidates are randomly sampled.

4.3.1.2 MSDialog

In addition to UDC, we also crawled another technical support conversation data from the Microsoft Answer community, which is a QA forum on topics about a variety of Microsoft products. We firstly crawled 35,536 dialogs about 76 different categories of Microsoft products including "Windows", "IE", "Office", "Skype", "Surface", "Xbox", etc. [9] Then we filtered dialogs whose number of turns are out of the range [3, 99]. After that we split the data into training/validation/testing partitions by time. Specifically, the training data contains 25,019 dialogs from "2005-11-12" to "2017-08-20". The validation data contains 4,654 dialogs from "2017-08-21" to "2017-09-20". The testing data contains 5,064 dialogs from "2017-09-21" to "2017-10-04".

The next step is to generate the dialog context and response candidates. For each dialog, we assigned "User" label to the first participant who proposed the question leading to this information-seeking conversation, and "Agent" label to the other participants who provided responses. The "Agent" in our data could be Microsoft customer service staff, a Microsoft MVP (Most Valuable Professional) or a user from the Microsoft Answer community. Then for each utterance by the "User" u_i^t [10], we

[8]The data can be downloaded from `https://www.dropbox.com/s/2fdn26rj6h9bpvl/ubuntu%20data.zip?dl=0`

[9]Note that some categories are more fine-grained, such as"Outlook_Calendar", "Outlook_Contacts", "Outlook_Email", "Outlook_Messaging", etc.

[10]We consider the utterances by the user except the first utterance, since there is no associated dialog context with it.

collected the previous c utterances as the dialog context, where $c = \min(t-1, 10)$ and $t-1$ is the total number of utterances before u_i^t. The true response by the "Agent" becomes the positive response candidate. For the negative response candidates, we adopted negative sampling to construct them following previous work (Wan et al., 2016; Lowe et al., 2015; Wu et al., 2017). For each dialog context, we firstly used the true response as the query to retrieve the top $1,000$ results from the whole response set of agents with BM25. Then we randomly sampled 9 responses from them to construct the negative response candidates. For data preprocessing, we performed tokenization and punctuation removal. Then we removed stop words and performed word stemming. For neural models, we also removed words that appear less than 5 times in the whole corpus.

4.3.1.3 AliMe Data

We collected the chat logs between customers and a chatbot AliMe from "2017-10-01" to "2017-10-20" in Alibaba. The chatbot is built based on a question-to-question matching system [11] (Li et al., 2017), where for each query, it finds the most similar candidate question in a QA database and return its answer as the reply. It indexes all the questions in our QA database using Lucence[12]. For each given query, it uses TF-IDF ranking algorithm to call back candidates. To form our data set, we concatenated utterances within three turns [13] to form a query, and used the chatbot system to call back top-K [14] most similar candidate questions as candidate "responses". [15] We then asked a business analyst to annotate the candidate responses, where a "response"

[11] Interested readers can access AliMe Assist through the Taobao App, or the web version via `https://consumerservice.taobao.com/online-help`

[12] `https://lucene.apache.org/core/`

[13] The majority (around 85%) of conversations in the data set are within 3 turns.

[14] We set K=15.

[15] A "response" here is a question in our system.

Table 4.4: The statistics of data sets used in experiments.

Data	UDC			MSDialog			AliMe		
Items	Train	Valid	Test	Train	Valid	Test	Train	Valid	Test
# Context-response pairs	1000K	500K	500K	173K	37K	35K	51K	6K	6K
# Candidates per context	2	10	10	10	10	10	15	15	15
# Positive candidates per context	1	1	1	1	1	1	2.9	2.8	2.9
Min # turns per context	1	2	1	2	2	2	2	2	2
Max # turns per context	19	19	19	11	11	11	3	3	3
Avg # turns per context	10.1	10.1	10.1	5.0	4.9	4.4	2.4	2.1	2.2
Avg # words per context	116.0	115.6	115.9	271	263	227	38.3	35.3	34.2
Avg # words per utterance	22.1	22.1	22.1	66.7	67.6	66.8	4.9	4.7	4.6

is labeled as positive if it matches the query, otherwise negative. In all, we have annotated 63,000 context-response pairs, where we use 51,000 as training, 6,000 for testing, and 6,000 for validation shown in Table 4.4. Note that we have included human evaluation in AliMe data. Furthermore, if the confidence score of answering a given user query is low, the system will prompt three top related questions for users to choose. We collected such user click logs as our external data, where we treat the clicked question as positive and the others as negative. We collected 510,000 clicked questions with answers from the click logs in total as the source of external knowledge.

4.3.2 Experimental Setup

4.3.2.1 Baselines

We consider different types of baselines for comparison, including traditional retrieval models, deep text matching models and the state-of-the-art multi-turn conversation response ranking method as the following:

- BM25: this method uses the dialog context as the query to retrieve response candidates for response selection. We consider BM25 model (Robertson and Walker, 1994) as the retrieval model.

- ARC-II: ARC-II is an interaction focused deep text matching architectures proposed by Hu et al. (2014), which is built directly on the interaction matrix between the dialog context and response candidates. A CNN is running on the interaction matrix to learn the matching representation score.

- MV-LSTM: MV-LSTM (Wan et al., 2016) is a neural text matching model that matches two sequences with multiple positional representations learned by a Bi-LSTM layer.

- DRMM: DRMM (Guo et al., 2016) is a deep relevance matching model for ad-hoc retrieval. We implemented a variant of DRMM for short text matching. Specifically, the matching histogram is replaced by a top-k max pooling layer and the remaining part is the same with the original model.

- Duet: Duet (Mitra et al., 2017) is the state-of-the-art deep text matching model that jointly learns local lexical matching and global semantic matching between the two text sequences.

- SMN: Sequential Matching Network (SMN) (Wu et al., 2017) is the state-of-the-art deep neural architecture for multi-turn conversation response selection. It matches a response candidate with each utterance in the context on multiple levels of granularity and then adopts a CNN network to distill matching features. We used the TensorFlow [16] implementation of SMN shared by authors (Wu et al., 2017) [17].

We also consider a degenerated version of our model, denoted as **DMN**, where we do not incorporate external knowledge via pseudo-relevance feedback or QA correspondence knowledge distillation. Finally, we consider a baseline **BM25-PRF**, where we incorporate external knowledge into BM25 by matching conversation context with the expanded responses as in Section 4.2.3.2 using BM25 model.

---

[16] `https://www.tensorflow.org/`

[17] The reported SMN results with the code from authors are on the raw data sets of UDC and MSDialog without any over sampling of negative training data.

4.3.2.2 Evaluation Methodology

For the evaluation metrics, we adopted mean average precision (MAP), Recall@1, Recall@2, and Recall@5 following previous related work (Wu et al., 2017; Lowe et al., 2015). For UDC and MSDialog, MAP is equivalent to the mean reciprocal rank (MRR) since there is only one positive response candidate per dialog context. For AliMe data, each dialog context could have more than one positive response candidates.

4.3.2.3 Parameter Settings

All models were implemented with TensorFlow and MatchZoo[18] toolkit. Hyper-parameters are tuned with the validation data. For the hyper-parameter settings of DMN-KD and DMN-PRF models, we set the window size of the convolution and pooling kernels as $(3, 3)$. The number of convolution kernels is 8 for UDC and 2 for MSDialog. The dimension of the hidden states of BiGRU layer is set as 200 for UDC and 100 for MSDialog . The dropout rate is set as 0.3 for UDC and 0.6 for MSDialog . All models are trained on a single Nvidia Titan X GPU by stochastic gradient descent with Adam(Kingma and Ba, 2014) algorithm. The initial learning rate is 0.001. The parameters of Adam, β_1 and β_2 are 0.9 and 0.999 respectively. The batch size is 200 for UDC and 50 for MSDialog. The maximum utterance length is 50 for UDC and 90 for MSDialog. The maximum conversation context length is set as 10 following previous work (Wu et al., 2017). We padded zeros if the number of utterances in a context is less than 10. Otherwise the most recent 10 utterances will be kept. For DMN-PRF, we retrieved top 10 QA posts and extracted 10 terms as response expansion terms. For DMN-KD, we retrieved top 10 question posts with accepted answers. For the word embeddings used in our experiments, we trained word embeddings with the Word2Vec tool with the Skip-gram model using our training

[18] `https://github.com/faneshion/MatchZoo`

Table 4.5: Comparison of different models over Ubuntu Dialog Corpus (UDC) and MSDialog data sets. Numbers in bold font mean the result is better compared with the best baseline. ‡ means statistically significant difference over the best baseline with $p < 0.05$ measured by the Student's paired t-test.

Data	UDC				MSDialog			
Methods	MAP	Recall@5	Recall@1	Recall@2	MAP	Recall@5	Recall@1	Recall@2
BM25	0.6504	0.8206	0.5138	0.6439	0.4387	0.6329	0.2626	0.3933
BM25-PRF	0.6620	0.8292	0.5289	0.6554	0.4419	0.6423	0.2652	0.3970
ARC-II	0.6855	0.8978	0.5350	0.6959	0.5398	0.8662	0.3189	0.5413
MV-LSTM	0.6611	0.8936	0.4973	0.6733	0.5059	0.8516	0.2768	0.5000
DRMM	0.6749	0.8776	0.5287	0.6773	0.5704	0.9003	0.3507	0.5854
Duet	0.5692	0.8272	0.4756	0.5592	0.5158	0.8481	0.2934	0.5046
SMN	0.7327	0.9273	0.5948	0.7523	0.6188	0.8374	0.4529	0.6195
DMN	0.7363	0.9196	0.6056	0.7509	0.6415	0.9155	0.4521	0.6673
DMN-KD	**0.7655**‡	**0.9351**‡	**0.6443**‡	**0.7841**‡	**0.6728**‡	**0.9304**‡	**0.4908**‡	**0.7089**‡
DMN-PRF	**0.7719**‡	**0.9343**‡	**0.6552**‡	**0.7893**‡	**0.6792**‡	**0.9356**‡	**0.5021**‡	**0.7122**‡

data. The max skip length between words and the number of negative examples is set as 5 and 10 respectively. The dimension of word vectors is 200. Word embeddings will be initialized by these pre-trained word vectors and updated during the training process.

4.3.3 Evaluation Results

4.3.3.1 Performance Comparison on UDC and MSDialog

We present evaluation results over different methods on UDC and MSDialog in Table 4.5. We summarize our observations as follows: (1) DMN-PRF model outperforms all the baseline methods including traditional retrieval models, deep text matching models and the state-of-the-art SMN model for response ranking on both conversation datasets. The results demonstrate that candidate response expansion with pseudo-relevance feedback could improve the ranking performance of responses in conversations. The main difference between DMN-PRF model and SMN model is the information extracted from retrieved feedback QA posts as external knowledge. This indicates the importance of modeling external knowledge with pseudo-relevant feedback beyond the dialog context for response selection. (2) DMN-KD model also outperforms all the baseline methods on MSDialog and UDC. These results show

Table 4.6: Comparison of different models over the AliMe data. Numbers in bold font mean the result is better compared with the best baseline. ‡ means statistically significant difference over the best baseline with $p < 0.01$ measured by the Student's paired t-test.

Data	AliMe			
Methods	MAP	Recall@5	Recall@2	Recall@1
BM25	0.6392	0.6407	0.4204	0.2371
BM25-PRF	0.6412	0.6510	0.4209	0.2545
ARC-II	0.7306	0.6595	0.3671	0.2236
MV-LSTM	0.7734	0.7017	0.4105	0.2480
DRMM	0.7165	0.6575	0.3616	0.2212
Duet	0.7651	0.6870	0.4088	0.2433
SMN	0.8145	0.7271	0.4680	0.2881
DMN	0.7833	0.7629	0.5012	0.3568
DMN-KD	**0.8323**	**0.7631**	**0.5122**‡	**0.3596**‡
DMN-PRF	**0.8435**‡	**0.7701**‡	**0.5323**‡	**0.3601**‡

that the extracted QA correspondence matching knowledge could help the model select better responses. Comparing DMN-KD and DMN-PRF, their performances are very close. (3) If we compare the performances of DMN-PRF, DMN-KD with the degenerated model DMN, we can see that incorporating external knowledge via both pseudo-relevance feedback and QA correspondence knowledge distillation could improve the performance of the deep neural networks for response ranking with large margins. For example, the improvement of DMN-PRF against DMN on UDC is 4.83% for MAP, 1.60% for Recall@5, 8.19% for Recall@1, 5.11% for Recall@2 respectively. The differences are statistically significant with $p < 0.05$ measured by the Student's paired t-test.

4.3.3.2 Performance Comparison on AliMe Data

We further compare our models with the competing methods on the AliMe data in Table 4.6. We find that: (1) our DMN model has comparable results in terms of MAP when compared with SMN, but has better Recall; (2) DMN-KD shows comparable or better results than all the baseline methods; (3) DMN-PRF significantly outperforms other competing baselines which shows the effectiveness of adding external pseudo-

relevance feedback to the task; (4) both DMN-PRF and DMN-KD show better results than DMN, which demonstrates the importance of incorporating external knowledge via both pseudo-relevance feedback and QA correspondence knowledge distillation.

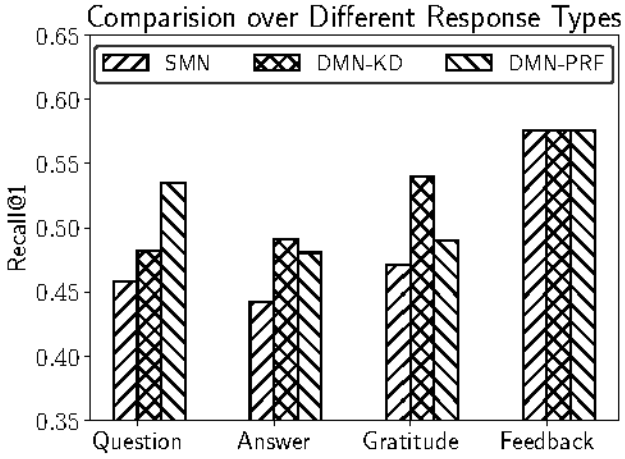


Figure 4.3: Performance comparison over different response types on MSDialog data.

4.3.3.3 Performance Comparison over Different Response Types

We conduct fine-grained analysis on the performance of different models on different response types. We annotated the user intents in 10,020 MSDialog utterances using Amazon Mechanical Turk [19]. We defined 12 user intent types including several types related to "questions" (original question, follow-up question, information request, clarifying question, and etc.), "answers" (potential answer and further details), "gratitude" (expressing thanks, greetings) and "feedback" (positive feedback and negative feedback). Then we trained a Random Forest classifier with TF-IDF features and applied this classifier to predict the response candidate types in the testing data of MSDialog. The dialog contexts were grouped by the type of the true response candidate. Finally we computed the average Recall@1 over different groups. Figure 4.3 shows the results. We find that both DMN-KD and DMN-PRF

[19] https://www.mturk.com/

improve the performances of SMN for responses with type "questions", "answers" and "gratitude". This indicates that incorporating external knowledge with PRF or QA correspondence knowledge distillation can help the model select better responses, especially for QA related responses. For responses with type "Feedback", DMN-KD and DMN-PRF achieved similar performances comparing with SMN.

4.3.4 Model Ablation Analysis

We investigate the effectiveness of different components of DMN-PRF and DMN-KD by removing them one by one from the original model with UDC and MSDialog data. We also study the effectiveness of different interaction types for **M1**/**M2**/**M3**. Table 4.7 shows the results. We summarize our observations as follows: 1) For the interaction matrices, we find that the performance will drop if we remove any one of **M1**/**M2** for DMN-PRF or **M1**/**M2**/**M3** for DMN-KD. This indicates that all of word level interaction matching, sequence level interaction matching and external QA correspondence interaction matching are useful for response selection in information-seeking conversation. 2) For interaction types, we can find that dot product is the best setting on both UDC and MSDialog except the results of DMN-KD on MSDialog. The next best one is cosine similarity. Bilinear product is the worst, especially on MSDialog data. This is because bilinear product will introduce a transformation matrix $\mathbf{A}$ as an additional model parameter, leading to higher model complexity. Thus the model is more likely to overfit the training data, especially for the relatively small MSDialog data. 3) If we only leave one channel in the interaction matrices, we can find that **M1** is more powerful than **M2** for DMN-PRF. For DMN-KD, **M1** is also the best one, followed by **M2**. **M3** is the last one, but it stills adds additional matching signals when it is combined with **M1** and **M2**. The matching signals **M3** from external collection could be supplementary features to the word embedding based matching matrix **M1** and BiGRU representation based matching matrix **M2**.

Table 4.7: Evaluation results of model ablation. "TB4.5" means the setting is the same with the results in Table 4.5. For DMN-KD, the model is the same with DMN if we remove M3. Numbers in bold font mean the result is better compared with other settings.

	Data	UDC				MSDialog			
Model	Change	MAP	Recall@5	Recall@1	Recall@2	MAP	Recall@5	Recall@1	Recall@2
DMN-PRF	Only M1	0.7599	0.9294	0.6385	0.7761	0.5632	0.8509	0.3654	0.5579
	Only M2	0.7253	0.9271	0.5836	0.7440	0.4996	0.8584	0.2595	0.5021
	Inter-Dot (TB4.5)	**0.7719**	**0.9343**	**0.6552**	**0.7893**	**0.6792**	**0.9356**	**0.5021**	**0.7122**
	Inter-Cosine	0.7507	0.9260	0.6248	0.7675	0.6729	0.9356	0.4944	0.7027
	Inter-Bilinear	0.7228	0.9199	0.5829	0.7401	0.4923	0.8421	0.2647	0.4744
DMN-KD	Only M1	0.7449	0.9247	0.6167	0.7612	0.5776	0.8673	0.3805	0.5779
	Only M2	0.7052	0.9203	0.5538	0.7260	0.5100	0.8613	0.2794	0.5011
	Only M3	0.3887	0.6017	0.2015	0.3268	0.3699	0.6650	0.1585	0.2957
	M1+M2 (DMN)	0.7363	0.9196	0.6056	0.7509	0.6415	0.9155	0.4521	0.6673
	M1+M3	0.7442	0.9251	0.6149	0.7612	0.6134	0.8860	0.4224	0.6266
	M2+M3	0.7077	0.9198	0.5586	0.7263	0.5141	0.8659	0.2885	0.5069
	Inter-Dot (TB4.5)	**0.7655**	**0.9351**	**0.6443**	**0.7841**	0.6728	**0.9304**	0.4908	0.7089
	Inter-Cosine	0.7156	0.9121	0.5770	0.7268	**0.6916**	0.9249	**0.5241**	**0.7249**
	Inter-Bilinear	0.7061	0.9135	0.5590	0.7225	0.4936	0.8224	0.2679	0.4814

4.3.5 Impact of Conversation Context Length

We further analyze the impact of the conversation context length on the performance of our proposed DMN-KD and DMN-PRF models. As presented in Figure 4.4, we find the performance first increases and then decreases, with the increase of conversation context length. The reason for these trends is that the context length controls the available previous utterances in the dialog context modeled by DMN-KD and DMN-PRF. If the context length is too small, there would be not enough information for the model to learn the matching patterns between the context and response candidates. However, setting the context length too large will also bring noise into the model results, since the words in utterances a few turns ago could be very different due to the topic changes during conversations.

4.3.6 Case Study

We perform a case study in Table 4.8 on the top ranked responses by different methods including SMN, DMN-KD and DMN-PRF. In this example, both DMN-KD and DMN-PRF produced correct top ranked responses. We checked the retrieved QA posts by the correct response candidate and found that "*settings, regional, change, windows, separator, format, excel, panel, application*" are the most frequent terms.

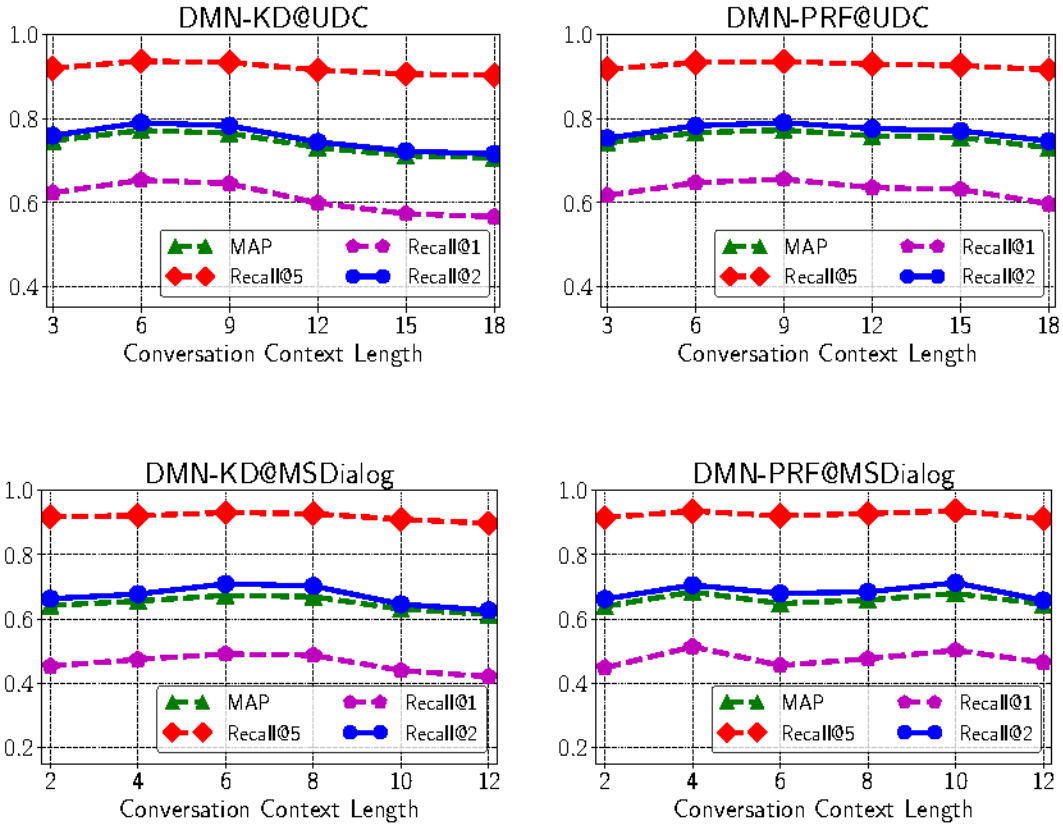


Figure 4.4: Performance of DMN-KD and DMN-PRF with different choices of context length over UDC and MSDialog data.

