Facilitating Information Access for Heterogeneous Data Across Many Languages

by

Peng Shi

A thesis
presented to the University of Waterloo
in fulfillment of the
thesis requirement for the degree of
Doctor of Philosophy
in
Computer Science

Waterloo, Ontario, Canada, 2023

Examining Committee Membership

The following served on the Examining Committee for this thesis. The decision of the Examining Committee is by majority vote.

Supervisor:	Jimmy Lin Professor, David R. Cheriton School of Computer Science, University of Waterloo
Internal Member:	Ming Li Professor, David R. Cheriton School of Computer Science, University of Waterloo
	Charles L. A. Clarke Professor, David R. Cheriton School of Computer Science, University of Waterloo
Internal-External Member:	Lukasz Golab Professor, Department of Management Science, University of Waterloo
External Member:	Xiaodan Zhu Associate Professor, Department of Electrical and Computer Engineering, Queen's University

Author's Declaration

This thesis consists of material all of which I authored or co-authored: see Statement of Contributions included in the thesis. This is a true copy of the thesis, including any required final revisions, as accepted by my examiners.

I understand that my thesis may be made electronically available to the public.

Statement of Contributions

This thesis consists of seven peer-reviewed publications. Peng Shi is the first author of six papers, who is responsible for the implementation, conducting experiments, and paper writing. These six papers are introduced in Chapter 3, 4, 6, 7, and 8. Peng Shi is the co-first author of the paper presented in Chapter 5, who is responsible for the implementation, experiment, and paper writing on the part of "Cross-Lingual Textual Embeddings".

List of publications:

- Peng Shi, Patrick Ng, Feng Nan, Henghui Zhu, Jun Wang, Jiarong Jiang, Alexander Hanbo Li, Rishav Chakravarti, Donald Weidner, Bing Xiang, Zhiguo Wang. Generation-focused Table-based Intermediate Pre-training for Free-form Question Answering. *In Proceedings of AAAI (Thirty-Fifth AAAI Conference on Artificial Intelligence)*, February 2022.
- Peng Shi, Patrick Ng, Zhiguo Wang, Henghui Zhu, Alexander Hanbo Li, Jun Wang, Cicero Nogueira dos Santos, Bing Xiang. Learning Contextual Representations for Semantic Parsing with Generation-Augmented Pre-Training. *In Proceedings of AAAI (Thirty-Fifth AAAI Conference on Artificial Intelligence)*, February 2021.
- Hsiu-Wei Yang*, Yanyan Zou*, Peng Shi*, Wei Lu, Jimmy Lin, Xu Sun. Aligning Cross-Lingual Entities with Multi-Aspect Information. *In Proceedings of EMNLP (Empirical Methods in Natural Language Processing)*, November, 2019.
- Peng Shi, Rui Zhang, He Bai, Jimmy Lin. Cross-Lingual Training of Dense Retrievers for Document Retrieval. *Proceedings of the 1st Workshop on Multilingual Representation Learning*, November, 2021.
- Peng Shi, He Bai, Jimmy Lin. Cross-Lingual Training of Neural Models for Document Ranking. *In Proceedings of EMNLP (Empirical Methods in Natural Language Processing): Findings*, November, 2020.
- Peng Shi, Linfeng Song, Lifeng Jin, Haitao Mi, He Bai, Jimmy Lin and Dong Yu, Cross-lingual Text-to-SQL Semantic Parsing with Representation Mixup. *Findings of EMNLP, 2022.*
- Peng Shi, Rui Zhang, He Bai and Jimmy Lin, XRICL: Cross-lingual Retrieval-Augmented In-Context Learning for Cross-lingual Text-to-SQL Semantic Parsing. *Findings of EMNLP, 2022.*

Abstract

Information access, which enables people to identify, retrieve, and use information freely and effectively, has attracted interest from academia and industry. Systems for document retrieval and question answering have helped people access information in powerful and useful ways. Recently, natural language technologies based on neural network have been applied to various tasks for information access. Specifically, transformer-based pre-trained models have pushed tasks such as document and passage retrieval to new state-of-the-art effectiveness. (1) Most of the research has focused on helping people access passages and documents on the web. However, there is abundant information stored in other formats such as semi-structured tables and domain-specific relational databases in companies. Development of the models and frameworks that support access information from these data formats is also essential. (2) Moreover, most of the advances in information access research are based on English, leaving other languages less explored. It is insufficient and inequitable in our globalized and connected world to serve only speakers of English.

In this thesis, we explore and develop models and frameworks that could alleviate the aforementioned challenges. This dissertation consists of three parts. We begin with a discussion on developing models designed for accessing data in formats other than passages and documents. We mainly focus on two data formats, namely semi-structured tables and relational databases. In the second part, we discuss methods that can enhance the user experience for non-English speakers when using information access systems. Specifically, we first introduce model development for multilingual knowledge graph integration, which can benefit many information access applications such as cross-lingual question answering systems and other knowledge-driven cross-lingual NLP applications. We further focus on multilingual document dense retrieval and reranking that boost the effectiveness of search engines for non-English information access. Last but not least, we take a step further based on the aforementioned two parts by investigating models and frameworks that can facilitate non-English speakers to access structured data. In detail, we present cross-lingual Text-to-SQL semantic parsing systems that enable non-English speakers to query relational databases with queries in their languages.

Acknowledgements

I am deeply grateful to my supervisor, Jimmy Lin, for his guidance, encouragement, and unwavering support throughout my PhD journey. His expertise and insight have been invaluable and I am truly thankful for the time and energy he dedicated to my research.

I would like to express my gratitude to Zhiguo Wang and Patrick Ng for giving me the opportunity to delve into the field of semantic parsing when I was just starting out in the area. Their guidance and support have been invaluable to me.

I would like to acknowledge the supports and contributions of my colleagues: Salman Mohammed, Michael Azmy, Jinfeng Rao, Hsiu-Wei Yang, Yanyan Zou, Yusen Zhang, Xiangyu Dong, Chang Shu, Naihao Deng, Shuaichen Chang, Tao Yu, Rui Zhang, Xinyu Zhang, Xueguang Ma, Feng Nan, Henghui Zhu, Jun Wang, Jiarong Jiang, Tianbao Xie, Chen Henry Wu, Torsten Scholak, Bailin Wang, Linfeng Song, Zhoujun Cheng, Chengzu Li, Yimu Wang, and many others. Their inputs have greatly improved the research quality and enriched my overall research experience.

I would like to express my appreciation to He Bai and Jimmy Lin for supporting me and encouraging me during the Covid-19 pandemic. Their supports were vital in helping me navigate through this challenging time.

I would like to express my sincerest gratitude to my PhD thesis committee for their guidance and feedback: Jimmy Lin, Ming Li, Charles L. A. Clarke, Lukasz Golab, and Xiaodan Zhu. Their insights have been invaluable to me.

I would like to extend my appreciation to all my friends: Luchen Tan, Haotian Zhang, Jian Li, Meixin Cheng, Wei Tu, Xiaowei Kuang, Masijia Qiu, Ji Xin, Kaisong Huang, Yuan Chen, Wei Zhong, and many others. I am deeply grateful for their presence in my life. I would also thank Tim Hortons for the companionship during the journey.

I am deeply appreciative of the guidance of Yue Zhang and Zhiyang Teng, who brought me into the world of Natural Language Processing.

In conclusion, I would like to express my heartfelt appreciation to everyone who supported me to reach this stage.

Dedication

To my parents and my partner.

Table of Contents

List of Figures

List of Tables

Chapter 1

Introduction

In this era of "information explosion",[1] the amount of data is increasing rapidly. By July 2022, the indexed Web contains at least 4.77 billion pages.[2] This large amount of data can provide knowledge and information in all aspects, benefiting people's daily life, from travel planning to business decision-making. However, finding useful and relevant information in the middle of so much data becomes challenging. To help users access the information and data they require, various research efforts have been undertaken to make the access process simple and effective, referred to as information access. Formally, based on the definition of Wikipedia, information access refers to the freedom or ability to identify, obtain and make use of databases or information effectively.[3] Information access systems are expected to understand the information needs of users, fetch relevant data from heterogeneous data sources, and display the data to users in a user-friendly way.

1.1 Challenges

Recently, the state-of-the-art natural language technologies have enabled many people to access information in powerful and useful ways, especially in document retrieval and open-domain question answering. However, there are still challenges in information access research, including the dataset construction, domain generalization, support for low-resource languages, etc. In this thesis, we discuss the following two challenges in information access research.

[1] https://en.wikipedia.org/wiki/Information_explosion

[2] https://www.worldwidewebsize.com

[3] https://en.wikipedia.org/wiki/Information_access

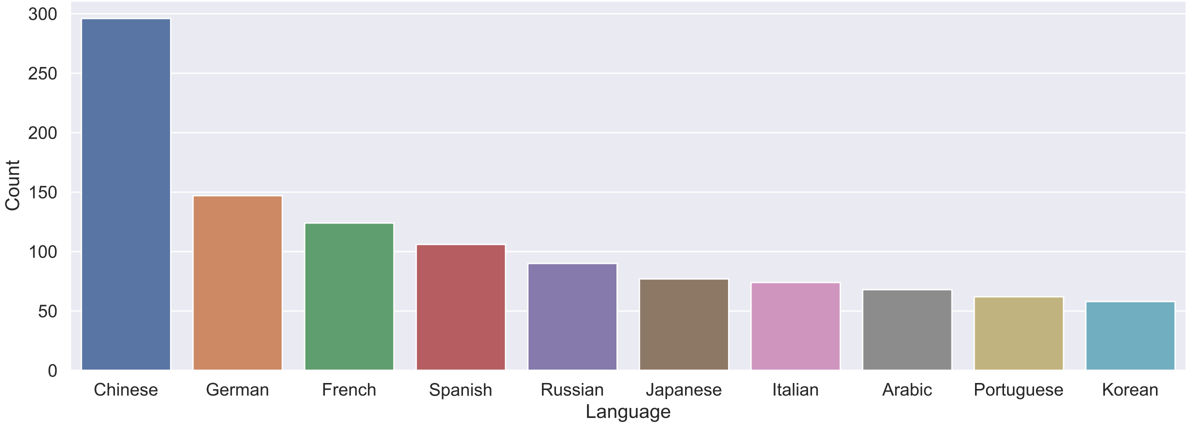


Figure 1.1: The number of public datasets in each language (Top 10 non-English languages) based on the statistics from PaperWithCode website by January 2023.

.

Data are heterogeneous. Knowledge and information can be stored in different formats, such as documents, semi-structured tables, structured databases, images, speech, and videos. However, a large portion of research in information access, such as text retrieval, focuses on fetching relevant passages or documents based on user queries. The models for accessing heterogeneous data sources such as semi-structured tables and structured databases are impotent compared to the ones for passages and documents.[4] Moreover, most of these data sources are private and domain-specific, the public search engines can not build indexes over them. To this end, it is vital to further improve the models that are tailed for these data sources, helping users to access these heterogeneous data when powerful commercial search engines are not applicable.

Support for non-English speakers. Recent years have witnessed advances in dataset and model development for information access applications. However, a large portion of these works are based on English, without further exploring other languages [460, 370]. For example, in academia, more human efforts are committed to building a large number of English datasets than those in other languages. Figure 1.1 shows the number of datasets in the top 10 languages (non-English) based on the statistics from PaperWithCode.[5] The number of datasets in English (2072 datasets) far exceeds that of datasets in these non-English languages. Systematic surveys and analysis were also conducted in [201] and [547]. Specifically, [201] classified languages into six classes, as shown in Table 1.1. They found

[4] In this thesis, heterogeneous data refers to data in different formats instead of in different languages.

[5] https://paperswithcode.com/datasets

Class	Examples of Languages	# Languages	# Speakers	% of Total Languages
0	Dahalo, Warlpiri	2191	1.2B	88.38%
1	Cherokee, Fijian	222	30M	5.49%
2	Zulu, Konkani	19	5.7M	0.36%
3	Indonesian, Ukranian	28	1.8B	4.42%
4	Russian, Vietnamese	18	2.2B	1.07%
5	English, Spanish	7	2.5B	0.28%

Table 1.1: Statistics of six classes of languages are shown in the table. The statistics include the number of languages, the number of speakers, and the percentage of total languages for each language class. The table is adopted from [201].

that Class 6 languages, especially English, have much more resources than other classes based on the dataset counts on LDC catalog,[6] ELRA Map,[7] and Web (Refer to Figure 3 in [201] for more details.) We can also observe that Class 0 has the largest percentage of languages and represents 15% of all speakers. It is insufficient and inequitable in our globalized and connected world for these to serve only speakers of English. For fulfilling the demand for information access technologies that serve diverse populations, it is essential to improve the model effectiveness in many languages, particularly in non-English scenarios. For example, an effective cross-lingual semantic parser can help people all over the world to access the US government's open data with questions in different languages.

1.2 Thesis Overview

In general, research on information access covers broad topics, such as retrieval and reranking for documents, tables, and passages [521, 167], question answering for documents [370, 235], tables [338], and knowledge graphs [298], semantic parsing for databases [542], document summarization [269], fact verification [448, 63], etc. In this thesis, we discussed the frameworks and models for alleviating the aforementioned two challenges with some of the information access applications, such as document retrieval and semantic parsing. Overall, the thesis consists of three parts.

In the first part, we explore information access applications for seeking knowledge from heterogeneous data, including web tables, and structured databases. The systems

[6] https://catalog.ldc.upenn.edu/

[7] https://catalog.elra.info/en-us/

for these applications are required to model the *interactions* between user utterances and data sources, such as semi-structured tables and database schema.[8] We first explore table-based question answering systems that can generate free-form answers for questions which require explanations with pre-trained sequence-to-sequence models. However, these pre-trained models are optimized based on long documents, leading to weaker modeling ability for the interaction of user utterances and tables; tabular data have their own structures to express the semantics, which are usually not captured by these pre-trained models. In this thesis, we demonstrate that intermediate-pretraining over large-scale tables from the web and synthesized user utterances can mitigate the aforementioned issue by enhancing encoding ability over structured data input (Chapter 3). We also explore the application of Text-to-SQL semantic parsing, which can help users to access databases with natural language questions. With a transformer-based model, we conduct pre-training for learning the joint representations of user utterance and database schema; the model is also trained with large-scale tables from the web and synthesized user utterances. The model can implicitly capture the interaction between the utterances and database schema, leading to better query understanding ability (Chapter 4).

In the second part, we explore methods and frameworks to facilitate information access systems for non-English languages. Concretely, we first investigate the application of cross-lingual entity alignment. The target of this task is to match entities in a source language with their counterparts in target languages. For example, the entity University of Toronto in an English knowledge graph is matched with entity トロント大学 in a Japanese knowledge graph. To solve this problem, we combine the Graph Neural Network and BERT-based model to encode the entities by integrating their context, such as topological connections, relation types, attributes, and literal descriptions expressed in different languages (Chapter 5), leading to better matching effectiveness. This is a component in information access systems that eases the multilingual information seeking process by interconnecting data from different languages. The integrated data can benefit cross-lingual information access models such as multilingual question answering over knowledge graphs. We also explore document retrievers that enable searching information from multilingual document collections. We start with dense retriever that represents the user queries and documents as dense vectors, and conduct the matching in dense vector space. We compare the effectiveness of several training strategies for adapting English dense retrieval models to non-English languages, known as cross-lingual training strategies. Similarly, we also investigate different cross-lingual training strategies for document reranking models (Chapter 6).

In the third part, we delve deeper by exploring models and frameworks that support

[8] In this thesis, utterance and user query are used interchangeably.

non-English speakers to access structured data such as relational databases. We propose a framework to improve parsing effectiveness by integrating the signals provided by the external translation systems. In detail, we manage to leverage the information from translations and reduce the negative influence of noisy translations at the same time (Chapter 7). More recently, in-context learning with large language models (LLMs), such as GPT-3 [42] and Codex [55], has become a new learning paradigm. Recent papers have also shown promising results of in-context learning on Text-to-SQL for English utterances. However, their parsing ability for non-English utterances remains unknown; in this thesis, we examine the parsing effectiveness of these large LLMs for non-English user queries. To further boost their performance, we explore a retrieval-reranker pipeline to obtain better exemplars for few-shot learning (Chapter 8).

1.3 Contributions

Overall, the contributions of this thesis are summarized as follows:

We improve the information access to heterogeneous data.

- We propose novel training objectives for adapting pre-trained language models to have better structured knowledge (tables or database schema) encoding abilities, achieving state-of-the-art performance on the table-based free-form question answering and Text-to-SQL semantic parsing benchmarks (Chapter 3 and Chapter 4).
- We propose to use generation models and synchronous context-free grammar to overcome the pre-training data challenges (Chapter 3 and Chapter 4). To the best of our knowledge, we are the first to use both crawled SQL and crawled tables to boost the performance of Text-to-SQL semantic parsers (Chapter 4).

We facilitate the development of information access systems for non-English speakers

- We present a BERT-based bi-encoder architecture for cross-lingual entity alignment by leveraging the literal descriptions of entities (Chapter 5). To the best of our knowledge, this is one of the earliest BERT-based bi-encoder architectures for the matching task.
- Our work is one of the earliest studies that apply multilingual BERT on non-English document reranking (Chapter 6).

We advance research that enables non-English speakers to access structured data.

- We propose a novel framework that reduces the influence of noise in machine translation, leading to state-of-the-art performance on cross-lingual Text-to-SQL semantic parsing benchmarks (Chapter 7).

- We propose a novel retrieval-reranking framework to improve the example selection process for in-context learning for cross-lingual semantic parsing. To the best of our knowledge, we are the first to explore the effectiveness of large pre-trained language models for cross-lingual Text-to-SQL semantic parsing (Chapter 8).

- We construct two new benchmarks for facilitating the cross-lingual evaluation of the Text-to-SQL semantic parsing (Chapter 8).

Chapter 2

Background

In this chapter, we provide a short introduction to the applications we explore in this thesis, including dense retrieval and reranking, question answering, semantic parsing, and cross-lingual entity matching. We also give an overview of multilingual training and cross-lingual training.

2.1 Dense Retrieval and Reranking

The goal of dense retrieval and reranking is to retrieve relevant information based on keyword queries or natural language queries to fulfill the information needs of users.

Traditional document retrieval and passage retrieval rely greatly on the term-based matching signal between the queries and documents, where the Okapi BM25 [84, 386, 387] and TF-IDF models are usually applied. However, these traditional sparse vector space models suffer from the vocabulary mismatch issue between the queries and documents, or not well representing the semantics of texts. Furthermore, these methods are unsupervised and hardly improved by leveraging more human-annotated data (neural weighting schemes [143, 566, 325] is one possible direction for improving sparse retrieval with annotation).

As the development of neural models, an alternative method is to translate the query and passages into dense vector space, and conduct the search by matching the *embeddings* with predefined distance functions, such as L1 distance or cosine similarity. These methods are denoted as *dense* retrieval. The dense retrieval models are usually implemented with dual-encoder architecture (denoted as query encoder and passage encoder depending on their model inputs), which encode the query and passage independently. This architecture is also called "Siamese" network [41]. To train the encoders, large-scale human-annotated datasets, such as passage retrieval datasets [318, 83], question answering datasets [370, 219, 200], etc., are leveraged to obtain these high-quality encoders. Compared to classical term-based retrieval models, dense retrieval models generally have a superior ability to capture the semantics of text. Recent advances in contextual pre-trained models also improve the quality of encoders significantly [208, 131, 46, 255, 443]. One example is the Dense Passage Retrieval [205], which leveraged the pre-trained language model BERT [103] as the text encoder to embed the queries and passages into dense vector space for retrieval, improving over BM25 baselines by a large margin.

After the first stage retrieval, a subsequent procedure of reranking is widely adopted to further improve the retrieval results by incorporating a reranker. Instead of using a bi-encoder architecture that is used in dense retrieval, cross-encoder architecture is leveraged for modeling the query-candidate interaction in token level [144, 283, 176, 323, 324]. This retrieve-and-rerank fashion obtains state-of-the-art performance on various datasets [384, 423].

2.2 Question Answering

The goal of question answering is to produce answers to natural language questions based on various knowledge sources.

Question answering systems try to handle numerous question types such as factoid question [370, 200, 219], choice questions [75], why-question [331, 184], etc. Based on the different types of knowledge sources, the task of question answering can be categorized into Text-based (answer questions based on the unstructured texts) [53], Table-based (answer questions based on semi-structure tables) [167], KG-based (answer questions based on the knowledge graph) [298] and hybrid (combination of different sources) [61].

With given or retrieved knowledge sources, there are two popular architectures for producing the answer: extraction-based architecture and generation-based architecture. The extraction-based architecture is now built upon on bidirectional pre-trained language model, such as BERT [103] and Roberta [273], where the answer span is predicted based on the representation of tokens. This architecture is intuitive for entity-centric factoid questions, while it is inadequate for free-form based questions.[1] More recently, sequence-

[1] Free-form based questions require the systems to generate sentence-length or paragraph-length answers that provide sufficient explanations.

to-sequence pre-trained language models such as BART [233] and T5 [366] advance the effectiveness of question answering with the generation-based architecture. This architecture unifies answers, either entity or free-form sentences, as target sequence, and the model is trained end-to-end to learn to generate answers in any format, obtaining state-of-the-art performance on different types of QA datasets [497, 191].

2.3 Semantic Parsing

The goal of semantic parsing is to interpret the natural language utterance into a formal meaning representation.

Based on the types of formal meaning representations, the semantic parsing task can be categorized into general-purpose semantic parsing (transducing natural language utterances into logical forms such as abstract meaning representation) [214, 561], and task-oriented semantic parsing (transducing natural language utterances into logical forms such as SQL) [542]; The latter is our focus. These logical forms can be further executed against databases or knowledge graphs, enabling users to access databases or knowledge graphs with natural language interfaces. This technique empowers some of the most popular commercial AI products, such as Apple Siri, Google Assistant, and Amazon Alexa.

Advances in neural networks enable the simple modeling of semantic parsing, by formulating it as a sequence-to-sequence problem [110, 111]. The simplest form of such models is the standard attention-based sequence-to-sequence LSTM model [173, 23]. The later work mainly focused on improving the encoder and decoder networks. More recently, the large-scale pre-trained language models have achieved significant improvements in many NLP tasks; the semantic parsing task also benefits from them. Instead of using word embeddings, integrating the contextual embeddings into sequence-to-sequence architectures significantly boost the model effectiveness [147, 463]; this falls in the category that improves the encoder networks. More recently, the sequence-to-sequence pre-trained language model T5 are applied on this task, achieving state-of-the-art performance on single-turn and conversational Text-to-SQL tasks [397]; both the encoder and decoder are upgraded with pre-trained language models.

2.4 Cross-lingual Entity Matching

The goal of cross-lingual entity matching is to align entities in the source language with the corresponding counterparts in target languages.

Embedding-based alignment has become the mainstream as the development of deep learning technologies. To match the entities, efforts have been committed to leverage the graph structures such as knowledge graph embedding models [58, 575, 438, 341], and graph neural networks [500, 491, 492, 439, 475]. Using side information has also been investigated in recent years. For example, [56] leveraged both multilingual entity and description embeddings for entity alignment. [447] also utilized the entity name and description for enhancing representation learning.

2.5 Multilingual Training

The multilingual training technique is commonly used in multilingual translation, where the model is trained with a mixed dataset from multiple languages [198], for example, many-to-one translation (e.g., translate Chinese, French, and German into English with one single model). When annotations are available in multiple languages, this joint training method can perform better than single model training, especially for low resource languages [241, 405].

This technique is also used for learning multilingual word representation [35, 220], contextual representation [103, 77, 272, 509], and sentence representation [519, 122]. One representative example is the large-scale pre-trained language models that use multilingual texts from over 100 languages to learn contextual representations, such as mBERT [103], XLM-Roberta [77], mBART [272], and mT5 [509]. These pre-trained language models achieved impressive performance on various cross-lingual transfer settings, e.g., zero-shot cross-lingual transfer [489].

A common problem with multilingual training is that the data from different languages are both heterogeneous (different languages may exhibit very different properties) and imbalanced (there may be wildly varying amounts of training data for each language). Thus, while low-resource languages will often benefit from transfer from other languages, for languages where sufficient monolingual data exists, performance will often decrease due to interference from the heterogeneous nature of the data. How to balance the training process becomes an open question [471, 574].

2.6 Cross-Lingual Transfer

2.6.1 Zero-shot Transfer

Zero-shot transfer means that the models work for a language without annotated data in that language. To achieve this, the models are allowed to leverage the unannotated monolingual data in that language (target language) or annotated data in other languages (source language). Using document retrieval as an example and considering English as source language while Hindi as target language, pairs of user queries and documents in Hindi without relevance labels are regarded as unannotated monolingual data. On the other hand, query-document-label triples in English are regarded as annotated data in source language. As we discussed in Section 2.5, the multilingual large-scale pre-trained language models become the backbone of zero-shot transfer technique. Based on these, several techniques are proposed to improve the model effectiveness.

Multilingual Representation Alignment. The goal of multilingual representation alignment is to constrain the embedding space of different languages into a unified space. This was originally studied for word vectors with the goal of enabling cross-lingual transfer, where the embeddings for two languages are in alignment if word translations, e.g. cat and Katze, have similar representations [292, 424]. For the contextual representation produced by pre-trained language models, [220] proposed a cross-lingual pre-training objective that uses parallel data, and inspired following work [70, 71, 47, 380], leading to improved downstream cross-lingual transfer. Adversarial training is also widely adopted to filter away language-related information and align the representation of tokens or sentences in different languages [494, 495, 65]. More specifically, a discriminator is trained to distinguish the language of the hidden states they represent, while the encoder is trained to fool the discriminator, by unifying the embedding space of different languages.

Teacher-student approach and Distillation. Teacher-student approach is related to “knowledge distillation” [171], where a student classifier is trained using the predictions of a teacher classifier. [505] applied knowledge distillation for cross-lingual text classification but required expensive parallel corpora. Instead of using expensive parallel corpora, the following work tried to use minimal resources to achieve knowledge distillation, e.g., seed word translation [204], or without any extra resources [488].

There are some other methods that pseudo training data is synthesized for the target language in zero-shot manner. We also regard these techniques as zero-shot transfer; we discussed these in Section 2.6.2.

2.6.2 Target Language Fine-tuning

Fine-tuning on target language to bridge the language gap is the most direct way for cross-lingual transfer. Pre-trained language models also achieved impressive performance in this setting. However, to make this setting possible, we need to focus on the following two issues.

How to obtain the target language data annotations automatically? One common method is "translate train", where the source language text is translated into the target language, and the labels are also required to be projected to the target language. The label transfer is easy for classification tasks such as sentence/document classification; it is more challenging for structure prediction tasks such as part of speech (POS) tagging, named entity recognition (NER), dependency parsing, and semantic role labeling (SRL), where the token alignment is hard to obtain with translators [193, 333]. Another common method is to generate labels from unlabeled data via automatic labeling functions. The automatic labeling function can be obtained via "translate train" or zero-shot manner. For example, [230] used neural machine translators to translate reading comprehension training data in the source language to the target language, and trained a question generator with these data in the target language as the automatic labeling function. [403] leverage cross-lingual question generators to generate questions for passages in the target language in zero-shot manner, producing the training corpus for multilingual reading comprehension task.

How to select training examples for language transfer? Two cases are considered for this issue. In the case where the training examples are synthesized automatically, researchers tend to select high-quality examples, via uncertainty estimation [504], back-translation [172], scoring function [230], etc. If training examples are obtained via human annotation, how to select representative data points for annotating to save budgets is the problem that requires investigation [50], especially for the tasks that need annotations from domain experts.

2.7 Terminology Note

In the methodology perspective, adjective *cross-lingual* indicates that the task is solved with cross-lingual learning/transfer. In the task definition perspective, *cross-lingual* indicates that the models are required to process the inputs that are composed of different languages, e,g, cross-lingual information retrieval where the query is in one language while the document is in another language, or cross-lingual semantic parsing where the query is in one language while the schema is in another language. On the other hand, for information retrieval, if the query and document are in the same language, we call it *mono-lingual* information retrieval. *Multilingual* is often used for systems that can deal with multiple languages. Some authors also use the term *polyglot* to refer to models that are trained multilingually [307].

Chapter 3

Free-form Question Answering over Table

In this chapter, we discuss an information access application that has received little attention: free-form question answering over table. Question answering over semi-structured tables has attracted significant attention in the NLP community. However, most of the existing work focus on questions that can be answered with short-form answer, i.e. the answer is often a table cell or aggregation of multiple cells. This can mismatch with the intents of users who want to ask more complex questions that require free-form answers such as explanations. To bridge the gap, most recently, pre-trained sequence-to-sequence language models such as T5 are used for generating free-form answers based on the question and table inputs. However, these pre-trained language models have weaker encoding abilities over table cells and schema. We propose an intermediate pre-training framework, Generation-focused Table-based Intermediate Pre-training (GenTaP), that jointly learns representations of natural language questions and tables. GenTaP learns to generate via two training objectives to enhance the question understanding and table representation abilities for complex questions. Based on experimental results, models that leverage GenTaP framework outperform the existing baselines on FeTaQA benchmark. This work is based on:

- Peng Shi, Patrick Ng, Feng Nan, Henghui Zhu, Jun Wang, Jiarong Jiang, Alexander Hanbo Li, Rishav Chakravarti, Donald Weidner, Bing Xiang, Zhiguo Wang. Generation-focused Table-based Intermediate Pre-training for Free-form Question Answering. *In Proceedings of AAAI (Thirty-Fifth AAAI Conference on Artificial Intelligence)*, February 2022.

3.1 Introduction

Question Answering (QA) [370, 217] is an important natural language processing task that enables the interactions between the users and large-scale knowledge sources. Based on the different forms of the knowledge sources, the QA task is categorized into different sub-tasks, such as Text-based QA that answer questions based on the unstructured texts, Table-based QA where semi-structure tables are the knowledge source, and Semantic Parsing where logic-form is generated to answer question from structured knowledge graphs and databases.

For Text-based QA and Table-based QA, existing work primarily focused on extracting relevant portion of the text/table to answer the question, which are usually short-form facts or entities [370, 338, 189]. However, these QA systems may not meet the needs of the users, who tend to ask more complex questions that require free-form answers (e.g. explanations) rather than short entities.[1]

Efforts have been made in addressing the shortcoming of the QA systems. For the Text-based QA, [215, 118, 217] proposed to leveraged sequence-to-sequence architectures to generate free-form answers based on the retrieved documents. However, the free-form Table-based QA remains largely unexplored. More recently, [308] used pre-trained language model T5 [366] — a sequence-to-sequence architecture — to generate long form answers from the table knowledge source.

However, the sequence-to-sequence pre-trained language models, such as BART [233] or T5 [366], have weaker encoding ability over table cells and schema. These language models usually employ long documents as the training corpus, obtaining impressive encoding ability over unstructured text. On the other hand, tabular data have their own structures to express the semantics, which are usually not captured by these language models.

Recently, several solutions are proposed for alleviating the aforementioned issue by introducing pre-training or intermediate training strategies for tables. For example, [168] proposed TAPAS that used Masked Language Model (MLM) as pre-training objective for improving the contextual representation of BERT [103] over table inputs. They showed the pre-trained model obtained state-of-the-art performance for Table-based QA where entities are extracted from the table. They achieved large improvements over the table entailment task. Albeit the improvements, these pre-training models were designed and evaluated for the short-form answer, where the answer is often a table cell or aggregation of multiple cells. Thus pre-training strategies to solve complex questions that require long-form answers remain unexplored.

[1] Complex question in our work refers to the question that requires long-form explanation to answer.

In this chapter, we present an intermediate language model pre-training framework, Generation-focused Table-based Intermediate Pre-training (GenTaP), that exploits different learning strategies, including short-form entities and long-form explanations. We demonstrate that our learning strategies enhance question understanding and table representation abilities of the pre-trained language models for complex questions. Instead of using a bidirectional contextual encoder such as BERT to exploit the potential of the text generation task, our framework is based on the BART [233] encoder-decoder architecture, which was trained with denoising training objectives. Specifically, our two different learning targets are designed for improving different aspects of the pre-trained language model, including, but not limited to, long-form answer generation augmentation (LongAug) and factual accurate answer generation augmentation (ShortAug). *LongAug* leverages table knowledge enriched long sentence as the learning target. *ShortAug* uses short entities that precisely answer the corresponding question as the target; this learning target is to improve the model's accuracy in generating key facts based on the knowledge contained in the table.

One key challenge of employing the aforementioned intermediate pre-training tasks is the training data. Although it is easy to obtain large-scale tables from web sources such as Wikipedia Tables, it is difficult to obtain the questions and answers (long form or short form) pairs that are interrelated with the tables. Recent work used the surrounding text of the tables as a proxy for related natural language utterances [168, 528]. However, this causes a mismatch between the intermediate pre-training and downstream tasks where questions are one essential component of the tasks. More recently, [415] confirmed that the surrounding text is far from optimal because those texts are dissimilar to the natural language questions in terms of text length, composition and content. The surrounding text of the tables can be quite noisy and may be irrelevant to the tables. In this work, following [415] and [116], we leverage both sequence-to-sequence generation model and synchronous context-free grammar to generate the question-answer pairs for intermediate pre-training.

