Natural language is a fundamental form of information and communication. My ultimate research goal is to design Machine Learning approaches to Natural Language Processing systems that can understand language and communicate with people in different contexts, domains, text styles, and languages. To be specific, I focus on building intelligent systems that (1) understand the meaning of language grounded in various contexts where the language is used and (2) generate effective language responses in different forms for information request and human-computer communication. To this end, during my Ph.D. study, I decomposed my long-term goal into the following research questions:

1. How can we enable machines to understand language at different levels of granularity, including entities, sentences, documents, and conversations [1, 2, 3, 4]?
2. How can we synthesize formal programs (e.g., SQL) from natural language for question answering between humans and machines [5, 6, 7, 8, 9, 10, 11]?
3. How can agents summarize the information in emails, news, and scientific articles [12, 13, 14]?
4. How can we retrieve information relevant to a user's query from documents in low-resource languages [15, 16]?

The challenges and solutions are inherently multi-disciplinary, spanning areas including Natural Language Processing, Deep Learning, Information Retrieval, Databases, and Human-Computer Interaction. The sections below present my completed projects towards the answers to each question published as 16 papers in top AI conferences, followed by my future research directions.

1. Deep Neural Modeling of Text Units. Natural language consists of text units at different granularity levels, such as entity mentions, sentences, documents, and conversations. Small text units make up larger ones to convey meaning, and the semantics of text units depend on each other. Therefore, modeling text units and their dependency relationships is fundamental for natural language understanding. In particular, I am interested in how deep learning models can help us better understand and utilize the dependency relationships among text units in different NLP tasks. I have designed end-to-end deep neural networks for (1) entity extraction and coreference resolution in documents [2], (2) sentiment analysis and text classification for sentences and documents [1], and (3) addressee and response selection for multi-turn multi-party conversations [3]. The third one is described in detail as follows.

Real-world conversations often involve more than two speakers. In the Ubuntu Internet Relay Chat channel (Ubuntu IRC), for example, one user can initiate a discussion about an Ubuntu-related technical issue, and many other users can work together to solve the problem. Such a multi-party dialog can have complex speaker interactions: at each turn, users play one of three roles (sender, addressee, observer), and those roles vary across turns. In this project, I study the problem of addressee and response selection in multi-party conversations: given a responding speaker and a dialog context, the task is to select an addressee and a response from a set of candidates for the responding speaker. The task requires modeling multi-party conversations and can be directly used to build retrieval-based dialog systems. A task example is given in Table 1.

thread id	Sender	Addressee	Utterance
1	codepython	wafflejock	thanks
1	wafflejock	codepython	yup np
2	wafflejock	theoletom	you can use sudo apt-get install packagename – reinstall, to have apt-get install reinstall some package/metapackage and redo the configuration for the program.
3	codepython	-	i installed ubuntu on a separate external drive. now when i boot into mac, the external drive does not show up as bootable. the blue light is on. any ideas?
4	Guest54977	-	hi. wondering who knows where an ubuntu backup can be retrieved from.
2	theoletom	wafflejock	it's not a program. it's a desktop environment.
4	Guest54977	-	did some searching on my system and googling, but couldn't find an answer
2	theoletom	-	be a trace of it left yet there still is.
5	releaf	-	what's your opinion on a $500 laptop that will be a dedicated ubuntu machine?
3	codepython	-	my usb stick shows up as bootable (efi) when i boot my mac. but not my external hard drive on which i just installed ubuntu. how do i make it bootable from mac hardware?
3	Jordan_U	codepython	did you install ubuntu to this external drive from a different machine?
5	Umeaboy	releaf	what country you from?
5	wafflejock		

Model Prediction	Addressee	Response
SI-RNN	⋆ releaf	⋆ there are a few ubuntu dedicated laptop providers like umeaboy is asking depends on where you are

Table 1: An example of addressee and response selection in Ubuntu IRC multi-party dialog. My SI-RNN engages in a new sub-conversation by suggesting a solution to "releaf" about Ubuntu dedicated laptops. ⋆ denotes the ground-truth.

To model the complexity of multi-party dialogues, I designed the Speaker Interaction Recurrent Neural Network (SI-RNN). SI-RNN uses its dialog encoder to maintain speaker embeddings in a role-sensitive way. Speaker embeddings are updated in different GRU-based units based on their roles (sender, addressee, observer). Furthermore, since the addressee and response are mutually dependent, SI-RNN models the conditional probability (of an addressee given the response and vice versa) and selects the addressee and response pair by maximizing the joint probability. On the Ubuntu IRC benchmark, SI-RNN significantly improves the addressee and response selection performance by 10% accuracy, particularly in complex conversations with many speakers and with responses to distant messages many turns in the past. Table 1 also shows our SI-RNN output for the task example.