Table 4.8: Examples of Top-1 ranked responses by different methods. y_i^k means the label of a response candidate.

Context	[User] I open Excel and it automatically formats my dates into American formatting. I have changed and saved the formatting to NZ style. However everytime I pull the document out of office 365 it reverts back to the American format. How do I stop this? [Agent] Is it one file or all files in Excel? [User] It does seem to be all Excel files. How do I change the global date format setting?	
Method	y_i^k	Top-1 Ranked Response
SMN	0	Go to Settings ->System ->Tablet Mode....Change setting as indicated in the snapshot below.
DMN-KD	1	That is a Windows setting. Go to Control Panel >Regional settings. This will change date settings for all applications.
DMN-PRF	1	That is a Windows setting. Go to Control Panel >Regional settings. This will change date settings for all applications.

Among them "*excel*" is especially useful for promoting the rank of the correct response candidate, since this term which is included multiple times by the dialog context does not actually appear in the raw text of the correct response candidate. This gives an example of the effectiveness of incorporating external knowledge from the retrieved QA posts into response candidates.

4.4 Summary

In this chapter, we propose a learning framework on top of deep matching networks that leverages external knowledge for response ranking in information-seeking conversation systems. We incorporate external knowledge into deep neural models with pseudo-relevance feedback and QA correspondence knowledge distillation. Extensive experiments with information-seeking conversation data sets including both open benchmarks and commercial data show that our methods outperform various baselines including the state-of-the-art method on response selection in multi-turn conversations. We also perform analysis over different response types and model variations to provide insights on model applications.

CHAPTER 5

USER INTENT IN INFORMATION-SEEKING CONVERSATIONS

5.1 Introduction

In this Chapter, we study response retrieval in information-seeking conversations from a different perspective compared with Chapter 4. Significant progress has been made on the integration of conversation context by generating reformulated queries with contexts (Yan et al., 2016a), learning with both word sequence view and utterance sequence view (Zhou et al., 2016), enhancing context-response matching with sequential interactions between context utterances and response candidates (Wu et al., 2017), learning with external knowledge (Yang et al., 2018). However, much less attention is paid on the user intent in conversations and how to leverage user intent for response ranking in information-seeking conversations.

To illustrate user intent in information-seeking conversations, we show an example dialog from the Microsoft Answers Community[1] in Table 5.1. Microsoft Answers Community is a customer support QA forum where users can ask questions relevant to Microsoft products. Agents like Microsoft employees or other experienced users will reply to these questions. There could be multi-turn conversation interactions between users and agents. We define a taxonomy of user intent following previous research (Qu et al., 2018, 2019). We can observe that there are diverse user intent like "Original Question (OQ)", "Information Request (IR)", "Potential Answers (PA)", "Follow-up Questions (FQ)", "Further Details (FD)", etc. in an information-seeking

[1] https://answers.microsoft.com

conversation. Moreover, several transition patterns can happen between different user intent. For example, given a question from the user, an agent could provide a potential answer directly or ask for some information as clarification questions before providing answers. Users will provide further details regarding the information requests from agents. At the beginning of a conversation, the agent would like to greet customers or express gratitude to users before they move on to next steps. Near the end of a conversation, the user may provide a positive or negative feedback towards answers from agents, or ask a follow-up question to continue the conversation interactions.

Such user intent patterns can be helpful for conversation models to select good responses due to the following reasons:

(1) The intent sequence in conversation context utterances can provide additional signals to promote correct responses. Intent sequence patterns can promote response candidates with correct intent and demote response candidates with wrong intent. It can help prevent conversation models from producing responses with wrong intent. For example, in Table 5.1, given the intent sequence **[OQ]** $\rightarrow$ **[IR/ PA]** $\rightarrow$ **[PA/ FQ]** $\rightarrow$ **[FD]**, we know that the user is still expecting an answer to solve her question. Although both Response-1 and Response-2 show some lexical and semantic similarities with context utterances, only Response-1 is with the intent "Potential Answers" (PA). In this case, the model should have the capability to promote the rank of Response-1 and demote Response-2 with wrong intent "Greetings/ Gratitude" (GG).

(2) Intent information can help the model to derive an importance weighting scheme over context utterances with attention mechanisms. In the given example dialog in Table 5.1, the model should learn to assign larger weights to utterances on question descriptions (OQ and FQ) and further details (FD) in order to address the information need of the user. In other cases, the model may also assign larger attention weights on utterances with intent related to questions/ answers instead

Table 5.1: An example dialog to illustrate user intent transition patterns in information-seeking conversations from the Microsoft Answers Community. We define different user intent types following previous research (Qu et al., 2018, 2019). We show a conversation context with 4 utterances and two response candidates where there is one correct candidate and one wrong candidate. The user intent of utterances and response candidates are labeled. "OQ", "IR", "PA", "FQ", "FD", "GG" denote "Original Question", "Information Request", "Potential Answer", "Follow-up Question", "Further Details", "Greetings/ Gratitude" respectively. We also highlight some lexical match between utterances and response candidates using colorful underlines. This table can be more readable with color print.

ID	Role	Utterances	Intent
Utterance-1	User	Windows downloaded this update "2018-02 Cumulative Update for Windows 10 ....." But during the restart it says "we couldn't complete the update, undoing changes". So what can I do to stop this? Thanks	**OQ**
Utterance-2	Agent	Is there any other pending updates? Try Download troubleshooter for Win 10.	**IR/ PA**
Utterance-3	User	Yes, pending updates the same one. I already used the built in troubleshooter, it did fix some 3 issues, but doing a restart the problem persists. Can I stop updates from installing this particular one? Thanks.	**PA/ FQ**
Utterance-4	User	Not sure if related but I just saw that Malicious Software Removal of March did not install	**FD**
Response-1 (***Correct***)	Agent	Try run troubleshooter and then restart your PC. If problem persist, open start and search for Feedback and open Feedback Hub app and report this issue.	**PA**
Response-2 (***Wrong***)	Agent	Glad to know that you fixed the issue, and as I said downloading the "Show or hide updates" troubleshooter and restarting the PC will help you. Thank you for asking questions and providing feedback here!	**GG**

of greetings/ gratitude. Most existing neural conversation models do not explicitly model user intent to weight context utterances.

More research needs to be done to understand the role of user intent in response retrieval and to develop effective models for intent-aware response ranking in information-seeking conversations, which is exactly the goal of this chapter. There are some existing related works from the Dialog System Technology Challenge (formerly the Dialog State Tracking Challenge, DSTC)[2]. Many DSTC tasks focus on goal oriented conversations like restaurant reservation. These tasks are typically tackled with slot filling (Zhang and Wang, 2016; Hori et al., 2019), which is not applicable to information-seeking conversations because of the diversity of information needs. Recently in DSTC7 of 2018,[3] an end-to-end response selection challenge has been introduced, which shared similar motivation with our work. However, the evaluation treated response selection as a classification task and there was no explicit modeling of user intent in conversations.

In this chapter, we analyze user intent in information-seeking conversations and propose neural ranking models with the integration of user intent modeling. Different user intent types are defined and characterized following previous research (Qu et al., 2018, 2019). Then we propose an intent-aware neural ranking model for response retrieval, which is built on the top of the recent breakthroughs with natural language representation learning with the Transformers (Vaswani et al., 2017; Devlin et al., 2018). Transformers are model architectures which are based entirely on multi-head self-attention mechanisms instead of recurrent neural nets for modeling the global dependencies between the input and output, in order to speed up model training with parallel computing. They have achieved state-of-the-art results on several tasks like machine translation. We referred to the proposed model as

[2] https://www.microsoft.com/en-us/research/event/dialog-state-tracking-challenge/

[3] http://workshop.colips.org/dstc7/

"**IART**", which is "**I**ntent-**A**ware **R**anking with **T**ransformers". IART incorporates intent-aware utterance attention to derive the importance weighting scheme of utterances in conversation context towards better conversation history understanding. Given input conversation context utterances and response candidates, IART firstly generates representations from two different perspectives: user intent representations with a trained neural classifier and semantic information encoded with Transformers. Then self-attention matching and cross-attention matching will be performed over encoded representations from Transformers to extract important matching features, which will be weighted by the intent-aware attention mechanism and aggregated into a matching tensor. Finally a two-layer 3D convolutional neural network will distill features over the matching tensor to generate the final ranking score.

We conduct extensive experiments with three information-seeking conversation data sets: **MSDialog** (Qu et al., 2018) which contains crawled customer service dialogs on Microsoft products from Microsoft Answers community, a popular open benchmark data Ubuntu Dialog Corpus (**UDC**) (Lowe et al., 2015), and another commercial customer service data from a large eCommerce company (**AliMe**). We compare our methods with various neural ranking models and the state-of-the-art baselines on response selection in multi-turn conversations including Deep Attention Matching Network (DAM) (Zhou et al., 2018b). Experimental results show our methods outperform all baselines. We also perform visualization and deep analysis of learned user intent in information-seeking conversations to provide insights.

To sum up, our contributions can be summarized as follows:

(1) We analyze user intent in information-seeking conversations for intent-aware response ranking. To the best of our knowledge, our work is the first one to explicitly define and model user intent for response retrieval in information-seeking conversations.

(2) We propose an intent-aware response ranking model with Transformers (Vaswani et al., 2017): IART. IART derives the importance weighting scheme of utterances in conversation context with user intent signals for better conversation history modeling.

(3) Experimental results with three different information-seeking conversation data sets show that our methods outperform various baselines including the state-of-the-art method. We also perform analysis on learned user intent and ranking examples to provide insights.

5.2 Intent-aware Response Ranking

5.2.1 Problem Formulation

The research problem of response ranking in information-seeking conversations is defined as follows. We are given an information-seeking conversation data set $\mathcal{D} = \{(\mathcal{U}_i, \mathcal{R}_i, \mathcal{Y}_i)\}_{i=1}^{N}$, where $\mathcal{U}_i = \{u_i^1, u_i^2, \ldots, u_i^{t-1}, u_i^t\}$ in which u_i^t is the utterance in the t-th turn of the i-th dialog. $\mathcal{R}_i$ and $\mathcal{Y}_i$ are a set of response candidates $\{r_i^1, r_i^2, \ldots, r_i^k\}_{k=1}^{M}$ and the corresponding labels $\{y_i^1, y_i^2, \ldots, y_i^k\}$, where $y_i^k = 1$ denotes r_i^k is a true response for $\mathcal{U}_i$. Otherwise $y_i^k = 0$. For user intent information, there are sequence level user intent labels for both dialog context utterances and response candidates $\mathcal{E} = \{(\mathcal{I}_i^u, \mathcal{I}_i^r)\}_{i=1}^{N}$, where $\mathcal{I}_i^u$ and $\mathcal{I}_i^r$ are user intent labels for context utterances and response candidates for the i-th dialog respectively. Our task is to learn a ranking model $f(\cdot)$ with $\mathcal{D}$ and $\mathcal{E}$. For any given $\mathcal{U}_i$, the model should be able to generate a ranking list for the candidate responses $\mathcal{R}_i$ with $f(\cdot)$. Note that in practice, $\mathcal{E}$ can come from predicted results of user intent classifiers to reduce human annotation costs. In this chapter, $\mathcal{E}$ are the predicted results of the user intent classifier from a previous work (Qu et al., 2019) for MSDialog and Ubuntu Dialog Corpus. For AliMe data, there is an intention classifier in an eCommerce assistant bot to identify the intention of each customer question (Li et al., 2017). $\mathcal{E}$ is the output of the intention classifier which is a probabilistic distribution over 40 intention scenarios.

Table 5.2: A summary of key notations in this chapter. Note that all vectors are denoted with bold cases.

$\mathcal{D}$	The conversation data set used for training/validation/testing
$\mathcal{E}$	The user intent labels from prediction results of intent classifiers
$u_i^t, \mathcal{U}_i, \mathcal{U}$	The t-th utterance of the i-th dialog, all utterances of the i-th dialog and the set of all dialog utterances
$r_i^k, \mathcal{R}_i, \mathcal{R}$	The k-th response candidate for the i-th dialog, all response candidates of the i-th dialog and the set of all candidate responses
$y_i^k, \mathcal{Y}$	The label for the k-th response candidate for the i-th dialog and the set of all labels
$\mathbf{I}_u^t, \mathbf{I}_r^k$	The user intent representation for u_i^t and r_i^k
$f(\cdot)$	The ranking model learned with $\mathcal{D}$ and $\mathcal{E}$
$f(\mathcal{U}_i, r_i^k)$	The predicted matching score between $\mathcal{U}_i$ and r_i^k
N	The total number of dialogs in $\mathcal{D}$
M	The total number of response candidates for $\mathcal{U}_i$
l_r, l_u	The length of a response candidate and the length of a context utterance
l_t	The number of dimensions of user intent vectors, which is also the number of different user intent labels
l_c	The window size for the utterances in dialog context, which is the maximal number of utterance turns in context modeled
L	The number of stacked layers in the Transformer encoder
d	The number of dimensions of word embedding vectors
$\mathbf{M}_s$,$\mathbf{M}_c$	Interaction matrices between dialog utterance u_i^t and candidate response r_i^k based on self-attention and cross-attention matching
$\mathcal{B}$	The matching tensor which stacks the self-attention matching matrices and cross-attention matching matrices

5.2.2 Method Overview

In following sections, we describe the proposed neural ranking models with user intent modeling for intent-aware response ranking in information-seeking conversations. A summary of key notations in this chapter in presented in Table 5.2. IART incorporates intent-aware utterance attention to derive the importance weighting scheme of different context utterances. Given input context utterances and response candidates, we firstly generate representations from two different perspectives: user intent representations with a trained neural classifier and semantic information encoding with Transformers. Then self-attention matching and cross-attention matching will be performed over encoded representations from Transformers to extract important matching features. These matching features will be weighted by the intent-aware attention mechanism and aggregated into a matching tensor. Finally a two-layer 3D

convolutional neural network will distill final representations over the matching tensor and generate the ranking score for the conversation context/ response candidate pair. We will present details of different components of IART in following sections.

5.2.3 User Intent Taxonomy

We use the MSDialog dataset[4] that consists of technical support dialogs for Microsoft products developed by Qu et al. (2018). The dataset contains two sets, a complete set that consists of all the crawled dialogs and a labeled subset that contains dialogs with user intent annotation. The complete set consists of $35,000$ multi-turn QA dialogs in the technical support domain. Over $2,000$ dialogs with $10,020$ utterances were sampled for user intent annotation on Amazon Mechanical Turk.[5] A taxonomy of 12 labels presented in Table 5.3 were developed in Qu et al. (2018) to characterize the user intent in information-seeking conversations. The user intent labels include question related labels (e.g. Original Questions, Clarifying Quesiton, Follow-up Question, etc.), answer related labels (e.g. Potential Answer, Further Details, etc.), feedback related labels (e.g. Positive Feedback, Negative Feedback) and greeting related labels (e.g. Greetings/ Gratitude), which cover most of user intent types in information-seeking conversations. Inter-rater agreement score was used to ensure the annotation quality. Intent annotations with low agreement scores were filtered. In addition to MSDialog, we also consider another open benchmark data Ubuntu Dialog Corpus (UDC) (Lowe et al., 2015), which consists of almost one million two-person technical support conversations about Ubuntu. User intent annotation is also performed for randomly sampled $4,063$ UDC utterances in the dialogs adopting the same user intent taxonomy.

---

[4] `https://ciir.cs.umass.edu/downloads/msdialog/`

[5] `https://www.mturk.com/`

Table 5.3: Descriptions of user intent taxonomy.

Code	Label	Description
OQ	Original Question	The first question that initiates a QA dialog
RQ	Repeat Question	Questions repeating a previous question
CQ	Clarifying Question	Users or agents ask for clarification
FD	Further Details	Users or agents provide more details
FQ	Follow Up Question	Follow-up questions about relevant issues
IR	Information Request	Agents ask for information from users
PA	Potential Answer	A potential solution to solve the question
PF	Positive Feedback	Positive feedback for working solutions
NF	Negative Feedback	Negative feedback for useless solutions
GG	Greetings/Gratitude	Greet each other or express gratitude
JK	Junk	No useful information in the utterance
O	Others	Utterances that cannot be categorized

5.2.4 Utterance/ Response Input Representations

Given a response candidate r_i^k and an utterance u_i^t in the context $\mathcal{U}_i$, the model firstly looks up a global initial embedding dictionary to represent r_i^k and u_i^t as two sequences of embedding vectors $\mathbf{E}(r_i^k) = [\mathbf{e}_{r,1}, \mathbf{e}_{r,2}, \cdots, \mathbf{e}_{r,l_r}]$ and $\mathbf{E}(u_i^t) = [\mathbf{e}_{u,1}, \mathbf{e}_{u,2}, \cdots, \mathbf{e}_{u,l_u}]$, where $\mathbf{e}_{r,i} \in \mathbb{R}^d$, $\mathbf{e}_{u,i} \in \mathbb{R}^d$ are the embedding vectors of the i-th word in r_i^k and u_i^t respectively. We then represent the utterance/ response pair from two different perspectives to perform intent-aware response ranking: 1) user intent representation with intent classifiers (Section 5.2.4.1); 2) utterance/ response semantic information encoding with Transformers (Section 5.2.4.2).

5.2.4.1 User Intent Representation

To represent user intent, we adopt the best setting of the neural classifiers CNN-Context-Rep proposed by Qu et al. (2019) for user intent classification. Specifically, given sequences of embedding vectors for context utterances and response candidate $\mathbf{E}(u_i^t)$ and $\mathbf{E}(r_i^k)$, convolutional filters with the shape (f, d) are applied to a window of f words to produce a new feature c_i. This operation is applied to every possible window of words in the utterance u_i^t and generates a feature map $\mathbf{c} = \{c_1, c_2, \ldots, c_{n-f+1}\}$. Max pooling is applied to select the most salient feature of a window of p features

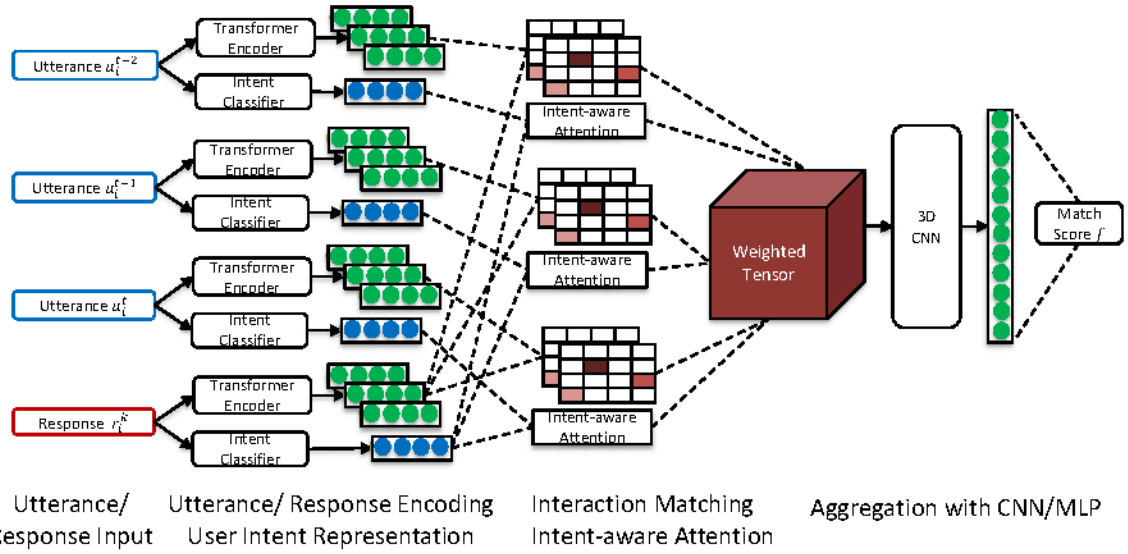


Figure 5.1: The architecture of IART model for intent-aware conversation response ranking.

by taking the maximum value $\hat{c}_i = max\{\mathbf{c}_{i:i+p-1}\}$, where p denotes the max pooling kernel size. The model uses multiple filters with varying window sizes to obtain multiple features in different granularity. These features will be concatenated and flatten into an output tensor, which will be projected into a tensor with shape $(l_t, 1)$ with a fully connected layer. Here l_t is the number of different user intent labels.[6]

As shown in the previous work (Qu et al., 2018), the user intent of a given utterance is closely related to the utterances around it, which compose the context for the given utterance. In order to incorporate context information, CNN-Context-Rep applies convolution operations and max pooling to the utterances $(u_i^{t-1}, u_i^t, u_i^{t+1})$ separately. After global pooling following the last convolutional layer, the three one-dimensional tensors are concatenated for final predictions. Note that this approach only applies to context utterances since response candidates don't have context information.The final outputs of this intent classifier are the intent representation vectors $\mathbf{I}_u^t$ and

[6]In our experiments for MSDialog and UDC, $l_t = 12$ as presented in Section 5.2.3.