The outcome of the GenTaP is a sequence-to-sequence pre-trained model that has the enhanced ability of generating long-form answers for complex questions from tabular knowledge sources. The experimental results show that the models outperform the state-of-the-art models on FeTaQA dataset. We also find that our models have transfer ability for the few-shot data-to-text generation task by outperforming existing baselines. In summary, our work shows the following contributions:

- We propose a new framework for table-based long-form answer generation that exploits two different learning targets with synthetic data.

- We leverage a novel strategy to overcome pre-training data challenges by leveraging a generation model and synchronous context-free grammar to generate synthetic data for learning joint representations of textual data and tables.

- Our pre-trained model obtains state-of-the-art performance on the table-based free-form question answering dataset FeTaQA .

- Our pre-trained model demonstrates good transfer ability by achieving better effectiveness than baselines on few-shot data-to-text (FSD2T) generation task .

3.2 Related Work

Table-based Pre-training. Recently, table-based pre-training received a lot of attention [168, 116, 415, 100, 528, 537, 183, 270]. Large scale crawled tables are used for pre-training to enhance the table representation ability of language models. Different from these work, we focus on the pre-training for free-form question answering, by leveraging the context-table alignments and question generation model.

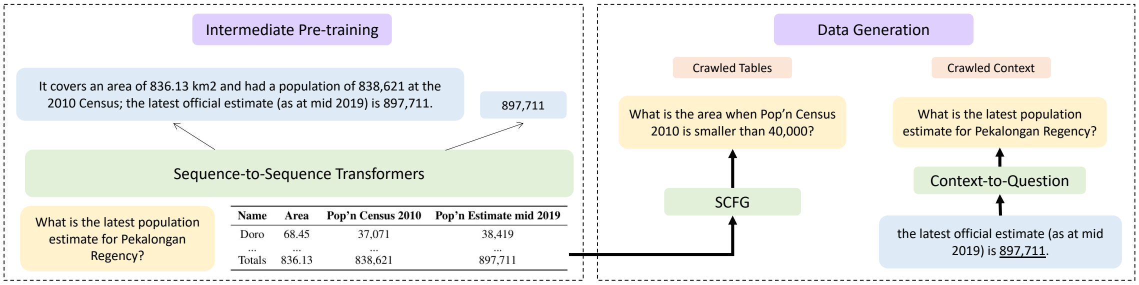
Generation-based Question Answering. By leveraging the powerful sequence-to-sequence pre-trained language model, several question answering tasks are formulated as the generation problem [233, 366, 404, 295, 191, 132, 236]. Free-form question answering have also been received increasing attention [118, 217, 308] as it can handle more complex questions. More recently, [497] unified structured knowledge based tasks (e.g. table-based question answering, semantic parsing, data-to-text generation) with sequence-to-sequence models.

Data-to-Text. Data-to-Text generation requires the model to produce precise and fluent description given the structured data input, such as tables [225, 337], triples [133, 326, 309], or logic forms [93, 501, 422]. Recently, large scale pre-trained models are actively applied on these tasks, obtaining new state of the art [385, 62, 244].

3.3 Models

3.3.1 Baseline Models

To answer complex questions based on tabular content, one of the two methods is usually exploited: pipeline model and end-to-end model. For the pipeline model, a semantic parser


Intermediate Pre-training
It covers an area of 836.13 km2 and had a population of 838,621 at the 2010 Census; the latest official estimate (as at mid 2019) is 897,711.
897,711
Sequence-to-Sequence Transformers
What is the latest population estimate for Pekalongan Regency?
Data Generation
Crawled Tables
Crawled Context
What is the area when Pop'n Census 2010 is smaller than 40,000?
What is the latest population estimate for Pekalongan Regency?
SCFG
Context-to-Question
the latest official estimate (as at mid 2019) is 897,711.

Name	Area	Pop'n Census 2010	Pop'n Estimate mid 2019
Doro	68.45	37,071	38,419
...	...	...	...
Totals	836.13	838,621	897,711

Figure 3.1: GenTaP Framework. The left figure shows our Intermediate Pre-training stages: LongAug and ShortAug. The right figure shows our synthetic training data generation methods: Context-to-Question for LongAug, and SCFG for ShortAug.

is first leveraged to generate denotations (which are usually entities from the table), and then a data-to-text generation model is used to compose a coherent and fluent sentence from the table schema and denotations. This pipeline model relies heavily on the semantic parser to produce accurate denotations; otherwise error propagation may lead to poor performance. The second method, an end-to-end model, is formulated as a sequence-to-sequence learning problem where free-form answers are directly generated conditioned on the question and table input, without producing intermediate results. [308] showed the latter approach yielded significantly better performance.

Thus, in this work, we use the BART sequence-to-sequence pre-trained language model as our baseline architecture, by leveraging its potential on text generation. More specifically, the table is linearized into a sequence T by separating the rows with special token `[ROW]` and separating cell values with vertical bar. This linearized table is appended to the question tokens q with `[SEP]` in between. In addition, we provide the positional embeddings for each token, including the segment embedding (for question segment and table segment), row embedding and column embedding [168]. These embeddings are added on top of the token embeddings as model inputs and optimized during the training. The free-form answer is regarded as target sequence. The Data-to-Text generation task is similar to the Free-form Question Answering, just without the prepended question. The input of the sequence-to-sequence model is the linearized table and the learning target is the table summary. We can regard a hidden question *"What is the summary of the table?"* is prepended.

3.3.2 Intermediate Pre-training

For the pre-training model, we use a similar architecture as the baseline systems. The questions and tables are fed into the transformer encoder; the tables are linearized with same strategy as the baseline systems.

Two types of augmentations are employed in the intermediate pre-training stages: LongAug and ShortAug. In the LongAug, table-enriched sentences are regarded as our learning target, where the sentences express some facts that are based on some parts of the table. This learning target is expected to improve (include but not limit to) the natural sentence generation ability in the context of table-based question answering scenario. In the ShortAug, short entities are the learning target. If multiple entities are generated, they are separated with vertical bars. This learning target is expected to help the model to improve the factual accuracy of the pre-trained models. Because the essential component in the long-form answer is still the key entities that answer the questions. In terms of model architecture, we use same architecture as the baseline model, a positional embedding augmented sequence-to-sequence model. Note that during pre-training, we use two separate decoders for these two learning targets, and the model is trained with multitask learning fashion. Our preliminary experimental results show that two separate decoders outperformed unified decoder.

3.3.3 Pre-training Data Synthesis

Data is one key part in this intermediate pre-training. As discussed, the question-answer (long or short form) pairs are expected in our pre-training stage, while they are not available in large scale for representation learning. In this work, we exploit two methods for synthesizing the pairs from large scale tables from Wikipedia: Context-to-Question Generation and Synchronous Context-free Grammar.

LongAug Synthetic Data: The target of Context-to-Question Generation is to synthesize *(Question, Long-form Answer)* pairs for intermediate pre-training stage LongAug. For each table we crawled from the Wikipedia page, we retrieve the statements that are relevant to the specific table from Wikipedia articles. We note these statements as *table knowledge enriched sentences* and these sentences are used as the proxy for long-form answers. Because the relevant statements usually come from the same article as the table appears in, we only consider each sentence in the specific Wikipedia page, without examining other articles. We compute the relevance level for each sentence and the table, by using the lexical matching strategy: if there are several cell values in the table appearing in the

$$\begin{aligned}
\text{[question]} &\rightarrow \text{What is [select] when [where]} \;\mid \\
&\ \text{What is [select]} \\
\text{[select]} &\rightarrow \text{the [column]} \mid \\
&\ \text{the [aggregation] of the [column]} \\
\text{[where]} &\rightarrow \text{[column] [comparison] [value]} \mid \\
&\ \text{[where] and [where]} \\
\text{[aggregation]} &\rightarrow \text{smallest} \mid \text{largest} \mid \text{sum} \mid \text{average} \\
\text{[comparison]} &\rightarrow \text{is} \mid \text{is smaller than} \mid \text{is larger than}
\end{aligned}$$

Figure 3.2: The SCFG for ShortAug Data Sampling.

sentence (more than the threshold), we regard it as a relevant statement candidate. We note these overlapped entities as *key entities*. For each key entity, we generate a question for it by leveraging a context-to-question generator.

In particular, the input of the generator is the table knowledge enriched sentence and the key entity; the output of the generator is the corresponding question — see Figure 3.4 for an example. We use the BART model as the generator. To train the generator, we use the SQUAD [370] dataset. The SQUAD dataset is designed for reading comprehension task where (question, paragraph, short-form answer) triples are provided. We adapt the SQUAD dataset for our purpose: for each example, we first identify the sentence from the paragraph where the short-form answer is found; the input to train the generator is the concatenation of the article title, the identified sentence and short-form answer; the training target is the question. In this way, we generate large scale (question, table, long-form answer) triples by leveraging the alignment between the table and the context and context-to-question generator, without using extra table-based QA datasets.

ShortAug Synthetic Data: Similar to [116], we build table-dependent question that are SQL-like. We define a synchronous context-free grammar (SCFG) as shown in Figure 3.2 and questions are sampled from it. The corresponding answers can be easily obtained during the sampling process. These answers are all cell values from the table, or the numerical aggregation results such as `SUM`, `MAX` and `MIN`. As the example shown in right side of data generation in Figure 3.1, a question "What is the [area] when [Pop'n Census 2010] is smaller than 40,000" can be composed based on the table. In this way, we synthesize large scale (question, table, short-form answer) triples for the intermediate pre-training stage ShortAug.

Model	**BLEU**	**ROUGE-1**	**ROUGE-2**	**ROUGE-L**	**METEOR**
TAPAS + T5-large	11.00	0.40	0.22	0.35	0.24
T5-small (fine-tuned by [308])	21.60	0.55	0.33	0.47	0.40
T5-base (fine-tuned by [308])	28.14	0.61	0.39	0.51	0.47
T5-large (fine-tuned by [308])	30.54	0.63	0.41	0.53	0.49
BART (fine-tuned by us)	32.14	0.658	0.432	0.551	0.512
Zero-shot (ours)	27.12	0.566	0.351	0.469	0.422
GenTaP (ours)	**36.74**	**0.689**	**0.476**	**0.587**	**0.545**
- ShortAug	36.07	0.683	0.470	0.582	0.541
- LongAug & ShortAug	33.87	0.668	0.443	0.563	0.520

Table 3.1: Results on the test split of FeTaQA dataset.

3.4 Experimental Setup

For all experiments, we train our GenTaP model with underlying transformers initialized with BART-large model [233]. 250K LongAug examples are generated via Context-to-Question Generation and 250K ShortAug examples are generated via SCFG. The tables that are used in the downstream tasks are removed in the pre-training stage.

3.4.1 Data Preprocessing

We leverage several heuristics to collect the tables and the contexts pairs. More specifically, for each sentence in the same page of the table, if one of the conditions is satisfied, then it is a valid *(table, context)* pair. A sentence is valid **1)** if it has tokens matching at least 3 key entities from the same row of the table. **2)** if it has tokens matching with 2 key entities from the same row of the table for more than two times (two different rows).

3.4.2 Training Details

For intermediate pre-training, we use 8 Tesla V100 GPUs to train at most 100K steps with initial learning rate of 2e-5 and batch size of 64. For FeTaQA dataset finetuning, 4 Tesla V100 GPUs are used to train the model, with initial learning rate of 1e-5 and batch size of 32. For FSD2T dataset finetuning, 1 Tesla V100 GPU is used to train with initial learning rate of 1e-5 and batch size of 8.

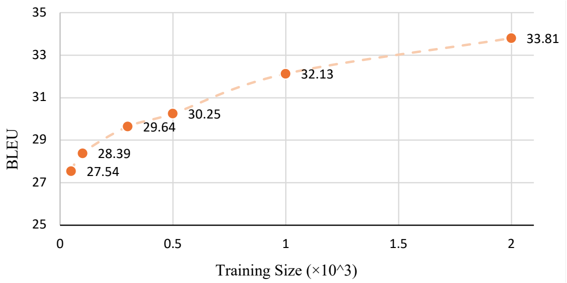


Figure 3.3: Low-data regimes. We finetuned GenTaP on 50, 100, 300, 500, 1000 and 2000 sampled training examples.

Model	**Precision**	**Recall**
T5-small	-2.8093	-2.3946
T5-base	-2.4989	-2.2686
T5-large	-2.3428	-2.1451
Zero-shot	-2.8333	-2.3555
GenTaP	**-2.0627**	**-1.8609**
- ShortAug	-2.0801	-1.8932
- LongAug & ShortAug	-2.1482	-1.9941

Table 3.2: BARTScore results on FeTaQA test split. Scores are shown in log probability. Higher is better.

3.4.3 Tasks, Datasets and Baselines.

We evaluate our model on the FeTaQA [308] dataset. FeTaQA is a table-based free-form question answering dataset that contains large scale *(question, table, long-form answer, supporting table cells)* pairs. Compared with WikiSQL [570] or WTQ [338], the questions in FeTaQA are more complex — requiring elaborations and explanations. The state-of-the-art systems on FeTaQA are based on the T5 models end-to-end models that generate answers directly from the question and table inputs. We also compare our models with pipeline baselines that first leverage state-of-the-art weakly supervised parser TAPAS [168] to generate denotations, and then leverage the T5-large as data-to-text generator.

We also evaluate transfer ability of our model by testing it on the few-shot Data-to-Text generation task. That is, we examine if our pre-training model is helpful on the related task of generating natural sentences based on the knowledge of table. We evaluate our model on Data-to-Text generation Dataset (FSD2T) [68]. The FSD2T includes data in three different domains, including the Humans, Books and Songs. We experiment on different training size, including 50, 100, 200 and 500 training examples in each domain. The models are chosen based on the performance of the development set with 1000 examples. Test sets for Humans, Books and Songs consist of 13587, 5252, and 11879 examples. We compared our models with Base [68], Base+Switch+LM [68], and TableGPT [139] that are all based on GPT2 [364].

In the ablation study, the –ShortAug denotes the intermediate pre-training without ShortAug. The –LongAug & ShortAug denotes the baseline model without intermediate pre-training — note that this model does include the positional, segment, column and row embeddings.[2]

3.5 Results

3.5.1 FeTaQA Main Results.

The main results of FeTaQA dataset are shown in Table 3.1. We evaluate the models with *unsupervised matching* in the *discrete string space* [548], such as BLEU, ROUGE-{1,2,L} and METEOR. The previous state-of-the-art performance (before the paper submission) is obtained by T5-large with 770M parameters, which achieves 30.54 BLEU score, outperforming other variants of T5 such as T5-base (220M parameters) and T5-small (60M parameters). For ROUGE-1, ROUGE-2, ROUGE-L and METEOR, the T5-large achieves 0.63, 0.41, 0.53 and 0.49 respectively. More recently (after paper submission), [497] obtained 33.44 BLUE score with T5-3B (3B parameters). For the baseline that leverages the table-based pre-trained model such as TAPAS, the experimental results are obtained with the TAPAS + T5-large architecture. TAPAS + T5-large is a pipeline architecture that leverages the state-of-the-art models in two worlds: the weakly semantic parsing and the data-to-text generation. The model firstly extracts denotations (key entities) based on the questions and tables input. Then a trained T5-large model performs the data-to-text generation based on the produced denotations, together with other meta information of the tables. This baseline only obtains 11.00 BLEU score, due to imperfect parsing system and error propagation issue.

Our framework is based on the BART architecture with 406M parameters, that is smaller than the T5-large architecture. We finetune the BART model on the dataset, obtaining 32.14 BLEU score, exceeding the state-of-the-art T5-large model [308] of 30.54

[2] BART (fine-tuned by us) in Table 3.1 and 3.3 do not leverage segment, column and row embeddings.

Model	Lexical level F1	Tuple level F1
T5-large	0.722	0.509
BART fine-tuned	0.725	0.515
GenTaP	**0.767**	**0.558**
- ShortAug	0.755	0.554
- LongAug & ShortAug	0.746	0.538

Table 3.3: Factual Consistency Evaluation.

BLEU score and comparable with more recent version finetuned by [497] of 32.45 BLEU score. For other metrics, our finetuned BART model also achieves new state-of-the-art performance. Augmenting with our pre-trained GenTaP model, the performance is further improved by large margins on different evaluate metrics, reaching 36.74 BLEU score, and 0.689, 0.476, 0.587, 0.545 on the ROUGE-{1,2,L} and METEOR, respectively.

3.5.2 Zero-shot and Few-shot FeTaQA Results.

Based on our intermediate pre-training objectives, our trained models already have the ability of answering the questions with free-form statements. Therefore, it is interesting to evaluate the zero-shot performance of the pre-trained models. Without finetuning, we directly feed the FeTaQA test set into the model and produce the answers. The results are shown in Zero-shot entry in Table 3.1, with 27.12 BLEU score and 0.566, 0.351, 0.469, 0.422 on the metrics of ROUGE-{1,2,L} and METEOR, respectively. Hence, the performance is on par with fully supervised T5-small model.

Through experiments in low-data regimes, we find that our pre-trained GenTaP model is an efficient learner. We finetuned GenTaP on 50, 100, 300, 500, 1000 and 2000 sampled training examples. Experimental results are shown in Figure 3.3. Using just 100-300 training examples, the model can achieve comparable performance against the T5-base model; while with 1000-2000 training examples, the model can obtain the similar effectiveness against the supervised BART baseline.

3.5.3 Model-based Evaluation.

Leveraging large scale pre-trained language model to evaluate the performance of generation models has become popular as its metric has been shown to have high correlation

Domain	Humans				Books				Songs			
# of training instances	50	100	200	500	50	100	200	500	50	100	200	500
GPT (Switch + LM)	25.7	29.5	36.1	41.7	34.3	36.2	37.9	40.3	36.1	37.2	39.4	42.2
Table-GPT	29.8	34.5	40.6	45.6	35.1	37.3	38.5	41.6	36.7	37.8	39.3	42.3
GenTaP (ours)	**39.4**	**45.9**	**47.4**	**50.8**	**39.8**	**41.6**	**43.1**	**46.7**	**38.3**	**42.0**	**44.0**	**45.1**
- LongAug & ShortAug	37.5	44.1	46.5	50.1	37.9	40.8	40.4	46.6	36.7	40.7	42.7	43.6

Table 3.4: Few-Shot Data-to-Text Generation results on different domains.

with human judgement. In this work, we further evaluate the models with the recent work, BARTScore [548]. Instead of relying on *token-level matching* on the *discrete string space*, the BARTScore formulates evaluating generated text as a text generation task from pre-trained language models. The log probability of BART generator is used to evaluate the quality of hypothesises (h) based on the references (r). Based on different input-output pairs, the following metrics can be evaluated by using the BARTScore. 1) **Precision**: The encoder input is the reference text and the decoder input is the generated text. The $P(h|r)$ is calculated and it accesses how likely the hypothesis can be generated based on the reference input. 2) **Recall**: The encoder input is the generated text and the decoder input is the reference text. The $P(r|h)$ is evaluated and it calculates how many semantic content units are covered by the hypothesis. We use the BART finetuned on ParaBank2 as the evaluation checkpoint.[3]

We evaluate the predictionsof T5 models and compared against our models.[4] As shown in the top section of Table 3.2, the T5-large obtains -2.3428 precision and -2.1451 recall. With FeTaQA dataset finetuning, our model obtains the best performance with -2.0627 precision and -1.8609 recall. Unsurprisingly, it also outperforms the zero-shot evaluation significantly, where the precision and recall scores are -2.8333 and -2.3555, respectively.

3.5.4 Human Evaluation.

To further evaluate the quality of the answers generated by the models, we conducted human evaluation based on the following criteria. We asked internal annotators to evaluate 50 samples of FeTaQA instances on a 1-5 scale. The average score of the answers is 3.84, with 32 out of 50 answers obtaining 4 or 5, which is higher than –ShortAug with score of 3.70 and –LongAug & ShortAug with score of 3.42.

[3] https://github.com/neulab/BARTScore

[4] https://github.com/Yale-LILY/FeTaQA

Example 1

Question: What films did Kevin James star in between Barnyard and Grown Ups?

Reference: James starred in I Now Pronounce You Chuck and Larry (2007) and Paul Blart: Mall Cop (2009) between Barnyard and Grown Ups.

Baseline: In 2006, Kevin James starred in Barnyard, and wrote, directed and starred in Grown Ups.

Our Model: Kevin James starred in Barnyard (2006) and I Now Pronounce You Chuck & Larry (2007).

Example 2

Question: Which animated characters were designed by Glen Keane in 1989 and 1990?

Reference: Glen Keane designed and animated the character of Ariel in the film The Little Mermaid (1989) and Marahute in The Rescuers Down Under (1990).

Baseline: Glen Keane designed the characters for The Little Mermaid (1989) and The Rescuers Down Under (1990).

Our Model: Glen Keane designed Ariel in The Little Mermaid (1989) and Marahute in The Rescuers Down Under (1990).

Table 3.5: Selected Examples for FeTaQA. Our Model refers to GenTaP while the Baseline refers to positional embedding augmented BART model without pre-training.

Few-Shot FSD2T Main Results.

The results of few-shot data-to-text generation task are shown in Table 3.4. We can observe that our baseline models already achieve the state-of-the-art BLEU score all three domains under different training settings. For our GenTaP models, even it is not pre-trained for the question answering purpose, the models showed good transfer ability by further improving the performance. By comparing the different training size, we can observe that with fewer training examples, such as 50 or 100, the model has larger improvement margins. When the training size is larger such as 500, the improvements are less significant.

3.6 Discussion and Analysis

3.6.1 Are the generated free-form answers factually consistent?

While metrics such as BLEU and ROUGE often serve as the primary metrics for assessing the quality of generated text, these metrics have been shown to be sometimes poorly correlated with answer correctness [106]. As a result, we leverage an alternate evaluation criteria which leverages the highlighted cells from the FeTaQA dataset's annotations. The highlighted cells are intended to capture key entities that the free-form answer should ideally make use of. So we measure the precision and recall of these key entities in the generated answer text. More specifically, we regard the highlighted cells that appear in the references as reference entity set; we extract the key entities from the generated text with string matching, denoted as hypothesis entity set. The precision, recall and F1 scores based on these two sets can be calculated; we call these scores are in lexical level. We can further regard the key entities that are from the same table row as a tuple; a tuple is correct only when all entities in the tuple are correct. Thus we can evaluate the tuple level precision, recall and F1 score. This is stricter evaluation for the models. These results are shown in Table 3.3 and demonstrate an improvement when GenTaP is used for pre-training. Our GenTaP obtains the 0.767 on the lexical level F1 score and 0.558 on the tuple level F1 score, outperforming the state-of-the-art T5-large model by large margin.

3.6.2 Error analysis.

To further understand the performance and behaviors of the models, we investigated the errors the models made. We classify the errors into the following types: lookup error and aggregation error. For the *lookup error*, the models fail to retrieve relevant rows/columns based on the header mentions or conditions. As shown in the Table 4.6, the two examples belong to this category. The question in the Example 1 requires the model to understand the condition "*between Barnyard and Grown Ups*" and retrieve the relevant rows in between from the table. The baseline model fails to understand the question and just extracts the information of movie "*Barnyard*" and "*Grown Ups*". Our GenTaP model is partially correct based on the answer it generates. It retrieves the movie "*I Now Pronounce You Chuck & Larry*" that is after the "*Barnyard*" but misses the other one. The question in the Example 2 asks the model to provide the information about the "*animated characters*". Our GenTaP model provides the corresponding information "*Ariel*" and "*Marahute*", however, the baseline does not answer with these key entities. On the other hand, the *aggregation type* questions are hard for the models. For example, the question "How much overall damage did the German submarine U-438 cause?" required the model to calculate the sum of the tonnage of the submarines and all the models failed. Further improving this type of questions is left for future work.

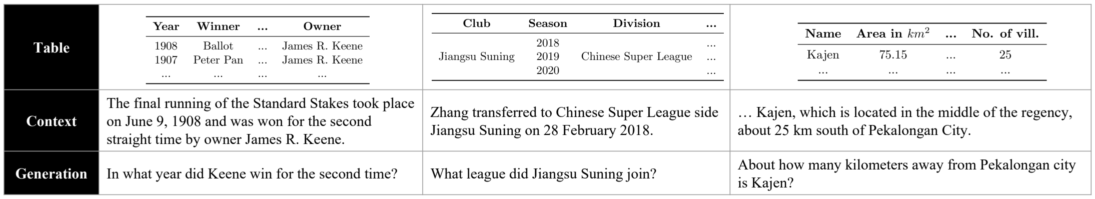


Table	Year Winner ... Owner 1908 Ballot ... James R. Keene 1907 Peter Pan ... James R. Keene	Club Season Division ... 2018 ... Jiangsu Suning 2019 Chinese Super League ... 2020 ...	Name Area in km^2 ... No. of vill. Kajen 75.15 ... 25 ... ...
Context	The final running of the Standard Stakes took place on June 9, 1908 and was won for the second straight time by owner James R. Keene.	Zhang transferred to Chinese Super League side Jiangsu Suning on 28 February 2018.	… Kajen, which is located in the middle of the regency, about 25 km south of Pekalongan City.
Generation	In what year did Keene win for the second time?	What league did Jiangsu Suning join?	About how many kilometers away from Pekalongan city is Kajen?

Figure 3.4: Examples of our LongAug synthetic data. The Generated Questions were synthesized using our Context-to-Question method.

3.6.3 Ablation Study for Pre-training

Data Synthesis Quality.

The LongAug synthetic data generator — Context-to-Question Generation — obtains 21.52 BLEU score on the SQUAD validation set. To assess the quality of the pre-training data, we further sampled 50 examples from the generated *(question, table, context)* triples and ask graduate students and practitioners who are working on NLP for judgement. We evaluate the data in the following aspects: 1) **Alignment**: whether the context is supported by the facts from the table. Because the contexts are aligned with tables automatically, false positive error will be introduced. 2) **Correctness**: whether the generated question is correct based on the context and sampled answer span. This evaluates the correctness aspect of the question generator. Out of 50 examples, there are 18.5 (averagely) context sentences aligning with the table. This indicates that the automatic alignment strategy imperfectly introduces errors for the data generation stage and can be further improved in the future work. For the Context-to-Question generator, 30.5 (averagely) out of 50 questions are in high quality based on the contexts and selected key entities. More alignment and generation examples are shown in Figure 3.4. First row shows a high-quality *(Question, Table, Context)* pair. For the second one, the generator makes mistakes with the subject, being confused "*Zhang*" with "*Jiangsu Suning*". For the third one, the error happens on the automatic alignment where the distance "*25*" in the context is matched with the number of village "*25*" in the table.

How does LongAug synthetic data size affect model performance?

For LongAug, we analyze the effectiveness of the generation-based training data in terms of the scale. The Table 3.6 shows the performance of FeTaQA with different scales of pre-training corpus.

Training Size	10K	50K	100k	250K
BLEU	34.58	35.01	35.49	36.07

Table 3.6: Results on different LongAug synthetic data sizes

Training Task Design.

In this section, we show the ablation study of the training targets. Based on the automatic evaluation metrics, the LongAug improve the BLEU score from 33.87 to 36.07 by large margin. The ShortAug can further improve the metric to 36.74. The effectiveness of the LongAug and ShortAug is also shown from the BARTScore, Lexical level F1, Tuple Level F1 and the human evaluation.

Instead of using the generated questions as the text for the model input in our proposed GenTaP framework, we also explored design choices for pre-training: 1) Random Token Masking, and 2) Key Entity Masking. **Random Token Masking (RTM)** is analogous to the Masked Language Model and we randomly mask the token in the context as the model input. We keep the table unchanged and use the original context as the learning target. We expect the model to capture the alignments between the context and table by learning to recover the incomplete context. **Key Entity Masking (KEM):** Instead of masking random tokens which may be unimportant, we try to mask the key entities. More specifically, based on the context-table alignment aforementioned, we masked the co-occurrent entities in the context, making it a proxy of natural questions. Again, we use the unmasked context as the training target. In this way, we can enforce the model to learn to capture more alignments between context and table by recovering the context, because all missing tokens come from the table content. We pre-train the models in the same way as the GenTaP with the *(context, table)* pairs. We use BLEU score to evaluate the model performance. **RTM** obtains 34.26 BLEU score while **KEM** obtains 34.85 BLEU score. Based on the results, we find that using the generated question as text input is a better choice than these two proposals, thus we did not use them in our main experiments.

3.7 Summary

In this chapter, we present an intermediate pre-training framework, GenTaP, that improves the joint encoding ability of question and table for pre-trained sequence-to-sequence language model. With two augmentation strategies, our models achieve state-of-the-art performance on the free-form table-based question answering task. Also, the GenTaP models show good transfer ability to the few-shot data-to-text generation task, by outperforming existing models on FSD2T dataset in various domains.

Chapter 4

Learning Contextual Representation for Semantic Parsing with Generation-Augmented Pre-training

Most recently, there has been significant interest in learning contextual representations for various NLP tasks, by leveraging large-scale text corpora to train large neural language models with self-supervised learning objectives, such as Masked Language Model (MLM). However, based on a pilot study, we observe three issues of existing general-purpose language models when they are applied to text-to-SQL semantic parsers: fail to detect column mentions in the utterances, fail to infer column mentions from cell values, and fail to compose complex SQL queries. To mitigate these issues, we present a model pre-training framework, Generation-Augmented Pre-training (GAP), that jointly learns representations of natural language utterances and table schemas by leveraging generation models to generate pre-train data. GAP Model is trained on 2M utterance-schema pairs and 30K utterance-schema-SQL triples, whose utterances are produced by generative models. Based on experimental results, neural semantic parsers that leverage GAP Model as a representation encoder obtain new state-of-the-art results on both SPIDER and CRITERIA-TO-SQL benchmarks. This work is presented in:

- Peng Shi, Patrick Ng, Zhiguo Wang, Henghui Zhu, Alexander Hanbo Li, Jun Wang, Cicero Nogueira dos Santos, Bing Xiang. Learning Contextual Representations for Semantic Parsing with Generation-Augmented Pre-Training. *In Proceedings of AAAI (Thirty-Fifth AAAI Conference on Artificial Intelligence)*, February 2021.

Pain Point 1: Fail to match and detect the column mentions.
Utterance: Which professionals live in a city containing the substring ’West’? List his or her role, street, *city* and state.
Prediction: `SELECT role_code, street, state FROM Professionals WHERE city LIKE ’%West%’`
Error: Missing column `city` in `SELECT` clause.

Pain Point 2: Fail to infer columns based on cell values.
Utterance: Give the average life expectancy for countries in Africa which are *republics*?
Prediction: `SELECT Avg(LifeExpectancy) FROM country WHERE Continent = ’Africa’`
Error: Missing `GovernmentForm = ’Republic’`.

Pain Point 3: Fail to compose complex target SQL.
Utterance: Which semesters do not have any student enrolled? List the semester name.
Prediction: `SELECT semester_name FROM Semesters WHERE semester_id NOT IN (SELECT semester_name FROM Student_Enrolment)`
Error: Should use `semester_id` in nested SQL to align with the column in `WHERE` clause.

Table 4.1: Error examples collected from the Spider development set based on the RAT-SQL + BERT [463].

4.1 Introduction

Recently, deep contextual language models [103, 273, 233, 113, 366] have shown their effective modeling ability for text, achieving state-of-the-art results in series of NLP tasks. These models capture the syntactic and semantic information of the input text, generating fine-grained contextual embeddings, which can be easily applied to downstream models. Despite the success of large scale pre-trained language models on various tasks, it is less clear how to extend them to semantic parsing tasks such as text-to-SQL [477, 355, 354, 245], which requires joint reasoning of the natural language utterance and structured database schema information. Recent work [147, 463, 34, 33] shows that with more powerful pre-trained language models, the highly domain-specific semantic parsers can be further improved, even though these language models are trained for pure text encoding.

However, based on error analysis on the output of neural language model-based text-to-SQL systems, we observe that these models can be further enhanced if we could mitigate the following three pain points, which are also illustrated in Table 7.8. (1) *The model is ineffective to match and detect column names in utterances.* The model should learn to detect column names mentioned in utterances by matching utterance tokens with the schema, and use the matched columns in the generated SQL. The error analysis indicates that, in some cases, models miss some columns when synthesizing the target SQL, while the column is mentioned explicitly in the utterance. (2) *The model fails to infer the columns implicitly from cell values.* This problem is trickier than the first one, because the model is expected to infer the column name based on some cell values mentioned in the utterance, instead of just matching the utterance tokens with the schema. This requires the model to have more domain knowledge. For example, as presented in the second section of Table 7.8, the model should know `republics` is a `GovernmentForm`. (3) *The model should learn to compose complex queries.* Besides the column selection, to generate a correct SQL, the model should learn to attach the selected columns to the correct clauses. This is a non-trivial task, especially when the target SQL is complex, e.g., when the query is nested. As shown in the last section of Table 7.8, the model should learn to use the corresponding column `semester_id` in the nested SQL, instead of using column `semester_name`.