2. Multi-turn Text-to-SQL Semantic Parsing. Generating SQL queries from user utterances is important to help people acquire information from databases. Such a text-to-SQL semantic parsing system bridges the data and the user through an intelligent natural language interface, greatly promoting the possibility and efficiency of information access for many users besides database experts. Furthermore, in real-world applications, users often access information in a multi-turn interaction with the system by asking a sequence of related questions. The users may explicitly refer to or omit previously mentioned entities and constraints, and may introduce refinements, additions, or substitutions to what has already been said. This requires text-to-SQL systems to process context information to synthesize correct SQL queries.

To advance the state-of-the-art in this field, I contributed to two multi-turn text-to-SQL data sets with Tao Yu and other collaborators: (1) SParC [6] for cross-domain **S**emantic **Par**sing in **C**ontext which contains 4,298 unique multi-turn question sequences, comprised of 12k+ questions annotated with SQL queries, and (2) CoSQL [7] (Figure 1) for building database-querying dialogue systems, which consists of 30k+ turns with 10k+ annotated SQL queries obtained from a Wizard-of-Oz collection of 3k dialogues. Both of them are built on top of our Spider dataset [8], the largest cross-domain context-independent text-to-SQL dataset available in the field, and thus span 200 complex databases over 138 domains. The large number of domains provide rich contextual phenomena and thematic relations between the questions, which general-purpose natural language interfaces to databases have to address. In addition, it enables us to test the generalization of the trained systems to unseen databases and domains. We are actively maintaining the datasets and leaderboards of our Text-to-SQL Challenge Series including Spider (https://yale-lily.github.io/spider), SParC (https://yale-lily.github.io/sparc), and CoSQL (https://yale-lily.github.io/cosql).

D_1 :	Database about student dormitories containing 5 tables	
Q_1 :	What are the names of all the dorms?	INFORM_SQL
S_1 :	`SELECT dorm_name FROM dorm`	
A_1 :	(Result table with many entries)	
R_1 :	This is the list of the names of all the dorms.	CONFIRM_SQL
Q_2 :	Which of those dorms have a TV lounge?	INFORM_SQL
S_2 :	`SELECT T1.dorm_name FROM dorm AS T1 JOIN has_amenity AS T2 ON T1.dormid = T2.dormid JOIN dorm_amenity AS T3 ON T2.amenid = T3.amenid WHERE T3.amenity_name = ‘TV Lounge’`	
A_2 :	(Result table with many entries)	
R_2 :	This shows the names of dorms with TV lounges.	CONFIRM_SQL
Q_3 :	What dorms have no study rooms as amenities?	AMBIGUOUS
R_3 :	Do you mean among those with TV Lounges?	CLARIFY
Q_4 :	Yes.	AFFIRM
S_4 :	`SELECT T1.dorm_name FROM dorm AS T1 JOIN has_amenity AS T2 ON T1.dormid = T2.dormid JOIN dorm_amenity AS T3 ON T2.amenid = T3.amenid WHERE T3.amenity_name = ‘TV Lounge’ EXCEPT SELECT T1.dorm_name FROM dorm AS T1 JOIN has_amenity AS T2 ON T1.dormid = T2.dormid JOIN dorm_amenity AS T3 ON T2.amenid = T3.amenid WHERE T3.amenity_name = ‘Study Room’`	
A_4 :	Fawlty Towers	
R_4 :	Fawlty Towers is the name of the dorm that has a TV lounge but not a study room as an amenity.	CONFIRM_SQL
	. . .	
Q_8 :	Thanks!	THANK_YOU
R_8 :	You are welcome.	WELCOME

Figure 1: A dialog from CoSQL. Gray boxes separate the user inputs (Q_i) querying the database (D_i) from the SQL queries (S_i), execution results (A_i), and expert responses (R_i).

Furthermore, I proposed an editing-based model for our cross-domain multi-turn text-to-SQL task [5]. Based on the observation that adjacent natural language questions are often linguistically dependent and their corresponding SQL queries tend to overlap, I utilized the interaction history by editing the previous predicted query to improve the generation quality. This editing mechanism views SQL as sequences and reuses generation results at the token level in a simple manner. It is flexible to change individual tokens and robust to error propagation. Moreover, to deal with complex table structures in different domains, I employed an utterance-table encoder and a table-aware decoder to incorporate the context of the user utterance and the table schema. Experimental results on SParC showed that by generating from the previous query, the model delivered an improvement of 7% question match accuracy and 11% interaction match accuracy over the previous state-of-the-art.