$\mathbf{I}_r^k$ for context utterances and response candidates, where $\mathbf{I}_u^t \in \mathbb{R}^{l_t \times 1}, \mathbf{I}_r^k \in \mathbb{R}^{l_t \times 1}$. These intent representation vectors will become the basis of the intent-aware attention presented in Section 5.2.5. We show the evaluation results of different user intent classification models on MSDialog from Qu et al. (2018) in Table 5.4. We can find that CNN-Context-Rep outperforms all baseline methods. The accuracy and F1 are 0.69 and 0.71 respectively. Since we have larger training data with MSDialog, we firstly train an intent classification model with annotated 10K MSDialog utterances and predict the user intent for all MSDialog utterances/ response candidates in the response ranking data. Then we fine-tune the model with the annotated UDC data to adapt the classifier for UDC. The model after fine-tuning will be used to predict the user intent of all UDC utterances/ response candidates.

Table 5.4: Evaluation results of different user intent classification models on MSDialog from Qu et al. (2018). The significance test can only be performed on accuracy. In a multi-label classification setting, accuracy gives a score for each individual sample, while other metrics evaluate the performance over all samples. ‡ means statistically significant difference over the best baseline with $p < 10^{-4}$ measured by the Student's paired t-test.

Method Types	Methods	Accuracy	Precision	Recall	F1
Feature based	Random Forest	0.6268	**0.7657**	0.5903	**0.6667**
Baselines	AdaBoost	**0.6399**	0.7247	**0.6030**	0.6583
	BiLSTM	0.5515	0.6284	0.5274	0.5735
Neural	CNN	**0.6364**	0.7152	**0.6054**	**0.6558**
Baselines	CNN-MFS	0.6342	**0.7308**	0.5919	0.6541
	Char-CNN	0.5419	0.6350	0.4940	0.5557
	BiLSTM-Context	0.6006	0.6951	0.5640	0.6227
Neural	CNN-Feature	0.6509	0.7619	0.6110	0.6781
Classifiers	CNN-Context	0.6555	0.7577	0.6070	0.6740
	CNN-Context-Rep	**0.6885**‡	**0.7883**	**0.6516**	**0.7134**

5.2.4.2 Utterance/ Response Encoding and Matching with Transformers

Self-attention based models like Transformers (Vaswani et al., 2017) have shown impressive performances for NLP tasks including machine translation, natural language inference and question answering. These models require significantly less time

to train compared with other neural models like RNN, since the computation of attention mechanism can be parallelized. We also adopt the encoder architecture in Transformers to encode the semantic dependency information in utterance/ response pairs. We firstly introduce the Scaled Dot-Product Attention used in Transformers (Vaswani et al., 2017), which performs transformation from a query and a set of key-value pairs to an output representation. The output representation is defined as a weighted sum of the values, where the weight to each value is computed as the interaction score between the query and the corresponding key normalized by the softmax function. Specifically, given the input query embeddings $\mathcal{Q}$, key embeddings $\mathcal{K}$ and value embeddings $\mathcal{V}$, where $\mathcal{Q} \in \mathbb{R}^{l_\mathcal{Q} \times d}$, $\mathcal{K} \in \mathbb{R}^{l_\mathcal{K} \times d}$, $\mathcal{V} \in \mathbb{R}^{l_\mathcal{V} \times d}$, the scaled dot-prod attention is defined as:

$$\text{Attention}(\mathcal{Q}, \mathcal{K}, \mathcal{V}) = \text{softmax}\Big(\frac{\mathcal{Q}\mathcal{K}^T}{\sqrt{d}}\Big)\mathcal{V} \tag{5.1}$$

where $l_\mathcal{Q}$, $l_\mathcal{K}$, $l_\mathcal{V}$ are the number of words in each sentence and $l_\mathcal{K} = l_\mathcal{V}$. To avoid pushing the softmax function into regions with extremely small gradients when d is large, the dot product between $\mathcal{Q}$ and $\mathcal{K}$ is scaled by $\frac{1}{\sqrt{d}}$. In practice, we usually set $\mathcal{K} = \mathcal{V}$. The output of the attention function has the same shape with the query sentence $\mathcal{Q}$. Following the design of Transformers, we also add a feed-forward network FFN with ReLU activation over the layer normalized (Ba et al., 2016) sum of the output $\text{Attention}(\mathcal{Q}, \mathcal{K}, \mathcal{V})$ and the query $\mathcal{Q}$, which is defined by:

$$FFN(x) = \max(0, x\mathbf{W}_1 + b_1)\mathbf{W}_2 + b_2 \tag{5.2}$$

Here x is a tensor in the same shape with the query sentence $\mathcal{Q}$ and $\mathbf{W}_1, b_1, \mathbf{W}_2, b_2$ are model parameters of FFN to be learned. $\text{FFN}(x)$ is residually added with x and is then normalized as the final representations. We refer to this module as the TransformerEncoder module $\text{TransformerEncoder}(\mathcal{Q}, \mathcal{K}, \mathcal{V})$, which will be used as a feature extractor for utterances and responses to capture both the dependency

information within words in the same sequence and interactions between words in two different sequences.

To encode context utterances and response candidates, we consider two different types of attention interactions:

- Interaction Matching with Self-attention: for self-attention, we let an utterance or a response to attend to itself to capture dependency within words in the same sequence. We achieve this by setting $\mathcal{Q}, \mathcal{K}, \mathcal{V}$ to the same input sentence. Specifically, the self-attention interaction matching matrix is computed as:

$$\mathbf{M}_s^l = \{\mathbf{S}_{u_i^t}^l[p] \cdot \mathbf{S}_{r_i^k}^l[q]^T\}_{l_u \times l_r} \tag{5.3}$$

$$\mathbf{S}_{u_i^t}^l = \text{TransformerEncoder}(\mathbf{U}_{i,t}^{l-1}, \mathbf{U}_{i,t}^{l-1}, \mathbf{U}_{i,t}^{l-1}) \tag{5.4}$$

$$\mathbf{S}_{r_i^k}^l = \text{TransformerEncoder}(\mathbf{R}_{i,k}^{l-1}, \mathbf{R}_{i,k}^{l-1}, \mathbf{R}_{i,k}^{l-1}) \tag{5.5}$$

  where $\mathbf{S}_{u_i^t}^l$ and $\mathbf{S}_{r_i^k}^l$ is the representation learned in the l-th stacked layer of Transformers for utterance u_i^t and response r_i^k from self-attention. l ranges from 1 to L. When $l = 1$, $\mathbf{U}_{i,t}^0 \in \mathbb{R}^{l_u \times d}$ and $\mathbf{R}_{i,k}^0 \in \mathbb{R}^{l_r \times d}$ denote the initial word embedding sequence of utterance u_i^t and r_i^k. Each element in the self-attention interaction matching matrix $\mathbf{M}_s^l$ is the dot product of the p-th embedding in $\mathbf{S}_{u_i^t}^l$ and the q-th embedding in $\mathbf{S}_{r_i^k}^l$.

- Interaction Matching with Cross-attention: in order to capture the similarity and alignment information between context utterance/ response candidate pairs, we also incorporate cross-attention in IART as follows:

$$\mathbf{M}_c^l = \{\mathbf{C}_{u_i^t}^l[p] \cdot \mathbf{C}_{r_i^k}^l[q]^T\}_{l_u \times l_r} \tag{5.6}$$

$$\mathbf{C}_{u_i^t}^l = \text{TransformerEncoder}(\mathbf{U}_{i,t}^{l-1}, \mathbf{R}_{i,k}^{l-1}, \mathbf{R}_{i,k}^{l-1}) \tag{5.7}$$

$$\mathbf{C}_{r_i^k}^l = \text{TransformerEncoder}(\mathbf{R}_{i,k}^{l-1}, \mathbf{U}_{i,t}^{l-1}, \mathbf{U}_{i,t}^{l-1}) \tag{5.8}$$

  where $\mathbf{C}_{u_i^t}^l$ and $\mathbf{C}_{r_i^k}^l$ is the representation learned in the l-th stacked layer of Transformers for utterance u_i^t and response r_i^k from cross-attention. Each ele-

ment in the cross-attention interaction matching matrix $\mathbf{M}_c^l$ is the dot product of the p-th embedding in $\mathbf{C}_{u_i^t}^l$ and the q-th embedding in $\mathbf{C}_{r_i^k}^l$. Cross-attention can extract dependency features between utterances and responses since the value segments with closer embedding representations to the query segment will be assigned larger attention weights in the scaled dot-product attention of Transformers.

5.2.5 Intent-aware Attention Mechanism

Given the self-attention/ cross-attention interaction matching matrices for different utterances/ response pairs for a dialog, we firstly stack them to aggregate them as a 4D matching tensor as follows:

$$\mathcal{B} = \{\mathbb{B}_{t,p,q,l}\}_{l_c \times l_u \times l_r \times (2L+2)} \tag{5.9}$$

where l_c, l_u, l_r, L are the number of utterance turns in conversation context, number of words in the context utterance, number of words in the response candidate and number of stacked layers in TransformerEncoder. t, p, q, l are indexes along these 4 dimensions of the matching tensor.

We propose intent-aware attention mechanism to weight matching representations of different utterance turns in a conversation context, so that the model can learn to attend to different utterance turns in context. The motivation is to incorporate a more flexible way to weight and aggregate matching features of different turns with intent-aware attention. Specifically, let $\mathbf{I}_u^t \in \mathbb{R}^{l_t \times 1}, \mathbf{I}_r^k \in \mathbb{R}^{l_t \times 1}$ denote the intent representation vectors defined in Section 5.2.4.1 for context utterances and response candidates, we design three different types of intent-aware attention as follows:

- Dot Product: we firstly concatenate the two intent representation vectors of the utterance/ response pair, and then compute the dot product between a model weight parameter $\mathbf{w}$ and the concatenated vector:

$$\mathcal{A}_t = \frac{\exp(\mathbf{w}^T[\mathbf{I}_u^t, \mathbf{I}_r^k])}{\sum_{t'} \exp(\mathbf{w}^T[\mathbf{I}_u^{t'}, \mathbf{I}_r^k])} \tag{5.10}$$

where $\mathbf{w} \in \mathbb{R}^{2l_t \times 1}$ is the model parameter to be learned.

- Bilinear: we compute the bilinear interaction between $\mathbf{I}_u^t$ and $\mathbf{I}_r^k$ and then normalize the result with a softmax function:

$$\mathcal{A}_t = \frac{\exp({\mathbf{I}_u^t}^T \mathbf{w} \mathbf{I}_r^k)}{\sum_{t'} \exp({\mathbf{I}_u^{t'}}^T \mathbf{w} \mathbf{I}_r^k)} \tag{5.11}$$

where $\mathbf{w} \in \mathbb{R}^{l_t \times l_t}$ is the bilinear interaction matrix to be learned.

- Outer Product: we compute the outer product between $\mathbf{I}_u^t$ and $\mathbf{I}_r^k$ and then flat the result matrix to a feature vector. Finally we project this feature vector into an attention score with a fully connected layer and a softmax function:

$$\mathcal{A}_t = \frac{\exp(\mathbf{w}^T \cdot \text{flat}(\mathbf{I}_u^t \otimes {\mathbf{I}_r^k}^T))}{\sum_{t'} \exp(\mathbf{w}^T \cdot \text{flat}(\mathbf{I}_u^{t'} \otimes {\mathbf{I}_r^k}^T))} \tag{5.12}$$

where flat and $\otimes$ denote the flatten layer which transforms a matrix with shape $(l_t \times l_t)$ into a vector with shape $(l_t^2 \times 1)$ and out product operation. $\mathbf{w} \in \mathbb{R}^{l_t^2 \times 1}$ is a model parameter to be learned.

Note that the normalization in the softmax function is performed over all utterance turns within a conversation context. Thus the result $\mathcal{A}_t$ is the attention weight corresponding to the t-th utterance turn in a conversation context. We also add masks over the padded utterance turns to avoid introducing noise matching feature representations. With the computed attention weights over context utterance turns, we can scale the 4D matching tensor to generate a weighted matching tensor:

$$\widehat{\mathcal{B}} = \{\mathbb{B}_{t,p,q,l} \cdot \mathcal{A}_t\}_{l_c \times l_u \times l_r \times (2L+2)} \tag{5.13}$$

Finally IART adopts a two layer 3D convolution neural network (CNN)[7] to extract important matching features from this weighted matching tensor $\widehat{\mathcal{B}}$. 3D CNN requires 5D input and filter tensors, as we can add one more input dimension corresponding to the batched training examples over the 4D weighted matching tensor. We compute the final matching score $f(\mathcal{U}_i, r_i^k)$ with a MLP over the flatten output of the 3D CNN.

5.2.6 Loss and Model Training

For model training, we consider the cross-entropy loss between the predicted matching scores $f(\mathcal{U}_i, r_i^k)$ and the ground truth matching labels as follows:

$$\mathcal{L}(\mathcal{D}, \mathcal{E}; \Theta) = \sum_{i=1}^{P} -y_i^k \log(f(\mathcal{U}_i, r_i^k)) - (1 - y_i^k) \log(1 - f(\mathcal{U}_i, r_i^k)) \tag{5.14}$$

Where P is the total number of context utterance/ response pairs. y_i^k is the ground truth matching label. The parameters of IART are optimized using back-propagation with *Adam* algorithm (Kingma and Ba, 2014).

5.3 Experiments

5.3.1 Data Set Description

We evaluated our method with three data sets: Ubuntu Dialog Corpus (UDC), MSDialog, and an internal commercial data AliMe consisting of a set of customer service conversations in Chinese from a large eCommerce company. They have also been used for the response ranking experiments in Chapter 4. The data statistics is shown in Table 4.4. UDC contains multi-turn technical support conversation data collected from the chat logs of the Freenode Internet Relay Chat (IRC) network. It

[7] `https://www.tensorflow.org/api_docs/python/tf/nn/conv3d`

consists of 1 million context-response pairs for training, 0.5 million pairs for validation and 0.5 million pairs for testing. MSDilaog is released by Qu et al. (2018). It contains QA dialogs on various Microsoft products crawled from the Microsoft Answer community. Yang et al. (2018) processed the data and created a version which is suitable for response ranking experiments. We use the same data version for the experiments in this Chapter. The ground truth responses returned by the real agents are the positive response candidates. Negative sampling has been adopted to create negative response candidates. For the AliMe data, they are the chat logs between customers and a chatbot from "2017-10-01" to "2017-10-20" in a large eCommerce company. Detailed descriptions and statistics of these three data sets can be found in Chapter 4. We skip these detailed descriptions here to avoid duplicated content. Note that we have included human evaluation in AliMe data.

5.3.2 Experimental Setup

5.3.2.1 Baselines

We consider different types of baselines for comparison, including traditional retrieval models, neural ranking models and the state-of-the-art multi-turn conversation response ranking method as follows[8]:

- Traditional retrieval models: these methods treat the dialog context as the query to retrieve response candidates for response selection. We consider BM25 model (Robertson and Walker, 1994) as the retrieval model. We also consider BM25-PRF (Yang et al., 2018), which matches conversation context with the expanded responses using BM25 model.

[8]We did not compare with (Tao et al., 2019) since the code of the proposed MRFN model is not available until this submission, although the authors presented that they would release the code of the MRFN model in the paper.

- Neural ranking models: in recent years there are some neural ranking models (Guo et al., 2019) proposed for ad-hoc retrieval and question answering. We consider several representative methods in this category: ARC-II (Hu et al., 2014), MV-LSTM (Wan et al., 2016), DRMM (Guo et al., 2016) and Duet (Mitra et al., 2017). MV-LSTM is a representation focused model and ARC-II, DRMM are interaction focused models. Duet is a hybrid method of both representation focused and interaction-focused models.

- Deep Matching Network (DMN) with External Knowledge (Yang et al., 2018): these models incorporate external knowledge into deep neural ranking models with pseudo-relevance feedback (DMN-PRF) and QA correspondence knowledge distillation (DMN-KD) for response ranking in multi-turn conversations. DMN is the version of model without the integration of external knowledge information.

- Deep Attention Matching Network (DAM) (Zhou et al., 2018b): DAM is the state-of-the-art model for response ranking in multi-turn conversations with open source code released[9] until this submission. DAM also represents and matches a response with its multi-turn context using dependency information learned by Transformers. But it does not explicitly model user intent information in information-seeking conversations.

5.3.2.2 Evaluation Methodology

For evaluation metrics, we adopted mean average precision (MAP) and $R_n@k$ which is the recall at top k ranked responses from n available candidates for a given conversation context following previous related works (Zhou et al., 2018b; Yang et al., 2018; Wu et al., 2017; Lowe et al., 2015). We reported R10@1, R10@2, and R10@5.

[9] `https://github.com/baidu/Dialogue/tree/master/DAM`

5.3.2.3 Parameter Settings and Implementation Details

All models are implemented with TensorFlow[10] and MatchZoo[11] toolkit. Hyper-parameters are tuned with the validation data. For the hyper-parameter settings of IART, we set the size of the convolution and pooling kernels as $(3, 3, 3)$. The number of stacked Transformer layers is set as 5 for UDC and 4 for MSDialog. The batch size is 128 for UDC and 32 for MSDialog. All models are trained on a single Nvidia Titan X GPU. Learning rate is initialized as 1e-3 with exponential decay during training process. The decay steps and decay rate are set as 400 and 0.9 respectively. The maximum utterance length is 50 for UDC and 200 for MSDialog. The maximum number context utterance turns is set as 9 for UDC and 6 for MSDialog. We padded zeros if the number of utterance turns in a context is less than the maximum number of utterance turns. For user intent labels, there are 12 different types for UDC/ MSDialog, and 40 different types for AliMe data. For the word embeddings, we trained word embeddings with the Word2Vec tool with the CBOW model using our training data following previous work (Wu et al., 2017; Zhou et al., 2018b). The max skip length between words and the number of negative examples is set as 10 and 25. The dimension of word embeddings is 200. Word embeddings will be initialized by these pre-trained word vectors and updated during the training process.

5.3.3 Evaluation Results

5.3.3.1 Performance Comparison on UDC and MSDialog

We present evaluation results over different methods on UDC and MSDialog in Table 5.5. We summarize our observations as follows: (1) On MSDialog, all three variations of IART with dot, outer product and bilinear based intent-aware attention mechanism show significant improvements over all baseline methods including the

[10] https://www.tensorflow.org/

[11] https://github.com/NTMC-Community/MatchZoo

Table 5.5: Comparison of different models over Ubuntu Dialog Corpus (UDC) and MSDialog. Numbers in bold font mean the result is better compared with the best baseline DAM. † and ‡ means statistically significant difference over the best baseline DAM with $p < 0.1$ and $p < 0.05$ measured by the Student's paired t-test respectively.

Data	UDC				MSDialog			
Methods	R10@1	R10@2	R10@5	MAP	R10@1	R10@2	R10@5	MAP
BM25	0.5138	0.6439	0.8206	0.6504	0.2626	0.3933	0.6329	0.4387
BM25-PRF	0.5289	0.6554	0.8292	0.6620	0.2652	0.3970	0.6423	0.4419
ARC-II	0.5350	0.6959	0.8978	0.6855	0.3189	0.5413	0.8662	0.5398
MV-LSTM	0.4973	0.6733	0.8936	0.6611	0.2768	0.5000	0.8516	0.5059
DRMM	0.5287	0.6773	0.8776	0.6749	0.3507	0.5854	0.9003	0.5704
Duet	0.4756	0.5592	0.8272	0.5692	0.2934	0.5046	0.8481	0.5158
DMN	0.6056	0.7509	0.9196	0.7363	0.4521	0.6673	0.9155	0.6415
DMN-KD	0.6443	0.7841	0.9351	0.7655	0.4908	0.7089	0.9304	0.6728
DMN-PRF	0.6552	0.7893	0.9343	0.7719	0.5021	0.7122	0.9356	0.6792
DAM	0.7686	0.8739	0.9697	0.8527	0.7012	0.8527	0.9715	0.8150
IART-Dot	**0.7703**	**0.8746**	0.9688	**0.8535**	**0.7234**‡	**0.8650**‡	**0.9772**‡	**0.8300**‡
IART-Outerprod	**0.7717**‡	**0.8766**‡	0.9691	**0.8548**‡	**0.7212**‡	**0.8664**‡	**0.9749**	**0.8289**‡
IART-Bilinear	**0.7713**‡	**0.8747**	0.9688	**0.8542**†	**0.7317**‡	**0.8752**‡	**0.9792**‡	**0.8364**‡

state-of-the-art method DAM. On UDC, IART with three different intent-aware attention mechanisms also show improvements under all metrics except R10@5. With the comparison between the results of DAM and IART, we can find that incorporating user intent modeling and intent-aware attention weighting scheme to combine the self-attention and cross-attention interaction matching matrices from Transformers can help improve the response ranking performance. (2) If we compare three variations of IART, we can find that the bilinear based intent-aware attention mechanism works better for MSDialog and outer product based intent-aware attention mechanism works better for UDC. The overall performances of these three model variations are close to each other. (3) For statistical significance testing results, we find that most improvements of IART over the best baseline DAM on MSDialog are statistically significant with $p < 0.05$ measured by the Student's paired t-test. For UDC, the difference between IART and outer product based intent-aware attention mechanism and DAM is statistically significant. In general, our proposed model IART shows larger performance improvements on MSDialog. One possible reason is that the intent classifier on MSDialog is more accurate due to the larger annotated training data

Table 5.6: Comparison of different models over the AliMe data. Numbers in bold font mean the result is better compared with the best baseline DAM. † and ‡ means statistically significant difference over the best baseline DAM with $p < 0.1$ and $p < 0.05$ measured by the Student's paired t-test respectively.