Recent work has demonstrated that jointly pre-training on utterances and table contents (e.g., column names and cell values) can benefit downstream tasks such as table parsing and semantic parsing [528, 168]. These models are pre-trained using the Masked Language Modeling (MLM) task by either masking tokens from the *utterance* input or tokens from the *schema* input. However, this learning objective can only model the alignment between the utterance and schema implicitly. We hypothesize that, in order to cope with the three pain points previously listed, it is necessary to use pre-training objectives that enforce the learning of contextual representations that better capture the alignment between utterances and schema/table contents.

In this chapter, we present a language model pre-training framework, **G**eneration-**A**ugmented **P**re-training (GAP), that exploits multiple learning objectives (pre-training tasks) and synthetic data generation to jointly learn contextual representations of natural language utterances and table schema. We propose the following three new learning objectives that not only enforce joint learning but also improve the ability of the model to grasp more domain knowledge, which is helpful in cross-domain scenarios: (1) *column prediction task*, which is a pre-training task that consists in giving a label for each column in the input schema to decide whether it is used in the input utterance or not. This task is designed to improve the column detection ability of the model. (2) *column recovery task*, which consists in randomly replacing some of the column names with one of their cell values and asking the model to recover the original column name either based on the cell value itself or based on the contextual information of the utterance when the column is explicitly mentioned in the utterance. This learning objective is meant to enhance the column inferring ability of the model. (3) *SQL generation*, which consists in generating SQL queries given utterances and schema. This task can boost the ability of the model to compose complex queries by leveraging large scale SQL datasets from the Web.

A key challenge to use the proposed pre-training tasks is training data. Although it is easy to obtain large scale datasets of crawled tables and SQL queries, it is difficult to obtain high-quality utterances interrelated with the tables or logically consistent with crawled SQL queries. Recent work used the surrounding text of tables as a proxy of natural language utterances [528, 168]. However, this option is far from optimal because those texts are dissimilar to user utterances in terms of text length, composition and content. The surrounding text of a table is usually a paragraph, while natural language utterances in the downstream task are short sentences. Furthermore, the content of surrounding text of tables can be quite noisy because the text may be irrelevant to the table. In GAP , we overcome the pre-training data challenge through the use of synthetic data. We propose two sequence-to-sequence (seq2seq) generative models, *SQL-to-text* and *table-to-text*, that can produce large scale datasets with enough quality for pre-training. We train our generative models by finetuning BART [233], a state-of-the-art pre-trained language model. Concurrently, [537] and [99] utilized synthetic data generated from synchronized context-free grammar and existing data-to-text datasets [337] for pre-training, respectively, which requires extra crowd and expert annotation efforts.

The outcome of GAP is a pre-trained model that can be plugged into neural semantic parsers to compute contextual representations of utterances and schema. We apply GAP to text-to-SQL semantic parsing datasets, and experimental results show that systems augmented with GAP outperform state-of-the-art semantic parsers on Spider and Criteria-to-SQL datasets. In summary, our work presents the following main contributions: 1) Based on an error analysis, we spot three main issues in pre-trained LM-based text-to-SQL semantic parsers. 2) We propose a new framework for pre-training semantic parsers that exploits multiple pre-training tasks and synthetic data. 3) We present three novel learning objectives that alleviate the three main issues spotted with pre-trained LMs for semantic parsing. 4) We propose a novel strategy to overcome pre-training data challenges by leveraging SQL-to-Text and Table-to-Text generative models to generate synthetic data for learning joint representations of textual data and table schema. 5) To the best of our knowledge, this is the first work to effectively use both crawled SQL and crawled tables to enhance the text-to-SQL semantic parsing task.

4.2 Related Work

Semantic Parsing: The semantic parsing task is framed as mapping the natural language utterances to meaning representations. The meaning representations can be executed in a variety of environments such as data analysis by translating the natural language queries into database queries. Based on different meaning representations, the semantic parsing task can be classified into three regimes [203]: logic based formalism such as λ-DCS [248], graph based formalism such as AMR [25] and UCCA [3], and programming languages such as Python and SQL. Recently, more interests are concentrated on the SQL-based semantic parsing, and most of the work try to solve the problem with general encoder-decoder architecture. Overall, they enhance the models based on following aspects: (1) Improving the decoding mechanism [525, 111, 390]; (2) Improving the decoding target [147]; (3) Improving the model encoding ability [463, 33, 528, 396, 280, 99, 537]; (4) Reranking over the generated candidates to improve parses quality [206, 527]. GAP advances the line of (3) by leveraging generation models and three novel learning objectives to enhance the utterance-schema representations.

Question Generation and Table-to-Text Generation: The question generation task is to generate grammatically and semantically correct questions. The generated questions are usually used for enhancing the question answering models [114, 142, 532, 569]. The table-to-text generation task is to generate declarative sentences that describe the information provided by the table [271, 138, 337, 362]. Our Table-to-Text model is a combination of these two directions, focusing on generating questions from table, i.e., composing questions based on the sampled columns and cell values, without providing detailed information about "what to ask".

Pre-training Models: Recent pre-training techniques exploit external knowledge (e.g. entity-level information, commonsense knowledge, knowledge graph) into large-scale pre-trained language models [499, 470, 346, 388]. More recently, [528], [168], leverage the semi-structured table data to enhance the representation ability of language models. Concurrently, [537] and [99] leveraged synchronous context-free grammar to generate synthetic data and utilized existing high-quality data-to-text dataset for pre-training, respectively. Different from these work, we explore the direction of utilizing the generators to enhance the joint utterances and structured schema encoding ability of the pre-trained models.

4.3 Models

We first present the architecture of the semantic parsers, and then introduce the pre-training model in the GAP framework. Lastly, we describe how to obtain the synthetic pre-training data with generative models.

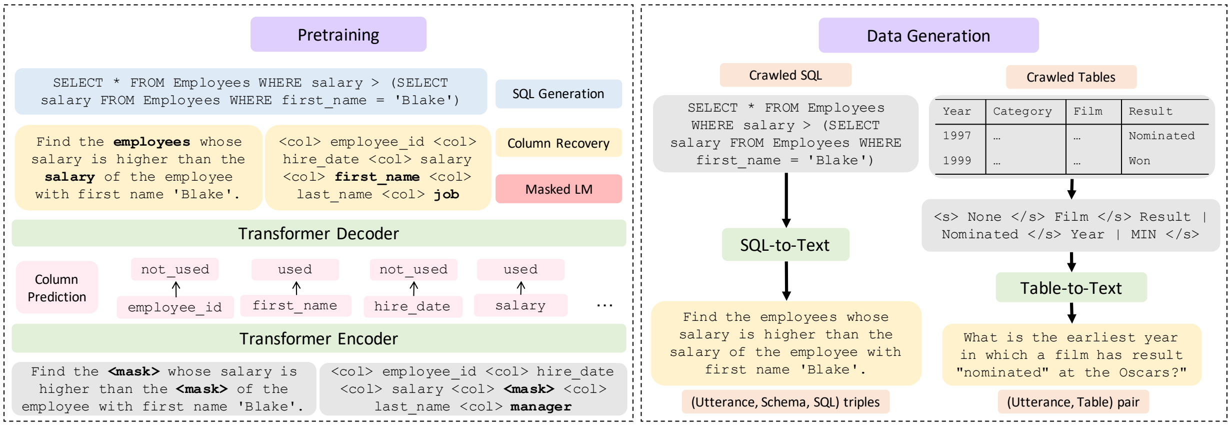

Pretraining
SELECT * FROM Employees WHERE salary > (SELECT salary FROM Employees WHERE first_name = 'Blake')
SQL Generation
Find the employees whose salary is higher than the salary of the employee with first name 'Blake'.
<col> employee_id <col> hire_date <col> salary <col> first_name <col> last_name <col> job
Column Recovery
Masked LM
Transformer Decoder
Column Prediction
not_used
employee_id
used
first_name
not_used
hire_date
used
salary
…
Transformer Encoder
Find the <mask> whose salary is higher than the <mask> of the employee with first name 'Blake'.
<col> employee_id <col> hire_date <col> salary <col> <mask> <col> last_name <col> manager
Data Generation
Crawled SQL
SELECT * FROM Employees WHERE salary > (SELECT salary FROM Employees WHERE first_name = 'Blake')
SQL-to-Text
Find the employees whose salary is higher than the salary of the employee with first name 'Blake'.
(Utterance, Schema, SQL) triples
Crawled Tables
Year
Category
Film
Result
1997
…
…
Nominated
1999
…
…
Won
<s> None </s> Film </s> Result | Nominated </s> Year | MIN </s>
Table-to-Text
What is the earliest year in which a film has result "nominated" at the Oscars?"
(Utterance, Table) pair

Figure 4.1: Building blocks of GAP framework. On the left side, we illustrate our proposed pre-training tasks. On the right side, we depict our proposed data generation strategies.

4.3.1 Text-to-SQL Semantic Parser

The Text-to-SQL semantic parser translates natural language utterances to SQL queries. The semantic parsers in our experiments are based on the encoder-decoder architecture. Given an utterance $U = \{x_1, x_2, ..., x_n\}$ and a schema S consisting of tables $T = \{t_1, t_2, ..., t_{|T|}\}$ and columns $C = \{c_1, c_2, ..., c_{|C|}\}$, we leverage the contextual encoder to obtain the representations of utterance tokens and schema. The decoder is required to compute a distribution $P(Y|X, S)$ over SQL programs. Based on different model designs, the decoder learning target Y can be raw SQL tokens [560] or other intermediate representations such as SemQL [147] or AST tree [34, 528].

4.3.2 Pre-training Model

The left part of Figure 8.1 presents an overview of GAP in the pre-training stage. Given an utterance U and schema S, GAP MODEL takes as input the concatenation of U and the column names c in S in the following format $X = \{$`<s>` U `<col>` c_1 `<col>` c_2 ... `<col>` $c_{|C|}$ `</s>`$\}$, where c_i denotes the i-th column in schema S. With the 12-layer transformers, each token in the input can be encoded as contextual representations, denoted as $\mathbf{h}$. For different learning objectives, the representations are utilized by different decoders. To jointly learn contextual representations for utterances and schemas and mitigate the three pain points discussed in the intro, we leverage four learning objectives in the pre-training: Besides the Masked Language Model (MLM), we propose learning objectives including

Column Prediction (CPred), Column Recovery (CRec), and SQL Generation (GenSQL). Multi-task learning is leveraged for these learning objectives,

Column Prediction (CPred): The Column Prediction learning objective encourages the model to capture the alignment signals between the utterance and schema, by predicting whether a column is used in the utterance or not. An illustration is shown in the pink component of Figure 8.1. Specifically, based on the representations obtained from the transformer encoder, a two-layer MLP is applied on each column representation $\mathbf{g}_{col}$, which is obtained from the output of an average pooling layer that aggregates all subtokens of the corresponding column. Afterward, a sigmoid activation function is applied to obtain the probability that the corresponding column is mentioned in the utterance. The GAP Model maximizes $P_{\theta_{enc}}(Y_c|X)$ where Y_c is a 0/1 label for a column and X is in its unmasked version.

Column Recovery (CRec): The Column Recovery learning objective strengthens the model's ability to discover the connections between the cell values and the column names, by recovering the column name based on a sampled cell value. For example, as shown in the left yellow part of Figure 8.1, the model recovers the column name `job` from cell value `manager`. Generally, the transformer decoder recovers column names based on two information sources: one is the actual cell value, and the other one is the column name mention in the utterance. We design the following rules for the value replacement:

- If a column is not mentioned in the utterance, we will replace the column name with its cell value with a probability of 0.5. In this case, the column name will be recovered from cell value without other contextual information.
- If a column is mentioned, we will directly replace the column name with its cell value. In this case, the model can leverage the contextual information from the utterance and the cell value to recover the column name.

SQL Generation (GenSQL): This learning objective is directly related to the downstream task. Based on the representation from the transformer encoder, the GAP Model decoder maximizes $p_{dec}(y_{sql}|\mathbf{h})$. This learning target encourages the model to learn to compose complex SQL that requires logical reasoning, considering that there are a large number of sophisticated SQLs in crawled data. For example, the GAP Model decoder needs to generate the column in the appropriate position such as in the `ORDER BY` clause or `WHERE` clause, instead of just predicting the column is used or not. Specifically, the GAP Model decoder emits the target SQL token by token with a close vocabulary set, which is composed of the SQL keywords vocabulary and column names. The embeddings of the SQL keywords are randomly initialized and trained during the pre-training phase. The column representations are obtained in the same way as the one used in Column Prediction learning objective, by averaging the column's sub-tokens representations. At each decoding step, the decoder generates a hidden vector and then a dot-product operation is applied to it and the target vocabulary representations, yielding a probability distribution over the vocabulary set.

Masked Language Model(MLM): We use the standard MLM objective, with a masking rate of 35% sub-tokens in the whole input sequence, including the utterance and schema. Based on the representation from transformer encoder, GAP Model employs a transformer decoder to maximize $p_\theta(x|x_m)$ on large-scale utterance-schema pairs, where x_m is the masked version of x.

4.3.3 Pre-training Data Generation

As discussed, previous pre-training approaches such as TaBERT [528] and TAPAS [168] use the surrounding texts of the tables as a proxy of natural language utterance. However, those texts are noisy and sometimes are not directly related to the table contents. In the downstream task, the input texts are usually utterances/user queries, which are short and highly dependent on the schema and contents of the structured data. In order to minimize the gap between pre-training and downstream tasks, we adopt a state-of-the-art pre-trained sequence-to-sequence model, such as BART, to generate high-quality utterances based on crawled SQLs or structured tables.

As shown in the right part of Figure 8.1, we design two different models, namely SQL-to-Text generation model and Table-to-Text generation model, for handling the two different inputs. Specifically, the SQL-to-text generation model takes the SQL as input and generates the utterance that explains the query intent. The other model, the Table-to-Text generation model, generates utterances based on a set of sampled column names and cell values from tables. In this way, we can generate utterances interrelated with tables without composing queries that might be suspicious.

SQL-to-Text Generation: We crawl 30K SQLs from GitHub.[1] To generate utterances for these SQL queries, we train a SQL-to-Text model on the Spider dataset. The input is the original SQL and it is directly tokenized by the BART tokenizer without additional pre-processing. After finetuning BART, the model can generate high-quality utterances logically consistent with the input SQL, achieving a 0.1934 BLEU score on the development set. Then we use the model to generate utterances for crawled SQLs. We further extract

[1] https://github.com

columns and tables in each SQL as positive schema candidates, denoted as $\texttt{schema}_{pos}$. We also sample columns and tables from the pool which are extracted from other SQLs as negative candidates, denoted as $\texttt{schema}_{neg}$. The final schema is composed of these two parts. The utterance-schema-SQL triples are then collected for the GenSQL learning objective in the pre-training phase.

Table-to-Text Generation: Generating utterances from tables is different because query intents are not given. Instead of synthesizing noisy SQLs and then translating into natural language utterances, we propose a Table-to-Text generation model that can directly transform a set of column names and cell values into user queries without query intent constraints. Specifically, we sample column names and cell values (both are referred to as candidates) from tables. For example, based on the table in the right part of Figure 8.1, we can sample columns `Year`, `Film` and `Result`, and a cell value `Nominated`. We then linearize the sampled candidates into {`column name | associated cell value list`} and concatenate them into a sequence, separated by `<sep>` token. Furthermore, to control the complexity and diversity of the generated text, we integrate three types of control codes into the model input:

- Aggregator-based control code: Including `COUNT`, `MAX`, `MIN`, `AVG`, and `SUM`. For the first two sampled columns, we randomly sample an aggregator for each with the probability γ_1 (we use γ_1 as 0.5) if the column type matches with the selected aggregator, e.g., aggregator `SUM` should be applied on the numerical type column. If the control codes are sampled, they will be appended to the associated cell value list of the corresponding column.

- Structure control code: Including `IN`, `NOT IN`, `INTERSECT`, `UNION`, and `EXCEPT`. For each example, with a probability of γ_2 (we use γ_2 as 0.35), we randomly sample one of them with uniform distribution. Otherwise, `NONE` is used. This control code is used as the first item of the input sequence.

- Order-based control code: We add {`LIMIT` : number} as a part of the control code, which is usually used in an `ORDER BY` based query. With this control code, the generated utterances usually contain phrases that constrain the number of query results should be returned, e.g., *Show the name of aircrafts with top three lowest speed.*.

We fine-tune a BART model on Spider dataset to create the generator. To align with our designed input, we convert the SQL into the format we expected. We extract all the columns and their associated aggregators and values from the SQL. We also obtain any special control codes that appear in the SQL. After fine-tuning, the model achieves 0.1821 BLEU score on the development set. Afterward, we apply the finetuned model to the crawled tables and generate high-quality utterances. The utterance-schema pairs are collected for the learning objectives including MLM, CPred, and CRec in pre-training phase.

For the pre-training step, we need to decide whether a column is mentioned in the utterance or not. To create the label for this, we directly regard all the sampled columns to have a positive label. This is based on the assumption that the generation model uses all the columns to synthesize the utterance, and does not have the hallucination issue that models generate some columns names or cell values that are not presented in the input.

4.4 Experimental Setup

4.4.1 Pre-training Data

Utterance-Table Pairs: We extract the tables from English Wikipedia. We further apply the following heuristic strategies to pre-process the extracted tables: (1) Removing tables with less than 4 columns; (2) Removing tables with less than 3 rows; (3) Removing columns whose names have more than 10 tokens; (4) Removing columns whose cell values have more than 50% empty string; (5) Filtering cell values with more than 5 tokens or containing any pre-defined non-ASCII characters. After the pre-processing, we obtain 500K tables.

For each table, we then randomly sample the column names, cell values, and control codes as the Table-to-Text generation model input to produce the utterances. We apply the following strategies to sample inputs: (1) We randomly generate an integer from 2 to 6, denoting the number of columns we will sample; (2) We sample the wildcard $*$ with probability of 0.2; (3) We sample one of the structure control codes with a probability of 0.35; (4) We sample the order-based control code with a probability of 0.25; (5) For the first two sampled columns, we randomly sample one of the aggregators with a probability of 0.5; (6) For each column without any associated aggregator-based control code, we sample one value from that column with a probability of 0.4. We then generate 4 instances per table and we finally obtain 2M training instances.

Utterance-Schema-SQL Triples: We crawl the SQL from GitHub repositories if the SQL can be parsed by one of the SQL parsers: moz-sql-parser[2] and sqlparse.[3] We apply

[2] https://github.com/mozilla/moz-sql-parser

[3] https://github.com/andialbrecht/sqlparse

the trained SQL-to-Text generation model to the SQL and obtain 30K utterance-SQL pairs. To obtain the schema, for each SQL, we extract the table names and column names from the SQL as positive candidates and randomly sample table names and column names from other SQL as negative candidates. The combination of these two components are regarded as the associated schema. We then obtain utterance-schema-SQL triples for GenSQL learning objective training.

4.4.2 Training Details

In the pre-training, we train our GAP Model with the underlying transformers initialized with BART [233] model. During the fine-tuning phase, we only leverage the encoder component of the GAP Model with 12-layer transformers as the contextual encoder for the semantic parsers. As discussed in the previous section, each epoch contains 2M utterance-table pairs and 30K utterance-schema-SQL triples. We train the GAP Model with multi-task training strategies: 30K utterance-schema-SQL triples are for GenSQL learning objective and 2M utterance-table pairs are evenly split for the other three learning objectives, including MLM, CPred and CRec. We train the model for 6 epochs with a batch size of 64 on 4 Tesla V100 GPUs. The model is optimized with Adam optimizer [211] with a learning rate of $1e-5$ and linearly decayed learning rate schedule.

4.4.3 Tasks, Datasets and Baseline Systems

For the downstream tasks, we conduct experiments on two datasets to show the effectiveness of our framework.

Spider: Spider dataset [542] is a text-to-SQL dataset with 10,181 annotated parallel utterance-database-SQL triples. Different from WikiSQL, the examples in the Spider dataset is more complex, involving nested query, set operation, and multiple tables joining. The exact set match accuracy is the evaluation metric. The test set is not publicly available. For the baseline parser, we use RAT-SQL [463] model as our baseline system to report the end-to-end performance. RAT-SQL model is the state-of-the-art parser in the Spider dataset, which leverages the 8-layer relation-aware transformer to model the connections among tables and utterances. To show that the GAP Model can be plugged into different neural semantic parsers, we further use IRNet [147] model for ablation study. IRNet semantic parser is based on SemQL grammar, which is an effective intermediate representation for SQL. IRNet is efficient in terms of training time, which requires 1 day for training, while RAT-SQL model requires approximately 5 days for training. We augment the encoder part of our GAP Model to these base parsers, by replacing their original contextual encoders.

Criteria-to-SQL: Criteria-to-SQL is a dataset to facilitate retrieving eligible patients for a trial from the electronic health record database. The task is to translate the eligibility criteria to executable SQL queries. For example, a criteria statement *any infection requiring parenteral antibiotic therapy or causing fever (i.e., temperature* $>$ *100.5f)* $\leqslant$ *7 days prior to registration* is required to be interpreted into SQL `SELECT id FROM records WHERE active_infection = 1 AND (parenteral_antibiotic_therapy = 1 OR causing_fever = 1 OR temperature > 100.5)`. The dataset contains 2003 annotated examples, and the evaluation metrics are the SQL accuracy and execution accuracy. Our baseline system for Criteria-to-SQL dataset is adopted from [546], a slot-filling based model that takes advantage of the prior grammar knowledge to design the sketch. We denote this system as **YXJ** model. The system uses the BERT-base as the contextual encoder.

4.5 Results

4.5.1 Spider Results

Table 4.2 shows the end-to-end results on Spider dataset. Based on the codebase provided by [463],[4] we replicate the RAT-SQL + BERT large model, achieving 0.665 exact set match accuracy on the development set. This matches the RAT-SQL V2 + BERT but still worse than its V3. By replacing the BERT-large with the encoder of BART,[5] we obtain accuracy of 0.676 on the development set and 0.651 on test set. The BART Encoder based model achieves comparable results with RAT-SQL V3 + BERT large model on the hidden test set with less encoder layer (BART encoder has 12-layer transformers while BERT large model has 24-layer transformers). With our GAP Model, the RAT-SQL can be further augmented, benefiting from enhanced contextual encoding ability. The model achieves an accuracy of 0.718 on the development set and 0.697 on the hidden test set. This confirms the effectiveness of the Generation-augmented pre-training. This performance achieves the state-of-the-art performance on the hidden test set with fewer model parameters on Spider dataset at the time of writing. Comparing scores of the development set and the test set, we observe BART based models (+BARR Encoder or GAP Model) have better generalization ability on the hidden test, considering that the gap between the development

[4] https://github.com/microsoft/rat-sql

[5] The encoder of BART has 12-layer transformers while BERT-large has 24-layer transformers.

Model	Dev	Test
EditSQL + BERT [560]	0.576	0.534
IRNet + BERT [147]	0.619	0.547
RyanSQL V2 + BERT [72]	0.706	0.606
RAT-SQL V2 + BERT [463]	0.658	0.619
AuxNet + BART	0.700	0.619
ShadowGNN†	-	64.8
YCSQL + BERT†	-	65.3
RAT-SQL V3 + BERT [463]	0.697	0.656
RAT-SQL + STRUG [99]	0.727	-
RAT-SQL + GraPPa [537]†	**0.734**	0.696
RAT-SQL + BERT (our replicate)	0.665	-
RAT-SQL + BART Encoder (ours)	0.676	0.651
RAT-SQL + GAP Model (ours)	0.718	**0.697**

Table 4.2: Exact set match accuracy on the public development set and hidden test set of Spider. † denotes that the algorithms are concurrent work and leaderboard results are public after our paper submission.

set and test set is smaller than the model such as RAT-SQL V3 + BERT. Concurrently, [537] used synchronized context-free grammar to generate synthetic data for pre-training; [99] leveraged existing large-scale data-to-text dataset for enhancing the structured data representations. Both of them achieve comparable performance as ours, but require more model parameters (24-layer transformers in the pre-trained model) and extra crowd and expert annotation efforts.

RAT-SQL	Easy	Medium	Hard	Extra	All
+BERT	0.830	0.713	0.583	0.384	0.656
+BART Encoder	0.826	0.711	0.581	0.370	0.651
+GAP Model	0.872	0.751	0.637	0.412	0.697

Table 4.3: Breakdown results on Spider hidden test set.

Based on the complexity of the SQL, the examples in Spider are classified into four types: Easy, Medium, Hard, and Extra Hard. Here, we provide a breakdown analysis on the Spider test set, as shown in Table 4.3. The BERT results are adopted from [463], which is the state-of-the-art system on Spider dataset. Comparing the RAT-SQL+BERT model and RAT-SQL+BART Encoder model, we can find that the performance of RAT-SQL+BART is comparable with the state of the art, but with fewer model parameters (12-layer transformers in BART encoder v.s. 24-layer transformers in BERT-large encoder). We also find that the RAT-SQL+GAP Model Encoder can have significant improvement over its baseline RAT-SQL+BART Encoder on each hardness level.

RAT-SQL	Selection	Inferring	Composing
+BART Encoder	14	10	16
+GAP Model	5	6	7

Table 4.4: Error counts of different types for RAT-SQL+BART Encoder and RAT-SQL+GAP Model Encoder.

For comparison, we sample 40 examples from Spider development set which the baseline system RAT-SQL+BART Encoder fails in. Because we focus more on the following three error types as we discussed in the introduction part: *column selection error*, *column inferring error* and *SQL composing error*, we ignore other error types during the sampling. We analyze the predictions of both the RAT-SQL+BART Encoder and RAT-SQL+GAP Model Encoder. The statistics are shown in Table 4.4. The numbers in the Table represent the error count of each error type. Based on the analysis results, we can find that the GAP Model Encoder can alleviate all three error types, especially the *column selection* and *SQL composing error*.

4.5.2 Criteria-to-SQL Results

Model	SQL Acc.	Exec. Acc.
SQLNet	0.132	0.139
YXJ [546]	0.142	0.158
YXJ + Roberta (ours)	0.294	0.538
YXJ + BART Encoder (ours)	0.307	0.558
YXJ + GAP Model (ours)	**0.327**	**0.594**

Table 4.5: Test results of Criteria-to-SQL. The SQL accuracy and the execution accuracy are reported.

Table 4.5 shows the test results of the Criteria-to-SQL dataset. The YXJ model [546] is built upon BERT-base encoder and sketch-based decoder, achieving the state-of-the-art performance of 0.142 SQL accuracy and 0.158 execution accuracy. We use this system as our baseline. Instead of using the BERT encoder, we augment the model with more powerful pre-trained language models such as RoBERTa and BART. These two pre-trained language models yield significant improvement over the BERT baseline, achieving 0.294 and 0.307 on the SQL accuracy, respectively. After executing the generated SQL queries against the database, these two models obtain 0.538 and 0.558 execution accuracy, respectively. By replacing the BART encoder with GAP Model, the parser obtains 2.0% improvement on the SQL accuracy and 3.6% improvement on the execution accuracy, which registers new state-of-the-art performance. This also confirms our assumption that the parsers can benefit from better quality of contextual encoders that jointly reason over utterances and schemas.

4.6 Discussion and Analysis

4.6.1 Analysis for Spider Dataset

We further select examples from the Spider development set, presented in Table 4.6, to show the improved prediction of our model. The *baseline* system refers to RAT-SQL+BART Encoder model and *our model* refers to the RAT-SQL+GAP Model Encoder. Overall, our model achieves better column selection performance, either explicit matching between the schema and utterance (e.g. in *Example 1*, *how much does it weigh* should match `weight` instead of `pet_age`), or implicit matching (e.g. in *Example 4*, *arriving in ASY Airport* should match `DestAirport` instead of `Airline`). Furthermore, our model can handle complex questions better (e.g. in *Example 5*, our model can generate `HAVING avg(LifeExpectancy) < 72` condition).

4.6.2 Impact of Learning Objectives

We investigate four different learning objectives in this work, namely Masked Language Model (MLM), Column Prediction (CPred), Column Recovery (CRec) and SQL Generation (GenSQL). We conduct the ablation study on Spider development set to compare the first three learning objectives under two different conditions: One is with GenSQL learning objective and the other one is without. We use the IRNet based model in the ablation study because it is more efficient in training than RAT-SQL based model, and it can achieve comparable performance. We also want to show that our GAP Model is plugin-able and can augment different semantic parsers. Table 4.7 shows the ablation results.

Example 1
Utterance: What type of pet is the youngest animal, and how much does it weigh?
Baseline: `SELECT PetType, pet_age FROM Pets ORDER BY pet_age ASC LIMIT 1`
Our Model: `SELECT PetType , weight FROM pets ORDER BY pet_age ASC LIMIT 1`

Example 2
Utterance: What is the average and maximum age for each pet type?
Baseline: `SELECT T2.PetType, Avg(T3.Age), Max(T2.pet_age) FROM Has_Pet AS T1 JOIN Pets AS T2 ON T1.pet_id = T2.pet_id JOIN Student AS T3 ON T1.student_id = T3.student_id GROUP BY T2.PetType`
Our Model: `SELECT avg(pet_age) , max(pet_age) , pettype FROM pets GROUP BY pettype`

Example 3
Utterance: What are the names of the singers who performed in a concert in 2014?
Baseline: `SELECT T2.Name FROM singer_in_concert AS T1 JOIN singer AS T2 ON T1.singer_id = T2.singer_id JOIN concert AS T3 ON T1.convert_id = T3.concert_id WHERE T1.concert_ID = 2014`
Our Model: `SELECT T2.name FROM singer_in_concert AS T1 JOIN singer AS T2 ON T1.singer_id = T2.singer_id JOIN concert AS T3 ON T1.concert_id = T3.concert_id WHERE T3.year = 2014`

Example 4
Utterance: Count the number of United Airlines flights arriving in ASY Airport.
Baseline: `SELECT Count(*) FROM airlines AS T1 JOIN flights AS T2 ON T2.Airline = T1.uid WHERE T1.Airline = 'United Airlines' AND T2.Airline = 'ASY'`
Our Model: `SELECT count(*) FROM AIRLINES AS T1 JOIN FLIGHTS AS T2 ON T2.Airline = T1.uid WHERE T1.Airline = 'United Airlines' AND T2.DestAirport = 'ASY'`

Example 5
Utterance: What are the different continents and the total popuation and average life expectancy corresponding to each, for continents that have an average life expectancy less than 72?
Baseline: `SELECT Count(*), Avg(LifeExpectancy), Avg(LifeExpectancy) FROM country WHERE LifeExpectancy < 72 GROUP BY country.Continent`
Our Model: `SELECT sum(Population) , avg(LifeExpectancy) , Continent FROM country GROUP BY Continent HAVING avg(LifeExpectancy) < 72`

Example 6
Utterance: Give the ids of documents that have between one and two paragraphs.
Baseline: `SELECT T2.Document_ID FROM Paragraphs AS T1 JOIN Documents AS T2 ON T1.Document_ID = T2.Document_ID GROUP BY T1.Document_ID HAVING Count(*) < 2`
Our Model: `SELECT Document_ID FROM Paragraphs GROUP BY Document_ID HAVING count(*) BETWEEN 1 AND 2`

Table 4.6: Selected Examples.

Model	Dev. Acc.	
IRNet + BERT (Ours)	0.620	
IRNet + TaBERT	0.652	
IRNet + RoBERTa (Ours)	0.658	
Learning Objectives	w/o GenSQL	w/ GenSQL
baseline	0.680	0.699
MLM	0.697	0.717
CPred	0.699	0.710
CRec	0.705	0.719
MLM + CPred	0.704	0.716
MLM + CRec	0.711	0.728
MLM + CPred+ CRec	0.715	0.723

Table 4.7: Ablation study on different learning objectives.

The first section of the Table 4.7 shows the results of three baseline systems that are based on IRNet model: IRNet + BERT, IRNet + TaBERT and IRNet + RoBERTa. These results confirm that improving the encoder quality of the semantic parser is a promising direction to pursue.

In the second section of the Table 4.7, we present detailed ablation study results. Without the GenSQL learning objective, compared with baseline (IRNet + BART Encoder), the three learning objectives (MLM, CPred, CRec) can improve the performance of the parser, with a 1.7%, 1.9% and 2.5% increase, respectively. This indicates that these learning objectives improve the encoding quality of the transformer encoder. Based on the standard unsupervised learning objective MLM, we observe that the CPred and CRec learning objectives are helpful, which lead the model to the accuracy of 0.704 and 0.711, respectively. When we further combine the three learning objectives, the semantic parser's effectiveness is further boosted, achieving the accuracy of 0.715, a 3.5% increase over its baseline.