3. Text Summarization in Different Domains and Styles. Text summarization aims to produce fluent and coherent synopses covering salient information in documents. Recently, neural methods have shown promising results for both extractive and abstractive approaches. However, most work focuses on the single-document setting in the news domain and relies on huge amounts of training data, while little effort has been made on summarization in other settings or domains with limited amounts of data. To bridge the gap, my research has focused on extending neural summarization towards various domains and text styles in the following scenarios: (1) Summarizing multiple long news articles using data-efficient graph convolutional neural networks to capture sentence semantic relations and achieves state-of-the-art performance on DUC with only a few hundred documents for training [12] (co-lead with Michihiro Yasunaga), (2) Summarizing scientific articles by integrating the authors' original highlights and the article's citations that describe its impacts on the community [13] (led by Michihiro Yasunaga), and (3) Generating subject lines for personal email messages by optimizing quality estimation scores via reinforcement learning [14], which is described below in more detail.

An email message consists of two main elements: an *email subject line* and an *email body*. The subject line should tell what the email body is about and what the sender wants to convey. I proposed the task of subject line generation, which aims to automatically produce email subjects given the email body, and built the first dataset, Annotated Enron Subject Line Corpus. Table 2 shows an email body with three possible subject lines. Compared with news headline generation or news single document summarization, generating email subjects is more challenging because email subjects are generally much shorter and the dataset contains far fewer training examples. This requires systems that can summarize with a high compression ratio and can be trained in a data-efficient manner. Furthermore, the community also lacks proper metrics for this task beyond ROUGE and METEOR which simply measure n-gram overlappings with references.

Email Body: Hi All, I would be grateful if you could get to me today via email a job description for your current role. I would like to get this to the immigration attorneys so that they can finalise the paperwork in preparation for INS filing once the UBS deal is signed. Kind regards,
Subject 1: Current Job Description Needed *(COMMENT: This is good because it is both informative and succinct.)*
Subject 2: Job Description *(COMMENT: This is okay but not informative enough.)*
Subject 3: Request *(COMMENT: This is bad because it does not contain any specific information about the request.)*

Table 2: An email with three possible subject lines.

My solution involves both an evaluation metric and a data-efficient model. First, to properly evaluate the subject, I built a neural network for Email Subject Quality Estimator (ESQE) to score the quality of an email subject given the email body. Statistical analysis showed that ESQE has a higher correlation with human evaluation than metrics based on n-gram matching. Next, to summarize messages to short subjects with a high compression ratio, I combined extractive and abstractive approaches. My model first used an extractor to select multiple sentences from the input email body, capturing salient information such as named entities and dates. Then it used an abstractor to rewrite multiple selected sentences into a succinct subject line while preserving key information. Furthermore, to make use of limited amounts of training data with only about 10k examples, I used a two-stage training strategy instead of an end-to-end fashion. At the first stage, I created proxy sentence labels by checking the word overlap between the subject and the body sentence. The extractor and the abstractor are separately trained using supervised cross-entropy loss. The second step is to train the extractor in a reinforcement learning framework to directly optimize the ESQE metric. Both automatic metrics (ROUGE and ESQE) and human evaluations (Informative and Fluent) demonstrated that our method outperformed other state-of-the-art summarization models and approached human-level quality.

4. Low-Resource Cross-lingual Information Retrieval. Cross-lingual Information Retrieval (CLIR) is the task of ranking documents in one language for responsiveness to a user query in another language. As multilingual documents become more accessible, CLIR is increasingly important whenever relevant information is in other languages. Traditional CLIR systems consist of two components: machine translation and monolingual information retrieval. This approach first solves the translation problem and then performs retrieval in a monolingual setting. However, while conceptually simple, the performance of this modular approach is fundamentally limited by the quality of machine translation.

Recently, many deep neural learning-to-rank models have shown promising results in monolingual information retrieval. They learn a scoring function directly from the relevance label of query-document pairs. However, applying these techniques to CLIR has been challenging, primarily for two reasons. First, when queries and documents are in different languages, it is not clear how to measure their similarity in word embedding representation space. Furthermore, deep neural networks need a large amount of training data to achieve decent performance. Annotations are prohibitively expensive for low-resource language pairs in our cross-lingual case.

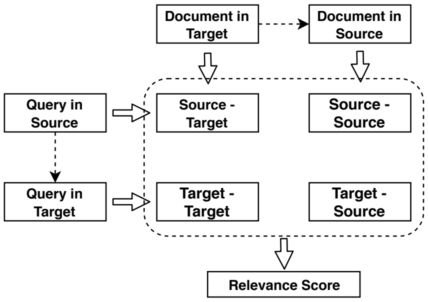

Figure 2: Deep Cross-lingual Relevance Ranking with Bilingual Query and Document Representation.