Data	AliMe			
Methods	R10@1	R10@2	R10@5	MAP
BM25	0.2371	0.4204	0.6407	0.6392
BM25-PRF	0.2454	0.4209	0.6510	0.6412
ARC-II	0.2236	0.3671	0.6595	0.7306
MV-LSTM	0.2480	0.4105	0.7017	0.7734
DRMM	0.2212	0.3616	0.6575	0.7165
Duet	0.2433	0.4088	0.6870	0.7651
DMN	0.3568	0.5012	0.7629	0.7833
DMN-KD	0.3596	0.5122	0.7631	0.8323
DMN-PRF	0.3601	0.5323	0.7701	0.8435
DAM	0.3819	0.5567	0.7717	0.8452
IART-Dot	**0.3821**	0.5547	**0.7802**†	**0.8454**
IART-Outerprod	**0.3901**‡	**0.5649**‡	**0.7812**†	**0.8493**†
IART-Bilinear	**0.3892**†	**0.5592**†	**0.7801**†	**0.8471**

of MSDialog for user intent prediction and more formal language used in MSDialog, as shown in evaluation results by Qu et al. (2019).

5.3.3.2 Performance Comparison on AliMe Data

We further compare our models with the competing methods on AliMe data in Table 5.6. We have similar findings with the experiments on UDC and MSDialog datasets. (1) On AliMe dataset, all three variations of IART show comparable or better results than all baseline methods including the state-of-the-art method DAM. This further demonstrates the effectiveness of our proposed methods. (2) If we compare three variations of IART, we can find that the outer product based intent-aware attention mechanism work better than the other two variations. But still, the overall performances of these three model variations are close to each other.

5.3.4 Impact of Different Context Utterance Number and Utterance Length

We further analyze the impact of different hyper-parameter settings on the performances of our proposed models. Figure 5.2 shows the performances of IART with

different choices of maximum context utterance number and maximum utterance length over the validation partition of UDC and MSDialog data. We find that when the maximum context utterance number is small (e.g. 3 for UDC and 2 for MSDialog), increasing this number will lead to better response ranking performances. Since larger value means the model can be potentially trained with longer conversation context. Thus more semantic encoding information and intent-aware attention weights of conversation context turns can be learned. However, when this value is larger than some threshold (e.g. 9 for UDC and 6 for MSDialog), continuing increasing this number won't add benefits to response ranking performances. One possible reason is that too large maximum context utterance number will be more likely to introduce noisy irrelevant historical conversation turns into the context. For maximum utterance length on UDC, the ranking metrics will increase if we increase the maximum utterance length from 20 to 50. Then the performance will not change if we continue increasing the maximum utterance length. For MSDialog, the performances are not as stable as those with UDC. The reason could be that, the validation data of MSDialog (37K context/ response pairs) is much smaller than that of UDC (500K context/ response pairs). So we can see some fluctuation of the ranking performance when we increase the maximum utterance length over MSDialog. For the choice of this model hyper-parameter, we find that the double of the average length of context utterances/ response candidates in the training data is usually a good setting.[12]

5.3.5 Case Study and User Intent Visualization

We perform a case study in Table 5.7 on the top ranked responses by different methods including the best baseline DAM and our proposed model IART with bilinear based intent-aware attention mechanism. We show the conversation context utter-

[12]For example, the average context utterance lengths in the training data of UDC and MSDialog are 22 and 106. We set the maximum utterance length as 50 for UDC and 200 for MSDialog.

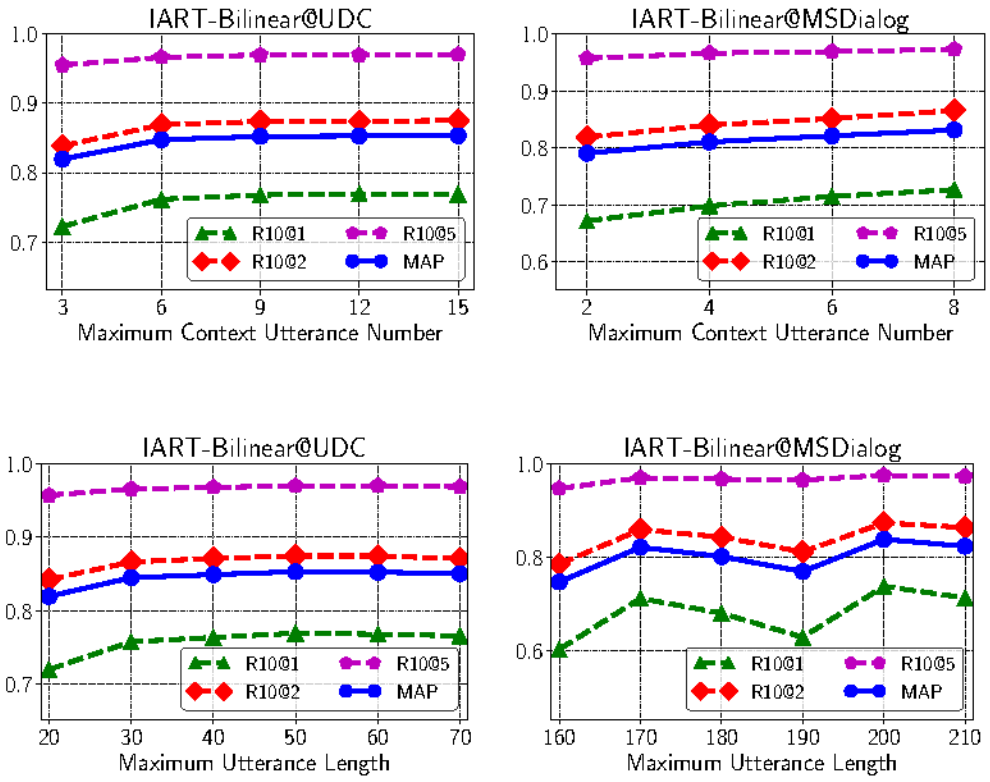


Figure 5.2: Performance of IART with different choices of maximum context utterance number and maximum utterance length over the validation partition of UDC and MSDialog data.

Table 5.7: A case study and examples of Top-1 ranked responses by different methods. y_i^k means the label of a response candidate. The predicted user intent and user roles are highlighted by bold font.

Context		**[User]** Hi, I have the new Outlook which updated a few days ago. I cannot find how to add senders to my blocked senders list manually. I'd like to block a domain which is spamming me. How do I do this on the new Outlook? Thanks **[Agent]** Hi, There are different ways to block senders on Outlook depending on the version of Outlook that you are using. May we know what version of Outlook are you using? We'll be looking forward for your reply. Regards. **[User]** Hi, I'm using the desktop website beta version. Thanks. **[Agent]** Desktop Website beta version? Are you referring to the Outlook Web App or the Windows mail? We'll look forward for your response. **[User]** I go to Outlook.com and sign in on there.
Context Intent		**[OQ]** → **[IR]** → **[PA]** → **[IR]** → **[FD/ OQ]**
Method	y_i^k	Top-1 Ranked Response
DAM	0	Thanks for the reply. Some email domain needs to be manually added to Outlook. However, it's good to know that the issue is resolved from your end. Should you need further assistance in the future, please do let us know. **[PF]**
IART	1	In Outlook Web App, moving an email from your Inbox to the Junk folder by clicking Junk button on the toolbar after viewing or selecting the said email will automatically redirect incoming emails from that sender to the Junk folder. And to manually block an email address, follow these steps: Let us know how things go. **[PA]**

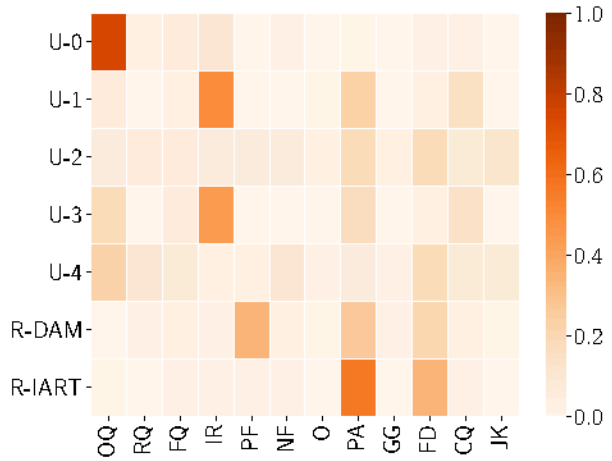


Figure 5.3: Visualization of learned user intent representation of context utterances and returned top-1 ranked response by DAM and IART from the case study in Table 5.7. U-0 to U-4 denotes the 0-th turn to the 4-th utterance turn in the context. R-DAM and R-IART denotes the top-1 ranked response returned by DAM and IART respectively. Darker spots mean higher predicted probabilities.

ances and top-1 ranked response by each method. In this example, IART produced the correct top ranked response. We visualized the learned user intent representation of context utterances and returned top-1 ranked response by DAM and IART in Figure 5.3. The predicted user intent of conversation utterances is **[OQ]** $\rightarrow$ **[IR]** $\rightarrow$ **[PA]** $\rightarrow$ **[IR]** $\rightarrow$ **[FD/ OQ]**, which means that there is an utterance on "Information Request (IR)" after the user proposed an "Original Question (OQ)". Then the user provided the answer to respond the information request from the agent. After that the agent performed another "Information Request (IR)" to confirm whether it is the Outlook Web app or the Windows desktop app. The user finally confirmed "Further Details (FD)" that the problem was related to the Outlook Web app (Outlook.com). This utterance is also relevant to the original question (OQ). Given such a user intent pattern in the conversation context, a reasonable response can be with intent "Potential Answers (PA)" on providing potential solutions to the user's question, which is captured by IART due to the integration of user intent modeling in response ranking

process. The DAM model, without user intent modeling, failed in such cases and selected a response candidate with "Positive Feedback (PF)" intent. The response returned by DAM assumed that "the issue is resolved", but actually the user was expecting an answer to her unsolved technical problem. On the other hand, both the returned response by DAM and IART show some lexical and semantic similarities with the conversation context. It is difficult to decide which one is better without the modeling of user intent information in the utterances. This gives an example and interpretation of why user intent modeling can be helpful for response ranking in conversations.

5.4 Summary

In this chapter, we analyze user intent in information-seeking conversations and propose an intent-aware neural ranking model with Transformers. Different user intent types are defined and characterized following previous research on user intent in information-seeking conversations. Then we propose an intent-aware neural ranking model for response retrieval, which is built on the top of the recent breakthrough of natural language representation learning with Transformers. Our proposed model incorporates intent-aware utterance attention to derive the importance weighting scheme of different utterances in conversation context towards better conversation history understanding. We conduct extensive experiments with three information-seeking conversation data sets including both standard benchmarks and commercial data. Our proposed methods outperform all baseline methods regrading a variety of metrics. We also perform case studies and analysis of learned user intent with their impact on response ranking in information-seeking conversations to provide insights and interpretation of experimental results.

CHAPTER 6

HYBRID RETRIEVAL-GENERATION NEURAL CONVERSATION MODELS

6.1 Introduction

Typical conversation systems are modularized systems with a natural language understanding module, a dialog state tracker, a dialog policy learning module, and a natural language generation module (Henderson, 2015). In recent years, fully data-driven end-to-end conversation models have been proposed to reduce hand-crafted features, rules or templates. These methods could be grouped into two different categories: generation-based approaches (Ritter et al., 2011; Shang et al., 2015; Sordoni et al., 2015; Vinyals and Le, 2015; Li et al., 2016b; Bordes et al., 2017) and retrieval-based approaches (Ji et al., 2014; Yan et al., 2016a,b, 2017; Yang et al., 2018).

Given some conversation context, retrieval-based models try to find the most relevant context-response pairs in a pre-constructed conversational history repository. Some of these methods achieve this in two steps: 1) retrieve a candidate response set with basic retrieval models such as BM25 (Robertson and Walker, 1994) or QL (Ponte and Croft, 1998); and 2) re-rank the candidate response set with neural ranking models to find the best matching response (Yan et al., 2016a,b, 2017; Wu et al., 2017; Yang et al., 2018). These methods can return natural human utterances in the conversational history repository, which is controllable and explainable. Retrieved responses often come with better diversity and richer information compared to generated responses (Song et al., 2018). However, the performance of retrieval-based methods is limited by the size of the conversational history repository, especially for

Table 6.1: A comparison of retrieval-based methods and generation-based methods for data driven conversation models.

Item	Retrieval-based methods	Generation-based methods
Main techniques	Retrieval models; Neural ranking models	Seq2Seq models
Diversity	Usually good if similar contexts have diverse responses in the repository	Easy to generate bland or universal responses
Response length	Can be very long	Usually short
Context property	Easy for similar context in the repository; Hard for unseen context	Easy to generalize to unseen context
Efficiency	Building index takes long time; Retrieval is fast	Training takes long time; Decoding is fast
Flexibility	Fixed response set once the repository is constructed	Can generate new responses not covered in history
Fluency	Natural human utterances	Sometimes bad or contain grammar errors
Bottleneck	Size and coverage of the repository	Specific responses; Long text; Sparse data
Informativeness	Easy to retrieve informative content	Hard to integrate external factual knowledge
Controllability	Easy to control and explain	Difficult to control the actual generated content

long tail contexts that are not covered in the history. Retrieval-based models lack the flexibility of generation-based models, since the set of responses of a retrieval system is fixed once the historical context/response repository is constructed.

On the other hand, the generation-based methods could generate highly coherent new responses given the conversation context. Much previous research along this line was based on the Seq2Seq model (Shang et al., 2015; Sordoni et al., 2015; Vinyals and Le, 2015), where there is an encoder to learn the representation of conversation context as a contextual vector, and a decoder to generate a response sequence conditioning on the contextual vector as well as the generated part of the sequence. The encoder/ decoder could be implemented by an RNN with long short term memory (LSTM) (Hochreiter and Schmidhuber, 1997) or gated recurrent units (GRU) (Chung et al.,

2014) hidden units. Although generation-based models can generate new responses for a conversation context, a common problem with generation-based methods is that they are likely to generate very general or universal responses with insufficient information such as "I don't know", "I have no idea", "Me too", "Yes please". The generated responses may also contain grammar errors. Ghazvininejad et al. (2018) proposed a knowledge-grounded neural conversation model in order to infuse the generated responses with more factual information relevant to the conversation context without slot filling. Although they show that the generated responses from the knowledge-grounded neural conversation model are more informative compared with responses from the vanilla Seq2Seq model, their model is still generation-based, and it is not clear how well this model will perform compared to retrieval-based methods. A comparison of retrieval-based methods and generation-based methods for end-to-end data driven conversation models is shown in Table 6.1. Clearly these two types of methods have their own advantages and disadvantages, it is thus necessary to integrate the merits of these two methods.

To this end, in this chapter we study the integration of retrieval-based and generation-based conversation models in an unified framework. The closest prior research to our work is the study on the ensemble of retrieval-based and generation-based conversation models by Song et al. (2018). Their proposed system uses a multi-seq2seq model to generate a response and then adopts a Gradient Boosting Decision Tree (GBDT) ranker to re-rank the generated responses and retrieved responses. However, their method still required heavy feature engineering to encode the context/ response candidate pairs in order to train the GBDT ranker. They constructed the training data by negative sampling, which may lead to sub-optimal performance, since the sampled negative response candidates could be easily discriminated from the positive response candidates by simple term-matching-based features.

We address these issues by proposing a hybrid neural conversational model with a generation module, a retrieval module and a hybrid ranking module. The generation module generates a response candidate given a conversation context, using a Seq2Seq model consisting of a conversation context encoder, a facts encoder and a response decoder. The retrieval module adopts a "context-context match" approach to recall a set of response candidates from the historical context/ response repository. The hybrid ranking module is built on top of neural ranking models to select the best response candidate among retrieved/ generated response candidates. The integration of neural ranking models, which can learn representations and matching features for conversation context/ response candidate pairs, enables us to minimize feature engineering costs during model development. To construct the training data of the neural ranker for response selection, we propose a distant supervision approach to automatically infer labels for retrieved/ generated response candidates. We evaluate our proposed approach with experiments on Twitter and Foursquare (Ghazvininejad et al., 2018) data. Experimental results show that the proposed model can outperform both retrieval-based models and generation-based models (including a recently proposed knowledge-grounded neural conversation model (Ghazvininejad et al., 2018)) on both automatic evaluation and human evaluation.[1]

In all, our contributions can be summarized as follows:

- We perform a comparative study of retrieval-based models and generation-based models for the conversational response generation task.
- We propose a hybrid neural conversational model to combine response generation and response retrieval with a neural ranking model to reduce feature engineering costs.

[1]Code will be released on Github.

- For model training, we propose a distant supervision approach to automatically infer labels for retrieved/ generated response candidates. We evaluate the effectiveness of different kinds of distant supervision signals and settings for the hybrid ranking of response candidates.

- We run extensive experimental evaluation on retrieval-based, generation-based and hybrid models using the Twitter and Foursquare data. Experimental results show that the proposed hybrid neural conversation model can outperform both retrieval-based and generation-based models on both automatic evaluation and human evaluation. We also perform qualitative analysis on top responses selected by the neural re-ranker and response generation examples to provide insights.

Roadmap. The rest of this chapter is organized as follows. Section 6.2 will present the details of the generation module, retrieval module and hybrid ranking module in the proposed model. Section 6.3 contains the experiments and results analysis. We will conclude in Section 6.4.

6.2 Hybrid Neural Conversation Models

6.2.1 Problem Formulation

We define the task of conversational response generation following the previous literature (Ghazvininejad et al., 2018). We are given a conversation context $u_i \in \mathcal{U}$, where u_i is the i-th context sequence which contains one or multiple utterances. There are also F factual snippets of text $\mathcal{F}_i = \{f_i^1, f_i^2, ..., f_i^F\}$ that are relevant to the i-th conversation context u_i. Based on the conversation context u_i and the set of external facts $\mathcal{F}_i$, the system outputs an appropriate response which provides useful information to users. Figure 1 shows an example of the conversational response generation task. Given an conversation context "*Going to Din Tai Fung Dumpling*

House tonight!", we can associate it with several contextually relevant facts from a much larger collection of external knowledge text (e.g. the Wikipedia dump, tips on Foursquare, product customer reviews on Amazon, etc.). A response that is both appropriate and informative in the given example could be "*The shrimp and pork wontons with spicy sauce are amazing!*".

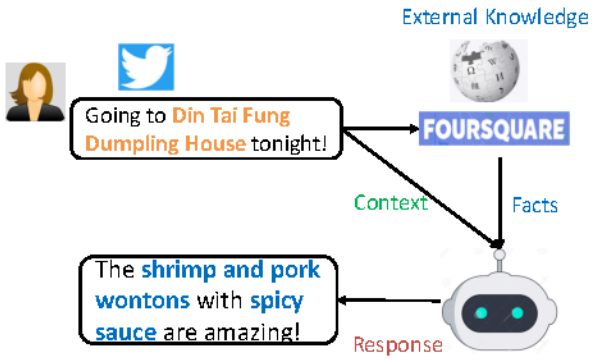


Figure 6.1: An example of the conversational response generation task. The factual information from external knowledge is denoted as blue color.

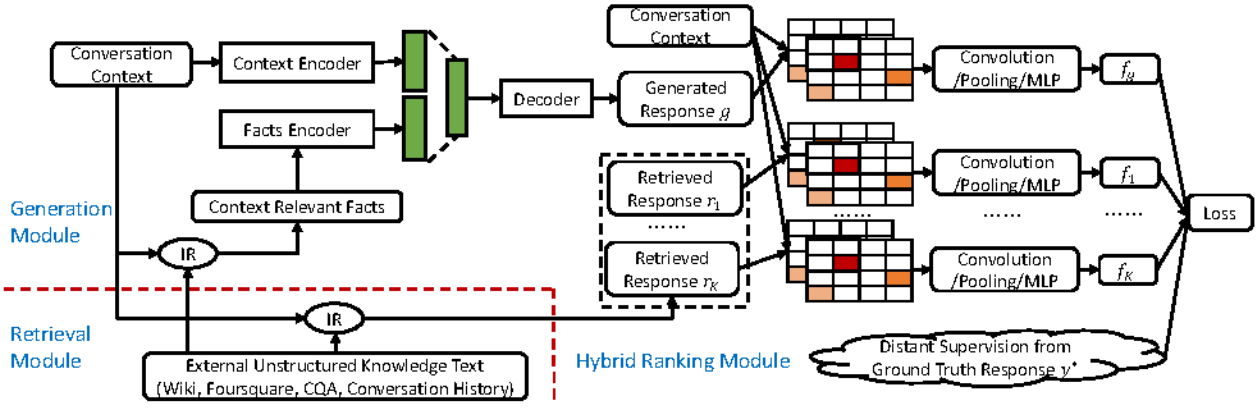


Figure 6.2: The architecture of the Hybrid Neural Conversation Model (HybridNCM).

6.2.2 Method Overview

In the following sections, we describe the proposed Hybrid Neural Conversation Model (HybridNCM) for response generation. Figure 6.2 shows the architecture of

Table 6.2: A summary of key notations in this chapter. Note that all vectors or matrices are denoted with bold cases.

u_i,$\mathcal{U}$	The context of the i-th conversation and the set of all conversation contexts
$f_i^k, \mathcal{F}_i, \mathcal{F}$	The k-th factual text relevant to context u_i, the factual texts relevant to context u_i and the set of all factual texts
$r_i^k, \mathcal{R}_i$, $\mathcal{R}$	the k-th retrieved response candidate to context u_i, the set of all retrieved response candidates for context u_i and the set of all retrieved response candidates
$g_i^k, \mathcal{G}_i$, $\mathcal{G}$	the k-th generated response candidate to context u_i, the set of all generated response candidates for context u_i and the set of all generated response candidates
$y_i^k, \mathcal{Y}_i$	the k-th response candidate and the union set of all the candidates for the i-th context, i.e., $y_i^k \in \mathcal{Y}, \mathcal{Y}_i = \mathcal{R}_i \cup \mathcal{G}_i$
$y_i^*, \mathcal{Y}^*$	The ground truth response candidate for the i-th context and the set of all ground truth response candidates
$f(\cdot)$	The neural ranking model learned in the hybrid ranking module
$f(u_i, y_i^k)$	The predicted matching score between u_i and y_i^k

the hybrid neural conversation model. In general, there are three modules in our proposed model:

(1) Generation Module: given the conversation context u_i and the relevant facts $\mathcal{F}_i$, this module is to generate a set of response candidates $\mathcal{G}_i$ using a Seq2Seq model which consists of a conversation context encoder, a facts encoder and a response decoder.