With the GenSQL learning objective, the comparison of these three learning objectives is based on a higher baseline with the accuracy of 0.699. This indicates that the GenSQL learning objective is valuable. Under this experimental condition, we observe that the MLM learning objective brings consistent improvement over the baseline with 1.8% increase in accuracy. For the CPred and CRec, the accuracy is boosted by 1.1% and 2.0%, respectively. When we combine the MLM with the CPred, we only observe comparable results with the MLM, without further significant improvement. However, the CRec learning objective brings the MLM a step forward, achieving 0.728 on the accuracy.

The combination of the three learning objectives under w/ GenSQL condition improves 2.4% on accuracy over the baseline. These results show that GenSQLand CRecare two salient learning objectives, leading the model to obtain an accuracy of more than 0.720, registering a new state-of-the-art performance on public development set on Spider.

4.6.3 Analysis of Pre-Training Inputs

Learning Objective	Dev. Acc.
baseline	0.680
MLM	0.697
MLM w/o utterance	0.678 (-1.7%)
MLM w/o schema	0.679 (-1.6%)
MLM (surrounding text)	0.679 (-1.6%)
CRev	0.705
CRev w/o utterance	0.688 (-1.7%)
CRev (surrounding text)	0.697 (-0.8%)

Table 4.8: The ablation study on different inputs for the pre-training based on the IRNet based model.

Whether to use utterance in pre-training: To prove that the utterance is beneficial in the pre-training, we conduct an ablation study by comparing the pre-trained models which are trained with and without utterance. Our experiments are based on the MLM and CRec learning objectives because the other two (CPred and GenSQL) require the utterance as the input based on their task definitions. Similarly, we use IRNet as our base parser.

The experimental results on Spider development set are shown in Table 4.8. As we can see, if the GAP Model is trained with MLM learning objective without utterances as part of the input, the semantic parser performance drops to 0.678 from 0.697, which is lower than the baseline (0.680) by 0.2%. For the CRec learning objective, the accuracy drops from 0.705 to 0.688, a 1.7% decrease, if the GAP is trained without utterance. Even though, CRec learning objective trained without utterances is still helpful, which improves the baseline model by 0.8%. This aligns with our analysis of the CRec learning objective: model can leverage two information sources to recover the column name. If there are no utterances, the model can only use the signals the cell values provide to recover the column name. Furthermore, when the model can access more contextual information, which is provided by the utterance, the model can learn better encoding ability by learning to align the cell values and the column names in the utterances.

Whether to use schema in pre-training: Another input choice is to only keep the utterances in the pre-training. This experimental setting is to justify that the model's improvement is not solely from better utterance representation. This input strategy is only applicable to the MLM learning objective as the schema is a necessary component for other learning objectives. As shown in the MLM w/o schema entry in Table 4.8, the model performance drops to 0.679, indicating that learning joint utterance and schema representation is necessary for this task.

Whether to use the generated text or the surrounding text of the table: The value of the generated text is already justified by the learning objectives such as CPred or GenSQL, because the definitions of these learning objectives require the generated utterances that cannot be obtained from the surrounding text of the table (denoted as surrounding text). Here, we further rationalize our generation-augmented framework on MLM and CRec learning objectives by replacing the generated text with the surrounding text.

The results are presented in the entries of MLM (surrounding text) and CRec (surrounding text) of Table 4.8. Overall, we can observe that the generation technique is superior to using the surrounding text as a proxy in the MLM and CRec learning objectives, considering the models drop 1.6% and 0.8% on the accuracy, respectively. We also find that the CReclearning objective is more robust for pre-training, given that the fine-tuned model performance gets less influence compared with the one with MLM learning objective.

4.6.4 Analysis of Pre-trained Model

As the GAP Model provides gains on the text-to-SQL benchmarks, understanding what they learn is important. Following previous work [268, 169, 170], we design a probing task, Column-Value Matching (CVM), to examine the extent to which the model can align the cell values in the utterances and the columns, i.e., the probes need to predict which column the cell value belongs to.

Specifically, given the column spans and cell value spans (part of utterances), we can obtain their representations with contextual encoders such as BART or GAP Model Encoder, and an average pooling layer. We further compress the representations into another space with linear transformation, denoted as $\{\mathbf{c}_j\}$ and $\mathbf{v}_i$, respectively. The probability of selecting column c_j given cell value v_i is determined by $p(\mathbf{c}_j|\mathbf{v}_i) \propto exp(\mathbf{v}_i \mathbf{c}_j)$. During training, the parameters of language model encoders are fixed. Here, we conduct the probing task training on the SPIDER dataset.

Note that the unavailability of span annotations of cell values in SPIDER dataset leads to further data pre-processing. Since human annotation is costly, we try to annotate the spans by automatically aligning the cell values in SQL to the utterance tokens. For a cell value used in SQL, assuming it has n tokens, we obtain all n-grams from the utterance, and select the best candidate based on the fuzzy matching score (determined by Levenshtein Distance) when the score is higher than a threshold (we use 60 in our experiment).[6] For integers in the SQL, we also leverage a dictionary to map it to English words when searching for their matches. If n-gram candidates are founded, the cell value will be used in the probing experiment. During the training, the encoder (e.g. BART Encoder or GAP MODEL Encoder) is fixed and only the parameters of probes are tune. The probes are optimized with Adam optimizer with cross-entropy loss function. The learning rate is $1e-5$ and the model is trained for 100 epochs on SPIDER dataset with a batch size of 96. The evaluation metric is instance-level accuracy, i.e., the prediction is correct if every cell value used in the utterance is matched with the correct column.

Model	Match Acc.
BART Encoder	23.17
GAP MODEL (MLM) Encoder	32.72
GAP MODEL (MLM + CRec) Encoder	36.78
GAP MODEL (MLM + CPred) Encoder	44.51

Table 4.9: Results of Value-Column Matching Probing Task.

The results are shown in Table 4.9. We report the accuracy of the BART Encoder model as our probing baseline, which achieves the accuracy of 23.17%. With GAP MODEL (MLM) Encoder, the accuracy raises to 32.72%, indicating that the model learns to align the cell values and column names implicitly. By providing stronger supervision, the MLM+CRec based model and MLM+CPred based models obtain higher accuracy (36.78% and 44.51%), showing that the models capture more alignment signals, contributing to better semantic parser performance.

[6] https://github.com/seatgeek/fuzzywuzzy

4.7 Summary

In this chapter, we spot three pain points in the Text-to-SQL semantic parsing task, and propose a generation-augmented pre-training framework to alleviate them, with four different learning objectives. Experimental results on Spider dataset and Criteria-to-SQL dataset show the effectiveness of this framework, which achieves state-of-the-art performance on both datasets.

Chapter 5

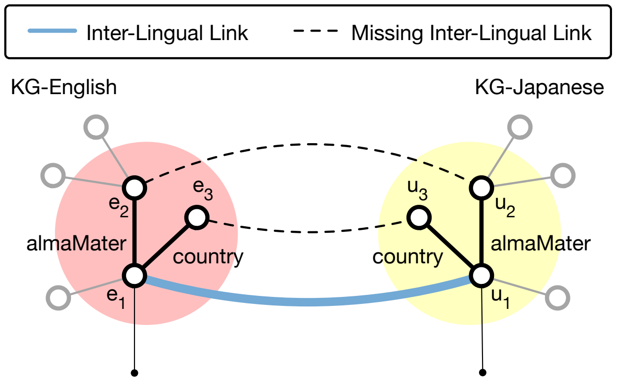
Aligning Cross-Lingual Entities with Multi-Aspect Information

Multilingual knowledge graphs (KGs), such as YAGO and DBpedia, represent entities in different languages. The task of cross-lingual entity alignment is to match entities in a source language with their counterparts in target languages. In this work, we investigate embedding-based approaches to encode entities from multilingual KGs into the same vector space, where equivalent entities are close to each other. Specifically, we apply graph convolutional networks (GCNs) to combine multi-aspect information of entities, including topological connections, relations, and attributes of entities, to learn entity embeddings. To exploit the literal descriptions of entities expressed in different languages, we propose two uses of a pre-trained multilingual BERT model to bridge cross-lingual gaps. We further propose two strategies to integrate GCN-based and BERT-based modules to boost performance. Extensive experiments on two benchmark datasets demonstrate that our method significantly outperforms existing systems. This work is based on:

- Hsiu-Wei Yang*, Yanyan Zou*, Peng Shi*, Wei Lu, Jimmy Lin, Xu Sun. Aligning Cross-Lingual Entities with Multi-Aspect Information. *In Proceedings of EMNLP (Empirical Methods in Natural Language Processing)*, November, 2019.

5.1 Introduction

A growing number of multilingual knowledge graphs (KGs) have been built, such as DBpedia [31], YAGO [431, 378], and BabelNet [311], which typically represent real-world knowl-



English: University of Toronto		Japanese: トロント大学	
Attribute	Value	Attribute	Value
Name	University of Toronto	大学名	トロント大学
Type	Public University	学校種別	州立
Found Date	1827-03-15	創立年	1827
Campus	Ontario	キャンパス	セントジョージ（トロント）
Former Name	King's College	旧名	キングスカレッジ
	⋮		⋮
Descriptions			
The University of Toronto is a public research university in Toronto, Ontario, Canada ⋯		トロント大学 は、オンタリオ州、トロントに本部を置くカナダの州立大学である ⋯	

Figure 5.1: An example fragment of two KGs (in English and Japanese) connected by an inter-lingual link (ILL). In addition to the graph structures (top) consisting of entity nodes and typed relation edges, KGs also provide attributes and literal descriptions of entities (bottom).

edge as separately-structured monolingual KGs. Such KGs are connected via inter-lingual links (ILLs) that align entities with their counterparts in different languages, exemplified by Figure 5.1 (top). Highly-integrated multilingual KGs contain useful knowledge that can benefit many knowledge-driven cross-lingual NLP tasks, such as machine translation [302] and cross-lingual named entity recognition [94]. However, the coverage of ILLs among existing KGs is quite low [56]: for example, less than 20% of the entities in DBpedia are covered by ILLs. The goal of cross-lingual entity alignment is to discover entities from different monolingual KGs that actually refer to the same real-world entities, i.e., discovering the missing ILLs.

Formally, in a multilingual knowledge graph $\mathcal{G}$, we use $\mathcal{L}$ to denote the set of languages that $\mathcal{G}$ contains and $\mathcal{G}_i = \{E_i, R_i, A_i, V_i, D_i\}$ to represent the language-specific knowledge graph in language $L_i \in \mathcal{L}$. E_i, R_i, A_i, V_i and D_i are sets of entities, relations, attributes, values of attributes, and literal descriptions, each of which portrays one aspect of an entity. The graph $\mathcal{G}_i$ consists of relation triples $\langle h_i, r_i, t_i \rangle$ and attribute triples $\langle h_i, a_i, v_i \rangle$ such that $h_i, t_i \in E_i$, $r_i \in R_i$, $a_i \in A_i$ and $v_i \in V_i$. Each entity is accompanied by a literal description consisting of a sequence of words in language L_i, e.g., $\langle h_i, d_{h,i} \rangle$ and $\langle t_i, d_{t,i} \rangle$, $d_{h,i}, d_{t,i} \in D_i$.

Given two knowledge graphs $\mathcal{G}_1$ and $\mathcal{G}_2$ expressed in source language L_1 and target language L_2, respectively, there exists a set of pre-aligned ILLs $I\left(\mathcal{G}_1, \mathcal{G}_2\right) = \{(e, u) \,| e \in E_1, u \in E_2\}$ which can be considered training data. The task of cross-lingual entity alignment is to align entities in $\mathcal{G}_1$ with their cross-lingual counterparts in $\mathcal{G}_2$, i.e., discover missing ILLs.

Traditional methods for this task apply machine translation techniques to translate entity labels [428]. The quality of alignments in the cross-lingual scenario heavily depends on the quality of the adopted translation systems. In addition to entity labels, existing KGs also provide multi-aspect information of entities, including topological connections, relation types, attributes, and literal descriptions expressed in different languages [31, 496], as shown in Figure 5.1 (bottom). The key challenge of addressing such a task thus is how to better model and use provided multi-aspect information of entities to bridge cross-lingual gaps and find more equivalent entities (i.e., ILLs).

Recently, embedding-based solutions [59, 437, 575, 475, 56] have been proposed to unify multilingual KGs into the same low-dimensional vector space where equivalent entities are close to each other. Such methods only make use of one or two aspects of the aforementioned information. For example, [575] relied only on topological features while [437] and [475] exploited both topological and attribute features. [56] proposed a co-training algorithm to combine topological features and literal descriptions of entities. However, combining these multi-aspect information of entities (i.e., topological connections, relations and attributes, as well as literal descriptions) remains under-explored.

In this work, we propose a novel approach to learn cross-lingual entity embeddings by using all aforementioned aspects of information in KGs. To be specific, we propose two variants of GCN-based models, namely Man and Hman, that incorporate multi-aspect features, including topological features, relation types, and attributes into cross-lingual entity embeddings. To capture semantic relatedness of literal descriptions, we fine-tune the pre-trained multilingual BERT model [103] to bridge cross-lingual gaps. We design two strategies to combine GCN-based and BERT-based modules to make alignment decisions. Experiments show that our method achieves new state-of-the-art results on two benchmark datasets.

5.2 Related Work

KG Alignment. Research on KG alignment can be categorized into two groups: monolingual and multilingual entity alignment. As for monolingual entity alignment, main approaches align two entities by computing string similarity of entity labels [394, 458, 312] or graph similarity [367, 344, 22]. Recently, [452] proposed an embedding-based model that incorporates attribute values to learn the entity embeddings.

To match entities in different languages, [474] leveraged only language-independent information to find possible links cross multilingual Wiki knowledge graphs. Recent studies learned cross-lingual embeddings of entities based on TransE [36], which are then used to align entities across languages. [56] designed a co-training algorithm to alternately learn multilingual entity and description embeddings. [475] applied GCNs with the connectivity matrix defined on relations to embed entities from multilingual KGs into a unified low-dimensional space.

In this work, we also employ GCNs. However, in contrast to [475], we regard relation features as input to our models. In addition, we investigate two different ways to capture relation and attribute features.

Multilingual Sentence Representations. Another line of research related to this work is aligning sentences in multiple languages. Recent works [166, 79, 117] studied cross-lingual sentence classification via zero-shot learning. [198] proposed a sequence-to-sequence multilingual machine translation system where the encoder can be used to produce cross-lingual sentence embeddings [15]. Recently, BERT [103] has advanced the state-of-the-art on multiple natural language understanding tasks. Specifically, multilingual BERT enables learning representations of sentences under multilingual settings. We adopt BERT to produce cross-lingual representations of entity literal descriptions to capture their semantic relatedness, which benefits cross-lingual entity alignment.

5.3 Models

In this section, we first introduce two GCN-based models, namely Man and Hman, that learn entity embeddings from the graph structures. Second, we discuss two uses of a multilingual pre-trained BERT model to learn cross-lingual embeddings of entity descriptions: PointwiseBert and PairwiseBert. Finally, we investigate two strategies to integrate the GCN-based and the BERT-based modules.

5.3.1 Cross-Lingual Graph Embeddings

Graph convolutional networks (GCNs) [213] are variants of convolutional networks that have proven effective in capturing information from graph structures, such as dependency graphs [150], abstract meaning representation graphs [149], and knowledge graphs [475]. In practice, multi-layer GCNs are stacked to collect evidence from multi-hop neighbors. Formally, the l-th GCN layer takes as input feature representations $H^{(l-1)}$ and outputs $H^{(l)}$:

$$H^{(l)} = \phi\left(\tilde{D}^{-\frac{1}{2}}\tilde{A}\tilde{D}^{-\frac{1}{2}}H^{(l-1)}W^{(l)}\right) \tag{5.1}$$

where $\tilde{A} = A + I$ is the adjacency matrix, I is the identity matrix, $\tilde{D}$ is the diagonal node degree matrix of $\tilde{A}$, $\phi(\cdot)$ is ReLU function, and $W^{(l)}$ represents learnable parameters in the l-th layer. $H^{(0)}$ is the initial input.

GCNs can iteratively update the representation of each entity node via a propagation mechanism through the graph. Inspired by previous studies [565, 475], we also adopt GCNs in this work to collect evidence from multilingual KG structures and to learn cross-lingual embeddings of entities. The primary assumptions are: (1) equivalent entities tend to be neighbored by equivalent entities via the same types of relations; (2) equivalent entities tend to share similar or even the same attributes.

Multi-Aspect Entity Features. Existing KGs [31, 431, 378] provide multi-aspect information of entities. In this section, we mainly focus on the following three aspects: topological connections, relations, and attributes. The key challenge is how to utilize the provided features to learn better embeddings of entities. We discuss how we construct raw features for the three aspects, which are then fed as inputs to our model. We use X_t, X_r and X_a to denote the topological connection, relation, and attribute features, individually.

The topological features contain rich neighborhood proximity information of entities, which can be captured by multi-layer GCNs. As in [475], we set the initial topological features to $X_t = I$, i.e., an identity matrix serving as index vectors (also known as the featureless setting), so that the GCN can learn the representations of corresponding entities.

In addition, we also consider the relation and attribute features. As shown in Figure 5.1, the connected relations and attributes of two equivalent entities, e.g., "*University of Toronto*" (English) and "トロント大学" (Japanese), have a lot of overlap, which can benefit cross-lingual entity alignment. Specifically, they share the same relation types, e.g., "country" and "almaMater", and some attributes, e.g., "foundDate" and "創立年". To capture relation information, [395] proposed RGCN with relation-wise parameters. However, with respect to this task, existing KGs typically contain thousands of relation types but few pre-aligned ILLs. Directly adopting RGCN may introduce too many parameters for the limited training data and thus cause overfitting. [475] instead simply used the unlabeled GCNs [213] with two proposed measures (i.e., functionality and inverse functionality) to encode the information of relations into the adjacency matrix. They also considered attributes as input features in their architecture. However, this approach may lose information about relation types. Therefore, we regard relations and attributes of entities as bag-of-words features to explicitly model these two aspects. Specifically, we construct count-based *N-hot* vectors X_r and X_a for these two aspects of features, respectively, where the (i, j) entry is the count of the j-th relation (attribute) for the corresponding entity e_i. Note that we only consider the top-F most frequent relations and attributes to avoid data sparsity issues. Thus, for each entity, both of its relation and attribute features are F-dimensional vectors.

Man. Inspired by [475], we propose the *Multi-Aspect Alignment Network* (Man) to capture the three aspects of entity features. Specifically, three l-layer GCNs take as inputs the triple-aspect features (i.e., X_t, X_r, and X_a) and produce the representations $H_t^{(l)}$, $H_r^{(l)}$, and $H_a^{(l)}$ according to Equation 5.1, respectively. Finally, the multi-aspect entity embedding is:

$$H_m = [H_t^{(l)} \oplus H_a^{(l)} \oplus H_r^{(l)}] \tag{5.2}$$

where $\oplus$ denotes vector concatenation. H_m can then feed into alignment decisions.

Such fusion through concatenation is also known as *Scoring Level Fusion*, which has been proven simple but effective for capturing multi-modal semantics [43, 209, 76]. It is worth noting that the main differences between Man and the work of [475] are two fold: First, we use the same approach as in [213] to construct the adjacency matrix, while [475] designed a new connectivity matrix as the adjacency matrix for the GCNs. Second, Man explicitly regards the relation type features as model input, while [475] incorporated such relation information into the connectivity matrix.

Hman. Note that Man propagates relation and attribute information through the graph structure. However, for aligning a pair of entities, we observe that considering the relations and attributes of neighboring entities, besides their own ones, may introduce noise. Merely focusing on relation and attribute features of the current entity could be a better choice. Thus, we propose the *Hybrid Multi-Aspect Alignment Network* (Hman) to better model such diverse features, shown in Figure 5.2. Similar to Man, we still leverage the l-th layer of a GCN to obtain topological embeddings $H_t^{(l)}$, but exploit feedforward neural networks to obtain the embeddings with respect to relations and attributes. The feedforward neural networks consist of one fully-connected (FC) layer and a highway network layer [430]. The reason we use highway networks is consistent with the conclusions of [306], who conducted

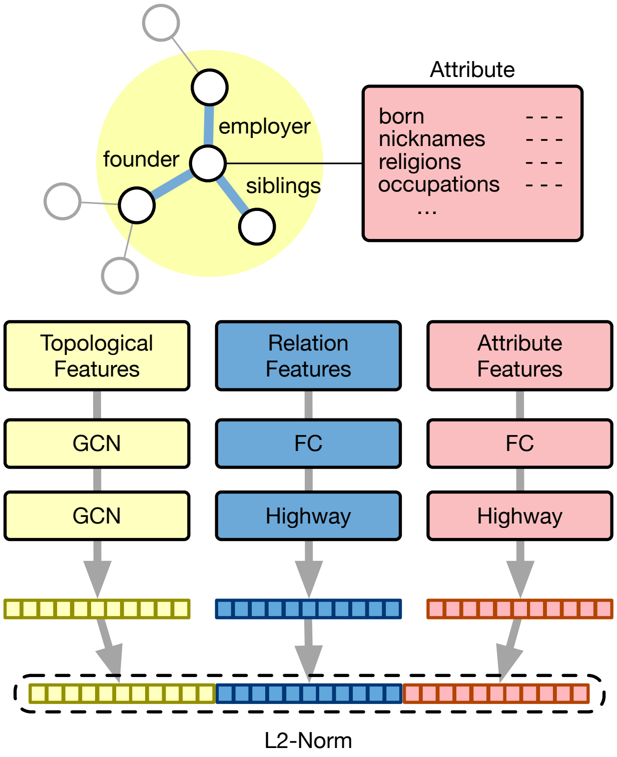


Figure 5.2: Architecture of Hman.

a design space exploration of neural models for entity matching and found that highway networks are generally better than FC layers in convergence speed and effectiveness.

Formally, these feedforward neural networks are defined as:

$$\begin{aligned} S_f &= \phi(W_f^{(1)} X_f + b_f^{(1)}) \\ T_f &= \sigma(W_f^t S_f + b_f^t) \\ G_f &= \phi(W_f^{(2)} S_f + b_f^{(2)}) \cdot T_f + S_f \cdot (1 - T_f) \end{aligned} \tag{5.3}$$

where $f \in \{r, a\}$ and X_f refer to one specific aspect (i.e., relation or attribute) and the corresponding raw features, respectively, $W_f^{(1,2,t)}$ and $b_f^{(1,2,t)}$ are model parameters, $\phi(\cdot)$ is ReLU function, and $\sigma(\cdot)$ is sigmoid function. Accordingly, we obtain the hybrid multi-aspect entity embedding $H_y = [H_t^{(l)} \oplus G_r \oplus G_a]$, to which ℓ_2 normalization is further applied.

Model Objective. Given two knowledge graphs, $\mathcal{G}_1$ and $\mathcal{G}_2$, and a set of pre-aligned entity

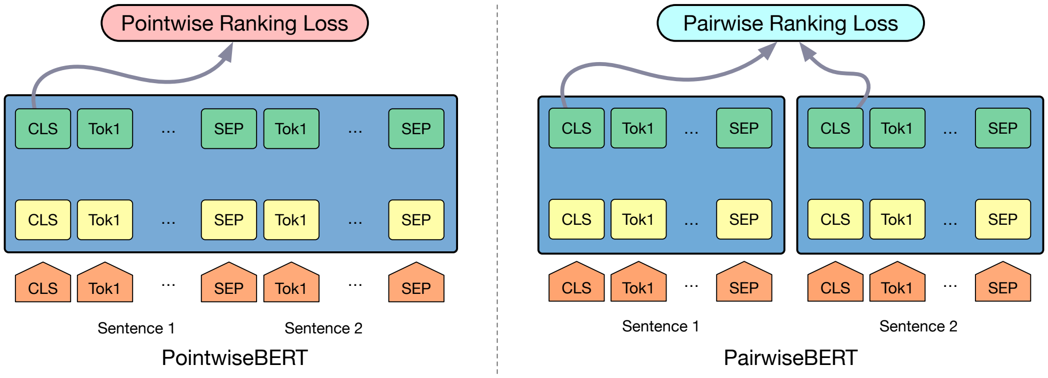


Figure 5.3: Architecture overview of PointwiseBert (left) and PairwiseBert (right).

pairs $I\left(\mathcal{G}_1, \mathcal{G}_2\right)$ as training data, our model is trained in a supervised fashion. During the training phase, the goal is to embed cross-lingual entities into the same low-dimensional vector space where equivalent entities are close to each other. Following [475], our margin-based ranking loss function is defined as:

$$\begin{aligned} J = \sum_{(e_1,e_2)\in I} \sum_{(e_1',e_2')\in I'} & [\rho(h_{e_1}, h_{e_2}) + \beta \\ & - \rho(h_{e_1'}, h_{e_2'})]_+ \end{aligned} \tag{5.4}$$

where $[x]_+ = \max\{0, x\}$, I' denotes the set of negative entity alignment pairs constructed by corrupting the gold pair $(e_1, e_2) \in I$. Specifically, we replace e_1 or e_2 with a randomly-chosen entity in E_1 or E_2. $\rho(x, y)$ is the ℓ_1 distance function, and $\beta > 0$ is the margin hyperparameter separating positive and negative pairs.

5.3.2 Cross-Lingual Textual Embeddings

Existing multilingual KGs [31, 311, 378] also provide literal descriptions of entities expressed in different languages and contain detailed semantic information about the entities. The key observation is that literal descriptions of equivalent entities are semantically close to each other. However, it is non-trivial to directly measure the semantic relatedness of two entities' descriptions, since they are expressed in different languages.

Recently, Bidirectional Encoder Representations from Transformer (BERT) [103] has advanced the state-of-the-art in various NLP tasks by heavily exploiting pre-training based on language modeling. Of special interest is the multilingual variant, which was trained with Wikipedia dumps of 104 languages. The spirit of BERT in the multilingual scenario is to project words or sentences from different languages into the same semantic space. This aligns well with our objective—bridging gaps between descriptions written in different languages. Therefore, we propose two methods for applying multilingual BERT, PointwiseBert and PairwiseBert, to help make alignment decisions.

PointwiseBert. A simple choice is to follow the basic design of BERT and formulate the entity alignment task as a text matching task. For two entities e_1 and e_2 from two KGs in L_1 and L_2, denoting source language and target language, respectively, their textual descriptions are d_1 and d_2, consisting of word sequences in two languages. The model takes as inputs [CLS] d_1 [SEP] d_2 [SEP], where [CLS] is the special classification token, from which the final hidden state is used as the sequence representation, and [SEP] is the special token for separating token sequences, and produces the probability of classifying the pair as equivalent entities. The probability is then used to rank all candidate entity pairs, i.e., ranking score. We denote this model as PointwiseBert, shown in Figure 5.3 (left).

This approach is computationally expensive, since for each entity we need to consider all candidate entities in the target language. One solution, inspired by the work of [417], is to reduce the search space for each entity with a *reranking strategy* (see Section 5.3.3).

PairwiseBert. Due to the heavy computational cost of PointwiseBert, semantic matching between all entity pairs is very expensive. Instead of producing ranking scores for description pairs, we propose PairwiseBert to encode the entity literal descriptions as cross-lingual textual embeddings, where distances between entity pairs can be directly measured using these embeddings.

The PairwiseBert model consists of two components, each of which takes as input the description of one entity (from the source or target language), as depicted in Figure 5.3 (right). Specifically, the input is designed as [CLS] $d_1(d_2)$ [SEP], which is then fed into PairwiseBert for contextual encoding. We select the hidden state of [CLS] as the textual embedding of the entity description for training and inference. To bring the textual embeddings of cross-lingual entity descriptions into the same vector space, a similar ranking loss function as in Equation 5.4 is used.

5.3.3 Integration Strategy

Sections 5.3.1 and 5.3.2 introduce two modules that separately collect evidence from knowledge graph structures and the literal descriptions of entities, namely graph and textual embeddings. In this section, we investigate two strategies to integrate these two modules to further boost performance.

Reranking. As mentioned in Section 5.3.2, the PointwiseBert model takes as input the concatenation of two descriptions for each candidate–entity pair, where conceptually we must process every possible pair in the training set. Such a setting would be cost prohibitive computationally.

One way to reduce the cost of PointwiseBert would be to ignore candidate pairs that are unlikely to be aligned. [371] showed that uncertainty-based sampling can provide extra improvements in ranking. Following this idea, the GCN-based models (i.e., Man and Hman) are used to generate a candidate pool whose size is much smaller than the entire universe of entities. Specifically, GCN-based models provide top-q candidates of target entities for each source entity (where q is a hyperparameter). Then, the PointwiseBert model produces a ranking score for each candidate–entity pair in the pool to further rerank the candidates. However, the weakness of such a reranking strategy is that performance is bounded by the quality of (potentially limited) candidates produced by Man or Hman.

Weighted Concatenation. With the textual embeddings learned by PairwiseBert denoted as H^B and graph embeddings denoted as H^G, a simple way to combine the two modules is by weighted concatenation:

$$H^C = \tau \cdot H^G \oplus (1 - \tau) \cdot H^B \tag{5.5}$$

where H^G is the graph embeddings learned by either Man or Hman, and τ is a factor to balance the contribution of each source (where τ is a hyperparameter).

5.3.4 Entity Alignment

After we obtain the embeddings of entities, we leverage ℓ_1 distance to measure the distance between candidate–entity pairs. A small distance reflects a high probability for an entity pair to be aligned as equivalent entities. To be specific, with respect to the reranking strategy, we select the target entities that have the smallest distances to a source entity in the vector space learned by Man or Hman as its candidates. For weighted concatenation, we employ the ℓ_1 distance of the representations of a pair derived from the concatenated embedding, i.e., H^C, as the ranking score.

5.4 Experimental Setup

5.4.1 Datasets and Settings

We evaluate our methods over two benchmark datasets: DBP15K and DBP100K [437]. Table 5.1 outlines the statistics of both datasets, which contain 15,000 and 100,000 ILLs, respectively. Both are divided into three subsets: Chinese-English (ZH-EN), Japanese-English (JA-EN), and French-English (FR-EN).

Following previous work [437, 475], we adopt the same split settings in our experiments, where 30% of the ILLs are used as training and the remaining 70% for evaluation. *Hits@k* is used as the evaluation metric [36, 437, 475], which measures the proportion of correctly aligned entities ranked in the top-k candidates, and results in both directions, e.g., ZH-EN and EN-ZH, are reported.

In all our experiments, we employ two-layer GCNs and the top 1000 (i.e., F=1000) most frequent relation types and attributes are included to build the N-hot feature vectors. For the MAN model, we set the dimensionality of topological, relation, and attribute embeddings to 200, 100, and 100, respectively. When training HMAN, the hyperparameters are dependent on the dataset sizes due to GPU memory limitations. For DBP15K, we set the dimensionality of topological embeddings, relation embeddings, and attribute embeddings to 200, 100, and 100, respectively. For DBP100K, the dimensionalities are set to 100, 50, and 50, respectively. We adopt SGD to update parameters and the numbers of epochs are set to 2,000 and 50,000 for MAN and HMAN, respectively. The margin β in the loss function is set to 3. The balance factor τ is determined by grid search, which shows that the best performance lies in the range from 0.8 to 0.7. For simplicity, τ is set to 0.8 in all associated experiments. Multilingual BERT-base models with 768 hidden units are used in POINTWISEBERT and PAIRWISEBERT. We additionally append one more FC layer to the representation of [CLS] and reduce the dimensionality to 300. Both BERT models are fine-tuned using the Adam optimizer.

5.5 Results

5.5.1 Results on Graph Embeddings

We first compare MAN and HMAN against previous systems [155, 58, 437, 475]. As shown in Table 5.2, MAN and HMAN consistently outperform all baselines in all scenarios, especially

Datasets		DBP15K				
		Entities	Rel.	Attr.	Rel.triples	Attr.triples
ZH-EN	Chinese	66,469	2,830	8,113	153,929	379,684
	English	98,125	2,317	7,173	237,674	567,755
JA-EN	Japanese	65,744	2,043	5,882	164,373	354,619
	English	95,680	2,096	6,066	233,319	497,230
FR-EN	French	66,858	1,379	4,547	192,191	528,665
	English	105,889	2,209	6,422	278,590	576,543

Datasets		DBP100K				
		Entities	Rel.	Attr.	Rel.triples	Attr.triples
ZH-EN	Chinese	106,517	4,431	16,152	329,890	1,404,615
	English	185,022	3,519	14,459	453,248	1,902,725
JA-EN	Japanese	117,836	2,888	12,305	413,558	1,474,721
	English	118,570	2,631	13,238	494,087	1,738,803
FR-EN	French	105,724	1,775	8,029	409,399	1,361,509
	English	107,231	2,504	13,170	513,382	1,957,813

Table 5.1: Statistics of DBP15K and DBP100K. Rel. and Attr. stand for relations and attributes, respectively.

Hman. It is worth noting that, in this case, Man and Hman use as much information as [475], while [437] require extra supervised information (relations and attributes of two KGs need to be aligned in advance). The performance improvements confirm that our model can better utilize topological, relational, and attribute information of entities provided by KGs.