Motivated by these observations, I proposed a deep relevance ranking model to combine different translations by using cross-lingual word embeddings in a low-resource setting [15]. As shown in Figure 2, the model first translates queries and documents and then uses four components to match them in both the source and target language. The final relevance score adds up all components to combine complementary evidence from different translations. Each component is implemented as a term interaction network that learns relevance scores from a similarity matrix of each pair of a query term and a document term. To measure the similarity of two words in different languages, I built cross-lingual embeddings by aligning monolingual embeddings onto one shared space. Furthermore, to deal with small amounts of training data, I first performed query likelihood retrieval and included the score as an extra feature in the model. In this way, the model effectively learns to rerank from only a few hundred relevance labels. In addition, by aligning word embedding spaces for multiple languages, the model can be directly applied under a zero-shot transfer setting when no training data is available for another language pair. On the Material CLIR dataset with three language pairs (English to Swahili, English to Tagalog, and English to Somali), the model outperformed other translation-based query likelihood retrieval models and state-of-the-art monolingual deep relevance ranking approaches by 2%-4% mAP (mean average precision) scores.

Future Research Directions

Building upon my past work, I plan to explore the following new directions and challenges.

Multilingual Transfer Learning for Low-resource Languages. Deep neural networks still heavily rely on large amounts of in-domain labeled training data. However, only a few languages have large amounts of labeled data, and generalization in low-resource scenarios is still an open challenge. Based on my CLIR project, I would like to continue developing multilingual transfer learning techniques that can leverage annotations in high-resource languages to boost the performance of low-resource languages. I am particularly interested in models that require minimal cross-lingual supervision, leverage knowledge from multiple source languages, and transfer to the target language and task. Such an intelligent system can quickly adapt to other low-resource languages with minimal annotation cost to greatly improve the efficiency of information access for millions of people speaking different languages around the world.

Grounding Language to User Interface Actions. Mobile applications are widely used in daily life. To accomplish a task, the user interacts with an application via a sequence of low-level user interface actions, such as clicking an element, swiping to check the rest of a list, and typing a string. I am interested in building an agent that receives a high-level user instruction in natural language (e.g., "book me a flight from Boston to Seattle departing tomorrow morning.") and learns to perform a sequence of actions to accomplish the goal. This would enable people to talk to their phones in natural language to complete tasks such as booking flights or setting up calendar events, thus improving efficiency and accessibility for mobile application users. A complete representation of a mobile phone application is complex, containing structured properties (e.g., the internal tree representation of a screen and the spatial relations among elements), textual information (e.g., button descriptions), and visual elements (e.g., image icons). Moreover, the agent needs to reason about the semantics of a high-level user goal and the context of screens, and then performs the correct sequence of actions. Therefore, this presents a challenging natural language grounding and navigation problem in this diverse and open-domain platform.

Natural Language for AI Interpretability. While deep learning has become the de facto approach to build intelligent systems, the improvement of performance often comes at a cost of interpretability. Complex neural networks permit easy architectural and operational variations for state-of-the-art accuracy, yet they provide little transparency about their inner decision-making mechanisms. I am interested in how natural language can promote interpretable AI: language is not only the means of communication between humans, but it also offers a medium for an intelligent system to explain and rationalize its solutions. To this end, I would like to empower intelligent systems with abilities to automatically extract or generate human-readable language explanations to justify their predictions or actions.

Controllable and Personalized Text Generation. While deep learning models have shown promising results in text generation tasks such as summarization, translation, and dialog response generation, progress remains to be made towards controllable and personalized text generation. In particular, I would like to develop more controllable models such that (1) the generated text stays faithful to the conditioned text input, (2) we can manipulate and transfer output text attributes (such as formal vs. informal style, positive vs. negative sentiment), and (3) we remove social bias and abusive content from the generated text. Furthermore, I would also like to incorporate personal information such as gender, age, social context, and background knowledge to generate text suitable to individual users.

References

[1] **Rui Zhang**, Honglak Lee, and Dragomir R. Radev. Dependency sensitive convolutional neural networks for modeling sentences and documents. In *NAACL*, 2016.

[2] **Rui Zhang**, Cícero Nogueira dos Santos, Michihiro Yasunaga, Bing Xiang, and Dragomir Radev. Neural coreference resolution with deep biaffine attention by joint mention detection and mention clustering. In *ACL*, 2018.