(2) Retrieval Module: this module adopts a "context-context match" approach to retrieve a few response candidates $\mathcal{R}$. The "context-context matching" approach matches the conversation context u_i with all historical conversation context. It then returns the corresponding responses of the top ranked historical conversation context as a set of the retrieved response candidates $\mathcal{R}_i$.

(3) Hybrid Ranking Module: given the generated and retrieved response candidates, i.e., $\mathcal{Y}_i = \mathcal{G}_i \cup \mathcal{R}_i$, this module is used to re-rank all the response candidates with a hybrid neural ranker trained with labels from distant supervision to find the best response as the final system output.

We will present the details of generating the responses for the i-th context u_i by these modules from Section 6.2.3 to Section 6.2.5. A summary of key notations in

this work is presented in Table 4.2. We use a bold letter for a vector or a matrix, and an unbold letter for a word sequence or a set.

6.2.3 Generation Module

We map a sequence of words to a sequence of embeddings by looking up the indices in an embedding matrix, e.g., $\mathbf{u} = \mathbf{E}(u_i) = [\mathbf{u}_1, \mathbf{u}_2, \cdots, \mathbf{u}_{L_u}]$ where L_u is the length of a word sequence u_i.

6.2.3.1 Context Encoder

Inspired by previous works on response generation with Seq2Seq models (Vinyals and Le, 2015; Shang et al., 2015; Ghazvininejad et al., 2018), we adopt a Seq2Seq architecture with attention mechanism (Bahdanau et al., 2014; Luong et al., 2015) in the hybrid neural conversation model. In the Seq2Seq architecture, a context encoder is used to transform a sequence of context vectors $\mathbf{u} = [\mathbf{u}_1, \mathbf{u}_2, \cdots, \mathbf{u}_{L_u}]$ into contextual hidden vectors $\mathbf{h} = [\mathbf{h}_1, \mathbf{h}_2, \cdots, \mathbf{h}_{L_u}]$ in Eq. (6.1).

$$\mathbf{h}_t = \mathrm{RNN}(\mathbf{u}_t, \mathbf{h}_{t-1}), \tag{6.1}$$

where $\mathbf{h}_t \in \mathbb{R}^H$ is the hidden state at time step t. In our implementation, we stack two layers of LSTM networks as the recurrent neural network. With the context encoder, we can summarize the conversation context by the last hidden vector $\mathbf{h}_{L_u}$ and maintain the detailed information at each time step by each hidden state $\mathbf{h}_t$.

6.2.3.2 Facts Encoder

For the facts encoder, we use the same architecture of the stacked LSTM as the context encoder in Section 6.2.3.1 to generate the hidden representations of relevant facts. Note that for each conversation context u_i, there are F sequences of facts $\mathcal{F} = \{f^1, f^2, \cdots, f^F\}$. We encode these facts into F sequences of hidden vectors

$\{\mathbf{f}^1, \mathbf{f}^2, \cdots, \mathbf{f}^F\}$ by the stacked LSTM, where $\mathbf{f}^j = [\mathbf{f}_1^j, \mathbf{f}_2^j, \cdots, \mathbf{f}_L^j]$ and $L = |\mathbf{f}^j|$. We summarize a fact into a fixed-size vector by averaging its hidden vectors, i.e., $\bar{\mathbf{f}}^j = \text{mean}(\mathbf{f}^j)$.

6.2.3.3 Response Decoder

The response decoder is trained to predict the next word g_t given the representations of conversation context $\mathbf{h}_{L_u}$, facts $\bar{\mathbf{f}}$, and all the previously generated words $g_{1:t-1}$ as follows:

$$p(g|u_i, \mathcal{F}) = \prod_{t=1}^{L_g} p(g_t|g_{1:t-1}, u_i, \mathcal{F}) \tag{6.2}$$

$$\mathbf{E} = [\mathbf{h}_1, \cdots, \mathbf{h}_{L_u}, \bar{\mathbf{f}}^1, \cdots, \bar{\mathbf{f}}^F] \in \mathbb{R}^{H \times (L_u + F)} \tag{6.3}$$

$$\mathbf{a}_t = \text{softmax}\left(\mathbf{E}^T \mathbf{s}_{t-1}\right) \tag{6.4}$$

$$\mathbf{c}_t = \mathbf{E}\mathbf{a}_t \tag{6.5}$$

$$\mathbf{v}_t = \tanh\left([\mathbf{s}_{t-1}, \mathbf{c}_t]\right) \tag{6.6}$$

$$\mathbf{s}_t = \text{RNN}(\mathbf{v}_t, \mathbf{s}_{t-1}) \tag{6.7}$$

$$\mathbf{s}_0 = \varphi\left(\tanh\left(\mathbf{h}_{L_u} + \frac{1}{F}\sum_{j=1}^{F} \bar{\mathbf{f}}^j\right)\right) \tag{6.8}$$

For the decoder, we stack two layers of LSTM networks with the attention mechanism proposed in (Luong et al., 2015). More specifically, we concatenate the hidden vectors of a context u_i and all factual vectors into a matrix $\mathbf{E}$ in Eq. (6.3). We then compute the attention weight $\mathbf{a}_t$ by the dot product between the decoder's previous hidden state $\mathbf{s}_{t-1}$ and all vectors in $\mathbf{E}$, followed by a softmax function in Eq. (6.4). The attention context summarizes the conversation context u_i and facts $\mathcal{F}$ by the weighted sum of $\mathbf{E}$ in Eq. (6.5). For the input to the decoder's RNN network, we concatenate the attention context $\mathbf{c}_t$ and the previous hidden state $\mathbf{s}_{t-1}$ that summarizes the partial generated response $g_{1:t-1}$, and apply a tanh function afterwards in

Eq. (6.6). The initial hidden vector of the decoder is initialized by the last hidden state of the context encoder and the average factual vectors in Eq. (6.8). $\varphi(\cdot)$ is a linear function that maps a vector from the encoder's hidden space to the decoder's hidden space. The conditional probability at the t-th time step can be computed by a linear function $\phi(\cdot)$, which is a fully connected layer, that maps the decoder's hidden state $\mathbf{s}_{t-1}$ to a distributional vector over the vocabulary, and a softmax function in Eq. (6.9).

$$p(g_t|g_{1:t-1}, u_i, \mathcal{F}) = \text{softmax}(\phi([\mathbf{s}_{t-1}, \mathbf{c}_t])) \tag{6.9}$$

where $\mathbf{s}_t$ is the hidden state of the decoder RNN at time step t.

6.2.3.4 Train and Decode

Given the ground-truth response y^* to a conversation context u_i with facts $\mathcal{F}$, the training objective is to minimize the negative log-likelihood over all the training data $\mathcal{L}_g$ in Eq. (6.10).

$$\mathcal{L}_g = -\frac{1}{|\mathcal{U}|} \sum_{y^*, u_i, \mathcal{F}} \log p(y^*|u_i, \mathcal{F}) \tag{6.10}$$

During prediction, we use beam search to generate response candidates and perform length normalization by dividing the output log-likelihood score with the length of generated sequences to add penalty on short generated sequences.

6.2.4 Retrieval Module

The retrieval module retrieves a set of response candidates from the historical conversation context-response repository. It adopts a "context-context match" approach to retrieve a few response candidates. We first index all context/ response pairs in

the training data with Lucene [2]. Then for each conversation context u_i, we match it with the "conversation context" text field in the index with BM25. We return the "response" text field of top K ranked context/ response pairs as the retrieved response candidates[3]. We would like to keep the retrieval module simple and efficient. The re-ranking process of response candidates will be performed in the hybrid ranking module as presented in Section 6.2.5.

6.2.5 Hybrid Ranking Module

6.2.5.1 Interaction Matching Matrix

We combine a set of generated response candidates $\mathcal{G}_i$ and a set of retrieved response candidates $\mathcal{R}_i$ as the set of all response candidates $\mathcal{Y}_i = \mathcal{G}_i \cup \mathcal{R}_i$. The hybrid ranking module re-ranks all candidates in $\mathcal{Y}_i$ to find the best one as the final system output. In our implementation, $\mathcal{G}_i$ contains one generated response and $\mathcal{R}_i$ contains K retrieved responses. We adopt a neural ranking model following the previous work (Pang et al., 2016; Yang et al., 2018). Specifically, for each conversation context u_i and response candidate $y_i^k \in \mathcal{Y}_i$, we first build an interaction matching matrix. Given y_i^k and u_i, the model looks up a global embedding dictionary to represent y_i^k and u_i as two sequences of embedding vectors $\mathbf{E}(y_i^k) = [\mathbf{y}_{i,1}^k, \mathbf{y}_{i,2}^k, \cdots, \mathbf{y}_{i,L_y}^k]$ and $\mathbf{E}(u_i) = [\mathbf{u}_{i,1}, \mathbf{u}_{i,2}, \cdots, \mathbf{u}_{i,L_u}]$, where $\mathbf{y}_{i,j}^k \in \mathbb{R}^d$, $\mathbf{u}_{i,j} \in \mathbb{R}^d$ are the embedding vectors of the j-th word in the word sequences y_i^k and u_i respectively. The model then builds an interaction matrix $\mathbf{M}$, which computes the pairwise similarity between words in y_i^k and u_i via the dot product similarity between the embedding representations. The interaction matching matrix is used as the input of a convolutional neural network (CNN) to learn important matching features, which are aggregated by the final multi-layer perceptron (MLP) to generate a matching score.

---

[2] `http://lucene.apache.org/`

[3] We set $K = 9$ in our experiments.

6.2.5.2 CNN Layers and MLP

The interaction matrices are fed into a CNN to learn high level matching patterns as features. CNN alternates convolution and max-pooling operations over these inputs. Let $\mathbf{z}^{(l,k)}$ denote the output feature map of the l-th layer and k-th kernel, the model performs convolution operations and max-pooling operations respectively in Eq. (6.11) and (6.12).

Convolution: let $r_w^{(l,k)} \times r_h^{(l,k)}$ denote the shape of the k-th convolution kernel in the l-th layer, the convolution operation can be defined as:

$$\mathbf{z}_{i,j}^{(l+1,k)} = \sigma\left(\sum_{k'=0}^{K_l-1}\sum_{s=0}^{r_w^{(l,k)}-1}\sum_{t=0}^{r_h^{(l,k)}-1} \mathbf{w}_{s,t}^{(l+1,k)} \cdot \mathbf{z}_{i+s,j+t}^{(l,k')} + b^{(l+1,k)}\right) \quad \forall l = 0, 2, 4, 6, \cdots, \tag{6.11}$$

where σ is the activation function ReLU, and $\mathbf{w}_{s,t}^{(l+1,k)}$ and $b^{(l+1,k)}$ are the parameters of the k-th kernel on the $(l+1)$-th layer to be learned. K_l is the number of kernels on the l-th layer.

Max Pooling: let $p_w^{(l,k)} \times p_h^{(l,k)}$ denote the shape of the k-th pooling kernel in the l-th layer, the max pooling operation can be defined as:

$$\mathbf{z}_{i,j}^{(l+1,k)} = \max_{0 \le s < p_w^{l+1,k}} \max_{0 \le t < p_h^{l+1,k}} \mathbf{z}_{i+s,j+t}^{(l,k)} \quad \forall l = 1, 3, 5, 7, \cdots, \tag{6.12}$$

Finally we feed the output feature representation vectors learned by CNN into a multi-layer perceptron (MLP) to calculate the final matching score $f(u_i, y_i^k)$.

6.2.5.3 Distant Supervision for Model Training

For model training, we consider a pairwise ranking learning setting. The training data consists of triples $(u_i, y_i^{k+}, y_i^{k-})$, where y_i^{k+} and y_i^{k-} denote the positive and the negative response candidate for dialog context u_i. A challenging problem here is that there is no ground truth ranking for all the candidate responses in $\mathcal{Y}_i$ given a conversation context u_i. The costs for annotating all context/ response candidates pairs

for model training would be very high. Thus, we generate training data to train the hybrid ranking module with distant supervision inspired by previous work on relation extraction (Mintz et al., 2009). Specifically we construct $\mathcal{Y}_i$ by mixing K retrieved response candidates $\{r_i^1, r_i^2, ..., r_i^K\}$ and one generated response candidate $\{g_i^1\}$. We then score these $K+1$ response candidates with metrics like BLEU/ ROUGE-L by comparing them with the ground truth responses in the training data. Finally we treat the top k' response candidates ranked by BLEU/ ROUGE-L as positive candidates and other responses as negative candidates. In this way, the training labels of response candidates can be inferred from distant supervision from the ground truth responses in the training data [4]. We perform experiments to evaluate the effectiveness of different kinds of distant supervision signals. In practice, there could be multiple appropriate and diverse responses for a given conversation context. Ideally, we need multiple reference responses for each conversation context, each for a different and relevant response. We leave generating multiple references for a conversation context for distant supervision to the future work. We have to point out that it is difficult to collect the data where each context is paired with comprehensive reference responses. Our proposed method can also be easily adapted to the scenario where we have multiple reference responses for a conversation context. Given inferred training labels, we can compute the pairwise ranking-based hinge loss, which is defined as:

$$\mathcal{L}_h = \sum_{i=1}^{I} \max(0, \epsilon - f(u_i, y_i^{k+}) + f(u_i, y_i^{k-})) + \lambda ||\Theta||_2^2 \tag{6.13}$$

where I is the total number of triples in the training data. $\lambda||\Theta||_2^2$ is the regularization term where λ denotes the regularization coefficient. ϵ denotes the margin in the hinge loss.

[4]Note that we do not have to do such inference during model testing, since we just need to use the trained ranking model to score response candidates instead of computing training loss during model testing.

Table 6.3: Statistics of experimental data used in this paper.

Items	Train	Valid	Test
# Context-response pairs	1,059,370	2,067	2,066
# Facts	43,111,643	79,950	79,915
Avg # facts per context	40.70	38.68	38.68
Avg # words per facts	17.58	17.42	17.47
Avg # words per context	16.66	17.85	17.66
Avg # words per response	11.65	15.58	15.89

6.3 Experiments

6.3.1 Data Set Description

We used the same grounded Twitter conversation data set from the study by Ghazvininejad et al. (2018). The data contains 1 million two-turn Twitter conversations. Foursquare tips[5] are used as the fact data, which is relevant to the conversation context in the Twitter data. The Twitter conversations contain entities that tie to Foursquare. Then the conversation data is associated with the fact data by identifying Twitter conversation pairs in which the first turn contained either a handle of the entity name or a hashtag that matched a handle appears in the Foursquare tip data. The validation and test sets (around 4K conversations) are created to contain responses that are informative and useful, in order to evaluate conversation systems on their ability to produce contentful responses. The statistics of data are shown in Table 6.3.

6.3.2 Experimental Setup

6.3.2.1 Competing Methods

We consider different types of methods for comparison including retrieval-based, generation-based and hybrid retrieval-generation methods as follows[6]:

[5] `https://foursquare.com/`

[6] We did not compare with (Song et al., 2018) since the code of both the state-of-the-practice IR system (Yan et al., 2016b) and the multi-seq2seq model, which are the two main components of the

- Seq2Seq: this is the standard Seq2Seq model with a conversation context encoder and a response decoder, which is the method proposed in (Vinyals and Le, 2015).

- Seq2Seq-Facts: this is the Seq2Seq model with an additional facts encoder, which is the generation module in the proposed hybrid neural conversational model.

- KNCM-MTask-R: KNCM-MTask-R is the best setting of the knowledge-grounded neural conversation model proposed in the research by Ghazvininejad et al. (2018) with multi-task learning. This system is trained with 23 million general Twitter conversation data to learn the conversation structure or backbone and 1 million grounded conversation data with associated facts from Foursquare tips. Since we used the same 1 million grounded Twitter conversation data set from this work, our experimental results are directly comparable with response generation results reported by Ghazvininejad et al. (2018).

- Retrieval: this method uses BM25 model (Robertson and Walker, 1994) to match the conversation context with conversation context/ response pairs in the historical conversation repository to find the best pair, which is the retrieval module in the proposed hybrid neural conversational model.

- HybridNCM: this is the method proposed in this paper. It contains two different variations: 1) **HybridNCM-RS** is a hybrid method by mixing generated response candidates from Seq2Seq and retrieved response candidates from the retrieval module in HybridNCM; 2) **HybridNCM-RSF** is a hybrid method by mixing generated response candidates from Seq2Seq-Facts and retrieved response candidates from the retrieval module in HybridNCM.

---

proposed ensemble model in (Song et al., 2018), is not available. The experimental data used in (Song et al., 2018) is also not available.

Table 6.4: The hyper-parameter settings in the generation-based baselines and the generation module in the proposed hybrid neural conversation model.

Models	Seq2Seq	Seq2Seq-Facts
Embedding size	512	256
# LSTM layers in encoder	2	2
# LSTM layers in decoder	2	2
LSTM hidden state size	512	256
Learning rate	0.0001	0.001
Learning rate decay	0.5	0.5
# Steps between validation	10000	5000
Patience of early stopping	10	10
Dropout	0.3	0.3

6.3.2.2 Evaluation Methodology

Following previous related work (Sordoni et al., 2015; Li et al., 2016a; Ghazvininejad et al., 2018), we use BLEU and ROUGE-L for the automatic evaluation of the generated responses. The corpus-level BLEU is known to better correlate with human judgments including conversation response generation (Galley et al., 2015) comparing with sentence-level BLEU. We also report lexical diversity as an automatic measure of informativeness and diversity. The lexical diversity metrics include Distinct-1 and Distinct-2, which are respectively the number of distinct unigrams and bigrams divided by the total number of generated words in the responses. In additional to automatic evaluation, we also perform human evaluation of the generated responses of different systems on the *appropriateness* and *informativeness* following previous works (Ghazvininejad et al., 2018).

6.3.2.3 Parameter Settings

All models are implemented with PyTorch[7] and MatchZoo[8] toolkit. Hyper-parameters are tuned with the validation data. The hyper-parameter settings in the generation-

[7] https://pytorch.org/

[8] https://github.com/NTMC-Community/MatchZoo

based baselines and the generation module in the proposed hybrid neural conversation model is shown in Table 6.4. For the hyper-parameter settings in the hybrid ranking module, we set the window size of the convolution and pooling kernels as $(6, 6)$. The number of convolution kernels is 64. The dropout rate is set as 0.5. The margin in the pairwise-ranking hinge loss is 1.0. All models are trained on a single Nvidia Titan X GPU by stochastic gradient descent with Adam (Kingma and Ba, 2014) algorithm. The initial learning rate is 0.0001. The parameters of Adam, β_1 and β_2 are 0.9 and 0.999 respectively. The batch size is 500. The maximum conversation context/ response length is 30. Word embeddings in the neural ranking model will be initialized by the pre-trained GloVe [9] word vectors and updated during the training process.

6.3.3 Evaluation Results

6.3.3.1 Automatic Evaluation

We present evaluation results over different methods on Twitter/ Foursquare data in Table 6.5. We summarize our observations as follows: (1) If we compare retrieval-based methods and HybridNCM with pure generation based methods such as Seq2Seq, Seq2Seq-Facts and KNCM-MTask-R, we find that retrieval-based methods and HybridNCM with a retrieval module achieve better performance in terms of all metrics. This verifies the competitive performance of retrieval-based methods for conversation response generation reported in previous related works (Song et al., 2018). (2) Both HybridNCM-RS and HybridHCM-RSF outperforms all the baselines including KNCM-MTask-R with multi-task learning proposed recently by by Ghazvininejad et al. (Ghazvininejad et al., 2018) under BLEU and ROUGE-L. The results demonstrate that combining both retrieved response candidates and generated response candidates could help produce better responses in conversation systems. For the two variations of HybridNCM, HybridNCM-RSF achieves better BLEU and worse

[9] `https://nlp.stanford.edu/projects/glove/`

Table 6.5: Comparison of different models over the Twitter/ Foursquare data. Numbers in bold font mean the result is the best under the metric corresponding to the column. ‡ means that the improvement from the model on that metric is statistically significant over all baseline methods with $p < 0.05$ measured by the Student's paired t-test. Note that we can only do significance test for ROUGE-L since the other metrics are corpus-level metrics.

Method	BLEU	ROUGE-L	Distinct-1	Distinct-2
Seq2Seq	0.5032	8.4432	2.36%	11.18%
Seq2Seq-Facts	0.5904	8.8291	1.91%	7.85%
KNCM-MTask-R	1.0800	\	7.08%	21.90%
Retrieval	1.2491	8.6302	**14.68%**	**58.71%**
HybridNCM-RS	1.3450	**10.4078**‡	11.30%	47.35%
HybridNCM-RSF	**1.3695**	10.3445‡	11.10%	46.01%

ROUGE-L. Overall the performances of these two variations of HybridNCM are similar to each other. One possible reason is that, the main gain over baselines comes from the retrieval module and the re-ranking process in hybrid ranking module. So the differences in the generation module do not change the results too much. (3) For lexical diversity metrics like 1-gram/ 2-gram diversity, generation-based methods are far behind retrieval-based methods and HybridNCM, even for KNCM-MTask-R with external grounded knowledge and multi-task learning. This result shows that retrieved response candidates have much better diversity comparing with generated response candidates by Seq2Seq models. Researchers have studied Maximum Mutual Information (MMI) object functions (Li et al., 2015) in neural models in order to generate more diverse responses. It would be interesting to compare MMI models with IR models for conversation response generation. We leave this study to our future work.