To explain why Hman achieves better results than Man, recall that Man collects relation and attribute information by the propagation mechanism in GCNs where such knowledge is exchanged through neighbors, while Hman uses feedforward networks to capture expressive features directly from the input feature vectors without propagation. As we discussed before, it is not always the case that neighbors of equivalent entities share similar relations or attributes. Propagating such features through linked entities in GCNs may introduce noise and thus harm performance.

Moreover, we perform ablation studies on the two proposed models to investigate the effectiveness of each component. We alternatively remove each aspect of features (i.e., topological, relation, and attribute features) and the highway layer in Hman, denoted as

Model	ZH → EN			EN→ ZH			JA → EN			EN→ JA			FR → EN			EN→ FR		
	@1	@10	@50	@1	@10	@50	@1	@10	@50	@1	@10	@50	@1	@10	@50	@1	@10	@50
	DBP15K																	
[155]	21.2	42.7	56.7	19.5	39.3	53.2	18.9	39.9	54.2	17.8	38.4	52.4	15.3	38.8	56.5	14.6	37.2	54.0
[58]	30.8	61.4	79.1	24.7	52.4	70.4	27.8	57.4	75.9	23.7	49.9	67.9	24.4	55.5	74.4	21.2	50.6	69.9
[437]	41.1	74.4	88.9	40.1	71.0	86.1	36.2	68.5	85.3	38.3	67.2	82.6	32.3	66.6	83.1	32.9	65.9	82.3
[475]	41.2	74.3	86.2	36.4	69.9	82.4	39.9	74.4	86.1	38.4	71.8	83.7	37.2	74.4	86.7	36.7	73.0	86.3
Man	46.0	79.4	90.0	41.5	75.6	88.3	44.6	78.8	90.0	43.0	77.1	88.7	43.1	79.7	91.7	42.1	79.1	90.9
Man w/o TE	21.5	55.0	79.4	20.2	53.6	78.8	15.0	44.0	69.9	14.3	44.0	70.6	10.2	34.5	59.5	10.8	35.2	60.3
Man w/o RE	45.6	79.1	89.5	41.1	75.0	87.3	44.2	78.7	89.8	43.0	76.9	88.1	42.8	79.7	91.4	42.1	78.9	90.6
Man w/o AE	43.7	77.1	87.8	39.2	72.9	85.5	43.2	77.6	88.4	41.2	74.9	86.6	42.9	79.6	91.0	41.5	78.9	90.5
Hman	**56.2**	**85.1**	**93.4**	**53.7**	**83.4**	**92.5**	**56.7**	**86.9**	**94.5**	**56.5**	**86.6**	**94.6**	**54.0**	**87.1**	**95.0**	**54.3**	**86.7**	**95.1**
Hman w/o TE	3.2	16.7	38.3	3.5	17.2	38.5	5.4	22.3	45.5	5.2	22.0	45.5	2.4	13.9	35.3	2.2	13.7	35.3
Hman w/o RE	50.2	78.4	86.5	49.3	78.6	87.0	52.6	81.6	89.1	52.4	81.1	89.8	52.7	84.2	91.4	52.0	83.9	91.1
Hman w/o AE	49.2	81.0	89.8	48.8	80.9	90.0	52.2	83.3	91.6	51.5	83.1	91.6	52.3	85.6	93.7	52.3	85.1	93.2
Hman w/o HW	46.8	76.1	84.1	46.0	76.2	84.6	50.5	79.5	87.5	49.9	79.1	87.5	51.9	82.7	90.9	51.6	82.5	90.6
	DBP100K																	
[155]	-	16.9	-	-	16.6	-	-	21.1	-	-	20.9	-	-	22.9	-	-	22.6	-
[58]	-	34.3	-	-	29.1	-	-	33.9	-	-	27.2	-	-	44.8	-	-	39.1	-
[437]	20.2	41.2	58.3	19.6	39.4	56.0	19.4	42.1	60.5	19.1	39.4	55.9	26.2	54.6	70.5	25.9	51.3	66.9
[475]	23.1	47.5	63.8	19.2	40.3	55.4	26.4	55.1	70.0	21.9	44.4	56.6	29.2	58.4	68.7	25.7	50.5	59.8
Man	27.2	54.2	**72.8**	24.7	50.2	**69.0**	30.0	60.4	**77.3**	26.6	54.4	71.2	31.6	64.0	77.3	28.8	59.3	73.4
Man w/o TE	11.8	28.6	47.7	11.2	28.3	47.9	7.4	21.7	39.4	7.2	21.6	39.8	5.4	19.4	38.2	5.1	18.8	37.1
Man w/o RE	26.5	53.4	72.1	23.9	49.2	67.9	29.8	60.3	77.1	26.3	53.9	70.6	31.0	63.2	76.4	28.4	58.4	72.2
Man w/o AE	25.5	51.7	70.4	22.8	47.6	66.3	29.4	59.4	76.1	25.9	52.9	69.7	30.8	62.7	75.8	28.1	57.8	71.5
Hman	**29.8**	**54.6**	69.5	**28.7**	**53.3**	**69.0**	**34.3**	**63.3**	76.1	**33.8**	**63.0**	**76.7**	**37.5**	**67.7**	**77.7**	**37.6**	**68.1**	**78.5**
Hman w/o TE	6.8	20.3	39.2	7.2	21.0	39.4	3.0	11.5	27.3	3.3	11.8	28.0	0.5	3.5	11.1	0.5	3.4	11.4
Hman w/o RE	28.0	50.3	62.3	28.2	50.6	62.9	30.3	54.9	64.8	30.2	55.9	66.9	32.8	60.3	69.1	33.3	60.9	69.8
Hman w/o AE	25.7	46.4	57.3	25.5	64.7	57.9	29.6	55.1	66.1	29.9	56.1	67.4	32.5	59.2	67.8	32.9	59.4	68.4
Hman w/o HW	25.2	46.0	57.9	25.2	45.9	57.9	28.6	52.6	62.2	28.5	53.0	63.0	32.8	60.9	70.0	32.9	60.2	70.3

Table 5.2: Results of using graph information on DBP15K and DBP100K. @1, @10 and @50 refer to Hits@1, Hits@10 and Hits@50, respectively.

w/o TE (RE, AE, and HW). As reported in Table 5.2, we observe that after removing relation or attribute features, the performance of Hman and Man drops across all datasets. These figures prove that these two aspects of features are useful in making alignment decisions. On the other hand, compared to Man, Hman shows more significant performance drops, which also demonstrates that employing the feedforward networks can better categorize relation and attribute features than GCNs in this scenario. Interestingly, looking at the two variants Man w/o TE and Hman w/o TE, we can see the former achieves better results. Since Man propagates relation and attribute features via graph structures, it can still implicitly capture topological knowledge of entities even after we remove the topological features. However, Hman loses such structure knowledge when topological features are excluded, and thus its results are worse. From these experiments, we can conclude that the topological information is playing an indispensable role in making alignment decisions.

Model	ZH → EN			EN→ ZH			JA → EN			EN→ JA			FR → EN			EN→ FR		
	@1	@10	@50	@1	@10	@50	@1	@10	@50	@1	@10	@50	@1	@10	@50	@1	@10	@50
	DBP15K																	
Translation∗	55.7	67.6	74.3	40.3	54.2	62.2	74.6	84.5	89.1	61.9	72.0	77.2	-	-	-	-	-	-
JAPE + Translation∗	73.0	90.4	96.6	62.7	85.2	94.2	82.8	94.6	98.3	75.9	90.7	96.0	-	-	-	-	-	-
PairwiseBert	74.3	94.6	98.8	74.8	94.7	99.0	78.6	95.8	98.5	78.3	95.4	98.4	95.2	99.2	99.6	94.9	99.2	99.7
Man (Rerank)	84.2	93.6	94.8	82.1	91.8	93.1	89.4	94.0	94.8	88.2	93.3	94.0	93.1	95.2	95.4	93.1	95.3	95.4
Hman (Rerank)	86.5	95.9	96.9	85.8	94.1	95.3	89.0	96.0	97.3	89.0	96.0	97.5	95.3	97.7	97.8	95.2	97.9	98.1
Man (Weighted)	85.4	98.2	99.7	83.8	97.7	99.5	90.8	98.8	99.7	89.9	98.5	99.5	96.8	99.6	99.8	96.7	99.7	**99.9**
Hman (Weighted)	**87.1**	**98.7**	**99.8**	**86.4**	**98.5**	**99.8**	**93.5**	**99.4**	**99.9**	**93.3**	**99.3**	**99.9**	**97.3**	**99.8**	**99.9**	**97.3**	**99.8**	**99.9**
	DBP100K																	
PairwiseBert	65.1	85.1	92.6	66.2	85.8	92.9	67.7	86.5	93.1	67.9	86.4	93.2	93.2	97.9	98.9	93.4	98.0	98.9
Man (Rerank)	59.5	62.1	62.2	55.9	58.2	58.2	65.5	68.2	68.4	59.9	62.1	62.3	69.7	70.4	70.5	65.5	66.2	66.2
Hman (Rerank)	58.9	61.2	61.3	57.9	60.2	60.3	66.9	69.4	69.6	67.0	69.6	69.8	72.1	72.9	73.0	72.7	73.5	73.5
Man (Weighted)	**81.4**	**94.9**	**98.2**	**80.5**	94.1	97.7	84.3	95.4	98.3	81.5	94.2	97.6	96.2	99.3	**99.7**	95.7	99.1	99.6
Hman (Weighted)	81.1	94.3	97.8	80.3	**94.5**	**97.9**	**85.2**	**96.1**	**98.4**	**84.6**	**96.1**	**98.5**	**96.5**	**99.4**	**99.7**	**96.5**	**99.5**	**99.8**

Table 5.3: Results of using both graph and textual information on DBP15K and DBP100K. @1, @10, and @50 refer to Hits@1, Hits@10, and Hits@50, respectively. ∗ indicates results are taken from [437].

5.5.2 Results with Textual Embeddings

In this section, we discuss empirical results involving the addition of entity descriptions, shown in Table 5.3. Applying literal descriptions of entities to conduct cross-lingual entity alignment is relatively under-explored. The recent work of [56] used entity descriptions in their model; however, we are unable to make comparisons with their work, as we do not have access to their code and data. Since we employ BERT to learn textual embeddings of descriptions, we consider systems that also use external resources, like Google Translate,[1] as our baselines. We directly take results reported by [437], denoted as "Translation" and "JAPE+Translation".

The PointwiseBert model is used with GCN-based models, which largely reduces the search space, as indicated by Man (Rerank) and Hman (Rerank), where the difference is that the candidate pools are given by Man and Hman, respectively. For DBP15K, we select top-200 candidate target entities as the candidate pool while for DBP100K, top-20 candidates are selected due to its larger size. The reranking method does lead to performance gains across all datasets, where the improvements are dependent on the quality of the candidate pools. Hman (Rerank) generally performs better than Man (Rerank) since Hman recommends more promising candidate pools.

The PairwiseBert model learns the textual embeddings that map cross-lingual de-

[1] https://cloud.google.com/translate/

	English	**Chinese**
ILL pair	Casino_Royale_(2006_film) (3)	007大戰皇家賭場 (3)
Features	starring, starring, distributor	starring, starring, language
Neighbors	Daniel_Craig (1), Eva_Green (4), Columbia_Pictures (9)	丹尼爾克雷格 (1), 伊娃格蓮 (4), 英語 (832)

Table 5.4: Case study of the noise introduced by the propagation mechanism.

scriptions into the same space, which can be directly used to align entities. The results are listed under PairwiseBert in Table 5.3. We can see that it achieves good results on its own, which also shows the efficacy of using multilingual descriptions. Moreover, such textual embeddings can be combined with graph embeddings (learned by Man or Hman) by weighted concatenation, as discussed in Section 5.3.3. The results are reported as Man (Weighted) and Hman (Weighted), respectively. As we can see, this simple operation leads to significant improvements and gives excellent results across all datasets. However, it is not always the case that KGs provide descriptions for every entity. For those entities whose descriptions are not available, the graph embeddings would be the only source for making alignment decisions.

5.6 Discussion and Analysis

In this section, we describe a case study to understand the performance gap between Hman and Man. The example in Table 5.4 provides insights potentially explaining this performance gap. We argue that Man introduces unexpected noise from heterogeneous nodes during the GCN propagation process. We use the number in parentheses (*) after entity names to denote the number of relation features they have.

In this particular example, the two entities "*Casino_Royale_(2006_film)*" in the source language (English) and "*007大戰皇家賭場*" in the target language (Chinese) both have three relation features. We notice that the propagation mechanism introduces some neighbors which are unable to find cross-lingual counterparts from the other end, marked in red. Considering the entity "英語" (English), a neighbor of "*007大戰皇家賭場*", no counterparts can be found in the neighbors of "*Casino_Royale_(2006_film)*". We also observe that "英語" (English) is a pivot node in the Chinese KG and has 832 relations, such as "語言" (Language), "官方語言" (Official Language), and "頻道語言" (Channel Language). In this case, propagating features from neighbors can harm performance. In fact, the feature sets of the ILL pair already convey information that captures their similarity (e.g., the “starring” marked in blue are shared twice). Therefore, by directly using feedforward networks, Hman is able to effectively capture such knowledge.

5.7 Summary

In this work, we focus on the task of cross-lingual entity alignment, which aims to discover mappings of equivalent entities in multilingual knowledge graphs. We proposed two GCN-based models and two uses of multilingual BERT to investigate how to better utilize multi-aspect information of entities provided by KGs, including topological connections, relations, attributes, and entity descriptions. Empirical results demonstrate that our best model consistently achieves state-of-the-art performance across all datasets. In the future, we would like to apply our methods to other multilingual datasets such as YAGO and BabelNet. Also, since literal descriptions of entities are not always available, we will investigate alternative ways to design graph-based models that can better capture structured knowledge for this task.

Chapter 6

Cross-Lingual Training for Relevance Transfer

Dense retrieval and reranking have shown great success in passage ranking in English. However, its effectiveness in document retrieval for non-English languages remains unexplored due to the limitation of training resources. In this work, we explore different transfer techniques for document retrieval and reranking from English annotations to multiple non-English languages. Our experiments on the test collections in six languages (Chinese, Arabic, French, Hindi, Bengali, and Spanish) from diverse language families reveal that zero-shot model-based transfer improves the search quality in non-English mono-lingual retrieval.

This chapter is presented in:

- Peng Shi, Rui Zhang, He Bai, Jimmy Lin. Cross-Lingual Training of Dense Retrievers for Document Retrieval. *Proceedings of the 1st Workshop on Multilingual Representation Learning*, November, 2021.

- Peng Shi, He Bai, Jimmy Lin. Cross-Lingual Training of Neural Models for Document Ranking. *In Proceedings of EMNLP (Empirical Methods in Natural Language Processing): Findings*, November, 2020.

6.1 Introduction

Relevance matching is one core problem in information retrieval, which is to rank the contents (document, passage, or table) by the relevance to a user's query. Recently, the retrieval-reranking pipeline has become a paradigm for information retrieval, where an efficient retriever is firstly leveraged to obtain a set of candidates and then an effective reranker is applied to those candidates to obtain better ranking results.

Traditionally, TF-IDF or BM25 are popular implementations of first-stage retriever, where the keywords are matched with an inverted index. However, these systems suffer from *vocabulary mismatch* problem [127]. Consider the question *"What is the body of water between England and Ireland?"*, which can be answered by the context *"Irish Sea, arm of the North Atlantic Ocean that separates Ireland from Great Britain."*. A term matching system fails to rank this relevant context in top positions because of the string mismatch between the question and the context, besides the key word *Ireland*. A recently proposed effective and efficient solution is the dense retriever. Dense retrieval uses dense vector representations for semantic encoding and matching. It has shown its effectiveness in open-domain question answering and passage ranking [325, 205]. However, the dense passage retrievers generate the question and context embeddings independently, without token-level interactions, which proves to be effective in the ranking task. Reranking is one component that alleviates the issue. By modeling the token-level matching signals between the query and context, the rerankers can push more relevant contexts in the top positions.

However, most of the existing work focus on high-resource languages such as English, where large-scale annotations are readily accessible. Widely available large-scale datasets such as Natural Questions (NQ) [219] and MS MARCO [318] are used for training dense retrieval encoders or rerankers to achieve state-of-the-art performances in English. Such data is especially hard to obtain for low-resource languages, considering that large amounts of annotations are required for training dense retrievers and rerankers.

In this work, we first explore techniques for leveraging the relevance judgments in a source language, usually a high-resource language such as English, to train dense retrievers for mono-lingual document retrieval in multiple target (non-English) languages. Note that this setting is different from cross-lingual information retrieval (CLIR), where queries and documents are in different languages [413, 533, 197, 265, 534, 66]. Specifically, we examine low-resource techniques including zero-shot model-based transfer and weakly-supervised target language transfer; we also explored the technique by leveraging public translators, such as Google Translate, to improve the language transfer of the dense retriever.

Furthermore, in this work, we also explore diverse methods to train neural document reranking models cross-lingually. While we are aware of two previous papers along these lines [413, 284], this work explores a far broader range of techniques and adds more nuance to previous findings. Beyond the basic approach proposed by these two papers, which we refer to as model-based transfer, we investigate additional approaches involving the translation of the training data, the translation of documents, hybrid models, as well as ensembles – which we broadly characterize into "high resource" and "low resource" settings. We show that various methods alone and in combination can yield robust increases in effectiveness across diverse languages with minimal resources, and that model-based cross-lingual transfer isn't the only way.

6.2 Related Work

Dense retrieval showed its superiority over the traditional term matching based methods such as BM25 or BM25+RM3 query expansion on passage retrieval task [89, 88, 205, 559, 52, 284]. A bi-encoder architecture is used for the dense retrievers, where the queries and documents are mapped into hidden vectors independently without any interaction between them. Compared with term-based sparse retrieval using TF-IDF or BM25, it can capture synonyms or paraphrases by incorporating contexts and provide additional flexibility to learn task-specific representations [254]. Document reranking is another topic in document retrieval. With pre-trained language models, the reranking effectiveness has been improved significantly [323, 520]. A cross-encoder architecture is often used for document reranking, where the tokens of query and the document can have full interactions in the encoder and the matching signals can be easily captured by the models. The multilingual BERT [103] has shown its language transfer abilities over different tasks [489]. [413] were the one of the first to build IR re-rankers based on the mBERT for non-English corpus by leveraging the relevance judgments in English. More recently, [20] and [19] leveraged mBERT and target language annotations to train cross-lingual DPRs.

6.3 Models

6.3.1 Dense Retriever

Dense retriever uses a dense encoder $E_P(\cdot)$ that encodes contexts, either text or table, to a d-dimension vectors and builds an index for collections that are used for retrieval. For the query, a different encoder $E_Q(\cdot)$ is used for mapping the query into d-dimension vector.

Based on the query vector, the closest top k contexts are retrieved from the pre-built index based on the pre-defined distance function. Following [205], we define the distance function as $sim(q, p) = E_Q(q)^\top E_P(p)$.

For the choice of the encoder, in principle, any neural networks can be used. By virtue of more powerful contextual encoding ability of recent large-scale pre-trained language models, models such as BERT and RoBERTa are popular choices of the encoder.

Here we use the BERT as an example. More specifically, the questions and contexts are linearized into sequence of tokens and fed into BERT and the hidden states of `[CLS]` are used as the representation. During inference, we apply both bag-of-words exact term matching such as BM25 or BM25+RM3 and dense retrieval. The relevance score of each candidate combines the term-matching scores with dense retrieval similarity $S_{context} = \alpha \cdot S_{term} + (1 - \alpha) \cdot S_{dense}$ where α is tuned via cross-validation. All candidates are sorted by the above score $S_{context}$ to produce the final output.

6.3.2 Reranker

The work on neural document ranking [521, 89] provides a general method for fine-tuning BERT: The input to the model comprises [`[CLS]`, Q `[SEP]` S `[SEP]`], which is the concatenation of the query Q and a piece of context S, with the special tokens `[CLS]` and `[SEP]`. The final hidden state of the `[CLS]` token is passed to a single layer neural network with a softmax, obtaining the probability that context S is relevant to the query Q.

6.3.3 Cross-Lingual Relevance Transfer of Dense Retriever

To perform cross-lingual transfer of dense retriever from a high-resource source language to low-resource target languages, we investigate two groups of strategies. The first strategy, model-based transfer, directly applies the retrieving model trained on the source language to other target languages in a zero-shot manner. The second strategy explores two data augmentation techniques to build the training data on target languages for finetuning.

Model-based Transfer

By exploiting the zero-shot cross-lingual transfer ability of pre-trained transformers such as mBERT [103] and XLM-Roberta [77], we train the dense retriever encoder in the source language and apply inference directly on target languages. These pre-trained transformers only require raw text in different languages, e.g. Wikipedia, and are trained in a self-supervised manner, so we characterize this approach as “low resource”.

Target Language Transfer

To bridge the language gap between the training and the inference, a direct solution is target language data augmentation. In this work, we explore two techniques for creating a target language transfer set, including generation-based query synthesis and weakly supervised query synthesis.

Generation-based Query Synthesis. The goal of the generation-based query synthesis is to leverage powerful generation models to predict reasonable queries given documents in the target language. We choose the multilingual version of BART (mBART) [272], a pre-trained sequence-to-sequence transformers, as our query generation model. The input of the model is the passage and its learning target is the corresponding query. We use the translate-train technique to obtain the generation models. More specifically, we leverage Google Translate to translate the query-document pairs in English to target languages. In the inference stage, we use the passages in the target language collections as the input and generate corresponding queries in the same language. In our preliminary experiments, we also tried zero-shot transfer that model is trained on English query-document pairs and directly applied to target languages for query inference. However, the generated queries are of low quality, and this observation is also confirmed by [69].

Weakly-supervised Query Synthesis. Wikipedia has documents in varies languages, and it is a good transfer set in cross-lingual training. We can automatically build the target language transfer set without manual annotation effort, by treating the titles of Wikipedia articles as queries and the corresponding documents as positive candidates. We also retrieve the top 1000 documents with BM25 for each query, and the documents except the positive candidate are labeled as negative candidates. Queries whose positive document is not in the retrieved set are removed. In this way, we can obtain query-document pairs in target languages.

Two-stage Training. We apply two-stage training to train the dense retriever encoders. The dense retriever encoders are firstly trained on source language annotated data which are available in a large scale; then the models are finetuned on the synthesized query-document pairs in the target language.

6.3.4 Cross-Lingual Relevance Transfer of Reranker

Our main research question is as follows: Given English (source) training data, how can we bootstrap a good document ranking model in non-English (target) languages? We discuss a number of approaches below, which we characterize as "high" or "low" resource in terms of *annotation effort*.

Model-based Transfer

Following Wu [489], the most obvious approach is to fine-tune mBERT using data in the source language, and apply inference directly on input in the target language. In essence, we follow the same setup as [521], with the exception that we use mBERT instead of (English) BERT. We characterize this approach as "low resource" given that mBERT is pre-trained in a self-supervised manner.

Translation-based Transfer

Training data translation. Instead of relying on mBERT to transfer models of relevance matching across languages, we can translate the English training data into the target language, and then fine-tune mBERT with the translated data.[1] At inference time, we directly apply the model on target-language documents. We considered two translation methods: Google Translate ($\mathrm{MB}_{\mathrm{gt}}$) and a simple embedding-based token-by-token translation approach ($\mathrm{MB}_{\mathrm{wt}}$). We characterize the first as "high resource" given the amount of bitext that is typically necessary to train a high-quality translation system, whereas the second as "low resource" since bilingual lexicons and aligned word embeddings are far easier to create.

Our token-based translation approach is inspired by [178]. The basic idea is to find the best token translation based on the cosine similarity between the token in the source language and candidate tokens in the target language. Specifically, for each token in the source language, the surface form is used for lookup in a bilingual dictionary. If the token has a unique translation, we use the translation directly. If it has multiple translations, we use an empirical scoring function $F(w, w_{t,i})$ to select the best translation. This scoring function calculates the cosine similarity between a candidate translation $w_{t,i}$ and the source

[1] Note that here we are using mBERT in a purely mono-lingual manner since mono-lingual BERT models are not widely available for all target languages.

token w based on its contextual tokens $w_{c,j}$ (in this work, we consider two words in the left context and two words in the right context), as follows:

$$\begin{aligned} F(w, w_{t,i}) &= \gamma \cdot \cos(\mathrm{E}(w), \mathrm{E}(w_{t,i})) \\ &+ (1-\gamma) \cdot \sum_{j=1}^{m} \frac{\cos(\mathrm{E}(w_{t,i}), \mathrm{E}(w_{c,j}))}{(d_j+1)^2} \end{aligned} \tag{6.1}$$

where $\mathrm{E}(w)$ is the bilingual embedding of the token w, d_j is the positional distance between the token w and its contextual token $w_{c,j}$, and γ is a hyperparameter for balancing the effects of the translation pair and the contextual tokens. Following previous work, we set γ to 0.5. If the source language token has no translations, the original surface form is kept unchanged.

Note that model-based transfer uses the *same* model across all languages, whereas this approach requires a separate model for *each* language.

Hybrid transfer. Both approaches above can be combined in a stage-wise fashion: We can first fine-tune mBERT on the English data, and then fine-tune again on the translated training data (we refer to this as the en→gt direction). Alternatively, we can switch the order of fine-tuning (the gt→en direction). In these experiments, we used the output of Google Translate (and hence these are "high resource" approaches).

Document translation. Another way to leverage existing translation capabilities is to translate the documents at search time from the target language into the source language (English), and directly apply the mBERT model that is trained on $\mathrm{MB}_{\mathrm{en}}$. We used Google Translate in this method, and thus it is "high resource".

Ensembles. Ensembles of the above approaches can exploit multiple signal and resources. One approach is to interpolate scores from multiple sources, on a per-document basis: $S_{\text{agg}} = \beta \cdot S_{\text{model-transfer}} + (1-\beta) \cdot S_{\text{doc-translation}}$. This method is denoted $\mathrm{ENS}_{\mathrm{INT}}$, which combines model-based transfer and document translation (from the results, the two most promising techniques). Alternatively, we also experimented with Reciprocal Rank Fusion [81] to aggregate two separate ranked lists, which is denoted $\mathrm{ENS}_{\mathrm{RRF}}$. These methods are "high resource".

For "low resource" ensembles, we aggregated signals from model-based transfer and the token-based approach for translating training data. These signals are either combined by per-document score interpolation or RRF, as per above.

Doc Language	Source	# Topics	# Docs
Chinese	NTCIR 8	73	308,832
Arabic	TREC 2002	50	383,872
French	CLEF 2006	49	171,109
Hindi	FIRE 2012	50	331,599
Bengali	FIRE 2012	50	500,122
Spanish	TREC 3	25	57,868

Table 6.1: Dataset statistics for test collections.

6.4 Experimental Setup

6.4.1 Experimental Setup for Dense Retrieval

We conduct experiments on six test collections in diverse languages (Chinese, Arabic, French, Hindi, Bengali, Spanish). Data statistics are shown in Table 6.1. For the evaluation metrics, we adopt the average precision (AP) up to rank 1000, precision at rank 20 (P@20) and nDCG at rank 20 (nDCG), computed with the `trec_eval` toolkit. The query was used to retrieve the top 1000 hits from the corpus using BM25 or BM25+RM3 query expansion; the default Anserini [516] settings were used. For the dense retrieval, the top 100 hits are retrieved from the index and the similarity scores are combined with the term-matching scores. For the documents that are not in the retrieved set, either from term-matching methods or dense retrieval, 0 score is assigned. Fisher's two-sided, paired randomization test [425] at $p < 0.05$ was applied to test for statistical significance.

For the model-based transfer, we explore two training datasets in the source language, English in our case, including the Natural Question and MS MARCO. Note that Natural Question is an open-domain question answering dataset, where the queries are usually long questions instead of a bag of keywords in the document ranking datasets. This introduces a new gap in query style besides language in the transfer process. For training the query generator in target languages, we obtain the training data by sampling 2000 query-passage pairs from MS MARCO and translate them into target languages same as target benchmarks. For two-stage training, we first train the dense encoders on the MS MARCO dataset and then further tune them on the transfer set.

For dense retrieval, documents are often too long to fit into BERT models for encoding. A common approach to address this issue is to split the long document into segments within fixed length (e.g. 512 tokens), and build an index based on the segments. In this work, we segmented the documents using a sliding window of 5 sentences and a stride of 1 sentence. For each query, the score of document S_{dense} is obtained by averaging the top-3 scores of the retrieved segments. We applied five-fold cross-validation on all datasets, choosing parameter α that yielded the highest AP.

6.4.2 Experimental Setup for Reranking

The test collections used for the reranking experiments are same as dense retrieval experiments. Following standard practice in information retrieval, average precision (AP) up to rank 1000 and precision at rank 20 (P@20) were adopted as the evaluation metrics, computed with the `trec_eval` tool.

For the token-based translation method, we used the MUSE bilingual dictionary [221] and the aligned word embeddings from fastText [202]. For fine-tuning mBERT, we followed the same experimental setup as [7]. We used data from the Microblog (MB) Tracks from TREC 2011–2014 [251] or its translated counterparts, setting aside 75% of the total data for training and the rest for validation, which was used for selecting the best model parameters. We trained each model using cross-entropy loss with a batch size of 16; the Adam optimizer was applied with an initial learning rate of 1×10^{-5}. During fine-tuning, the embeddings were fixed. The model with the highest AP on the validation set was chosen. We ran all experiments on an NVIDIA Tesla V100 16GB with PyTorch version 1.3.0. Each model was trained for up to 15 epochs, with an average running time of approximately two hours.

For retrieval, we used the open-source Anserini IR toolkit [517] with minor modifications based on version 0.6.0 to swap in Lucene Analyzers for different languages. Fortunately, Lucene provides analyzers for all the languages in our test collections. The query was used to retrieve the top 1000 hits from the corpus using BM25 or BM25+RM3 query expansion; default Anserini settings were used in both cases. Reranking with mBERT used the approach with higher AP (either BM25 or BM25+RM3); the top three sentences were considered in aggregating sentence-level evidence. We applied five-fold cross-validation on all datasets and the parameters α, the w_i's, and β were obtained by grid search, choosing the parameters that yielded the highest AP.

	AP	**P@20**	**nDCG**	**AP**	**P@20**	**nDCG**	**AP**	**P@20**	**nDCG**
Model		**NTCIR8-zh**			**TREC2002-ar**			**CLEF2006-fr**	
(**0**) BM25	0.4014	0.3849	0.4757	0.2932	0.3610	0.4056	0.3111	0.3184	0.4458
(**1**) BM25+RM3	0.3384	0.3616	0.4490	0.2783	0.3490	0.3969	0.3421	0.3408	0.4658
(**2**) NQ zero-shot	0.4221▲	0.4164▲	0.5235▲	0.2943	0.3560	0.4012	0.3470	0.3469	0.4726
(**3**) MS zero-shot	0.4167▲	0.4164▲	0.5095▲	0.3024	0.3810▲	0.4285	0.3332	0.3418	0.4573
(**4**) MS → QGen	0.4258▲	0.4336▲	0.5308▲	0.2988	0.3800	0.4276	0.3331	0.3429	0.4564
(**5**) MS → Wiki	0.4135	0.4123▲	0.5055▲	0.3060▲	0.3750	0.4293	0.3456	0.3480	0.4743
		FIRE2012-hi			**FIRE2012-bn**			**TREC3-es**	
(**0**) BM25	0.3867	0.4470	0.5310	0.2881	0.3740	0.4261	0.4197	0.6660	0.6851
(**1**) +RM3	0.3660	0.4430	0.5277	0.2833	0.3830	0.4351	0.4912	0.7040	0.7079
(**2**) NQ zero-shot	0.3939	0.4560	0.5408	0.2898	0.3980	0.4495▲	0.4910	0.6980	0.7007
(**3**) MS zero-shot	0.3944	0.4580	0.5461	0.2896▲	0.3900	0.4449	0.4950	0.7080	0.7171
(**4**) MS → QGen	0.3941	0.4660	0.5527	0.2887	0.3980	0.4486	0.4958▲	0.7180	0.7239
(**5**) MS → Wiki	0.3950	0.4630	0.5497	0.2898▲	0.4050	0.4549	0.4972▲	0.7180	0.7329

Table 6.2: Experimental results on the baselines and our cross-lingual transfer methods. Model (0) and (1) are term matching baselines. Model (2) NQ zero-shot: zero-shot transfer of models trained on Natural Questions. Model (3) MS zero-shot: zero-shot transfer of models trained on MS MARCO. Model (4) MS → QGen: trained on MS MARCO and then finetuned on query generation data requiring external translators. Model (5) MS → Wiki: trained on MS MARCO and then finetuned on query synthesis data from Wikipedia. For nDCG, we report nDCG@20. Significant gains against the baselines are denoted with ▲.