[3] **Rui Zhang**, Honglak Lee, Lazaros Polymenakos, and Dragomir Radev. Addressee and response selection in multi-party conversations with speaker interaction RNNs. In *AAAI*, 2018.

[4] Catherine Finegan-Dollak, Reed Coke, **Rui Zhang**, Xiangyi Ye, and Dragomir Radev. Effects of creativity and cluster tightness on short text clustering performance. In *ACL*, 2016.

[5] **Rui Zhang**, Tao Yu, He Yang Er, Sungrok Shim, Eric Xue, Xi Victoria Lin, Tianze Shi, Caiming Xiong, Richard Socher, and Dragomir Radev. Editing-based SQL query generation for cross-domain context-dependent questions. In *EMNLP*, 2019.

[6] Tao Yu, **Rui Zhang**, Michihiro Yasunaga, Yi Chern Tan, Xi Victoria Lin, Suyi Li, Heyang Er, Irene Li, Bo Pang, Tao Chen, Emily Ji, Shreya Dixit, David Proctor, Sungrok Shim, Jonathan Kraft, Vincent Zhang, Caiming Xiong, Richard Socher, and Dragomir Radev. SParC: Cross-domain semantic parsing in context. In *ACL*, 2019.

[7] Tao Yu, **Rui Zhang**, He Yang Er, Suyi Li, Eric Xue, Bo Pang, Xi Victoria Lin, Yi Chern Tan, Tianze Shi, Zihan Li, Youxuan Jiang, Michihiro Yasunaga, Sungrok Shim, Tao Chen, Alexander Fabbri, Zifan Li, Luyao Chen, Yuwen Zhang, Shreya Dixit, Vincent Zhang, Caiming Xiong, Richard Socher, Walter Lasecki, and Dragomir Radev. CoSQL: A conversational text-to-sql challenge towards cross-domain natural language interfaces to databases. In *EMNLP*, 2019.

[8] Tao Yu, **Rui Zhang**, Kai Yang, Michihiro Yasunaga, Dongxu Wang, Zifan Li, James Ma, Irene Li, Qingning Yao, Shanelle Roman, Zilin Zhang, and Dragomir Radev. Spider: A large-scale human-labeled dataset for complex and cross-domain semantic parsing and text-to-SQL task. In *EMNLP*, 2018.

[9] Catherine Finegan-Dollak, Jonathan K. Kummerfeld, Li Zhang, Karthik Ramanathan, Sesh Sadasivam, **Rui Zhang**, and Dragomir Radev. Improving text-to-SQL evaluation methodology. In *ACL*, 2018.

[10] Tao Yu, Zifan Li, Zilin Zhang, **Rui Zhang**, and Dragomir Radev. TypeSQL: Knowledge-based type-aware neural text-to-SQL generation. In *NAACL*, 2018.

[11] Tao Yu, Michihiro Yasunaga, Kai Yang, **Rui Zhang**, Dongxu Wang, Zifan Li, and Dragomir Radev. SyntaxSQLNet: Syntax tree networks for complex and cross-domain text-to-SQL task. In *EMNLP*, 2018.

[12] Michihiro Yasunaga, **Rui Zhang**, Kshitijh Meelu, Ayush Pareek, Krishnan Srinivasan, and Dragomir Radev. Graph-based neural multi-document summarization. In *CoNLL*, 2017.

[13] Michihiro Yasunaga, Jungo Kasai, **Rui Zhang**, Alexander R Fabbri, Irene Li, Dan Friedman, and Dragomir R Radev. Scisummnet: A large annotated corpus and content-impact models for scientific paper summarization with citation networks. In *AAAI*, 2019.

[14] **Rui Zhang** and Joel Tetreault. This email could save your life: Introducing the task of email subject line generation. In *ACL*, 2019.

[15] **Rui Zhang**, Caitlin Westerfield, Sungrok Shim, Garrett Bingham, Alexander Fabbri, William Hu, Neha Verma, and Dragomir Radev. Improving low-resource cross-lingual document retrieval by reranking with deep bilingual representations. In *ACL*, 2019.

[16] Douglas W. Oard, Petra Galuščáková, Kathleen McKeown, Marine Carpuat, Mohamed Elbadrashiny, Ramy Eskander, Kenneth Heafield, Efsun Kayi, Chris Kedzie, Smaranda Muresan, Suraj Nair, Xing Niu, Dragomir Radev, Anton Ragni, Han-Chin Shing, Yan Virin, Weijia Xu, **Rui Zhang**, Elena Zotkina, Joseph Barrow, and Mark Gales. Surprise languages: Rapid-response cross-language IR. In *EVIA*, 2019.