6.3.3.2 Human Evaluation

Automatic evaluation of response generation is still a challenging problem. To complement the automatic evaluation results, we also perform human evaluation to compare the performance of different methods following previous related works (Shang

Table 6.6: Comparison of different models with human evaluation on appropriateness. ‡ means that the improvement from the model on that metric is statistically significant over all baseline methods with $p < 0.05$ measured by the Student's paired t-test. The agreement score is evaluated by Fleiss' kappa (Fleiss et al., 1971) which is a statistical measure of inter-rater consistency. Agreement scores are comparable to previous results (0.2-0.5) as reported in (Shang et al., 2015; Song et al., 2018). Higher scores indicate higher agreement degree.

Comparison	Appropriateness				
Method	Mean	Bad (0)	Neutral (1)	Good (2)	Agreement
Seq2Seq	0.4733	61.67%	29.33%	9.00%	0.2852
Seq2Seq-Facts	0.4758	62.50%	27.42%	10.08%	0.3057
Retrieval	0.9425	34.42%	36.92%	28.67%	0.2664
HybridNCM-RS	**1.1175**‡	27.83%	32.58%	39.58%	0.3010
HybridNCM-RSF	1.0358	31.67%	33.08%	35.25%	0.2909

Table 6.7: Comparison of different models with human evaluation on informativeness. ‡ means that the improvement from the model on that metric is statistically significant over all baseline methods with $p < 0.05$ measured by the Student's paired t-test. The agreement score is evaluated by Fleiss' kappa (Fleiss et al., 1971) which is a statistical measure of inter-rater consistency. Agreement scores are comparable to previous results (0.2-0.5) as reported in (Shang et al., 2015; Song et al., 2018). Higher scores indicate higher agreement degree.

Comparison	Informativeness				
Method	Mean	Bad (0)	Neutral (1)	Good (2)	Agreement
Seq2Seq	0.2417	77.58%	20.67%	1.75%	0.4731
Seq2Seq-Facts	0.3142	70.75%	27.08%	2.17%	0.4946
Retrieval	0.8008	35.50%	48.92%	15.58%	0.3196
HybridNCM-RS	**1.0650**‡	18.42%	56.67%	24.92%	0.1911
HybridNCM-RSF	1.0292	20.42%	56.25%	23.33%	0.2248

Table 6.8: Side-by-side human evaluation results. Win/Tie/Loss are the percentages of conversation contexts a method improves, does not change, or hurts, compared with the method after "v.s." on human evaluation scores. HNCM denotes HybridNCM. Seq2Seq-F denotes Seq2Seq-Facts.

Type	Appropriateness	Informativeness
Comparision	Win/Tie/Loss	Win/Tie/Loss
HNCM-RS v.s. Seq2Seq	0.71/0.15/0.14	0.84/0.10/0.06
HNCM-RSF v.s. Seq2Seq	0.68/0.16/0.16	0.82/0.11/0.07
HNCM-RS v.s. Seq2Seq-F	0.70/0.15/0.15	0.80/0.12/0.08
HNCM-RSF v.s. Seq2Seq-F	0.65/0.19/0.17	0.77/0.15/0.09
HNCM-RS v.s. Retrieval	0.43/0.31/0.26	0.50/0.31/0.18
HNCM-RSF v.s. Retrieval	0.41/0.30/0.29	0.50/0.28/0.22

et al., 2015; Ghazvininejad et al., 2018; Song et al., 2018). We ask three educated annotators to do the human evaluation. We randomly sample 400 conversation contexts from the test data, and instruct the annotators to rate the output responses of different systems.[10] We hide the system ids and randomly permute the output responses to rule out human bias. In the annotation guidelines, we ask the annotators to evaluate the quality of output responses by different systems from the following 2 dimensions:

- *Appropriateness*: evaluate whether the output response is appropriate and relevant to the given conversation context.
- *Informativeness*: evaluate whether the output response can provide useful and factual information for the users.

Three different labels "0" (bad), "+1" (neural), "+2" (good) are used to evaluate the quality of system output responses. Table 6.6 and Table 6.7 show the compari-

[10]We mainly performed human evaluation on our methods and three baselines Seq2Seq, Seq2Seq-Facts and Retrieval. We didn't include KNCM-MTask-R into human evaluation since there is no open source code or official implementation from (Ghazvininejad et al., 2018). The results of KNCM-MTask-R in Table 6.5 are cited numbers from (Ghazvininejad et al., 2018) since we used the same experimental data sets.

son of different models with human evaluation. The table contains the mean score, ratio of three different categories of labels and the agreement scores among three annotators. The agreement score is evaluated by Fleiss' kappa (Fleiss et al., 1971) which is a statistical measure of inter-rater consistency. Most agreement scores are in the range from 0.2 to 0.5, which can be interpreted as "fair agreement" or "moderate agreement" [11]. The annotators have relative higher agreement scores for the informativeness of generation-based methods like Seq2Seq and Seq2Seq-Facts, since these methods are likely to generate short responses or even responses containing fluency and grammatical problems.

We summarize our observations on the human evaluation results in Table 6.6 and Table 6.7 as follows: (1) For the mean scores, we can see both HybridNCM-RS and HybridNCM-RSF achieve higher average rating scores compared with all baselines, in terms of both appropriateness and informativeness. These results from human evaluation verify that hybrid models could help improve the response generation performances of conversation systems. For baselines, the retrieval-based baseline is stronger than generation-based baselines. For HybridNCM-RS and HybridNCM-RSF, HybridNCM-RS achieves relatively higher average human rating scores with a small gap. (2) For the ratios of different categories of labels, we can see more than 72% of output responses by HybridNCM-RS (68% for HybridNCM-RSF) are labeled as "good (+2)" or "neural (+1)" for appropriateness, which means that most output responses of hybrid models are semantically relevant to the conversation contexts. Generation-based methods like Seq2Seq and Seq2Seq-Facts perform worse than both the retrieval-based method and hybrid models. The retrieval-based method, although quite simple, achieves much higher ratios for the categories "good (+2)" and "neural (+1)" compared with generation-based methods. For informativeness, the

[11] `https://en.wikipedia.org/wiki/Fleiss%27_kappa` (as of April 1st, 2019).

Table 6.9: The number and percentage of top responses selected by the hybrid ranking module from retrieved/ generated response candidates. #PickedGenRes is the number of selected responses from generated response candidates. #PickedRetRes is the number of selected responses from retrieved response candidates. #PickedTop1BM25 is the number of selected responses which is also ranked as top 1 responses by BM25.

Item	HybridNCM-RS		HybridNCM-RSF	
#TestQNum	2066	100.00%	2066	100.00%
#PickedGenRes	179	8.66%	275	13.31%
#PickedRetRes	1887	91.34%	1791	86.69%
#PickedTop1BM25	279	13.50%	253	12.25%

Table 6.10: The response generation performance when we vary the ratios of positive samples in distant supervision.

	Supervision	BLEU-1		BLEU-2		ROUGE-L	
Model	# Positive	BLEU	ROUGE-L	BLEU	ROUGE-L	BLEU	ROUGE-L
HybridNCM-RS	k'=1	0.9022	8.9596	0.7547	8.8351	1.0964	8.9234
	k'=2	1.0649	9.7241	1.1099	9.9168	1.1019	9.6216
	k'=3	**1.3450**	**10.4078**	**1.1165**	**10.1584**	**1.1435**	**10.0928**
HybridNCM-RSF	k'=1	1.0223	9.2996	**1.1027**	9.2453	1.0035	9.2812
	k'=2	1.3284	9.8637	1.0175	9.8562	**1.0999**	**9.8061**
	k'=3	**1.3695**	**10.3445**	0.8239	**9.8575**	0.9838	9.7961

hybrid models HybridNCM-RS and HybridNCM-RSF are still the best, beating both generation-based baselines and retrieval-based baselines. These results show that the re-ranking process in the hybrid ranking module trained with distant supervision in hybrid conversation models can further increase the informativeness of results by promoting response candidates with more factual content. (3) For the statistical significance test, both HybridNCM-RS and HybridNCM-RSF outperform all baseline methods with $p < 0.05$ measured by the Student's paired t-test in terms of human evaluation scores. We also show the side-by-side human evaluation results in Table 6.8. The results clearly confirm that performances of hybrid models are better than or comparable to the performances of all baselines for most test conversation contexts.

6.3.4 Analysis of Top Responses Selected by Re-ranker

The number and percentage of top responses selected from retrieved/ generated response candidates by the neural ranking model are shown in Table 6.9. We summa-

Table 6.11: The response generation performance when we vary different distant supervision signals. This table shows the results for the setting "k'=3", where there are 3 positive response candidates for each conversation context. "SentBLEU" denotes using sentence-level BLEU scores as distant supervision signals.

Model	HybridNCM-RS		HybridNCM-RSF	
Supervision	BLEU	ROUGE-L	BLEU	ROUGE-L
BLEU-1	**1.3450**	**10.4078**	**1.3695**	**10.3445**
BLEU-2	1.1165	10.1584	0.8239	9.8575
ROUGE-L	1.1435	10.0928	0.9838	9.7961
SentBLEU	0.8326	9.2887	1.0631	9.6338

rize our observation as follows: (1) most picked results (91.34% for HybridNCM-RS and 86.69% for HybridNCM-RSF) are from the retrieved response candidates. This is reasonable because we have multiple retrieved response candidates but only one generated response candidate. In some cases, generated responses are preferred to retrieved responses. (2) Although the percentage of generated responses is not high, this does not mean we can just directly use the results returned by the retrieval method. If we look at the row "PickedTop1BM25", we can find that only very few responses ranked as the 1st by BM25 are ranked as the 1st again by HybridNCM. Thus, HybridNCM changed the order of these responses candidates significantly. In particular, the hybrid ranking module in HybridNCM did the following two tasks: a) re-evaluate and re-rank the previous generated/ retrieved responses to promote the good response; b) try to inject some generated responses by Seq2Seq models into retrieved results if possible. (3) We notice that response candidates generated by Seq2Seq-Facts model are more likely to be picked compared to those generated by Seq2Seq. When a generated response contains rich factual content, the hybrid ranking module is more likely to pick it, which also helps boost the BLEU metrics.

6.3.5 Impact of Distant Supervision Signals

We investigate the impact of different distant supervision signals on the response generation performance in Table 6.11. We find that distant supervision signals like BLEU-1 is quite effective for training the hybrid ranking module. The sentence-level BLEU is not a good choice for the distant supervision signal. The reason is that the sentence-level BLEU is computed only based on the n-gram precision statistics for a given sentence pair. This score has a larger variance compared with the corpus-level BLEU. Since sentence-level BLEU scores would become very small smoothed values if there are no 4-gram or trigram matches between two sentences, which may happen frequently in short text pairs.

6.3.6 Impact of Ratios of Positive Samples

We further analyze the impact of the ratios of positive/ negative training samples on the response generation performance. Table 6.10 shows the results. The value of k' is the number of positive response candidates for each conversation context when we train the hybrid ranking module. When $k' = 1$, we select one positive candidate from the ground truth responses in the training data, which is equivalent to the negative sampling technique. As k' increases and $k' > 1$, we construct the positive candidates by selecting one positive sample from the ground truth responses and $k' - 1$ positive samples from the top ranked candidates by distant supervision. We find that larger k' can improve the response generation performance. This is reasonable since larger k' means the model can observe more positive training samples and positive/ negative response pairs in the pairwise ranking loss minimization process. However, increasing the value of k' also adds risks of introducing noisy positive training data. Thus, there is a trade-off for choices of values of k' in the practice of training with distant supervision.

Table 6.12: Examples of output responses by different methods. r means the response is retrieved. g means the response is generated. Entities marked with [ENTITY] have been anonymized to avoid potentially negative publicity. "HNCM" denotes "Hybrid-NCM".

Context	Donated to the [ENTITY] last night and now I have to listen to automated phone calls. It's enough to make me want to cancel.	
Method	r/g	System Output Response
Ground Truth	-	Ask them to put you on their internal dnc list. They will likely respect this, because future calls can get them charged.
Seq2Seq	g	I didn't get it. I didn't.
Seq2Seq-Facts	g	I'm sorry to hear that. Please dm us your email address so we can look into this. Thanks!
Retrieval	r	It's a known issue in LA county. I just got an email from my dm and tech and it should be good in 30 minutes or so.
HNCM-RS	r	We're listening and would like to know more and help with your experience. Please follow us so i can dm you our contact info. [ENTITY]
HNCM-RSF	g	We're sorry to hear this. Please dm us if you need assistance. Please dm us your contact info so we can look into this.

6.3.7 Examples and Case Study

We perform a case study in Table 6.12 on the outputs by different methods. In this example, we can find that the response produced by Seq2Seq is very general and it does not provide any useful information for the user. Seq2Seq-Facts generates a much better response by injecting more factual content into response generation process. The response returned by the Retrieval method is also relevant to the context. However, it provides very specific information like "LA county", "30 minutes", which may have negative impact on the appropriateness of this response for some users. The responses produced by hybrid models achieve a good balance between specificity and generalization. The response by HybridNCM-RS is from retrieved results and the response by HybridNCM-RSF is from generated results, which shows that both retrieval-based methods and generation-based methods have the capacity to produce good responses for certain context.

6.4 Summary

In this chapter, we perform a comparative study of retrieval-based methods and generation-based methods for building conversation systems. We propose a hybrid neural conversation model with the capability of both response retrieval and generation in order to combine the merits of these two types of methods. For the training of the hybrid ranking module, we propose a distant supervision approach to automatically infer labels for retrieved/ generated response candidates. Experimental results with the Twitter/ Foursquare data show that the proposed model can outperform both retrieval-based methods and generation-based methods including a recently proposed knowledge-grounded neural conversation model under both automatic evaluation and human evaluation. Our research findings provide insights on how to integrate text retrieval and text generation models for building conversation systems.

CHAPTER 7

CONCLUSIONS AND FUTURE WORK

7.1 Closing Remarks

In this dissertation, we investigated several aspects of single-turn answer retrieval and multi-turn information-seeking conversations to handle the new challenges of search on mobile Internet. In order to better satisfy the information needs of mobile Internet users who usually interact with mobile devices with a touch screen or a conversational interface, we studied effective methods to retrieve answers and perform information-seeking conversations. Many proposed methods in this dissertation are built on top of the recent advances of deep neural ranking and matching models. We started from the investigation of single-turn answer retrieval and analyzed the weaknesses of existing deep learning architectures for answer ranking. Then we proposed an attention based neural matching model with value-shared weighting scheme and attention mechanism for answer retrieval to improve existing deep neural answer ranking models. For multi-turn information-seeking conversations, we investigated a learning framework on top of deep neural matching networks that leverage external knowledge for response ranking. We also studied how to integrate user intent modeling into neural ranking models for response retrieval in information-seeking conversations. Finally, hybrid models of response retrieval and generation are also investigated in order to combine the merits of these two different paradigms of conversation models.

In Chapter 3, we analyzed existing deep learning approaches to automatically learn semantic matches between questions and answers. We found that existing deep models, either based on CNNs or LSTMs, need to be combined with additional fea-

tures such as word overlap features and BM25 to perform well. Without combining these additional features, their performance is significantly worse than the results obtained by the state-of-the-art methods based on linguistic feature engineering (Yih et al., 2013). This led us to develop a new deep learning model which can achieve comparable or even better performance than methods using feature engineering without additional features. We proposed an attention based neural matching model (aNMM) for answer retrieval. aNMM introduced a novel value-shared weighting scheme in deep neural networks as a counterpart of the position-shared weighting scheme in CNNs, based on the idea that semantic matching between a question and answer is mainly about the (semantic similarity) value regularities rather than spatial regularities. We also incorporated the attention scheme over the question terms using a gating function, so that we can explicitly discriminate question term importance. We evaluated the proposed model with the TREC QA dataset, which is one of the most widely used benchmarks for answer re-ranking. Our model can achieve better performance than a state-of-art method using linguistic feature engineering and comparable performance with previous deep learning models with combined additional features. If we combine our model with a simple additional feature like QL, our method can achieve the state-of-the-art performance for answer sentence retrieval.

In Chapter 4, we studied response retrieval in multi-turn information-seeking conversations beyond single-turn interactions. Most research on response selection in conversation systems model the matching patterns between user input (either with context or not) and response candidates, which ignores external knowledge beyond the dialog utterances. Similar to Web search, information-seeking conversations can be associated with massive external data collections that contain rich knowledge. We proposed a learning framework on top of deep neural matching networks that leverages external knowledge for response ranking in information-seeking conversation systems. We studied two different methods on integrating external knowledge into deep neural

matching networks with pseudo-relevance feedback and QA correspondence knowledge distillation. Inspired by the key idea of PRF, we proposed using the candidate response as a query to run a retrieval round on a large external collection. Then we extracted useful information from the (pseudo) relevant feedback documents to enrich the original candidate response representation. On the other hand, we also proposed to extract the "correspondence" regularities between question and answer terms from retrieved external QA pairs, which will be incorporated into deep matching networks as external knowledge to help response selection. Evaluation results on two benchmark conversation data sets and one commercial customer service data from Alibaba showed that, our methods outperformed all baseline methods including various deep text matching models and the state-of-the-art baseline on response selection in multi-turn conversations.

In Chapter 5, we investigated response retrieval in information-seeking conversations from a different perspective by looking at user intent in conversations. User intent transition patterns can be useful for conversation models to select good responses given conversation contexts. Different user intent types were defined and characterized following previous research (Qu et al., 2018, 2019). Then we proposed an intent-aware neural ranking model for response retrieval, which was built on the top of the recent breakthroughs with natural language representation learning with the Transformer (Vaswani et al., 2017; Devlin et al., 2018). We referred to the proposed model as "**IART**", which is "**I**ntent-**A**ware **R**anking with **T**ransformers". IART incorporates intent-aware utterance attention to derive the importance weighting scheme of utterances in conversation context towards better conversation history understanding. We conducted extensive experiments with three information-seeking conversation data sets. Experimental results showed our methods outperformed all baselines. We also performed visualization and deep analysis of learned user intent in information-seeking conversations to provide insights.

As we have said, there are two main paradigms to produce responses given conversation inputs from users: generation-based models and retrieval-based models. In Chapter 6, we performed a comparative study of retrieval-based models and generation-based models for building conversation systems. We found that both have pros and cons. Although retrieval-based models can return natural human utterances which are controllable and explainable, the performance of retrieval-based methods is limited by the size of the conversational history repository. On the other hand, the generation-based models can generate highly coherent new responses given the conversation context, but they are likely to generate very general or universal responses with insufficient information such as "I don't know". The generated responses may also contain grammar errors. Thus it is necessary to integrate the merits of these two different types of methods to let them complement each other. We proposed a hybrid neural conversational model with a generation module, a retrieval module and a hybrid ranking module. To construct the training data of the neural ranker for response selection, we proposed a distant supervision approach to automatically infer labels for retrieved/ generated response candidates. Experimental results showed that the proposed model can outperform both retrieval-based models and generation-based models for both automatic evaluation and human evaluation. We also performed qualitative analysis on top responses selected by the neural re-ranker and response generation examples to provide insights.

7.2 Future Work

Here we discuss the future directions for our work on single-turn answer retrieval and multi-turn information-seeking conversations.

7.2.1 On Single-Turn Answer Retrieval

In Chapter 3, we studied answer retrieval with an attention-based neural matching model. There are several directions to extend our work : (1) Investigate the effectiveness of attention-based neural matching models on non-factoid question answering data sets. Since TREC QA only contains factoid questions with short answers like noun phrases and named entities, it is interesting to study whether the performance gains can also be achieved on non-factoid QA data sets like WikiPassageQA (Cohen et al., 2018) and WebAP (Keikha et al., 2014a). (2) Investigate how to improve the recall in the first round retrieval. Most existing neural ranking models follow a two stage approach which includes the first round retrieval to recall answer candidates and the second round retrieval to re-rank answer candidates. Without correct answer candidates in the first round, the neural ranking models would also fail in the second re-ranking phrase. Thus it is interesting to study what the important factors on the first round retrieval are and how to improve the recall for the retrieval performance in the first round. (3) Another related direction for extension is machine reading comprehension (Rajpurkar et al., 2016), which focuses on accurately identifying the answer span given a question and a passage/ document to search. It is interesting to explore how to find answers with a "Retriever + Reader" (Chen et al., 2017) framework, which is a promising way to show direct answers given a large unstructured text collection.

7.2.2 On Response Retrieval with External Knowledge

In Chapter 4, we proposed a deep matching model with the integration of external knowledge for response ranking in information-seeking conversations. One possible extension of this work is looking at the user intent in information-seeking conversations and studying how to improve response retrieval with user intent modeling, which has been done in Chapter 5. Another direction is to study how to incorpo-

rate both structured and unstructured knowledge into deep matching networks for response ranking. The models proposed in Chapter 4 rely on extracted knowledge from unstructured external text collections with pseudo-relevance feedback and QA correspondence knowledge distillation. However, for some questions like factoid questions about a specific named entity, it could be easier to find answer responses from pre-constructed knowledge bases. It is interesting to study how to integrate both of these two types of external knowledge to produce responses given the diversity of various conversation contexts.