6.5 Results

6.5.1 Results for Dense Retrieval

Our results are shown in Table 6.3. Models (0) and Models (1) show the effectiveness of BM25 and BM25 with RM3 query expansion. For each language, we select the higher P@20 of the two models as the term-based matching baselines. That is, for the French, Bengali and Spanish collections, we use the BM25+RM3 as the term-based matching baseline and for others, we use the BM25.

Finding #1: Zero-shot model-based transfer improves term-based matching. The results of zero-shot model-based transfer are shown in Models (2) and Models (3).

Comparing with the corresponding baselines, we observe that the model-based transfer, either NQ zero-shot or MS zero-shot, can improve the retrieval effectiveness on P@20 for all collections, except the NQ zero-shot on TREC3-es dataset. We do not observe a clear winner between NQ and MS, though. For example, Models (2) perform better on Chinese and French collections; while Models (3) yield better retrieval effectiveness on Arabic and Spanish collections. Since mBERT is widely available, mono-lingual retrieval improvements can be obtained ‘‘for free‘‘ with annotated data in English. These results indicate that mBERT-based DPR effectively transfers relevance matching across languages.

Finding #2: Target language transfer benefits certain collections, and Wiki query synthesis is better than query generation. Target language transfer results are shown in Models (4) and Models (5). MS → QGen and MS → Wiki denote two-stage training strategy with different transfer sets, where QGen denotes generation-based query synthesis and Wiki denotes weakly supervised query synthesis from Wikipedia. By comparing the Models (4) with Models (3), we observe the second stage training with generation-based query-document pairs can improve the effectiveness of P@20 over the zero-shot model-based transfer on Chinese, French, Hindi, Bengali and Spanish collections. However, there is no difference over AP for all collections. By comparing the Models (5) with Models (3), we find that the second stage training with weakly-supervised training data can improve the P@20 over the zero-shot baselines on French, Hindi, Bengali and Spanish collections.

Furthermore, by comparing these two transfer sets, we observe that, except for the Chinese collection, the Wiki obtains better retrieval effectiveness than QGen, which requires external translators and training query generators on different languages.

6.5.2 Results for Reranking

Our results are shown in Table 6.3. Models (0) and (1) show the effectiveness of BM25 and BM25 with RM3 query expansion. We see that with the exception of the French and Spanish collections, RM3 actually decreases effectiveness. This interesting finding was not further investigated, as our goal was simply to establish a strong baseline; however, these results are consistent with [284]. For each language, we selected the higher of the two models as the starting point of reranking as well as the baseline for comparisons below. We organize results into five findings below. Unless otherwise stated, Fisher's two-sided, paired randomization test [425] at $p < 0.05$ was applied to test for statistical significance, with Bonferroni corrections as appropriate.

				AP	**P@20**	**AP**	**P@20**	**AP**	**P@20**
Model	**Train**	**Test**	**R**	**NTCIR8-zh**		**TREC2002-ar**		**CLEF2006-fr**	
(**0**) BM25				0.4014	0.3849	0.2932	0.3610	0.3111	0.3184
(**1**) +RM3				0.3384	0.3616	0.2783	0.3490	0.3421	0.3408
(**2**) mBERT	MB$_{\text{en}}$	doc	l	0.4488$^{\blacktriangle}$	0.4507$^{\blacktriangle}$	0.3081	0.4050$^{\blacktriangle}$	0.3631$^{\blacktriangle}$	0.3633$^{\blacktriangle}$
(**3**) mBERT	MB$_{\text{gt}}$	doc	h	0.4618	0.4616	0.3148	0.4120	0.3596	0.3531
(**4**) mBERT	MB$_{\text{wt}}$	doc	l	0.4220	0.4322	0.3022	0.3950	0.3557	0.3551
(**5**) mBERT	MB$_{\text{en}}$	doc$_{\text{gt}}$	h	0.4513	0.4534	0.3272¶	0.4020	0.3800¶	0.3745
(**6**) Hybrid	MB$_{\text{en}\to\text{gt}}$	doc	h	0.4525§	0.4534§	0.3209§	0.4140§	0.3706	0.3694§
(**7**) Hybrid	MB$_{\text{gt}\to\text{en}}$	doc	h	0.4423§	0.4438§	0.3075	0.4120§	0.3490	0.3459
(**8**) ENS$_{\text{INT}}$	MB$_{\text{en}}$	+doc$_{\text{gt}}$	h	0.4561	0.4521	0.3269	0.4060	0.3818	0.3694
(**9**) ENS$_{\text{RRF}}$	MB$_{\text{en}}$	+doc$_{\text{gt}}$	h	0.4582	0.4562	0.3237	0.4060	0.3767	0.3694
(**10**) ENS$_{\text{INT}}$	MB$_{\text{en+wt}}$	doc	l	0.4490	0.4507	0.3086	0.4030	0.3628	0.3622
(**11**) ENS$_{\text{RRF}}$	MB$_{\text{en+wt}}$	doc	l	0.4404	0.4486	0.3074	0.4010	0.3613	0.3500
				FIRE2012-hi		**FIRE2012-bn**		**TREC3-es**	
(**0**) BM25				0.3867	0.4470	0.2881	0.3740	0.4197	0.6660
(**1**) +RM3				0.3660	0.4430	0.2833	0.3830	0.4912	0.7040
(**2**) mBERT	MB$_{\text{en}}$	doc	l	0.4207$^{\blacktriangle}$	0.4800$^{\blacktriangle}$	0.3101$^{\blacktriangle}$	0.4060$^{\blacktriangle}$	0.5056$^{\blacktriangle}$	0.7240
(**3**) mBERT	MB$_{\text{gt}}$	doc	h	0.4150	0.4710	0.2975	0.3890	0.5051	0.7400
(**4**) mBERT	MB$_{\text{wt}}$	doc	l	0.4289	0.4860	0.3050	0.4070	0.5032	0.7300
(**5**) mBERT	MB$_{\text{en}}$	doc$_{\text{gt}}$	h	0.4240	0.4810	0.3419¶	0.4470	0.5238¶	0.7700¶
(**6**) Hybrid	MB$_{\text{en}\to\text{gt}}$	doc	h	0.4218§	0.4850§	0.3078§	0.4020	0.4996	0.7140
(**7**) Hybrid	MB$_{\text{gt}\to\text{en}}$	doc	h	0.4181§	0.4780	0.3030	0.3950§	0.5058	0.7220
(**8**) ENS$_{\text{INT}}$	MB$_{\text{en}}$	+doc$_{\text{gt}}$	h	0.4320	0.4910	0.3479	0.4530	0.5215	0.7660
(**9**) ENS$_{\text{RRF}}$	MB$_{\text{en}}$	+doc$_{\text{gt}}$	h	0.4283	0.4890	0.3406	0.4320	0.5209	0.7560
(**10**) ENS$_{\text{INT}}$	MB$_{\text{en+wt}}$	doc	l	0.4377	0.4860	0.3112	0.4020	0.5077	0.7260
(**11**) ENS$_{\text{RRF}}$	MB$_{\text{en+wt}}$	doc	l	0.4340	0.4900	0.3127	0.4090	0.5082	0.7240

Table 6.3: Ranking effectiveness of different cross-lingual training methods. "R" = Resource: high or low.

Finding #1: Model-based transfer, model (2), improves upon the baseline, with significant gains (denoted by $^{\blacktriangle}$) everywhere except for AP in Arabic and P@20 in Spanish. Since mBERT is widely available, mono-lingual retrieval improvements can be obtained "for free" with microblog relevance judgments in English. These results indicate that mBERT effectively transfers relevance matching across languages. This finding confirms previous work [413, 284], but see additional discussion below.

Finding #2: Comparing model-based transfer and the two approaches to translating training data, models (3) and (4), it is difficult to spot trends or reach definitive conclusions. Model-based transfer does not consistently beat simply translating the training data. In terms of AP, Google Translate, model (3), outperforms model-based transfer for Chinese and Arabic; token-based translation, model (4), beats model-based transfer in Hindi and achieves comparable scores in Arabic and Spanish. Interestingly, it is not always the case that Google Translate ("high resource") is better than token-based translation ("low resource"); the latter achieves higher AP for Hindi and Bengali. A Tukey's HSD test across models (2–4) showed no significant differences.

These results suggest that model-based transfer is not the only effective approach, and that simply translating the training data is at least competitive; neither [413] nor [284] explored this obvious baseline.

Finding #3: Results show that hybrid two stage training in the en→gt direction, model (6), can further improve over model-based transfer alone or translating training data with Google Translate alone, but the gains are not consistent; lower AP than either models (2) or (3) in Chinese, Bengali, and Spanish. When compared to the baseline, model (6) yields significant improvement on Chinese, Arabic, and Hindi (denoted by §). In the opposite direction, gt→en, while the hybrid model (7) significantly outperforms the baseline in a few cases, it doesn't seem to be consistently more effective than either models (2) or (3). Note that both hybrid approaches are "high resource" since they require Google Translate.

Finding #4: Document translation, model (5), generally beats model transfer, but it requires substantial resources, such as large amounts of parallel text for training a translation system. Because all our documents are in the newswire domain, the output of Google Translate is quite reasonable. Since this approach avoids language mismatch between training and test, it can outperform the model-based transfer approach: these improvements are significant (denoted by ¶) for the Spanish collection on both metrics, and for the Arabic, Bengali, and French collections on AP.

Finding #5: In general, ensembles outperform model transfer alone, with the "high resource" approaches beating the "low resource" approaches (as expected). Comparing the interpolation and RRF methods, we see no consistent trends. A Tukey's HSD test showed no significant differences between the four ensemble methods.

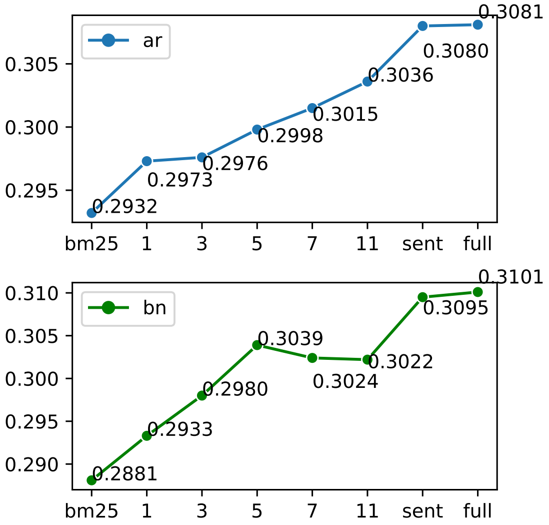


Figure 6.1: AP results on TREC02-ar and FIRE12-bn.

6.6 Discussion and Analysis

Given the effectiveness of model transfer, we additionally investigated a research question focused on model (2): How much contextual information does mBERT rely on besides term matching?

Inspired by the query-centric assumption [487] that relevance information is localized in the contexts around query terms, we conducted the following experiments: For each query term, we only kept the texts around the matched tokens in each sentence within a window size, and used only those contexts for reranking. We tried window size 1 (only the matched query terms are kept), 3 (the matched query terms with their left and right tokens), 5, 7, 11, and "sentence" (the entire sentence is kept if at least one query token matched). If the segments are from the same sentence, they are concatenated to form a new "sentence".

Experimental results are shown in Figure 6.1 for two representative collections. For comparison, we also repeat results of the baseline, either model (0) or (1), denoted *bm25* in the figure, and the results of model (2), denoted *full* in the figure. We can see that as the window size increases, AP tends to rise as well. This seems intuitive, as context is needed for relevance matching. Furthermore, results show that some words critical for determining relevance are located quite far from the query terms; these are discarded when the window size is too small, leading to lower AP scores. However, if we only keep sentences that have at least one query term, the ranking effectiveness is already comparable to using all sentences (0.3080 vs. 0.3081 in Arabic, 0.3095 vs. 0.3101 in Bengali). This simple filter can decrease the inference time needed for ranking 60% to 80% depending on the different characteristics of the collections.

6.7 Summary

For document dense retrieval, we investigate the effectiveness of three transfer techniques for document ranking from English training data to low-resource target languages. Our experiments in six languages demonstrate that zero-shot transfer of mBERT-based dense retrieval models improves traditional term-based matching method, and finetuning on augmented data in target languages can further benefit certain collections. For the document reranking, as a high-level summary, our experiments confirm that mBERT can enable cross-lingual training of document ranking models. However, mBERT's "multi-lingual capacity" for direct model-based transfer does not appear to be consistently better than other approaches of bridging language gaps. For example, simple approaches such as token-based translation of the training data also work well. However, model-based transfer requires only a single model, whereas the latter requires a model for each language. Overall, our work contributes to a better understanding of how relevance judgments in high-resource languages can be leveraged to improve search in low(er)-resources languages.

Chapter 7

Cross-lingual Text-to-SQL Semantic Parsing with Representation Mixup

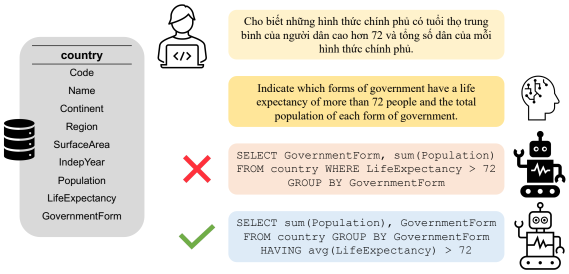
We focus on the cross-lingual Text-to-SQL semantic parsing task, where the parsers are expected to generate SQL for non-English utterances based on English database schemas. Intuitively, English translation as side information is an effective way to bridge the language gap, but noise introduced by the translation system may affect parser effectiveness. In this work, we propose a **Re**presentation Mi**x**up Framework (**Rex**) for effectively exploiting translations in the cross-lingual Text-to-SQL task. Particularly, it uses a general encoding layer, a transition layer, and a target-centric layer to properly guide the information flow of the English translation. Experimental results on CSpider and VSpider show that our framework can benefit from cross-lingual training and improve the effectiveness of semantic parsers, achieving state-of-the-art performance.

This work is based on:

- Peng Shi, Linfeng Song, Lifeng Jin, Haitao Mi, He Bai, Jimmy Lin and Dong Yu, Cross-lingual Text-to-SQL Semantic Parsing with Representation Mixup. *Findings of EMNLP, 2022.*

7.1 Introduction

The task of semantic parsing is to translate natural language utterances into meaning representations, such as lambda calculus [248] or a programming language [529, 570, 542]. More



Cho biết những hình thức chính phủ có tuổi thọ trung bình của người dân cao hơn 72 và tổng số dân của mỗi hình thức chính phủ.

Indicate which forms of government have a life expectancy of more than 72 people and the total population of each form of government.

✗

```
SELECT GovernmentForm, sum(Population)
FROM country WHERE LifeExpectancy > 72
GROUP BY GovernmentForm
```

✓

```
SELECT sum(Population), GovernmentForm
FROM country GROUP BY GovernmentForm
HAVING avg(LifeExpectancy) > 72
```

Figure 7.1: An illustration of the cross-lingual Text-to-SQL task. The utterance in English: "Find the government form name and total population for each government form whose average life expectancy is longer than 72.". The automatic translation fails to translate "trung bình" in Vietnamese into "average" in English.

recently, Text-to-SQL semantic parsing, using SQL queries as the meaning representation, has attracted increasing attention from both academia and industry researchers [570, 463, 537, 99, 415, 398].

Benefiting from recently annotated large-scale datasets [570, 542], research in Text-to-SQL has been greatly expedited. Moreover, due to the development of encoder-decoder pre-trained models [233, 366], semantic parsers have been improved significantly, benefiting from contextualized representations [257, 398]. However, these advances have been achieved mostly in English, leaving other languages underexplored. Systems that can handle non-English inputs well are in urgent need to enhance the user experience for non-English speakers. Nevertheless, the performance of current cross-lingual Text-to-SQL systems is still far from satisfactory. Taking Figure 7.1 as an example, a Vietnamese question is asked based on the English database schema, and the system is expected to generate the corresponding SQL query.

We first define the problem of cross-lingual Text-to-SQL formally. Given an utterance $X = (x_1, x_2, ..., x_n)$ and a database schema S, a Text-to-SQL model is expected to translate the utterance into a valid SQL query. Our framework is based on a standard encoder-decoder architecture. For the task, we assume the existence of an internationalized database where the database schema S is in English. The natural language queries from users are not in English; we denote these non-English languages as target languages. Here, we denote $X_t = \{x_1, x_2, ..., x_{n_t}\}$ as an utterance in the target language with n_t tokens. Similarly, the utterance in the *source language*[1] with n_s tokens is denoted $X_s = \{x_1, x_2, ..., x_{n_s}\}$. We assume the database schema S contains several tables $T \in D$

[1] In this work, the source language refers to English.

with column names $C = \{c_1, c_2, ..., c_{|T|}\}$, where $|T|$ denotes the number of columns in table T.

Because the utterances are non-English, English parsers cannot be directly applied. For tackling the cross-lingual issue, machine translation based methods can be effective solutions, e.g., first translating non-English utterances into English and then using English parsers to generate SQL queries. Here we denote these as *Translate-Test* methods. However, translation systems may introduce noise that causes further errors from the semantic parsers. For example, in Figure 7.1, Google Translate produces the English translation "Indicate which forms of government have a life expectancy of more than 72 people and the total population of each form of government." for the input Vietnamese utterance. One important information is missing in the translation process: "trung bình" should be translated into "average" but Google Translate fails to do so, resulting in the wrong `WHERE` condition prediction: `WHERE LifeExpectancy > 72` instead of `HAVING avg(LifeExpectancy) > 72`.

Another direction for solving the cross-lingual Text-to-SQL problem is to build *target language annotated datasets* and train a target language parser directly [294, 313]. However, these methods fail to leverage knowledge from English parsers (learned from annotated English data), which has the potential to benefit non-English Text-to-SQL parsing.

In this work, we propose **Rex**, a **Re**presentation mi**x**up framework for cross-lingual modeling that utilizes both the English data and the annotated data in the target language. First, REX adopts a two-stage training strategy where the target language models are first initialized with the pre-trained English parser and then trained with the target language data. Using this method, basic schema encoding ability and SQL decoding ability of the English parsers can be reused during target language training. Second, to further make use of English parsers' utterance encoding ability, we use English translations as context augmentation for bridging the cross-lingual gap and facilitating non-English model training.

Instead of simply concatenating the English translation and the target utterance, REX takes a general encoder, a transition layer, and a target-centric encoder to properly guide information flow of English translations, to mitigate the aforementioned issue around noisy translations. In detail, the general encoder generates contextual representations for bilingual utterances and database schemas. The transition layer is leveraged to obtain a cross-lingual mixup representation of the target language utterance, aiming to make the best use of English translations while minimizing the noise introduced. Lastly, the target-centric encoder focuses on the interaction between the target language utterance and the source language schema, by ignoring the side effects caused by translations.

We test our REX framework on two non-English Text-to-SQL semantic parsing datasets, CSpider and VSpider. Experimental results on the benchmarks show that our framework can further improve upon simple yet effective baselines. At the time of publication, Rex achieves state-of-the-art performance on the CSpider leaderboard based on the results on the hidden test set.

Our contributions are summarized as follows:

- We propose the Rex framework for leveraging knowledge from English parsers and information from machine translation by using representation mixup to reduce the negative side effects of automatic translation.
- We conduct a detailed ablation study to show how different configurations of the Rex framework affect parser effectiveness.
- Our framework obtains state-of-the-art performance on the CSpider and VSpider benchmarks.

7.2 Related Work

Cross-lingual Semantic Parsing: The goal of cross-lingual semantic parsing is to process user utterances in multiple languages and convert them into some type of logical representation. Much research progress has been made on this task in recent years.

Dataset creation represents a fundamental contribution that is useful for benchmarking progress [24, 241, 85, 409, 440, 453, 506]. On the other hand, for model development, multilingual pre-trained models are widely applied to the task in a supervised fashion or zero-shot fashion [409, 407, 241].

For example, [407] recently focused on cross-lingual transfer, where the model trained on English data is effectively adapted to other languages. However, their work focuses on single-database semantic parsing and the trained models do not generalize well across different databases. Instead, we focus on cross-database semantic parsing under the supervised learning setting.

Representation Mixup: The term "mixup" was first introduced by [557], referring to a data-agnostic data augmentation method for reducing the memorization issue and improving model robustness. Follow-up work tried to mix up the hidden representations [455] instead of the input. This technique has been widely applied in different directions [514, 120, 242]. Our approach is the first to introduce the idea of mixup to cross-lingual Text-to-SQL semantic parsing.

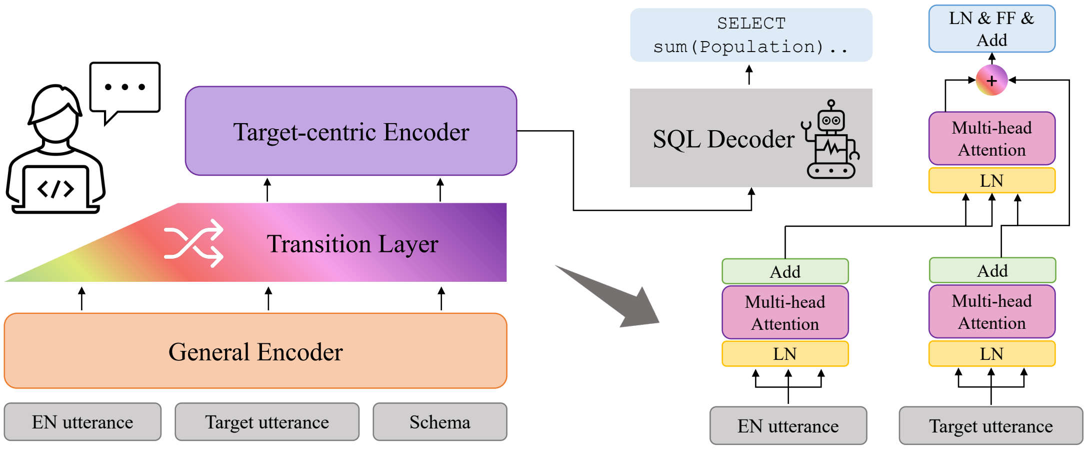


Figure 7.2: Illustration of the REX framework. The left part is the overall framework comprising a general encoder, a transition layer, and a target-centric encoder. The SQL query is decoded autoregressively from the SQL decoder. The right part demonstrates one of the implementations of the transition layer: Explicit Utterance Mixup.

7.3 Models

7.3.1 Baseline: Single-source Input

Here we introduce a sequence-to-sequence based semantic parsing model and a baseline that leverages English translation for non-English Text-to-SQL semantic parsing.

The model for Text-to-SQL semantic parsing has been continuously improved in recent years [525, 464, 147, 463, 390, 418, 73, 179, 358]. Specifically, [398] utilize the pre-trained sequence-to-sequence model T5 as the parser to directly generate SQL, simplifying the intermediate representation design for the grammar-based decoder. In detail, the model input is the concatenation of utterance X and linearized database schema S. With a pre-trained encoder, the contextualized utterance representation H_x and the database schema representation H_d can be obtained. The pre-trained decoder leverages these contextualized hidden states for generating SQL in an autoregressive fashion with constrained decoding. This architecture obtains state-of-the-art performance on English benchmarks without complex modeling. By replacing the T5 model with its multilingual version, mT5 [509], the model can be applied to non-English training data, to obtain parsers that have the ability to handle non-English utterances. We denoted this as single-source target-language training.

7.3.2 Representation Mixup Framework

Here, we propose a representation mixup framework for cross-lingual Text-to-SQL semantic parsing. As shown in Figure 8.1, the encoder stacks several general encoding layers with a transition layer, and some target-centric encoding layers.

First, utterances from the source and target languages can be encoded separately or jointly in the general encoder layers. Then, a transition layer implements the representation mixup between the input sequences, such as source language utterance, target language utterance, and database schema, in a specific layer. Then, a target-centric encoding layer will try to ignore the noise produced by machine translation systems by only focusing on the target language utterance. Finally, a SQL decoder leverages the information from the encoder component to generate a full SQL query.

General Encoder

The general encoder is used to generate basic representations of the utterances, including the source language and target language, and the database schema. Here we discuss two different methods, namely independent encoder [121] and joint encoder. The general encoder is parameterized with m-layer transformers.

Independent General Encoder: Formally, the source and target language utterances can be encoded with m-layer transformers to obtain hidden representation H_s and H_t:

$$\begin{aligned} H_s^m &= \mathtt{Transformers_s}(X_s) \\ H_t^m &= \mathtt{Transformers_t}(X_t) \end{aligned} \tag{7.1}$$

The database schema is first linearized into the token sequence S, following the method used in [398]. An m-layer transformer is applied on the linearized database schema tokens to obtain H_d:

$$H_d^m = \mathtt{Transformers_d}(S) \tag{7.2}$$

One benefit of independent encoding is that the representation of the schema can be shared and reused for all queries to the database, speeding up model inference. Note that the m-layer transformer parameters from different components can be either independent or tied.

Joint General Encoder: The interactions between the schema and the utterances are vital for training an effective semantic parser. Instead of encoding the information independently, joint encoding allows full information interaction between the utterances and schema. More specifically, the input of the joint encoder is the concatenation of the source language utterance, the target language utterance, and the schema:

$$H_s^m, H_t^m, H_d^m = \texttt{Transformers}([X_s; X_t; S]) \tag{7.3}$$

where [;] denotes the concatenation operation. Compared to the independent general encoder, this design requires re-encoding of the schema for each natural language query, where the model can benefit from interactions between the utterances and the schema.

Transition Layer

The transition layer is used to guide the information flow among the different components properly. The output of the transition layer is the representation of the target language utterance and the database schema. Formally, the transition layer is denoted as follows:

$$H_t^{m+1}, H_d^{m+1} = f(H_s^m, H_t^m, H_d^m). \tag{7.4}$$

We discuss different transition layer designs in detail in Section §4.2.

Target-centric Encoder

Only the hidden states of the target language utterance and the schema are kept for further modeling, eliminating the side effects of noisy translations. Formally, k-layer transformers are applied to the concatenation of target language utterance and schema representations:

$$H_t, H_d = \texttt{Transformers}([H_t^{m+1}; H_d^{m+1}]) \tag{7.5}$$

The output of the target-centric encoder is then used in the SQL decoder.

SQL Decoder

The transformer decoder is trained to generate SQL queries token by token. The SQL queries are directly tokenized without any preprocessing. The cross-attention of the transformer decoder is applied to the output of the target-centric encoder. Compared to a grammar-based SQL decoder, SQL queries generated token by token may have syntactic errors. For example, the SQL generator may hallucinate column names that are not from the corresponding database schema. To alleviate this issue, we apply the constrained decoding algorithm Picard [398] to improve the SQL generation quality.

7.3.3 Design of the Transition Layer

Here we introduce the transition layer in detail. The transition layer enhances interactions among the different components (source language utterance, target language utterance, and database schema). The transition layer serves as an information mixer and information flow controller, by fusing information from different components implicitly or explicitly and feeding it to the next layer. The output of the transition layer is the hidden representation of the utterance in the target language and the schema, ignoring the source language information. In this way, the source language utterance only serves as context for the target utterance and/or the schema, without interfering with the decoder behavior explicitly due to unexpected translation noise. Here, we discuss three different transition mechanisms, namely implicit full mixup, implicit utterance mixup, and explicit utterance mixup.

Implicit Full Mixup (IFM): For implicit full mixup, all three components are involved in the modeling. The implicit full mixup layer is parameterized with a single layer transformer:

$$\begin{aligned} &H_t^{m+1}, H_d^{m+1} \\ &= \texttt{Transformer}([H_s^m; H_t^m; H_d^m])[p:q], \end{aligned} \tag{7.6}$$

where $[p:q]$ is the span of the concatenated sequence of target language utterance and schema tokens. Note that the hidden states of the source language utterance only serve as keys and values, while the hidden states of the target language utterance and the schema serve as queries, keys, and values in the multi-head attention. This is different from a vanilla transformer layer.

Implicit Utterance Mixup (IUM): The implicit utterance mixup implements the information flow transition on the utterance part:

$$\begin{aligned} H_t^{m+1} &= \texttt{Transformer}([H_s^m; H_t^m])[p:q] \\ H_d^{m+1} &= H_d^m, \end{aligned} \tag{7.7}$$

where $[p:q]$ is the span of the target language utterance. For the schema representation, skip connections are applied. Similar to implicit full mixup, the hidden states of the utterance from the source language are specifically used as keys and values, while the target utterance hidden states are also used for queries in multi-head attention. The main goal is to enhance the representation of the target language utterance by integrating information from the source language counterparts. This can further reduce the cross-lingual representation discrepancy between the target language utterance and the source language (English) schema.

Explicit Utterance Mixup (EUM): Instead of using fully connected self-attention to learn representation mixup, explicitly controlling the information flow is another strategy. Formally, self-attention is applied independently to the source language utterance and the target language utterance:

$$\begin{aligned} H_s^{'} &= \texttt{MultiHead}(H_s^m, H_s^m, H_s^m) \\ H_t^{'} &= \texttt{MultiHead}(H_t^m, H_t^m, H_t^m). \end{aligned} \tag{7.8}$$

Manifold mixup [514] provides a way to obtain intermediate representations by conducting linear interpolation on the hidden states, leveraging a cross-attention layer:

$$H_{t|s}^{m+1} = \texttt{MultiHead}(H_t^{'}, h_s^{'}, h_s^{'}). \tag{7.9}$$

Following [514], the cross-attention layer shares parameters with the self-attention layer. With the cross-attention layers, by using the hidden states of the target language tokens as queries and the hidden states of the source language tokens as keys and values, the model can extract target-related signals from the source. With the interpolation operation controlled by a mixup ratio λ, the target representation can be enhanced:

$$H_t^{m+1} = \texttt{LayerNorm}(\lambda H_{t|s}^{m+1} + (1-\lambda) H_t^{m+1}), \tag{7.10}$$

where the mixup ratio λ is shared by each example in training and inference. Similar to Equation 7.7, a skip connection layer is applied to the schema representations.

7.3.4 Framework Configurations

By configuring parameters such as m and k, some basic model architectures can be obtained.

Multi-source Input with Concatenation: Concatenation is a simple yet effective baseline for leveraging both the source language utterance and the target language utterance at the same time. By setting $k = 0$ and `Transition Layer` = `None`, our framework is configured as a simple concatenation model. In this case, the decoder can leverage bilingual information to generate the SQL query.

Focused Simple Concatenation: By setting $k = 0$ and `Transition Layer` = `Implicit Full Mixup`, our Rex framework can obtain a new architecture, denoted as *focused* simple concatenation. Different from simple concatenation, the SQL decoder only focuses on the target utterance component and the database schema, ignoring the source language utterance. This can reduce the negative effects caused by noisy machine translations.

7.4 Experimental Setup

Datasets: In this work, we evaluate our framework on two Text-to-SQL semantic parsing datasets, CSpider [294] and VSpider [313], which are **C**hinese and **V**ietnamese cross-domain Text-to-SQL datasets adapted from the Spider benchmark [542].

For CSpider, we use **E**xact Set **M**atch (EM) accuracy as the evaluation metrics. For VSpider, we use both EM accuracy and **T**est-**s**uite (TS) [568] accuracy for evaluation. For Exact Set Match accuracy, the prediction is classified as correct only if all of the components (`SELECT` clause, `WHERE` clause, `HAVING` clause, etc.) are correct. The Test-suite accuracy, which is an improved version of execution accuracy (if the execution results of a predicted SQL query are the same as those of the ground truth SQL query, then it is classified as correct), serves as a tight upper bound for semantic accuracy [568].

Model Training: Our Rex framework is an adapted sequence-to-sequence transformer. Benefiting from pre-trained sequence-to-sequence language models such as BART [233] or T5 [366], performance is significantly improved by finetuning pre-trained models. Furthermore, because our Rex framework is expected to process utterances in multiple languages, our experiments are based on mT5-large, which has 24 layers.

To leverage English annotated data, we conduct two-stage training: We first train mT5-large on the English dataset to obtain a trained parser checkpoint. Note that this trained English parser is based on a standard sequence-to-sequence architecture instead of the Rex framework. The input of the parser is the concatenation of the English utterance and the linearized database schema. This model obtains state-of-the-art performance on the Spider benchmark. We use the checkpoint to initialize our Rex framework and further finetune on target language datasets. During inference, we translate the target language utterances into English as model inputs. To fix the number of model parameters, we always use $m + k + 1 = 24$ in our experiments. For hyper-parameters, we follow [398] in all our experiments. By default, we use Google Translate for all model training and inference.