7.2.3 On User Intent Modeling for Response Retrieval

For intent-aware response ranking models, there are also many interesting directions to extend our work: (1) Study a more fine-grained user intent taxonomy to capture user intent in more detail. Currently we defined 12 different types of user intent in information-seeking conversations, which are related to questions, answers, feedback, etc. in conversations. In some applications, more fine-grained classification is desired. Let's take customer service chat bots in eCommerce websites as an example. To learn a better semantic representation of user intent, we can classify the customer questions by shopping procedures like pre-sale consulting, new orders creation, payments, shipping, return/refund, etc. We can also group customer intent by different domains of products like electronics, sports, beauty, food, clothing & shoes, etc. In some cases, multi-level hierarchical taxonomies are better to be adopted to describe user intent. How to automatically learn such an optimal user intent taxonomy for different domain is an interesting direction to explore. (2) Study intent-aware response ranking models with pre-trained language models like BERT (Devlin et al., 2018), which have shown impressive performances on a variety of tasks including machine translation, question answering and natural language inference. We are interested in investigating why such models are better if they work for response retrieval.

7.2.4 On Hybrid Models of Response Retrieval and Generation

In Chapter 6, we compared retrieval-based methods and generation-based methods for conversation modeling and studied hybrid retrieval-generation neural conversation models in order to combine the merits of these two different types of methods. Some possible extensions following our work are as follows: (1) Study reinforcement learning methods for response selection in order to directly optimize metrics like BLEU/ ROUGE. Currently the generation module and hybrid ranking module in our proposed model are trained separately. To construct the training data for the hybrid ranking module, we compared the generated/retrieved response candidates with the ground truth response candidates and classified positive/negative examples by similarity scores. The loss function of the hybrid ranking module is the pairwise ranking hinge loss, which may be not strongly correlated with the final evaluation metrics like BLEU/ ROUGE. By adopting a reinforcement learning framework, we can define reward functions based on the final evaluation metrics like BLEU/ROUGE to optimize them directly. (2) Propose a better evaluation method for response quality in conversations with reasonable costs. The evaluation of response quality in conversations is more challenging than some other tasks like evaluation in machine translation. The same conversation context can be responded to by multiple diverse responses. Such response diversity problems make it very hard to collect comprehensive reference responses given a conversation context (Gao et al., 2018). To mitigate this issue, current research on conversation response generation relies on human evaluation to judge response qualities, as we did in Chapter 6. However, fast model development and iteration only based on human evaluation is not feasible given the high human annotation costs. Thus the study of a new evaluation method on response quality with reasonable costs is also an active research area.

BIBLIOGRAPHY

Mikhail Ageev, Dmitry Lagun, and Eugene Agichtein. The answer is at your fingertips: Improving passage retrieval for web question answering with search behavior data. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing, EMNLP 2013, 18-21 October 2013, Grand Hyatt Seattle, Seattle, Washington, USA, A meeting of SIGDAT, a Special Interest Group of the ACL*, pages 1011–1021, 2013. URL http://aclweb.org/anthology/D/D13/D13-1103.pdf.

Jaime Arguello, Bogeum Choi, and Robert Capra. Factors influencing users' information requests: Medium, target, and extra-topical dimension. *ACM Transactions on Information Systems*, 36(4), July 2018.

Lei Jimmy Ba, Ryan Kiros, and Geoffrey E. Hinton. Layer normalization. *CoRR*, abs/1607.06450, 2016.

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate. *CoRR*, abs/1409.0473, 2014. URL http://arxiv.org/abs/1409.0473.

Adam Berger, Rich Caruana, David Cohn, Dayne Freitag, and Vibhu Mittal. Bridging the Lexical Chasm: Statistical Approaches to Answer-finding. In *Proceedings of the 23rd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2000. ISBN 1-58113-226-3. doi: 10.1145/345508.345576. URL http://doi.acm.org/10.1145/345508.345576.

Sumit Bhatia, Prakhar Biyani, and Prasenjit Mitra. Classifying User Messages For Managing Web Forum Data. In *WebDB '12*, 2012.

Sumit Bhatia, Prakhar Biyani, and Prasenjit Mitra. Summarizing Online Forum Discussions-Can Dialog Acts of Individual Messages Help? In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2014.

Jiang Bian, Yandong Liu, Eugene Agichtein, and Hongyuan Zha. Finding the right facts in the crowd: Factoid question answering over social media. In *Proceedings of the 17th International Conference on World Wide Web*, WWW '08, pages 467–476, New York, NY, USA, 2008. ACM. ISBN 978-1-60558-085-2. doi: 10.1145/1367497.1367561. URL http://doi.acm.org/10.1145/1367497.1367561.

Matthew W. Bilotti, Jonathan Elsas, Jaime Carbonell, and Eric Nyberg. Rank learning for factoid question answering with linguistic and semantic constraints. In *Proceedings of the 19th ACM International Conference on Information and Knowledge Management*, CIKM '10, pages 459–468, New York, NY, USA, 2010. ACM. ISBN 978-1-4503-0099-5. doi: 10.1145/1871437.1871498. URL http://doi.acm.org/10.1145/1871437.1871498.

Antoine Bordes, Y-Lan Boureau, and Jason Weston. Learning end-to-end goal-oriented dialog. *5th International Conference on Learning Representations, ICLR '17*, 2017.

Guihong Cao, Jian-Yun Nie, Jianfeng Gao, and Stephen Robertson. Selecting good expansion terms for pseudo-relevance feedback. In *Proceedings of the 31st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2008.

Snigdha Chaturvedi, Vittorio Castelli, Radu Florian, Ramesh M. Nallapati, and Hema Raghavan. Joint question clustering and relevance prediction for open domain non-factoid question answering. In *Proceedings of the 23rd International Conference on World Wide Web*, WWW '14, pages 503–514, New York, NY, USA, 2014. ACM. ISBN 978-1-4503-2744-2. doi: 10.1145/2566486.2567999. URL http://doi.acm.org/10.1145/2566486.2567999.

Danqi Chen, Adam Fisch, Jason Weston, and Antoine Bordes. Reading Wikipedia to answer open-domain questions. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 2017.

Ruey-Cheng Chen, Damiano Spina, W. Bruce Croft, Mark Sanderson, and Falk Scholer. Harnessing semantics for answer sentence retrieval. In *Proceedings of the Eighth Workshop on Exploiting Semantic Annotations in Information Retrieval*, ESAIR '15, 2015.

Junyoung Chung, Çaglar Gülçehre, KyungHyun Cho, and Yoshua Bengio. Empirical evaluation of gated recurrent neural networks on sequence modeling. *CoRR*, 2014.

Daniel Cohen and W. Bruce Croft. End to end long short term memory networks for non-factoid question answering. In *Proceedings of the 2016 ACM on International Conference on the Theory of Information Retrieval, ICTIR 2016, Newark, DE, USA, September 12- 6, 2016*, pages 143–146, 2016. doi: 10.1145/2970398.2970438. URL http://doi.acm.org/10.1145/2970398.2970438.

Daniel Cohen, Liu Yang, and W. Bruce Croft. Wikipassageqa: A benchmark collection for research on non-factoid answer passage retrieval. *CoRR*, abs/1805.03797, 2018. URL http://arxiv.org/abs/1805.03797.

Kevyn Collins-Thompson, Jamie Callan, Egidio Terra, and Charles L.A. Clarke. The effect of document retrieval quality on factoid question answering performance. In *Proceedings of the 27th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '04, pages 574–575, New York, NY, USA, 2004. ACM. ISBN 1-58113-881-4. doi: 10.1145/1008992.1009127. URL http://doi.acm.org/10.1145/1008992.1009127.

Andrés Corrada-Emmanuel and W. Bruce Croft. Answer models for question answering passage retrieval. In *Proceedings of the 27th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '04, pages 516–517, New York, NY, USA, 2004. ACM. ISBN 1-58113-881-4. doi: 10.1145/1008992.1009098. URL http://doi.acm.org/10.1145/1008992.1009098.

W. Bruce Croft, Donald Metzler, and Trevor Strohman. *Search Engines: Information Retrieval in Practice*. Addison-Wesley Publishing Company, USA, 1st edition, 2009. ISBN 0136072240, 9780136072249.

Hang Cui, Renxu Sun, Keya Li, Min-Yen Kan, and Tat-Seng Chua. Question answering passage retrieval using dependency relations. In *Proceedings of the 28th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '05, pages 400–407, New York, NY, USA, 2005. ACM. ISBN 1-59593-034-5. doi: 10.1145/1076034.1076103. URL http://doi.acm.org/10.1145/1076034.1076103.

Debajyoti Datta, Valentina Brashers, John Owen, Casey White, and Laura E. Barnes. A Deep Learning Methodology for Semantic Utterance Classification in Virtual Human Dialogue Systems. In *International Conference on Intelligent Virtual Agents*, 2016.

Mostafa Dehghani, Hamed Zamani, Aliakei Severyn, Jaap Kamps, and W. Bruce Croft. Neural ranking models with weak supervision. In *Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '17, 2017.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: pre-training of deep bidirectional transformers for language understanding. *CoRR*, abs/1810.04805, 2018. URL http://arxiv.org/abs/1810.04805.

Bhuwan Dhingra, Lihong Li, Xiujun Li, Jianfeng Gao, Yun-Nung Chen, Faisal Ahmed, and Li Deng. Towards end-to-end reinforcement learning of dialogue agents for information access. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 484–495, Vancouver, Canada, July 2017. Association for Computational Linguistics. doi: 10.18653/v1/P17-1045. URL https://www.aclweb.org/anthology/P17-1045.

Fernando Diaz and Donald Metzler. Improving the estimation of relevance models using large external corpora. In *Proceedings of the 29th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '06, pages 154–161, New York, NY, USA, 2006. ACM. ISBN 1-59593-369-7. doi: 10.1145/1148170.1148200. URL http://doi.acm.org/10.1145/1148170.1148200.

Oren Etzioni. Search needs a shake-up. *Nature*, 476(7358):25–26, August 2011. ISSN 1476-4687. doi: 10.1038/476025a. URL http://dx.doi.org/10.1038/476025a.

Joseph L. Fleiss et al. Measuring Nominal Scale Agreement Among Many Raters. *Psychological Bulletin*, 76(5):378–382, 1971.

Michel Galley, Chris Brockett, Alessandro Sordoni, Yangfeng Ji, Michael Auli, Chris Quirk, Margaret Mitchell, Jianfeng Gao, and Bill Dolan. deltableu: A discriminative metric for generation tasks with intrinsically diverse targets. *CoRR*, abs/1506.06863, 2015. URL http://arxiv.org/abs/1506.06863.

Yasser Ganjisaffar, Rich Caruana, and Cristina Lopes. Bagging gradient-boosted trees for high precision, low variance ranking models. In *SIGIR '11*, pages 85–94, New York, NY, USA, 2011. ACM. ISBN 978-1-4503-0757-4. doi: http://doi.acm.org/10.1145/2009916.2009932.

Jianfeng Gao, Patrick Pantel, Michael Gamon, Xiaodong He, and Li Deng. Modeling interestingness with deep neural networks. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, EMNLP 2014, October 25-29, 2014, Doha, Qatar, A meeting of SIGDAT, a Special Interest Group of the ACL*, pages 2–13, 2014. URL http://aclweb.org/anthology/D/D14/D14-1002.pdf.

Jianfeng Gao, Michel Galley, and Lihong Li. Neural approaches to conversational AI. *CoRR*, abs/1809.08267, 2018. URL http://arxiv.org/abs/1809.08267.

Marjan Ghazvininejad, Chris Brockett, Ming-Wei Chang, Bill Dolan, Jianfeng Gao, Wen-tau Yih, and Michel Galley. A knowledge-grounded neural conversation model. In *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, (AAAI'18)*, pages 5110–5117, 2018.

John J. Godfrey and Edward Holliman. Switchboard-1 Release 2. *Journal of Environmental Health*, 1997.

Jiafeng Guo, Yixing Fan, Qingyao Ai, and W. Bruce Croft. A deep relevance matching model for ad-hoc retrieval. In *CIKM '16*, 2016.

Jiafeng Guo, Yixing Fan, Liang Pang, Liu Yang, Qingyao Ai, Hamed Zamani, Chen Wu, W. Bruce Croft, and Xueqi Cheng. A deep look into neural ranking models for information retrieval. *CoRR*, abs/1903.06902, 2019. URL http://arxiv.org/abs/1903.06902.

Matthew Henderson. Machine learning for dialog state tracking : a review. 2015.

Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural Comput.*, 9(8), November 1997.

Chiori Hori, Julien Perez, Ryuichiro Higashinaka, Takaaki Hori, Y-Lan Boureau, Michimasa Inaba, Yuiko Tsunomori, Tetsuro Takahashi, Koichiro Yoshino, and Seokhwan Kim. Overview of the sixth dialog system technology challenge: DSTC6. *Computer Speech & Language*, 55:1–25, 2019. doi: 10.1016/j.csl.2018.09.004. URL https://doi.org/10.1016/j.csl.2018.09.004.

Baotian Hu, Zhengdong Lu, Hang Li, and Qingcai Chen. Convolutional neural network architectures for matching natural language sentences. In *NIPS '14*, pages 2042–2050, 2014.

Po-Sen Huang, Xiaodong He, Jianfeng Gao, Li Deng, Alex Acero, and Larry P. Heck. Learning deep structured semantic models for web search using clickthrough data. In *CIKM '13*, 2013.

Mohit Iyyer, Jordan Boyd-Graber, Leonardo Claudino, Richard Socher, and Hal Daumé III. A neural network for factoid question answering over paragraphs. In *EMNLP '14*, 2014.

Aaron Jaech, Hetunandan Kamisetty, Eric K. Ringger, and Charlie Clarke. Match-tensor: a deep relevance model for search. *CoRR*, abs/1701.07795, 2017. URL http://arxiv.org/abs/1701.07795.

Peter Jansen, Mihai Surdeanu, and Peter Clark. Discourse Complements Lexical Semantics for Non-factoid Answer Reranking. In *ACL'14*, pages 977–986, 2014. URL http://www.aclweb.org/anthology/P14-1092.

Zongcheng Ji, Zhengdong Lu, and Hang Li. An information retrieval approach to short text conversation. *CoRR*, abs/1408.6988, 2014.

Nal Kalchbrenner, Edward Grefenstette, and Phil Blunsom. A convolutional neural network for modelling sentences. *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics*, June 2014. URL http://goo.gl/EsQCuC.

Mostafa Keikha, Jae Hyun Park, and W. Bruce Croft. Evaluating Answer Passages Using Summarization Measures. In *SIGIR'14*, 2014a. doi: 10.1145/2600428.2609485. URL http://doi.acm.org/10.1145/2600428.2609485.

Mostafa Keikha, Jae Hyun Park, W. Bruce Croft, and Mark Sanderson. Retrieving Passages and Finding Answers. In *ADCS'14*, pages 81–84, 2014b. ISBN 978-1-4503-3000-8. doi: 10.1145/2682862.2682877. URL http://doi.acm.org/10.1145/2682862.2682877.

Tom Kenter and Maarten de Rijke. Attentive memory networks: Efficient machine reading for conversational search. In *CAIR '17*. ACM, 2017.

Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *CoRR*, abs/1412.6980, 2014.

Philipp Koehn, Franz Josef Och, and Daniel Marcu. Statistical phrase-based translation. In *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology - Volume 1*, NAACL '03, pages 48–54, Stroudsburg, PA, USA, 2003. Association for Computational Linguistics. doi: 10.3115/1073445.1073462. URL https://doi.org/10.3115/1073445.1073462.

Siwei Lai, Liheng Xu, Kang Liu, and Jun Zhao. Recurrent convolutional neural networks for text classification. In *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence, January 25-30, 2015, Austin, Texas, USA.*, pages 2267–2273, 2015. URL http://www.aaai.org/ocs/index.php/AAAI/AAAI15/paper/view/9745.

Victor Lavrenko and W. Bruce Croft. Relevance Based Language Models. In *SIGIR'01*, 2001. ISBN 1-58113-331-6. doi: 10.1145/383952.383972.

Quoc V Le and Tomas Mikolov. Distributed representations of sentences and documents. *arXiv preprint arXiv:1405.4053*, 2014.

Yann LeCun, Yoshua Bengio, and Geoffrey E. Hinton. Deep learning. *Nature*, 521 (7553):436–444, 2015. doi: 10.1038/nature14539. URL https://doi.org/10.1038/nature14539.

Feng-Lin Li, Minghui Qiu, Haiqing Chen, Xiongwei Wang, Xing Gao, Jun Huang, Juwei Ren, Zhongzhou Zhao, Weipeng Zhao, Lei Wang, Guwei Jin, and Wei Chu. *AliMe Assist* : An intelligent assistant for creating an innovative e-commerce experience. In *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management, CIKM 2017, Singapore, November 06 - 10, 2017*, pages 2495–2498, 2017. doi: 10.1145/3132847.3133169. URL https://doi.org/10.1145/3132847.3133169.

Jane Li, Scott Huffman, and Akihito Tokuda. Good abandonment in mobile and pc internet search. In *Proceedings of the 32Nd International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '09, pages 43–50, New York, NY, USA, 2009. ACM. ISBN 978-1-60558-483-6. doi: 10.1145/1571941.1571951. URL http://doi.acm.org/10.1145/1571941.1571951.

Jiwei Li, Michel Galley, Chris Brockett, Jianfeng Gao, and Bill Dolan. A diversity-promoting objective function for neural conversation models. *CoRR*, abs/1510.03055, 2015. URL http://arxiv.org/abs/1510.03055.

Jiwei Li, Michel Galley, Chris Brockett, Georgios P. Spithourakis, Jianfeng Gao, and William B. Dolan. A persona-based neural conversation model. In *ACL'16*, 2016a.

Jiwei Li, Will Monroe, Alan Ritter, Dan Jurafsky, Michel Galley, and Jianfeng Gao. Deep reinforcement learning for dialogue generation. In *EMNLP'16*, 2016b.

Jimmy Lin. An exploration of the principles underlying redundancy-based factoid question answering. *ACM Trans. Inf. Syst.*, 25(2), April 2007. ISSN 1046-8188. doi: 10.1145/1229179.1229180. URL http://doi.acm.org/10.1145/1229179.1229180.

Ryan Lowe, Nissan Pow, Iulian Serban, and Joelle Pineau. The ubuntu dialogue corpus: A large dataset for research in unstructured multi-turn dialogue systems. *CoRR*, abs/1506.08909, 2015.

Zhengdong Lu and Hang Li. A deep architecture for matching short texts. In C. J. C. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 26*, pages 1367–1375. Curran Associates, Inc., 2013.

Thang Luong, Hieu Pham, and Christopher D. Manning. Effective approaches to attention-based neural machine translation. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1412–1421, Lisbon, Portugal, September 2015. Association for Computational Linguistics. doi: 10.18653/v1/D15-1166. URL https://www.aclweb.org/anthology/D15-1166.

Yuanhua Lv and ChengXiang Zhai. A comparative study of methods for estimating query language models with pseudo feedback. In *CIKM '09*, 2009.

Dhiraj Madan and Sachindra Joshi. Finding dominant user utterances and system responses in conversations. *CoRR*, abs/1710.10609, 2017. URL http://arxiv.org/abs/1710.10609.

Mary Meeker. Internet trends 2018, 2018. URL https://www.kleinerperkins.com/perspectives/internet-trends-report-2018/.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. *CoRR*, abs/1301.3781, 2013. URL http://arxiv.org/abs/1301.3781.

Mike Mintz, Steven Bills, Rion Snow, and Daniel Jurafsky. Distant supervision for relation extraction without labeled data. In *ACL 2009, Proceedings of the 47th Annual Meeting of the Association for Computational Linguistics and the 4th International Joint Conference on Natural Language Processing of the AFNLP, 2-7 August 2009, Singapore*, pages 1003–1011, 2009. URL http://www.aclweb.org/anthology/P09-1113.

Bhaskar Mitra, Fernando Diaz, and Nick Craswell. Learning to match using local and distributed representations of text for web search. In *WWW '17*, 2017.

Andrew Olney, Max Louwerse, Eric Matthews, Johanna Marineau, Heather Hite-Mitchell, and Arthur Graesser. Utterance classification in autotutor. In *Proceedings of the HLT-NAACL 03 Workshop on Building Educational Applications Using Natural Language Processing - Volume 2*, HLT-NAACL-EDUC '03, pages 1–8, Stroudsburg, PA, USA, 2003. Association for Computational Linguistics. doi: 10.3115/1118894.1118895. URL https://doi.org/10.3115/1118894.1118895.

Gaurav Pandey, Danish Contractor, Vineet Kumar, and Sachindra Joshi. Exemplar encoder-decoder for neural conversation generation. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1329–1338, Melbourne, Australia, July 2018. Association for Computational Linguistics. URL https://www.aclweb.org/anthology/P18-1123.

Liang Pang, Yanyan Lan, Jiafeng Guo, Jun Xu, Shengxian Wan, and Xueqi Cheng. Text matching as image recognition. In *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence, February 12-17, 2016, Phoenix, Arizona, USA.*, pages 2793–2799, 2016. URL http://www.aaai.org/ocs/index.php/AAAI/AAAI16/paper/view/11895.

Jay M. Ponte and W. Bruce Croft. A language modeling approach to information retrieval. In *SIGIR '98*, 1998. ISBN 1-58113-015-5.

Minghui Qiu, Feng-Lin Li, Siyu Wang, Xing Gao, Yan Chen, Weipeng Zhao, Haiqing Chen, Jun Huang, and Wei Chu. Alime chat: A sequence to sequence and rerank based chatbot engine. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics, ACL 2017, Vancouver, Canada, July 30 - August 4, Volume 2: Short Papers*, pages 498–503, 2017. doi: 10.18653/v1/P17-2079. URL https://doi.org/10.18653/v1/P17-2079.

Xipeng Qiu and Xuanjing Huang. Convolutional neural tensor network architecture for community-based question answering. In *Proceedings of the Twenty-Fourth International Joint Conference on Artificial Intelligence, IJCAI 2015, Buenos Aires, Argentina, July 25-31, 2015*, pages 1305–1311. AAAI Press, 2015. ISBN 978-1-57735-738-4. doi: http://ijcai.org/papers15/Abstracts/IJCAI15-188.html.

Chen Qu, Liu Yang, W. Bruce Croft, Johanne R. Trippas, Yongfeng Zhang, and Minghui Qiu. Analyzing and characterizing user intent in information-seeking conversations. In *The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval, SIGIR 2018, Ann Arbor, MI, USA, July 08-12, 2018*, pages 989–992, 2018. doi: 10.1145/3209978.3210124. URL https://doi.org/10.1145/3209978.3210124.

Chen Qu, Liu Yang, W. Bruce Croft, Yongfeng Zhang, Johanne R. Trippas, and Minghui Qiu. User intent prediction in information-seeking conversations. *CoRR*, abs/1901.03489, 2019. URL http://arxiv.org/abs/1901.03489.

Filip Radlinski and Nick Craswell. A theoretical framework for conversational search. In *CHIIR '17*, pages 117–126. ACM, 2017.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. Squad: 100, 000+ questions for machine comprehension of text. *CoRR*, abs/1606.05250, 2016. URL http://arxiv.org/abs/1606.05250.

Stefan Riezler, Alexander Vasserman, Ioannis Tsochantaridis, Vibhu Mittal, and Yi Liu. Statistical Machine Translation for Query Expansion in Answer Retrieval. In *ACL'07*, 2007. URL http://www.aclweb.org/anthology/P07-1059.

Alan Ritter, Colin Cherry, and William B. Dolan. Data-driven response generation in social media. In *ACL '11*, 2011.

Stephen Robertson and Stephen Walker. Some simple effective approximations to the 2-poisson model for probabilistic weighted retrieval. In *SIGIR '94*, 1994.

J. J. Rocchio. Relevance feedback in information retrieval. In Gerard. Salton, editor, *The Smart retrieval system - experiments in automatic document processing*, pages 313–323. 1971.

Andreas Rücklé and Iryna Gurevych. End-to-end non-factoid question answering with an interactive visualization of neural attention weights. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics, ACL 2017, Vancouver, Canada, July 30 - August 4, System Demonstrations*, pages 19–24, 2017. doi: 10.18653/v1/P17-4004. URL https://doi.org/10.18653/v1/P17-4004.

Holger Schwenk, Loïc Barrault, Alexis Conneau, and Yann LeCun. Very deep convolutional networks for text classification. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics, EACL 2017, Valencia, Spain, April 3-7, 2017, Volume 1: Long Papers*, pages 1107–1116, 2017. URL https://aclanthology.info/papers/E17-1104/e17-1104.

Aliaksei Severyn and Alessandro Moschitti. Learning to rank short text pairs with convolutional deep neural networks. In *Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '15, pages 373–382, New York, NY, USA, 2015. ACM. ISBN 978-1-4503-3621-5. doi: 10.1145/2766462.2767738. URL http://doi.acm.org/10.1145/2766462.2767738.

Lifeng Shang, Zhengdong Lu, and Hang Li. Neural responding machine for short-text conversation. In *ACL '15*, 2015.

Yelong Shen, Xiaodong He, Jianfeng Gao, Li Deng, and Grégoire Mesnil. A latent semantic model with convolutional-pooling structure for information retrieval. In *Proceedings of the 23rd ACM International Conference on Conference on Information and Knowledge Management, CIKM 2014, Shanghai, China, November 3-7, 2014*, pages 101–110, 2014. doi: 10.1145/2661829.2661935. URL http://doi.acm.org/10.1145/2661829.2661935.

Sosuke Shiga, Hideo Joho, Roi Blanco, Johanne R. Trippas, and Mark Sanderson. Modelling information needs in collaborative search conversations. In *Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '17, pages 715–724, New York, NY, USA, 2017. ACM. ISBN 978-1-4503-5022-8. doi: 10.1145/3077136.3080787. URL http://doi.acm.org/10.1145/3077136.3080787.

Hongya Song, Zhaochun Ren, Shangsong Liang, Piji Li, Jun Ma, and Maarten de Rijke. Summarizing answers in non-factoid community question-answering. In *Proceedings of the Tenth ACM International Conference on Web Search and Data Mining*, WSDM '17, pages 405–414, New York, NY, USA, 2017. ACM. ISBN 978-1-4503-4675-7. doi: 10.1145/3018661.3018704. URL http://doi.acm.org/10.1145/3018661.3018704.

Yiping Song, Cheng-Te Li, Jian-Yun Nie, Ming Zhang, Dongyan Zhao, and Rui Yan. An ensemble of retrieval-based and generation-based human-computer conversation systems. In *Proceedings of the 27th International Joint Conference on Artificial Intelligence*, IJCAI'18, pages 4382–4388. AAAI Press, 2018. ISBN 978-0-9992411-2-7. URL http://dl.acm.org/citation.cfm?id=3304222.3304379.

Alessandro Sordoni, Michel Galley, Michael Auli, Chris Brockett, Yangfeng Ji, Margaret Mitchell, Jian-Yun Nie, Jianfeng Gao, and Bill Dolan. A neural network approach to context-sensitive generation of conversational responses. In *NAACL '15*, 2015.

Radu Soricut and Eric Brill. Automatic question answering using the web: Beyond the factoid. *Inf. Retr.*, 9(2):191–206, March 2006. ISSN 1386-4564. doi: 10.1007/s10791-006-7149-y. URL http://dx.doi.org/10.1007/s10791-006-7149-y.

Damiano Spina, Johanne R Trippas, Lawrence Cavedon, and Mark Sanderson. Extracting audio summaries to support effective spoken document search. *JAIST '17*, 68(9):2101–2115, 2017.

Nitish Srivastava, Geoffrey E. Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: a simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, 15(1):1929–1958, 2014.

Andreas Stolcke, Noah Coccaro, Rebecca Bates, Paul Taylor, Carol Van Ess-Dykema, Klaus Ries, Elizabeth Shriberg, Daniel Jurafsky, Rachel Martin, and Marie Meteer. Dialogue act modeling for automatic tagging and recognition of conversational speech. *Comput. Linguist.*, 26(3):339–373, September 2000. ISSN 0891-2017. doi: 10.1162/089120100561737. URL https://doi.org/10.1162/089120100561737.

Min Su and M. Basu. Gating improves neural network performance. In *Neural Networks, 2001. Proceedings. IJCNN '01. International Joint Conference on*, volume 3, pages 2159–2164 vol.3, 2001. doi: 10.1109/IJCNN.2001.938501.

Sainbayar Sukhbaatar, Arthur Szlam, Jason Weston, and Rob Fergus. End-to-end memory networks. In *NIPS'15*, 2015.

Huan Sun, Hao Ma, Wen-tau Yih, Chen-Tse Tsai, Jingjing Liu, and Ming-Wei Chang. Open domain question answering via semantic enrichment. In *Proceedings of the 24th International Conference on World Wide Web*, WWW '15, pages 1045–1055, New York, NY, USA, 2015. ACM. ISBN 978-1-4503-3469-3. doi: 10.1145/2736277.2741651. URL http://doi.acm.org/10.1145/2736277.2741651.

Mihai Surdeanu, Massimiliano Ciaramita, and Hugo Zaragoza. Learning to rank answers on large online QA collections. In *Proceedings of the 46th Annual Meeting for the Association for Computational Linguistics: Human Language Technologies (ACL-08: HLT)*, pages 719–727, 2008.

Mihai Surdeanu, Massimiliano Ciaramita, and Hugo Zaragoza. Learning to rank answers to non-factoid questions from web collections. *Comput. Linguist.*, 37(2): 351–383, June 2011. ISSN 0891-2017. doi: 10.1162/COLI_a_00051. URL http://dx.doi.org/10.1162/COLI_a_00051.

Dinoj Surendran and Gina-Anne Levow. Dialog act tagging with support vector machines and hidden markov models. In *In Proceedings of Interspeech/ICSLP*, 2006.

Ming Tan, Bing Xiang, and Bowen Zhou. Lstm-based deep learning models for non-factoid answer selection. *CoRR*, abs/1511.04108, 2015. URL http://arxiv.org/abs/1511.04108.

Chongyang Tao, Wei Wu, Can Xu, Wenpeng Hu, Dongyan Zhao, and Rui Yan. Multi-representation fusion network for multi-turn response selection in retrieval-based chatbots. In *WSDM '19*, 2019.

Stefanie Tellex, Boris Katz, Jimmy Lin, Aaron Fernandes, and Gregory Marton. Quantitative evaluation of passage retrieval algorithms for question answering. In *Proceedings of the 26th Annual International ACM SIGIR Conference on Research and Development in Informaion Retrieval*, SIGIR '03, pages 41–47, New York, NY, USA, 2003. ACM. ISBN 1-58113-646-3. doi: 10.1145/860435.860445. URL http://doi.acm.org/10.1145/860435.860445.

Paul Thomas, Daniel McDu, Mary Czerwinski, and Nick Craswell. Misc: A data set of information-seeking conversations. In *CAIR '17*. ACM, 2017.

Henry S. Thompson, Anne Anderson, Ellen Gurman Bard, Gwyneth Doherty-Sneddon, Alison Newlands, and Cathy Sotillo. The hcrc map task corpus: Natural dialogue for speech recognition. In *Proceedings of the Workshop on Human Language Technology*, HLT '93, pages 25–30, Stroudsburg, PA, USA, 1993. Association for Computational Linguistics. ISBN 1-55860-324-7. doi: 10.3115/1075671.1075677. URL https://doi.org/10.3115/1075671.1075677.

Zhiliang Tian, Rui Yan, Lili Mou, Yiping Song, Yansong Feng, and Dongyan Zhao. How to make context more useful? an empirical study on context-aware neural conversational models. In *ACL '17*, 2017.

Kateryna Tymoshenko and Alessandro Moschitti. Assessing the impact of syntactic and semantic structures for answer passages reranking. In *CIKM '15*, pages 1451–1460, New York, NY, USA, 2015. ACM. ISBN 978-1-4503-3794-6. doi: 10.1145/2806416.2806490. URL http://doi.acm.org/10.1145/2806416.2806490.

Kateryna Tymoshenko, Daniele Bonadiman, and Alessandro Moschitti. Learning to rank non-factoid answers: Comment selection in web forums. In *Proceedings of the 25th ACM International on Conference on Information and Knowledge Management*, CIKM '16, pages 2049–2052, New York, NY, USA, 2016. ACM. ISBN 978-1-4503-4073-1. doi: 10.1145/2983323.2983906. URL http://doi.acm.org/10.1145/2983323.2983906.

Svitlana Vakulenko, Ilya Markov, and Maarten de Rijke. Conversational exploratory search via interactive storytelling. In *SCAI '17*, 2017.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. Attention is all you need. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems 30*, pages 5998–6008. Curran Associates, Inc., 2017. URL http://papers.nips.cc/paper/7181-attention-is-all-you-need.pdf.

Oriol Vinyals and Quoc V. Le. A neural conversational model. *CoRR*, abs/1506.05869, 2015.

Shengxian Wan, Yanyan Lan, Jiafeng Guo, Jun Xu, Liang Pang, and Xueqi Cheng. A deep architecture for semantic matching with multiple positional sentence representations. In *AAAI'16*, pages 2835–2841, 2016.

Di Wang and Eric Nyberg. A long short-term memory model for answer sentence selection in question answering. In *ACL '15*, pages 707–712, 2015. URL http://aclweb.org/anthology/P/P15/P15-2116.pdf.

Hao Wang, Zhengdong Lu, Hang Li, and Enhong Chen. A dataset for research on short-text conversations. In *EMNLP '13*, 2013.

Mengqiu Wang, Noah A. Smith, and Teruko Mitamura. What is the Jeopardy model? a quasi-synchronous grammar for QA. In *EMNLP-CoNLL*, pages 22–32, Prague, Czech Republic, 2007. Association for Computational Linguistics.

Tsung-Hsien Wen, David Vandyke, Nikola Mrksic, Milica Gasic, Lina M Rojas-Barahona, Pei-Hao Su, Stefan Ultes, and Steve Young. A network-based end-to-end trainable task-oriented dialogue system. *EACL '17*, pages 438–449, 2017.

Jason Weston, Sumit Chopra, and Antoine Bordes. Memory networks. *CoRR*, abs/1410.3916, 2014. URL `http://arxiv.org/abs/1410.3916`.

Qiang Wu, Christopher J. Burges, Krysta M. Svore, and Jianfeng Gao. Adapting Boosting for Information Retrieval Measures. *Inf. Retr.*, 13(3):254–270, June 2010. ISSN 1386-4564. doi: 10.1007/s10791-009-9112-1.

Yu Wu, Wei Wu, Chen Xing, Ming Zhou, and Zhoujun Li. Sequential matching network: A new architecture for multi-turn response selection in retrieval-based chatbots. In *ACL '17*, 2017.

Yu Wu, Furu Wei, Shaohan Huang, Zhoujun Li, and Ming Zhou. Response generation by context-aware prototype editing. *CoRR*, abs/1806.07042, 2018. URL `http://arxiv.org/abs/1806.07042`.

Chenyan Xiong, Zhuyun Dai, Jamie Callan, Zhiyuan Liu, and Russell Power. End-to-end neural ad-hoc ranking with kernel pooling. In *Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '17, pages 55–64, New York, NY, USA, 2017. ACM. ISBN 978-1-4503-5022-8. doi: 10.1145/3077136.3080809. URL `http://doi.acm.org/10.1145/3077136.3080809`.

Zhen Xu, Bingquan Liu, Baoxun Wang, Chengjie Sun, and Xiaolong Wang. Incorporating loose-structured knowledge into LSTM with recall gate for conversation modeling. *CoRR*, 2016.

Xiaobing Xue, Jiwoon Jeon, and W. Bruce Croft. Retrieval Models for Question and Answer Archives. In *SIGIR'08*, pages 475–482, 2008. ISBN 978-1-60558-164-4. doi: 10.1145/1390334.1390416. URL `http://doi.acm.org/10.1145/1390334.1390416`.

Rui Yan, Yiping Song, and Hua Wu. Learning to respond with deep neural networks for retrieval-based human-computer conversation system. In *SIGIR'16*, pages 55–64, 2016a.

Rui Yan, Yiping Song, Xiangyang Zhou, and Hua Wu. "shall I be your chat companion?": Towards an online human-computer conversation system. In *CIKM '16*, 2016b.

Rui Yan, Dongyan Zhao, and Weinan E. Joint learning of response ranking and next utterance suggestion in human-computer conversation system. In *SIGIR '17*, 2017.

Liu Yang, Minghui Qiu, Swapna Gottipati, Feida Zhu, Jing Jiang, Huiping Sun, and Zhong Chen. Cqarank: Jointly model topics and expertise in community question answering. In *CIKM '13*, 2013.

Liu Yang, Qingyao Ai, Jiafeng Guo, and W. Bruce Croft. anmm: Ranking short answer texts with attention-based neural matching model. In *CIKM '16*, 2016a.

Liu Yang, Qingyao Ai, Damiano Spina, Ruey-Cheng Chen, Liang Pang, W. Bruce Croft, Jiafeng Guo, and Falk Scholer. Beyond factoid QA: effective methods for non-factoid answer sentence retrieval. In *Advances in Information Retrieval - 38th European Conference on IR Research, ECIR 2016, Padua, Italy, March 20-23, 2016. Proceedings*, pages 115–128, 2016b. doi: 10.1007/978-3-319-30671-1_9. URL http://dx.doi.org/10.1007/978-3-319-30671-1_9.

Liu Yang, Hamed Zamani, Yongfeng Zhang, Jiafeng Guo, and W. Bruce Croft. Neural matching models for question retrieval and next question prediction in conversation. *CoRR*, abs/1707.05409, 2017. URL http://arxiv.org/abs/1707.05409.

Liu Yang, Minghui Qiu, Chen Qu, Jiafeng Guo, Yongfeng Zhang, W. Bruce Croft, Jun Huang, and Haiqing Chen. Response ranking with deep matching networks and external knowledge in information-seeking conversation systems. In *The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval*, SIGIR '18, pages 245–254, New York, NY, USA, 2018. ACM. ISBN 978-1-4503-5657-2. doi: 10.1145/3209978.3210011. URL http://doi.acm.org/10.1145/3209978.3210011.

Yi Yang, Wen-tau Yih, and Christopher Meek. WikiQA: A challenge dataset for open-domain question answering. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, Lisbon, Portugal, September 2015. Association for Computational Linguistics.

Xuchen Yao, Benjamin Van Durme, Chris Callison-Burch, and Peter Clark. Answer extraction as sequence tagging with tree edit distance. In *Human Language Technologies: Conference of the North American Chapter of the Association of Computational Linguistics, Proceedings, June 9-14, 2013, Westin Peachtree Plaza Hotel, Atlanta, Georgia, USA*, pages 858–867, 2013. URL http://aclweb.org/anthology/N/N13/N13-1106.pdf.

Wen-tau Yih, Ming-Wei Chang, Christopher Meek, and Andrzej Pastusiak. Question answering using enhanced lexical semantic models. In *ACL '13*, pages 1744–1753, Sofia, Bulgaria, August 2013. Association for Computational Linguistics. URL http://www.aclweb.org/anthology/P13-1171.

Jun Yin, Xin Jiang, Zhengdong Lu, Lifeng Shang, Hang Li, and Xiaoming Li. Neural generative question answering. In *Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence, IJCAI 2016, New York, NY, USA, 9-15 July 2016*, pages 2972–2978, 2016. URL http://www.ijcai.org/Abstract/16/422.

Wenpeng Yin and Hinrich Schütze. Multigrancnn: An architecture for general matching of text chunks on multiple levels of granularity. In *ACL '15*, pages 63–73, Beijing, China, July 2015. Association for Computational Linguistics. URL http://www.aclweb.org/anthology/P15-1007.

Steve Young, Milica Gašić, Simon Keizer, François Mairesse, Jost Schatzmann, Blaise Thomson, and Kai Yu. The hidden information state model: A practical framework for pomdp-based spoken dialogue management. *Comput. Speech Lang.*, 24(2), April 2010.

Jianfei Yu, Minghui Qiu, Jing Jiang, Jun Huang, Shuangyong Song, Wei Chu, and Haiqing Chen. Modelling domain relationships for transfer learning on retrieval-based question answering systems in e-commerce. In *Proceedings of the Eleventh ACM International Conference on Web Search and Data Mining, WSDM 2018, Marina Del Rey, CA, USA, February 5-9, 2018*, pages 682–690, 2018. doi: 10.1145/3159652.3159685. URL https://doi.org/10.1145/3159652.3159685.

Lei Yu, Karl Moritz Hermann, Phil Blunsom, and Stephen Pulman. Deep Learning for Answer Sentence Selection. *arXiv:1412.1632 [cs]*, December 2014. URL http://arxiv.org/abs/1412.1632. arXiv: 1412.1632.

Evi Yulianti, Ruey-Cheng Chen, Falk Scholer, and Mark Sanderson. Using semantic and context features for answer summary extraction. In *Proceedings of the 21st Australasian Document Computing Symposium, ADCS 2016, Caulfield, VIC, Australia, December 5-7, 2016*, pages 81–84, 2016. URL http://dl.acm.org/citation.cfm?id=3015031.

Hamed Zamani, Javid Dadashkarimi, Azadeh Shakery, and W. Bruce Croft. Pseudo-relevance feedback based on matrix factorization. In *CIKM '16*, 2016.

Chengxiang Zhai and John Lafferty. Model-based feedback in the language modeling approach to information retrieval. In *CIKM '01*, 2001.

Ruqing Zhang, Jiafeng Guo, Yixing Fan, Yanyan Lan, Jun Xu, and Xueqi Cheng. Learning to control the specificity in neural response generation. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics, ACL 2018, Melbourne, Australia, July 15-20, 2018, Volume 1: Long Papers*, pages 1108–1117, 2018a. URL https://aclanthology.info/papers/P18-1102/p18-1102.

Xiang Zhang, Junbo Jake Zhao, and Yann LeCun. Character-level convolutional networks for text classification. In *Advances in Neural Information Processing Systems 28: Annual Conference on Neural Information Processing Systems 2015, December 7-12, 2015, Montreal, Quebec, Canada*, pages 649–657, 2015.

Xiaodong Zhang and Houfeng Wang. A joint model of intent determination and slot filling for spoken language understanding. In *Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence*, IJCAI'16, pages 2993–2999. AAAI Press, 2016. ISBN 978-1-57735-770-4. URL http://dl.acm.org/citation.cfm?id=3060832.3061040.

Yizhe Zhang, Michel Galley, Jianfeng Gao, Zhe Gan, Xiujun Li, Chris Brockett, and Bill Dolan. Generating informative and diverse conversational responses via adversarial information maximization. In *Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018, 3-8 December 2018, Montréal, Canada.*, pages 1815–1825, 2018b.

Li Zhou, Jianfeng Gao, Di Li, and Heung-Yeung Shum. The design and implementation of xiaoice, an empathetic social chatbot. *CoRR*, abs/1812.08989, 2018a. URL http://arxiv.org/abs/1812.08989.

Xiangyang Zhou, Daxiang Dong, Hua Wu, Shiqi Zhao, Dianhai Yu, Hao Tian, Xuan Liu, and Rui Yan. Multi-view response selection for human-computer conversation. In *EMNLP '16*, 2016.

Xiangyang Zhou, Lu Li, Daxiang Dong, Yi Liu, Ying Chen, Wayne Xin Zhao, Dianhai Yu, and Hua Wu. Multi-turn response selection for chatbots with deep attention matching network. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics, ACL 2018, Melbourne, Australia, July 15-20, 2018, Volume 1: Long Papers*, pages 1118–1127, 2018b. URL https://aclanthology.info/papers/P18-1103/p18-1103.