7.5 Results

CSpider: We report the performance of Rex on CSpider, which can be compared with other state-of-the-art systems on the leaderboard.[2] As shown in Table 7.1, our Rex framework obtains 66.1% EM accuracy on the development set and 59.7% EM accuracy on

[2] https://taolusi.github.io/CSpider-explorer

Model	**Dev.**	**Test**
DG-SQL ([461])	50.4	46.9
XL-SQL	54.9	47.8
RAT-SQL + GraPPa + Adv	59.7	56.2
LGESQL + ELECTRA + QT	64.5	58.1
Single-source		
Target-language Training	63.7	-
Multi-source		
Concatenation	65.5	-
Rex	**66.1**	**59.7**

Table 7.1: Model performance on the CSpider development set and hidden test set with EM accuracy.

the hidden test set, exceeding the best-performing system, LGESQL+ELECTRA+QT, by 1.6% on both the development set and the test set. Our model is based on a joint general encoder, explicit utterance mixup with 0.1 mixup ratio, and setting $m = 16$ and $k = 7$. Our system achieves state-of-the-art performance on the CSpider benchmark at the time of writing.

Comparing the multi-source models to single-source ones, we observe that the extra information can improve parser effectiveness. With multi-source concatenation, the parser is already competitive with the state-of-the-art parser. Our Rex framework further improves the model by 0.6% over the multi-source input with concatenation.

VSpider: Because our data split on VSpider is different from [313], our results are not directly comparable.[3]

The main results for VSpider are shown in Table 7.2. As a single source baseline, target language training obtains 64.2% EM accuracy and 59.0% TS accuracy. Multi-source concatenation outperforms target-language training, with 1.4% improvement on EM accuracy and 2.9% improvement on TS accuracy. Our Rex framework achieves better effectiveness both on EM accuracy and TS accuracy, with 69.0% EM accuracy and 64.5% TS accuracy. The model is based on a joint general encoder, explicit utterance mixup with

[3] [313] split the dataset into training (6831), dev (954), test (1906) sets. In order to prevent data leak from the trained English parser, we keep our splits consistent with Spider: training set (8659) and dev set (1034).

Model	**EM**	**TS**
Single-source		
Target-language Training	64.2	59.0
Multi-source		
Concatenation	65.6	61.9
Rex	**69.0**	**64.5**

Table 7.2: Model performance on the VSpider development set with EM accuracy and TS accuracy.

Model	**zh**			**vi**		
	G.	M.	Δ	G.	M.	Δ
Concat.	65.5	61.4	4.1	65.6	63.0	2.6
Rex	66.1	65.0	1.1	69.0	66.8	2.2

Table 7.3: Robustness with respect to translation error. The performance comparison is conducted based on parsers that use Google Translate (G.) and parsers that use Marian Translate (M.). Concat. denotes multi-source input with the concatenation model. Marian Translate introduces more noise than Google Translate. EM accuracy is reported.

0.3 mixup ratio, and setting $m = 12$ and $k = 11$.

7.6 Discussion and Analysis

To investigate issues in cross-lingual semantic parsing and to better understand our Rex framework, we performed several ablation experiments.

7.6.1 Robustness to Translation Noise

We argue that the models with transition layers are more robust to translation noise than the baseline model using multi-source input with concatenation. To verify this, we conduct an ablation study by testing the models using English translations from different translation systems: Google Translate and Marian Translate [449]. More specifically, the models are first trained with translations obtained from Google Translate, using the simple

Model	**zh**	**vi**
Target language Training	62.5	62.8
En → Target language Training	63.7	64.2

Table 7.4: Ablation study of two-stage training. "Target language training" denotes using target language labeled data to train the parser from scratch and "En→ Target language Training" denotes two-stage training. EM accuracy is the evaluation metric.

concatenation model and the Rex architecture. These models are tested with different translation systems. The results in the G. and M. columns show the performance of models that are tested with data from Google Translate and Marian Translate, respectively. The Δ column shows the performance gap between using Google Translate and Marian Translate.

The experimental results are shown in Table 7.3. For Chinese, the concatenation based model is sensitive to translation noise, degrading from 65.5% to 61.4% when the translation system is switched from Google Translate to Marian Translate. However, performance of the Rex model only drops 1.1% on accuracy, showing better robustness towards the translation noise. Similarly, on the VSpider benchmark, our Rex model shows better robustness than the concatenation model or the model obtained from source-language training.

7.6.2 Effectiveness of Two-stage Training

Here we conduct an ablation study to show that two-stage training is an effective way to leverage annotated English data. With labeled data in the target language, we can train the parser from scratch or apply two-stage training. Results are shown in Table 7.4. We can observe that two-stage training can benefit the parser consistently under different settings. For example, two-stage training can improve 1.2% EM accuracy for Chinese and 1.4% for Vietnamese. These results rationalize our design choice for the Rex framework.

7.6.3 Transition Layer Index and Design

We explore choices of transition layer index under different transition layer designs, including implicit full mixup, implicit utterance mixup, and explicit utterance mixup. We configure $m \in \{4, 8, 12, 16, 20, 23\}$. Note that when $m = 23$, the `Target-centric Encoder` = `None`. For the explicit utterance mixup, we use a fixed ratio controller by setting $\lambda = 0.3$

Model	**4**	**8**	**12**	**16**	**20**	**23**
IFM	66.4	67.5	68.7	68.8	68.6	68.9
IUM	67.6	67.7	68.8	68.4	68.0	67.5
EUM	67.6	67.4	**69.0**	68.5	68.2	66.8

Table 7.5: Ablation study of the transition layer index on the VSpider dev set. IFM denotes implicit full mixup; IUM denotes implicit utterance mixup; EUM denotes explicit utterance mixup. EM accuracy is reported.

ratio	**Easy**	**Med.**	**Hard**	**Extra**	**All**
0.1	88.3	**73.1**	49.4	42.2	67.8
0.2	87.1	72.6	51.7	**47.0**	**68.5**
0.3	**89.1**	71.7	52.9	45.2	**68.5**
0.4	84.7	70.0	**54.6**	45.2	66.9
0.5	86.3	70.9	51.7	45.8	67.3

Table 7.6: Study of the mixup ratio. Experimental results are based on the VSpider benchmark, using EM accuracy as the evaluation metric. Accuracy on different difficulty levels are reported.

(see study of the mixup ratio in Section §8.4). We conduct the ablation study on the VSpider dataset and the experimental results are shown in Table 7.5.

The EM accuracy scores from the configurations shown in Table 7.5 are better than the concatenation baseline (65.6%). For implicit full mixup, we find that the transition layer can contribute to model effectiveness more when $m \geqslant 12$. Note that when $m = 23$, the framework is configured as focused simple concatenation (see §7.3.4). For implicit utterance mixup, when $m = 12$ or $m = 16$, the model can perform better. Similarly, for explicit utterance mixup, implementing in the middle layers benefits the most. Especially when $m = 12$, the parser achieves the best EM accuracy. One possible explanation is that utterance mixup changes the information flow more significantly than full mixup, requiring more target-centric layers to encode the mixed information. Regarding the different settings of transition layer design, there is no clear winner. For example, implicit full mixup usually outperforms the others when $m \in \{16, 20, 23\}$. When $m \in \{4, 8\}$, implicit utterance mixup achieves higher accuracy than full mixup.

Model	**zh**	**vi**
Independent Encoder	64.1	66.2
Joint Encoder	66.1	69.0

Table 7.7: Ablation study of the general encoder. EM accuracy is reported.

7.6.4 Choice of Mixup Ratio

The mixup ratio for the explicit utterance mixup is an important hyper-parameter that affects the information flow of English translations. Here, we conduct an analysis to see how the ratio influences parser effectiveness. The experiments are based on the supervised setting with the VSpider dataset, by configuring $m = 16$ and $\lambda \in \{0.1, 0.2, 0.3, 0.4, 0.5\}$.

The experimental results are shown in Table 7.6. Based on the overall results, we can observe that the parser obtains the best overall performance when the mixup ratio is 0.2 or 0.3. For different difficulty levels,[4] there is no single ratio setting that achieves the best performance on the four difficulty levels. For example, 0.3 is the best ratio for easy questions while 0.4 is the best for hard questions.

7.6.5 Choice of General Encoder Design

We compare different design choices for the general encoder in Table 7.7. Even though the independent encoder has the merit of efficient inference (the hidden states of the schema can be reused for all queries to the database), the performance drop is noticeable. The independent encoder is similar to the *local transformers* proposed in the FILTER architecture fang2021filter, which benefits POS tagging and multilingual QA tasks. However, we argue that for the semantic parsing task, full interactions between different components are more beneficial.

7.6.6 Case Studies

Here we conduct an analysis to see what cases the Rex framework can improve over the baseline and what cases it still fails.

[4] The difficulty level of a query is based on the complexity of the corresponding SQL; see [542] for more details.

The noise introduced by translation may affect the parser performance unexpectedly if the decoder accesses the translated English utterance representations directly, such as with the model using multi-source input with concatenation. For example 1 in Table 7.8, the English translation does not correctly translate 以数量升序 ("in ascending order of the count"), which causes the concatenation based model to fail to predict the `ORDER BY` clause. In contrast, our REX framework leverages the explicit utterance mixup transition layer and the target-centric encoder to ignore this translation noise and predict the SQL query correctly.

For example 2, both the baseline system and REX fail. However, comparing the REX prediction with the Concat. prediction, we see that the REX output is closer to the gold SQL query. Comparing the two, we see that the REX query fails to join `COUNTRIES` with `CONTINENTS` and uses `CONTINENTS.Continent = "欧洲"` in the `WHERE` clause instead of `COUNTRIES.Continent`, because `COUNTRIES.Continent` has the ID instead of the name. This suggests that the model can be further improved by proposing better encoding techniques for the schema information.

7.7 Summary

We propose the REX framework that effectively integrates information from English translations into the modeling of target language utterances. More specifically, we propose three different transition layer implementations that enhance the interactions among different components. We further compare their effectiveness with detailed ablation studies. Experiments show that our framework is robust to translation noise by controlling the information flow properly, outperforming existing baselines on the VSPIDER and CSPIDER benchmarks.

Example 1

Chinese Utterance:
请以数量升序显示管弦乐队的唱片格式。

English Translation:
Please display the orchestra record format in ascending order.

Concat. Prediction:

```
SELECT major_record_format FROM orchestra ORDER BY major_record_format ASC
```

Rex Prediction:

```
SELECT major_record_format FROM orchestra GROUP BY major_record_format ORDER
BY count(*) ASC
```

Gold:

```
SELECT major_record_format FROM orchestra GROUP BY major_record_format ORDER
BY count(*) ASC
```

Example 2

Chinese Utterance:
欧洲哪些国家至少有3家汽车制造商?

English Translation:
Which European countries have at least 3 car manufacturers?

Concat. Prediction:

```
SELECT Country FROM CAR_MAKERS GROUP BY Country HAVING COUNT(*) >= 3
```

Rex Prediction:

```
SELECT T1.CountryName FROM COUNTRIES AS T1 JOIN CAR_MAKERS AS T2 ON
T1.CountryId = T2.Country WHERE T1.Continent = "欧洲" GROUP BY T1.CountryName
HAVING COUNT(*) >= 3
```

Gold:

```
SELECT T1.CountryName FROM COUNTRIES AS T1 JOIN CONTINENTS AS T2 ON
T1.Continent = T2.ContId JOIN CAR_MAKERS AS T3 ON T1.CountryId = T3.Country
WHERE T2.Continent = "欧洲" GROUP BY T1.CountryName HAVING count(*) >= 3
```

Table 7.8: Case studies comparing Rex and the Concatenation baseline. Examples are selected from the CSpider dev set.

Chapter 8

Cross-lingual Retrieval-Augmented In-Context Learning for Cross-lingual Text-to-SQL Semantic Parsing

In-context learning using large language models has recently shown surprising results for semantic parsing tasks such as Text-to-SQL translation. Prompting GPT-3 or Codex using several examples of question-SQL pairs can produce excellent results. However, existing work primarily focuses on English datasets, and it is unknown whether large language models can serve as competitive semantic parsers for other languages. To bridge this gap, our work focuses on cross-lingual Text-to-SQL semantic parsing for translating non-English utterances into SQL queries based on an English schema. We consider a zero-shot transfer learning setting with the assumption that we do not have any labeled examples in the target language (but have annotated examples in English). This work introduces the XRICL framework, which learns to retrieve relevant English exemplars for a given query to construct prompts. We also include global translation exemplars for a target language to facilitate the translation process for large language models. To systematically evaluate our model, we construct two new benchmark datasets, XSpider and XKaggle-dbqa, which include questions in Chinese, Vietnamese, Farsi, and Hindi. Our experiments show that XRICL effectively leverages large pre-trained language models to outperform existing baselines. This work is based on:

- Peng Shi, Rui Zhang, He Bai and Jimmy Lin, XRICL: Cross-lingual Retrieval-Augmented In-Context Learning for Cross-lingual Text-to-SQL Semantic Parsing. *Findings of EMNLP, 2022.*

8.1 Introduction

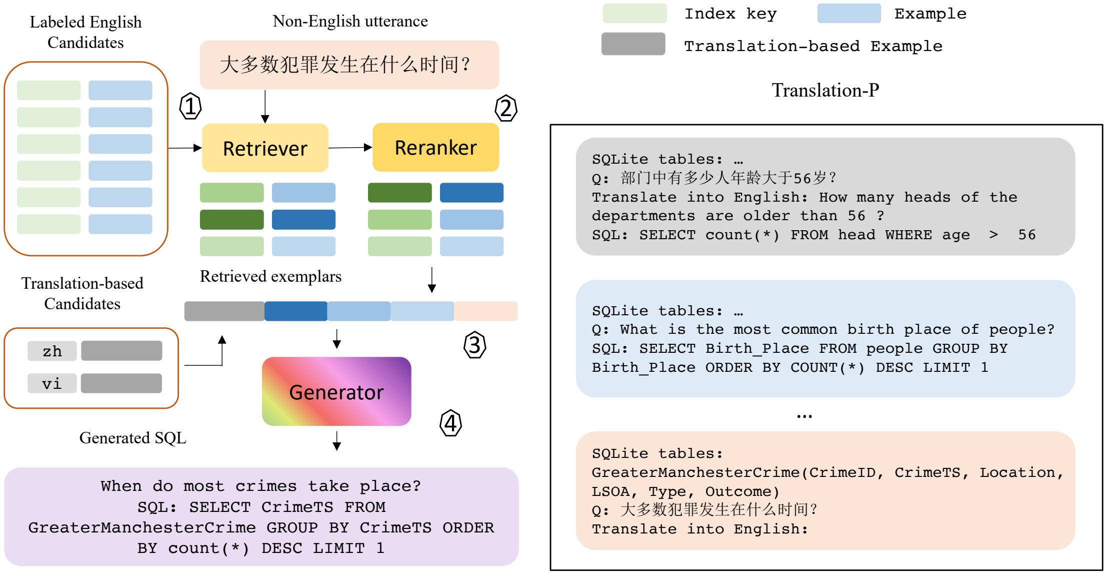
Semantic parsing is the task of translating natural language questions into meaning representations such as Lambda CDS [248], Python code [529], and SQL [542]. More recently, Text-to-SQL semantic parsing has attracted attention from academia and industry due to its challenging setup and practical applications. Cross-lingual Text-to-SQL semantic parsing [407, 294, 409] aims to translate non-English utterances into SQL queries based on an English schema (assuming we have an internationalized database), enabling users to query databases in non-English languages. For example, such a system could help people from around the world access the US government's open data[1] with natural language questions in different languages.

State-of-the-art approaches for Text-to-SQL semantic parsing have been greatly improved by finetuning pre-trained language models as a sequence-to-sequence problem [397, 528, 168, 537, 540, 415]. More recently, in-context learning with large language models (LLMs), such as GPT-3 [42] and Codex [55], has emerged as a new learning paradigm. This paradigm enables effective few-shot learning without model finetuning, showing its practical and scientific value [28]. Recent papers also have shown promising results applying in-context learning to the Text-to-SQL task. [368] studied if LLMs are already competitive Text-to-SQL semantic parsers without further finetuning on task-specific training data. Additionally, [350] and [391] investigated the exemplar retrieval problem for the semantic parsing task.

However, previous work mostly focused on English utterances, leaving other languages behind. It is unclear if LLMs are competitive for cross-lingual Text-to-SQL with English exemplars using in-context learning. Even in the mono-lingual setting (where the exemplars and the query are in the same language), many approaches are not practical beyond English due to the paucity of target language query-SQL exemplars.

To bridge this gap, we propose XRICL, a novel framework based on LLMs with in-context learning for cross-lingual Text-to-SQL semantic parsing. Specifically, the task is to generate SQL queries for non-English queries based on an English schema and an English query-SQL candidate pool. Our framework first constructs the context prompt by retrieving the most relevant English query-SQL exemplars for each target language query. Since we do not have any training data in the target language, we cannot train a retriever for target queries directly. Our solution is to train an English exemplar retriever with mT5 [509] and adopt a model-based cross-lingual transfer method for cross-lingual

[1] https://data.gov



```
When do most crimes take place?
SQL: SELECT CrimeTS FROM
GreaterManchesterCrime GROUP BY CrimeTS ORDER
BY count(*) DESC LIMIT 1
```

```
SQLite tables: …
Q: 部门中有多少人年龄大于56岁?
Translate into English: How many heads of the
departments are older than 56 ?
SQL: SELECT count(*) FROM head WHERE age  >  56
```

```
SQLite tables: …
Q: What is the most common birth place of people?
SQL: SELECT Birth_Place FROM people GROUP BY
Birth_Place ORDER BY COUNT(*) DESC LIMIT 1
```

…

```
SQLite tables:
GreaterManchesterCrime(CrimeID, CrimeTS, Location,
LSOA, Type, Outcome)
Q: 大多数犯罪发生在什么时间?
Translate into English:
```

Figure 8.1: Overview of our proposed XRICL framework. Given a labeled English question-SQL candidate pool and the non-English question as input, our framework uses in-context learning with a large pre-trained language model (e.g., Codex) to generate SQL queries in four steps: (1) Cross-lingual Exemplar Retrieval, (2) Exemplar Reranking, (3) Prompt Construction with Translation as Chain-of-Thought, and (4) Inference.

retrieval. The English exemplar retriever is trained with feedback from the LLM itself by distilling soft labels (likelihood).

Our framework introduces an additional exemplar into the LLM's input context, to instruct the model to translate the target query into English and then to translate the English query into SQL; this approach is inspired by recent work on chain-of-thought prompting [478, 410]. However, in our framework, this additional exemplar is identical for all test queries, which means that we only need a single pair of translations for any English-target language pair, requiring minimal translation effort.

During the inference process, the language model is expected to generate the English translation first and then the SQL query. In our experiments, we find that our proposed retriever and reranker can improve the LLMs' cross-lingual few-shot in-context learning performance by a large margin, and further improvements can be observed by adding an additional translation exemplar.

We further construct two benchmarks, XSpider and XKaggle-dbqa, to systematically evaluate the proposed framework in many languages. For XSpider, besides adopting existing work, including CSpider [294] and VSpider [313], we further translate the Spider dataset into Farsi and Hindi for evaluation. For XKaggle-dbqa, we translate the English Kaggle-dbqa dataset into Chinese, Farsi, and Hindi. Experimental results show that our proposed framework improves effectiveness compared to baseline systems.

Our contributions are summarized as follows: (1) We propose a novel retrieve-rerank framework to improve the exemplar selection process for in-context learning for cross-lingual Text-to-SQL semantic parsing. To the best of our knowledge, we are the first to explore the effectiveness of large pre-trained language models for cross-lingual Text-to-SQL semantic parsing. (2) We propose to use translation as a chain-of-thought prompt in the inference process, bridging the cross-lingual gap for large language models. (3) Last, we construct two new benchmarks, XSpider and XKaggle-dbqa, to facilitate evaluation of cross-lingual Text-to-SQL semantic parsing.

Before introducing the model in detail, we formally define the task here. Given a database where the schema s is in English (denoted as the source language), our task is to translate a non-English (denoted the target language) example x (x includes utterance u and schema s) into a SQL query a. In this work, we explore large pre-trained language models such as Codex for this Text-to-SQL task with in-context learning. To support in-context learning, labeled candidates of (utterance, schema, SQL) triples are required. Since more annotated resources are available in English, we assume that the labeled candidate set D is in English. Overall, in-context learning is an efficient method to leverage large pre-trained language models without expensive parameter fine-tuning. Furthermore, the candidate pool can be easily expanded for better generalization to new domains.

8.2 Related Work

In-context Learning: In-context learning is a relatively new paradigm for zero-shot and few-shot learning with large-scale pre-trained language models, first proposed in GPT-3 [42]. In-context learning for semantic parsing has been intensively investigated recently [339, 391, 421, 368, 174, 497, 55, 350]. However, most of the work considers only English, without examining the cross-lingual ability of the proposed methods. [482] evaluated the multilinguality of pre-trained language models on non-English multi-class classification with in-context learning. However, their task is simpler than semantic parsing tasks such as ours. To the best of our knowledge, we are the first to explore cross-lingual Text-to-SQL semantic parsing under the in-context learning setting.

Cross-lingual Semantic Parsing: Cross-lingual semantic parsing aims to handle user utterances from multiple languages and translate them into formal representations. Recent advances can be categorized into two threads: multilingual dataset creation and model development.

For example, [24] adapted a Chinese dialogue parsing dataset into English. [294] and [313] adapted the English Text-to-SQL dataset SPIDER [542] into Chinese and Vietnamese, which are used in this work for evaluation. Some multilingual datasets with different formal representations have also been created, such as SPARQL [85] and TOP [241].

In terms of model development, [405] is the most relevant to our work, which leveraged bilingual input for the semantic parsing task. However, they used RNN models and focused on multilingual representation alignment with pre-training. Instead, our work focuses on representation mixup with large multilingual pre-trained models. Improving cross-lingual zero-shot transfer is another direction [409, 407, 408].

Multilingual and Cross-lingual Retrieval: In multilingual retrieval, the task is to retrieve relevant documents where the user queries and the corpora are in the same language. Recent work takes advantage of cross-language transfer using pre-trained multilingual models [411, 419, 564, 563]. For example, [419] used DPR to retrieve documents based on ad-hoc queries in six languages. On the other hand, cross-lingual retrievers help users find relevant documents in languages that are different from that of the queries. This task has a long history that goes back several decades [320], but recent work includes [555, 266, 434]. For instance, [19] created a cross-lingual open-domain question answering dataset where the system is required to retrieve passages from different languages to answer user questions.

8.3 Models

Our XRICL framework is shown in Figure 8.1, consisting of four steps:

(1) *Cross-lingual Exemplar Retrieval*: Retrieve a list of N English exemplars that are relevant to the input non-English example x.

(2) *Exemplar Reranking*: Rerank the retrieved N exemplars and use the top K exemplars to construct prompts.

(3) *Prompt Construction with Translation as Chain of Thought*: Construct a prompt consisting of the translation exemplar as a chain of thought, the selected K exemplars, and the input example.

(4) *Inference*: Feed the prompt into a pre-trained language model to generate SQL.

8.3.1 Cross-lingual Exemplar Retriever

Given a non-English question, the goal of the cross-lingual exemplar retriever is to find *relevant* exemplars from the English candidate pool efficiently that can improve the predictions of the generators. Considering that we use labeled examples in English (a high-resource language) as candidates, we formulate this step as a cross-lingual retrieval problem, where the test question is in a non-English language. In this case, traditional term matching methods such as BM25 [386] or BM25 + RM3 query expansion [249] cannot be applied due to token mismatch. Instead, we propose to use a bi-encoder for cross-lingual semantic retrieval with model-based zero-shot transfer. We further improve the retriever with distillation-based training.

Model. Here, we leverage the popular bi-encoder architecture known as dense passage retriever (DPR) [205], where the query and candidates are mapped into representation vectors *independently*. The retriever uses a dense encoder $\mathrm{E}_u(\cdot)$ that converts an utterance into a d-dimensional vector and builds an index over the candidate pool that is used for retrieval.

For a test instance x, we use the same dense encoder to map the utterance into a d-dimensional vector (denoted the query vector). Based on the query vector, the closest top N exemplars are retrieved from the pre-built index based on the pre-defined distance function. Following [205], we define the distance function as

$$\mathrm{sim}(x, z) = \mathrm{E}_u(x)^\top \mathrm{E}_u(z) \tag{8.1}$$

where Z is the set of candidate exemplars and $z \in Z$. We use a transformer as the dense encoder, and the average of the contextual embeddings of the utterance tokens is taken as the representation of the encoded text.

Model-based Cross-lingual Transfer. Considering that we do not have training data in target languages, we adopt a model-based cross-lingual transfer method, where we leverage the zero-shot cross-lingual transfer ability of multilingual pre-trained transformers such as mBERT [103], XLM-Roberta [77], mBART [272], and mT5 [509]. Specifically, we train the dense retriever in the source language, where both the query utterance and candidate utterances are in English (in our case), and apply inference directly on query utterances in the target language and retrieve English exemplars in a cross-lingual manner.

Distillation-based Training. One common practice for bi-encoder training is contrastive learning. Given a query, positive examples and negative examples are required. The model is optimized such that examples from the positive class have similar representations and examples from the negative class have different representations.

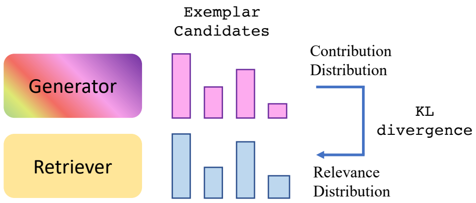


Figure 8.2: Illustration of distillation-based training. The contribution distribution is the likelihood distribution of the top-N exemplars produced by the LLM. The relevance distribution is the ranking score distribution produced by the retriever.

The key here is how to define positive and negative examples for the semantic parsing task. Recently, [174] used the similarity of target meaning representations to first rank the candidates and choose the top-k as positive examples and the bottom-k as negative examples. Instead of using human-designed relevance metrics, [391] proposed to use a language model to label positive and negative examples for contrastive learning; similar to [174], hard labels are used. Another way to train the bi-encoder is to use a regression-based loss function. [350] proposed to retrieve exemplars that have relevant program structures (tree edit distance of SQL abstract syntax trees is used as the relevance metric) for the test utterances and the model is optimized with mean-squared error loss for predicting the similarity score.

As an alternative to these above approaches, we train our retriever by distilling the LLM's scoring function. This scoring function calculates the ground-truth SQL query's likelihood given an English exemplar z_k and the input utterance x, which estimates the importance of this exemplar for parsing the given input utterance. Hence, we score the retrieved English exemplars with an LLM and optimize the KL divergence between the LLM's ranking scores and the retriever's ranking scores to update the retriever, as shown in Figure 8.2. This retriever is denoted DE-Retriever (Distillation-based Exemplar Retriever). Intuitively, with the KL divergence loss function, the model tries to match the probability of retrieving an exemplar z_k with the contribution of that exemplar to the generated SQL query a.

We first obtain N exemplars with the highest scores based on Equation (8.1), denoted as Z_{top-N}. Our loss function is defined as:

$$\begin{aligned}\mathcal{L}_{\texttt{distill}} = \texttt{KL}(\ &\texttt{SG}(p(z_n \mid x, a, Z_{top-N}; G)) \\ &\| \ p(z_n \mid x, Z; E)),\end{aligned} \tag{8.2}$$

where $\texttt{SG}$ denotes the stop gradient operation, G denotes the generator, and E denotes the retriever encoder. We further compute $p(z_n \mid x, a, Z_{top-N}; G)$ as follows:

$$\begin{aligned} & p(z_n \mid x, a, Z_{top-N}) \propto \\ & p(a \mid x, z_n, Z_{top-N}; G)\ p(z_n \mid x, Z_{top-N}) \end{aligned} \tag{8.3}$$

We approximate the posterior under the assumption that we have a uniform prior over the set of retrieved exemplars, so $p(z_n \mid x, Z_{top-N})$ is approximated as $\frac{1}{N}$. We further compute $p(a \mid x, z_n, Z_{top-N}; G)$ as:

$$\frac{\exp(p(a \mid x, z_n))}{\sum_{j=1}^{N} \exp(p(a \mid x, z_j))} \tag{8.4}$$

where $p(a \mid x, z_j)$ is computed with the generator.

More specifically, we use example z_j as the prompt and concatenate it with test instance u and target SQL a. Then we feed it to the generator to compute the log probability of each token $\log(p(a_i))$ in the target SQL query a; $p(a|x, z_j)$ can be computed as $\exp(\sum \log(p(a_i)))$.

8.3.2 Exemplar Reranking

For tasks such as information retrieval and open-domain question answering, reranking is widely adopted to further improve retrieval results by incorporating a reranker. Such a two-stage procedure is also useful in a variety of natural language processing tasks. In this work, following the retrieve-and-rerank idea, we propose to incorporate an exemplar reranker in our framework. This reranker can leverage token-level interactions between the utterances to better rank the exemplars.

More specifically, the query utterance u and the candidate utterance u_z are concatenated together with special tokens: `[CLS]` u `[SEP]` u_z `[SEP]`. The tokenized input is fed into a transformer model. An MLP with sigmoid activation is applied on top of the contextual embedding of the `[CLS]` token to obtain the relevance score of the candidate example [254]. Sigmoid cross-entropy loss is used and the model is optimized to produce a relevance score as $p(a|x, z_n, Z_{top-N}; G)$. This reranker is denoted DE-Reranker (Distillation-based Exemplar Reranker).

8.3.3 Prompt Construction with Translation as Chain of Thought

From the input instance x and the list of retrieved-and-reranked exemplars Z, we construct the augmented query by concatenating exemplars with the input instance following previous work [174, 391, 350, 267, 42, 339]. For the exemplar, we linearize the table schema, the question, and the SQL query. The exemplars are sorted by relevance score in descending order. For the test instance, only the table schema and the question are linearized. We denote this prompting approach Vanilla-P.

Translation as Chain of Thought: Recent work on chain-of-thought prompting is designed to solve the multi-step reasoning problem by providing intermediate reasoning steps before the final answer in the prompt [478]. Inspired by this, we use the translation pair (from non-English to English in our case) as an intermediate step for cross-lingual semantic parsing inference.

Specifically, a translation-based exemplar is inserted in front of Z. For example, in the right part of Figure 8.1, the grey box contains the Chinese version of the translation as a chain-of-thought prompt. The question in the prompt is in the target language, followed by an instruction `Translate into English` and the English translation of the question. Note that this translation-based exemplar is shared among all the test instances in that language, as shown in the left part of Figure 8.1. The translation-based examples are indexed by the language code, such as `zh` and `vi`. In this way, it only requires minimal translation effort to build the global translation-based exemplar. We denote this prompting approach Translation-P.

8.3.4 Inference

For inference, we feed the constructed prompt to a large pre-trained language model to generate the target SQL query with greedy decoding. In this work, we consider **Codex** (Codex-Davinci-001) [55] because it has shown superior performance for the English Text-to-SQL task [350].

8.4 Experimental Setup

In this section, we describe the datasets, implementation details, and baselines for our experiments.

8.4.1 Datasets

We create two benchmarks, XSpider and XKaggle-dbqa, by translating existing English Text-to-SQL datasets into other languages and evaluate our methods on these two benchmarks.

Model	zh-full	zh		vi		fa		hi	
	EM	EM	TS	EM	TS	EM	TS	EM	TS
(1) mT5 zero-shot	39.7	47.9	48.4	42.1	40.1	41.3	39.5	**41.2**	**39.7**
(2) mUSE	38.4	43.0	46.8	31.8	33.4	28.9	31.1	22.2	23.7
(3) mSBERT	37.9	41.3	47.1	34.6	33.5	29.3	31.8	22.0	22.3
(4) mT5-encoder	44.4	48.1	51.4	41.3	39.5	38.4	38.5	28.6	27.0
(5) DE-Retriever	46.0	50.4	53.9	42.2	40.7	38.2	40.0	29.9	27.9
(6) DE-R^2	46.4	52.1	55.3	**44.4**	41.9	40.0	40.6	30.0	28.2
(7) + Translation-P	**47.4**	**52.7**	**55.7**	43.7	**43.6**	**43.2**	**45.1**	**32.6**	**32.4**

Table 8.1: Results on the XSpider dev set. "zh-full" and "zh" are two different splits from CSpider [294]. EM and TS are exact match accuracy and test suite accuracy, respectively. Entry (5) is based on the DE-Retriever with Vanilla-P. Entry (6) is based on the DE-Retriever and DE-Reranker (denoted as DE-R^2) with Vanilla-P. Entry (7) is based on DE-R^2 with Translation-P.

XSpider: CSpider [294] and VSpider [313] are **C**hinese (zh) and **V**ietnamese (vi) cross-domain Text-to-SQL datasets translated from Spider [542]. More specifically, we use the English Spider training set as the candidate pool and training data for retriever-reranker models. We use the development sets of CSpider and VSpider for cross-lingual evaluation. We further translate the Spider development set into Farsi (fa) and Hindi (hi) for a more comprehensive evaluation.

XKaggle-dbqa: This is a recently constructed dataset for more realistic and challenging Text-to-SQL evaluation. The dataset is based on 8 databases from Kaggle. We translate the questions into Chinese (zh), Farsi (fa), and Hindi (hi) for cross-lingual evaluation. We use the English Spider training set as the candidate pool.

8.4.2 Experimental Details

For the exemplar retriever, we use 24-layer transformers initialized with the parameters of the mT5 encoder that is then fine-tuned on the English Spider dataset for the Text-to-SQL task. For the exemplar reranker, we use InfoXLM [70] as the starting point. We train the retriever and reranker on the English Spider dataset and then apply both models to cross-lingual retrieval and reranking in a zero-shot fashion. For the Codex configuration, we use greedy decoding by setting the temperature to zero. We use $N = 16$ and $K = 8$ for all experiments, which means that the DE-Retriever first retrieves 16 exemplars from the candidate pool and the DE-Reranker produces the top 8 exemplars for prompt construction.

In terms of evaluation metrics, we use **E**xact **M**atch (EM) accuracy for both the XSpider benchmark and the XKaggle-dbqa benchmark. Following [568], we report the **T**est-**s**uite (TS) accuracy. Only the datasets that are aligned with the Spider dev set can be evaluated with TS accuracy, so the XKaggle-dbqa benchmark is not applicable. Because the CSpider dev set is only partially aligned to the Spider dev set, the full CSpider (zh-full) dev set can be only evaluated with EM accuracy. We collect a subset of the CSpider dev set (zh) whose queries are aligned with the English Spider dev set, and further evaluate these using TS accuracy.

8.4.3 Baselines

mT5 zero-shot transfer is a baseline model that is trained with the English Spider training set. The model is based on the pre-trained sequence-to-sequence multilingual language model mT5-large [509]. This model has zero-shot cross-lingual transfer ability, with which the model can directly handle non-English utterances.

mUSE and mSBERT are baselines that use unsupervised retrievers to obtain exemplars: multilingual Universal Sentence Encoder [519] and multilingual Sentence-BERT [382]. Prompts are then constructed for in-context learning with Codex.

8.5 Results

8.5.1 Results on XSpider

Results on XSpider are shown in Table 8.1. We report the EM and TS accuracy. For the full CSpider dataset (zh-full), since TS Accuracy is not supported, we only report EM accuracy. We report both TS and EM accuracy on the subset of CSpider. Entry (1) reports the zero-shot performance of the mT5 model that is trained on the English Spider dataset. On `zh-full`, `vi`, `fa`, and `hi`, the mT5 zero-shot method obtains on average 41.1 EM accuracy and 39.8 TS accuracy (average TS accuracy is computed without `zh-full` because the metric cannot be computed on the full CSpider).

From entry (2) to entry (7), the methods are based on in-context few-shot learning. For entries (2–6), the prompting method is Vanilla-P. For entry (7), prompting with Translation-P is applied.

Model	zh	fa	hi
(1) mT5 zero-shot	9.7	8.1	7.6
(2) mUSE	20.7	12.4	16.2
(3) mSBERT	14.7	13.0	11.9
(4) mT5-Encoder	22.2	16.8	16.2
(5) DE-Retriever	26.5	18.4	16.8
(6) DE-R^2	27.0	18.4	17.8
(7) + Translation-P	**28.1**	**20.0**	**19.5**

Table 8.2: Results on the XKaggle-dbqa test set. We report exact match (EM) accuracy.

With unsupervised exemplar retrievers such as mUSE and mSBERT, shown in entries (2) and (3), Codex performs worse than mT5 zero-shot transfer, especially for Farsi (39.5→31.1/31.8 on TS accuracy) and Hindi (39.7→23.7/22.3 on TS accuracy). By switching the unsupervised exemplar retriever to the mT5-encoder, which is the encoder component of the fine-tuned mT5 model, the effectiveness of Codex improves by a large margin. For example, on the CSpider subset, TS accuracy improves to 51.4 from 47.1, outperforming mT5 zero-shot performance by 3 points. This indicates that the exemplar retrieval component is essential to take advantage of the competitive performance of LLMs such as Codex. For languages such as Vietnamese and Farsi, Codex is comparable to mT5 zero-shot transfer, while for Hindi, there is still a large gap (39.7 vs. 27.0 on TS accuracy).

By applying our proposed distillation based retriever-reranker pipeline (denoted as DE-R^2) for retrieving exemplars, impressive improvements can be observed in all four languages by comparing entry (6) with entry (4). Our end-to-end results are shown in entry (7), where we see that our proposed framework achieves the best results for most of the languages (except Vietnamese EM accuracy) in the in-context learning setting.

Comparing the best results of in-context learning with mT5 zero-shot results, we can see that Codex can achieve better performance in Chinese, Vietnamese, and Farsi. For example, XRICL outperforms mT5 zero-shot by 7.7 EM accuracy on the full dev set of CSpider. One exception is Hindi, where the best in-context learning performance cannot match mT5 zero-shot transfer. One possible explanation is that Codex has weaker modeling ability in Hindi because less Hindi data were accessible during the training.

8.5.2 Results on XKaggle-dbqa

There is agreement by researchers today that XKaggle-dbqa is a more realistic evaluation for the Text-to-SQL parsing task. The databases are real-world databases with abbreviated column names. We use the training set of English Spider as the candidate pool. In this case, both the model's generalization ability and its cross-lingual transfer capability can be tested.

The XKaggle-dbqa results are shown in Table 8.2. Entry (1) shows the zero-shot cross-lingual cross-domain transfer performance of the mT5 model trained on the English Spider dataset. For example, on Chinese Kaggle-dbqa, mT5 only obtains 9.7 EM accuracy. For comparison, mT5 reach 20.0 EM accuracy on the English test set in a zero-shot fashion, outperforming the previous state of the art obtained by RAT-SQL [463] with 18.4 EM accuracy [227] using column descriptions and model adaptation. This indicates that the mT5 model is more robust than RAT-SQL on domain transfer. However, the effectiveness degrades drastically when mT5 is applied to non-English languages. The mT5 zero-shot method on average obtains only 8.5 EM accuracy in the three languages.

For the Codex-based in-context learning methods, the results are shown in entries (2–7). With unsupervised retrieval methods such as mUSE, Codex can reach 20.7 EM accuracy in Chinese, improving over the zero-shot mT5 baseline. Comparing entries (2) and (3), there is no clear winner for these two unsupervised retrieval methods. Our end-to-end results are shown in entry (7), which achieves state-of-the-art performance on the XKaggle-dbqa benchmark, with 22.5 EM accuracy on average, which is better than the mT5 zero-shot method. For example, on Chinese Kaggle-dbqa, our framework obtains an 18.4 point improvement over mT5 zero-shot transfer.

8.6 Discussion and Analysis

8.6.1 Effectiveness on English Text-to-SQL

We show that our model is comparable to other in-context learning methods for English semantic parsing. Through this comparison, we show that our framework is built on a competitive backbone for Text-to-SQL. We use the DE-Retriever as the backbone model in the ablation study and compare with three recent methods, described as follows: [391] used hard labels obtained from the generator to train the retriever. [350] used the tree edit distance of SQL queries as a similarity function: a smaller distance means better exemplar quality for the specific test instance. The ranking model is optimized to predict the target

Model	EM	EX	TS
[391] (our impl.)	48.5	53.5	50.3
[350]	-	60.0	-
[368]	-	**67.0**	55.1
DE-Retriever (Ours)	53.5	60.3	**56.3**

Table 8.3: Results on the English SPIDER development set. Our system achieves results comparable to other state-of-the-art in-context learning methods for English Text-to-SQL. EM: Exact Match Accuracy. EX: Execution Accuracy. TS: Test-suite Accuracy [568].

SQL pair tree edit distance based on the utterance pair. [368] designed an efficient prompt that leverages table contents for zero-shot Text-to-SQL. We refer the reader to the original papers for more details.

Table 8.3 shows the results on the SPIDER development set. Our backbone system (DE-Retriever + Codex Generator) obtains 53.5 EM accuracy and 60.3 EX accuracy, which is comparable to the 60.0 EX accuracy reported by [350]. Comparing to [368], our system obtains comparable TS accuracy (56.3 vs. 55.1).

8.6.2 Effectiveness of DE-R^2

We analyze the effectiveness of DE-R^2 on the XSPIDER benchmark and the XKAGGLE-DBQA benchmark. By comparing entries (5) and (4) in Table 8.1 and Table 8.2, we can observe that the DE-Retriever can improve over the mT5-encoder baseline in most of the languages (except EM accuracy in Farsi). Comparing entries (6) and (5), we find that the reranker can further improve the EM accuracy and the TS accuracy. This indicates that our XRICL framework is effective in selecting good exemplars as prompts.

8.6.3 Effectiveness of Chain-of-Thought Prompt

By comparing entries (7) and (6) in Table 8.1 and Table 8.2, we find that Translation-P can further improve the semantic parsing ability of Codex on top of DE-R^2, except EM accuracy for Vietnamese.

Model	**zh-full**	**zh**	
	EM	EM	TS
(1) DE-R^2 + Translation-P	47.4	52.7	55.7
(2) T-Oracle	46.3	52.6	57.6
(3) TG-Oracle	52.5	58.0	62.2

Table 8.4: Results with oracles: T-Oracle is the Template Oracle and TG-Oracle is the Template+Generator Oracle. EM accuracy and TS accuracy are reported.

8.6.4 Oracle Performance

It is interesting to investigate the upper bound of Codex on cross-lingual Text-to-SQL semantic parsing. We design two pipelines to experiment with the capabilities of Codex when an oracle is available (i.e., the target SQL query is accessible to help the retrieval and reranking). We experiment with two different oracles:

Template Oracle: We retrieve exemplars using the *gold* parse. The template is extracted from the target SQL query and only exemplars with the same SQL template are retrieved. This is based on the assumption that utterances with the same SQL templates share the same query intent and the generator can benefit from these exemplars.

Template Oracle + Codex LM oracle: Here we introduce an oracle from the generator (Codex) into the pipeline. More specifically, we replicate the training process in the testing phase. The exemplars with the same SQL templates are first retrieved. For each retrieved exemplar, we use Codex to compute its contribution to the test instance as the reranking score. We then use the top-k as the exemplars.

The experimental results are shown in Table 8.4. Comparing entries (1) and (2), we can observe that our XRICL framework can outperform the Template Oracle in terms of EM accuracy on the full dataset and is comparable on the subset. Template Oracle + Codex LM Oracle reaches 52.5 on the full dataset and 58.0 on the subset in terms of EM accuracy. This suggests that signals from the Codex LM are useful and that there is additional room for improvement in our framework.

8.7 Summary

In this work, we proposed the XRICL framework that improves in-context learning for cross-lingual Text-to-SQL semantic parsing. The retrieve-and-rerank models that we propose can learn signals from large pre-trained models (Codex) to improve the quality of selected exemplars, which can further benefit the generator. By integrating prompts inspired by chain of thought, our proposed Translation-P method can bridge the cross-lingual gap for the generator. Extensive experiments on XSpider and XKaggle-dbqa demonstrate the effectiveness of our framework, which obtains state-of-the-art performance on few-shot in-context learning in most of the datasets, thus unlocking the potential of Codex.

Chapter 9

Conclusions and Future Work

9.1 Conclusions

This thesis investigates several aspects in information access research, and we summarized our contribution as follows:

Improving the Information Access Systems for Heterogeneous Data. Information and knowledge are residing in different formats. Some formats received more attention than others, such as news articles, Wikipedia passages, etc. However, improving the access to the information and knowledge that are located in other formats (e.g. tables or private databases) is also an important research problem, which can further benefit industrial applications. In this thesis, we present an intermediate pre-training framework that is trained over large-scale tables and synthesized user queries can improve the table-based free-form question answering(Chapter 3), achieving the state-of-the-art performance on FeTaQA benchmark. For the information access over structured databases, we propose a representation learning framework for Text-to-SQL semantic parsing that is trained on large-scale tables and crawled SQLs from the web. The pre-trained encoder can alleviate three pain points of existing Text-to-SQL semantic parsing models, outperforming existing baseline systems. To the best of our knowledge, we are the first to use both crawled SQL and crawled tables to boost the text-to-SQL semantic parsers (Chapter 4).

Facilitating the Information Access System Development for Non-English Speakers. To meet the information access needs of diverse populations, in this thesis, we first study the problem of cross-lingual entity matching (Chapter 5). This is an important component that benefits the effectiveness and efficiency of information access systems such as multilingual question answering over knowledge graphs, by integrating data from different languages into a unified view. Specifically, we improve GCN models and leverage pre-trained multilingual BERT model to better capture the relatedness of entities in multiple languages, by integrating multiple aspects of information from the knowledge graph, achieving state-of-the-art performance on benchmarks such as DBP15K and DBP100k. Also, to the best of our knowledge, the model is one of the earliest BERT-based bi-encoder designs for the matching task. We further explored the cross-lingual training strategies for the tasks of non-English dense retrieval and non-English document reranking. We are one of the earliest studies that use multilingual BERT on the non-English document reranking.

Extending Structured Data Access Systems to non-English Languages. It is received less attention from the community when it comes to the intersection of structured data access systems and multilingual models. In this thesis, we take a step further to explore models and frameworks that enable non-English speakers to access structured data. To achieve this, a common practice is to leverage external translation systems to facilitate the cross-lingual transfer. However, the noise in the translation outputs may cause error propagation for downstream tasks. In this thesis, we propose a representation mixup framework (Chapter 7) for Text-to-SQL semantic parsing that can guide the translation information flow properly within the model and reduce the negative influence of the noisy translation. As the development of large language models, in this thesis, we also evaluate their cross-lingual ability in the task of Text-to-SQL semantic parsing with in-context learning paradigm, when the training data in target languages are not available (Chapter 8). To the best of our knowledge, we are the first to explore the effectiveness of large pre-trained models for cross-lingual Text-to-SQL semantic parsing. We also construct new benchmarks for facilitating the cross-lingual Text-to-SQL semantic parsing evaluation.

9.2 Future Work

Conversational Information Seeking. Recent years have seen a huge development in conversational information seeking [92, 549, 256, 545]. When the information-seeking intents are complicated, users tend to issue multiple queries to achieve the goals. This requires the systems to handle the queries in context; for example, the models need to learn to refer to results in previous turns. By combining conversation information seeking with a unified framework [207, 281, 330, 497], the models need to learn to choose or combine knowledge and information from multiple sources during the conversation. This raises more challenges to the model development; however, this kind of system can further improve the user experience without composing complex queries or interacting with multiple systems for different information sources.

Effective Information Seeking in Many Languages. In this thesis, we explore several training techniques for transferring information access models in high-resource languages (e.g. English) to other languages (e.g. Hindi) without expensive annotations. Even though the model effectiveness is greatly improved with the benefit of the multilingual pre-training [509, 272], the performance is still far from perfect. It is still a challenge for improving the information access systems for non-English languages, especially for low-resource languages. One interesting direction to explore is to leverage multilingual resources [434] (e.g. bilingual dictionaries) in the data acquisition process and model training [359]. Moreover, integrating the model with the crowd-sourcing process (e.g. interactive annotation process) is also worth exploring.

References

[1] The hidden information state model: A practical framework for pomdp-based spoken dialogue management. *Computer Speech and Language*, 24(2):150 – 174, 2010.

[2] Nasreen Abdul-Jaleel, James Allan, W. Bruce Croft, Fernando Diaz, Leah Larkey, Xiaoyan Li, Donald Metzler, Mark D. Smucker, Trevor Strohman, Howard Turtle, and Courtney Wade. UMass at TREC 2004: Novelty and HARD. In *TREC*, 2004.

[3] Omri Abend and Ari Rappoport. Universal conceptual cognitive annotation (ucca). In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 228–238, 2013.

[4] Rakesh Agrawal and Ramakrishnan Srikant. Searching with numbers. *IEEE Trans. Knowl. Data Eng.*, 15(4):855–870, 2003.

[5] Alfred V. Aho and Jeffrey D. Ullman. *The Theory of Parsing, Translation and Compiling*, volume 1. Prentice-Hall, Englewood Cliffs, NJ, 1972.

[6] Zeynep Akkalyoncu Yilmaz, Shengjin Wang, Wei Yang, Haotian Zhang, and Jimmy Lin. Applying BERT to document retrieval with birch. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP): System Demonstrations*, pages 19–24, Hong Kong, China, 2019.

[7] Zeynep Akkalyoncu Yilmaz, Wei Yang, Haotian Zhang, and Jimmy Lin. Cross-domain modeling of sentence-level evidence for document retrieval. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3490–3496, Hong Kong, China, 2019.

[8] Miltiadis Allamanis, Daniel Tarlow, Andrew D. Gordon, and Yi Wei. Bimodal modelling of source code and natural language. In *ICML*, volume 37 of *JMLR Workshop and Conference Proceedings*, pages 2123–2132. JMLR.org, 2015.

[9] American Psychological Association. *Publications Manual*. American Psychological Association, Washington, DC, 1983.

[10] Bo An, Bo Chen, Xianpei Han, and Le Sun. Accurate text-enhanced knowledge graph representation learning. In *Proceedings of NAACL*, 2018.

[11] Oren Etzioni Ana-Maria Popescu and Henry Kautz. Towards a theory of natural language interfaces to databases. In *Proceedings of the 8th International Conference on Intelligent User Interfaces*, pages 149–157, 2003.

[12] Rie Kubota Ando and Tong Zhang. A framework for learning predictive structures from multiple tasks and unlabeled data. *Journal of Machine Learning Research*, 6:1817–1853, December 2005.

[13] Jacob Andreas. Good-enough compositional data augmentation. *arXiv preprint arXiv:1904.09545*, 2019.

[14] Galen Andrew and Jianfeng Gao. Scalable training of L1-regularized log-linear models. In *Proceedings of the 24th International Conference on Machine Learning*, pages 33–40, 2007.

[15] Mikel Artetxe and Holger Schwenk. Massively multilingual sentence embeddings for zero-shot cross-lingual transfer and beyond. *arXiv preprint arXiv:1812.10464*, 2018.

[16] Yoav Artzi and Luke Zettlemoyer. Weakly supervised learning of semantic parsers for mapping instructions to actions. *Transactions of the Association forComputational Linguistics*, 2013.

[17] Yoav Artzi and Luke Zettlemoyer. Weakly supervised learning of semantic parsers for mapping instructions to actions. *Transactions of the Association for Computational Linguistics*, 1:49–62, 2013.

[18] Yoav Artzi and Luke S. Zettlemoyer. Bootstrapping semantic parsers from conversations. In *EMNLP*, 2011.

[19] Akari Asai, Jungo Kasai, Jonathan H Clark, Kenton Lee, Eunsol Choi, and Hannaneh Hajishirzi. XOR QA: Cross-lingual open-retrieval question answering. *arXiv preprint arXiv:2010.11856*, 2020.

[20] Akari Asai, Xinyan Yu, Jungo Kasai, and Hanna Hajishirzi. One question answering model for many languages with cross-lingual dense passage retrieval. *Advances in Neural Information Processing Systems*, 34:7547–7560, 2021.

[21] Layla El Asri, Hannes Schulz, Shikhar Sharma, Jeremie Zumer, Justin Harris, Emery Fine, Rahul Mehrotra, and Kaheer Suleman. Frames: A corpus for adding memory to goal-oriented dialogue systems. *CoRR*, abs/1704.00057, 2017.

[22] Michael Azmy, Peng Shi, Jimmy Lin, and Ihab F. Ilyas. Matching entities across different knowledge graphs with graph embeddings. *arXiv preprint arXiv:1903.06607*, 2019.

[23] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate. In *ICLR*, 2015.

[24] He Bai, Yu Zhou, Jiajun Zhang, Liang Zhao, Mei-Yuh Hwang, and Chengqing Zong. Source critical reinforcement learning for transferring spoken language understanding to a new language. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 3597–3607, Santa Fe, New Mexico, USA, August 2018.

[25] Laura Banarescu, Claire Bonial, Shu Cai, Madalina Georgescu, Kira Griffitt, Ulf Hermjakob, Kevin Knight, Philipp Koehn, Martha Palmer, and Nathan Schneider. Abstract meaning representation for sembanking. In *Proceedings of the 7th linguistic annotation workshop and interoperability with discourse*, pages 178–186, 2013.

[26] Laura Banarescu, Claire Bonial, Shu Cai, Madalina Georgescu, Kira Griffitt, Ulf Hermjakob, Kevin Knight, Philipp Koehn, Martha Palmer, and Nathan Schneider. Abstract Meaning Representation for sembanking. In *Proceedings of the 7th Linguistic Annotation Workshop and Interoperability with Discourse*, pages 178–186, Sofia, Bulgaria, August 2013. Association for Computational Linguistics.

[27] Laura Banarescu, Claire Bonial, Shu Cai, Madalina Georgescu, Kira Griffitt, Ulf Hermjakob, Kevin Knight, Philipp Koehn, Martha Palmer, and Nathan Schneider. Abstract meaning representation for sembanking. In *Proceedings of the 7th Linguistic Annotation Workshop and Interoperability with Discourse*, 2013.

[28] Iz Beltagy, Arman Cohan, Robert Logan IV, Sewon Min, and Sameer Singh. Zero- and few-shot NLP with pretrained language models. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics: Tutorial Abstracts*, pages 32–37, Dublin, Ireland, May 2022. Association for Computational Linguistics.

[29] Jonathan Berant and Percy Liang. Semantic parsing via paraphrasing. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1415–1425, Baltimore, Maryland, June 2014. Association for Computational Linguistics.

[30] Núria Bertomeu, Hans Uszkoreit, Anette Frank, Hans-Ulrich Krieger, and Brigitte Jörg. Contextual phenomena and thematic relations in database qa dialogues: results from a wizard-of-oz experiment. In *Proceedings of the Interactive Question Answering Workshop at HLT-NAACL 2006*, pages 1–8. Association for Computational Linguistics, 2006.

[31] Christian Bizer, Jens Lehmann, Georgi Kobilarov, Sören Auer, Christian Becker, Richard Cyganiak, and Sebastian Hellmann. DBpedia: A crystallization point for the web of data. *Web Semantics: Science, Services and Agents on the World Wide Web*, 7(3):154–165, 2009.

[32] Ben Bogin, Jonathan Berant, and Matt Gardner. Representing schema structure with graph neural networks for text-to-sql parsing. In *ACL*, 2019.

[33] Ben Bogin, Matt Gardner, and Jonathan Berant. Global reasoning over database structures for text-to-sql parsing. *arXiv preprint arXiv:1908.11214*, 2019.

[34] Ben Bogin, Matt Gardner, and Jonathan Berant. Representing schema structure with graph neural networks for text-to-sql parsing. *arXiv preprint arXiv:1905.06241*, 2019.

[35] Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. Enriching word vectors with subword information. *Transactions of the Association for Computational Linguistics*, 5:135–146, 2017.

[36] Antoine Bordes, Nicolas Usunier, Alberto Garcia-Duran, Jason Weston, and Oksana Yakhnenko. Translating embeddings for modeling multi-relational data. In *Proceedings of NIPS*, 2013.

[37] Antoine Bordes and Jason Weston. Learning end-to-end goal-oriented dialog. *ICLR*, 2017.

[38] Benjamin Borschinger and Mark Johnson. A particle filter algorithm for Bayesian wordsegmentation. In *Proceedings of the Australasian Language Technology Association Workshop 2011*, pages 10–18, Canberra, Australia, December 2011.

[39] Léon Bottou. Stochastic gradient learning in neural networks. In *Proceedings of Neuro-Nimes*, 1991.

[40] Denny Britz, Anna Goldie, Thang Luong, and Quoc Le. Massive exploration of neural machine translation architectures. *arXiv preprint arXiv:1703.03906*, 2017.

[41] Jane Bromley, Isabelle Guyon, Yann LeCun, Eduard Säckinger, and Roopak Shah. Signature verification using a" siamese" time delay neural network. *Advances in neural information processing systems*, 6, 1993.

[42] Tom B Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *arXiv preprint arXiv:2005.14165*, 2020.

[43] Elia Bruni, Nam-Khanh Tran, and Marco Baroni. Multimodal distributional semantics. *Journal of Artificial Intelligence Research*, 49:1–47, 2014.

[44] Cristian Bucilu‘, Rich Caruana, and Alexandru Niculescu-Mizil. Model compression. In *Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 535–541, 2006.

[45] Pawel Budzianowski, Tsung-Hsien Wen, Bo-Hsiang Tseng, Iñigo Casanueva, Stefan Ultes, Osman Ramadan, and Milica Gasic. Multiwoz - a large-scale multi-domain wizard-of-oz dataset for task-oriented dialogue modelling. In *EMNLP*, 2018.

[46] Qingqing Cao, Harsh Trivedi, Aruna Balasubramanian, and Niranjan Balasubramanian. Deformer: Decomposing pre-trained transformers for faster question answering. *arXiv preprint arXiv:2005.00697*, 2020.

[47] Steven Cao, Nikita Kitaev, and Dan Klein. Multilingual alignment of contextual word representations. *arXiv preprint arXiv:2002.03518*, 2020.

[48] Li Zhang Karthik Ramanathan Sesh Sadasivam Rui Zhang Catherine Finegan-Dollak, Jonathan K. Kummerfeld and Dragomir Radev. Improving text-to-sql evaluation methodology. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 351–360, July 2018.

[49] Ashok K. Chandra, Dexter C. Kozen, and Larry J. Stockmeyer. Alternation. *Journal of the Association for Computing Machinery*, 28(1):114–133, 1981.

[50] Ernie Chang, Xiaoyu Shen, Hui-Syuan Yeh, and Vera Demberg. On training instance selection for few-shot neural text generation. *arXiv preprint arXiv:2107.03176*, 2021.

[51] Shuaichen Chang, Pengfei Liu, Yun Tang, Jing Huang, Xiaodong He, and Bowen Zhou. Zero-shot text-to-sql learning with auxiliary task.

[52] Wei-Cheng Chang, Felix X Yu, Yin-Wen Chang, Yiming Yang, and Sanjiv Kumar. Pre-training tasks for embedding-based large-scale retrieval. *arXiv preprint arXiv:2002.03932*, 2020.

[53] Danqi Chen, Adam Fisch, Jason Weston, and Antoine Bordes. Reading wikipedia to answer open-domain questions. *arXiv preprint arXiv:1704.00051*, 2017.

[54] David L Chen and Raymond J Mooney. Learning to interpret natural language navigation instructions from observations. In *Twenty-Fifth AAAI Conference on Artificial Intelligence*, 2011.

[55] Mark Chen, Jerry Tworek, Heewoo Jun, Qiming Yuan, Henrique Ponde de Oliveira Pinto, Jared Kaplan, Harri Edwards, Yuri Burda, Nicholas Joseph, Greg Brockman, et al. Evaluating large language models trained on code. *arXiv preprint arXiv:2107.03374*, 2021.

[56] Muhao Chen, Yingtao Tian, Kai-Wei Chang, Steven Skiena, and Carlo Zaniolo. Co-training embeddings of knowledge graphs and entity descriptions for cross-lingual entity alignment. In *Proceedings of IJCAI*, 2018.

[57] Muhao Chen, Yingtao Tian, Mohan Yang, and Carlo Zaniolo. Multilingual knowledge graph embeddings for cross-lingual knowledge alignment. *arXiv preprint arXiv:1611.03954*, 2016.

[58] Muhao Chen, Yingtao Tian, Mohan Yang, and Carlo Zaniolo. Multilingual knowledge graph embeddings for cross-lingual knowledge alignment. In *Proceedings of IJCAI*, 2017.

[59] Muhao Chen, Tao Zhou, Pei Zhou, and Carlo Zaniolo. Multi-graph affinity embeddings for multilingual knowledge graphs. In *Proceedings of NIPS Workshop on Automated Knowledge Base Construction*, 2017.

[60] Ruey-Cheng Chen, Luke Gallagher, Roi Blanco, and J Shane Culpepper. Efficient cost-aware cascade ranking in multi-stage retrieval. In *Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 445–454, 2017.

[61] Wenhu Chen, Ming-Wei Chang, Eva Schlinger, William Wang, and William W Cohen. Open question answering over tables and text. *arXiv preprint arXiv:2010.10439*, 2020.

[62] Wenhu Chen, Yu Su, Xifeng Yan, and William Yang Wang. Kgpt: Knowledge-grounded pre-training for data-to-text generation. *arXiv preprint arXiv:2010.02307*, 2020.

[63] Wenhu Chen, Hongmin Wang, Jianshu Chen, Yunkai Zhang, Hong Wang, Shiyang Li, Xiyou Zhou, and William Yang Wang. Tabfact: A large-scale dataset for table-based fact verification. *arXiv preprint arXiv:1909.02164*, 2019.

[64] Xilun Chen, Asish Ghoshal, Yashar Mehdad, Luke Zettlemoyer, and Sonal Gupta. Low-resource domain adaptation for compositional task-oriented semantic parsing, 2020.

[65] Xilun Chen, Yu Sun, Ben Athiwaratkun, Claire Cardie, and Kilian Weinberger. Adversarial deep averaging networks for cross-lingual sentiment classification. *Transactions of the Association for Computational Linguistics*, 6:557–570, 2018.

[66] Yanda Chen, Chris Kedzie, Suraj Nair, Petra Galuščáková, Rui Zhang, Douglas W Oard, and Kathleen McKeown. Cross-language sentence selection via data augmentation and rationale training. *arXiv preprint arXiv:2106.02293*, 2021.

[67] Yen-Chun Chen and Mohit Bansal. Fast abstractive summarization with reinforce-selected sentence rewriting. In *Proceedings of ACL*, 2018.

[68] Zhiyu Chen, Harini Eavani, Wenhu Chen, Yinyin Liu, and William Yang Wang. Few-shot nlg with pre-trained language model. *arXiv preprint arXiv:1904.09521*, 2019.

[69] Zewen Chi, Li Dong, Furu Wei, Wenhui Wang, Xian-Ling Mao, and Heyan Huang. Cross-lingual natural language generation via pre-training. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2020.

[70] Zewen Chi, Li Dong, Furu Wei, Nan Yang, Saksham Singhal, Wenhui Wang, Xia Song, Xian-Ling Mao, Heyan Huang, and Ming Zhou. Infoxlm: An information-theoretic framework for cross-lingual language model pre-training. *arXiv preprint arXiv:2007.07834*, 2020.

[71] Zewen Chi, Shaohan Huang, Li Dong, Shuming Ma, Saksham Singhal, Payal Bajaj, Xia Song, and Furu Wei. Xlm-e: Cross-lingual language model pre-training via electra. *arXiv preprint arXiv:2106.16138*, 2021.

[72] Donghyun Choi, Myeong Cheol Shin, Eunggyun Kim, and Dong Ryeol Shin. Ryansql: Recursively applying sketch-based slot fillings for complex text-to-sql in cross-domain databases. *ArXiv*, abs/2004.03125, 2020.

[73] DongHyun Choi, Myeong Cheol Shin, EungGyun Kim, and Dong Ryeol Shin. RYANSQL: Recursively applying sketch-based slot fillings for complex text-to-SQL in cross-domain databases. *Computational Linguistics*, 47(2):309–332, June 2021.

[74] Jonathan H Clark, Eunsol Choi, Michael Collins, Dan Garrette, Tom Kwiatkowski, Vitaly Nikolaev, and Jennimaria Palomaki. Tydi qa: A benchmark for information-seeking question answering in typologically diverse languages. *Transactions of the Association for Computational Linguistics*, 8:454–470, 2020.

[75] Peter Clark, Isaac Cowhey, Oren Etzioni, Tushar Khot, Ashish Sabharwal, Carissa Schoenick, and Oyvind Tafjord. Think you have solved question answering? try arc, the ai2 reasoning challenge. *arXiv preprint arXiv:1803.05457*, 2018.

[76] Guillem Collell, Ted Zhang, and Marie-Francine Moens. Imagined visual representations as multimodal embeddings. In *Proceedings of AAAI*, 2017.

[77] Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. Unsupervised cross-lingual representation learning at scale. *arXiv preprint arXiv:1911.02116*, 2019.

[78] Alexis Conneau, Douwe Kiela, Holger Schwenk, Loïc Barrault, and Antoine Bordes. Supervised learning of universal sentence representations from natural language inference data. In *EMNLP*, pages 670–680, 2017.

[79] Alexis Conneau, Ruty Rinott, Guillaume Lample, Adina Williams, Samuel Bowman, Holger Schwenk, and Veselin Stoyanov. XNLI: Evaluating cross-lingual sentence representations. In *Proceedings of EMNLP*, 2018.

[80] James W. Cooley and John W. Tukey. An algorithm for the machine calculation of complex Fourier series. *Mathematics of Computation*, 19(90):297–301, 1965.

[81] Gordon V. Cormack, Charles L. A. Clarke, and Stefan Büttcher. Reciprocal rank fusion outperforms Condorcet and individual rank learning methods. In *Proceedings of the 32nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 2009)*, pages 758–759, Boston, Massachusetts, 2009.

[82] Matt Crane. Questionable answers in question answering research: Reproducibility and variability of published results. *Transactions of the Association for Computational Linguistics 6 (2018)*, page 241“252, 2018.

[83] Nick Craswell, Bhaskar Mitra, Emine Yilmaz, Daniel Campos, and Ellen M Voorhees. Overview of the trec 2019 deep learning track. *arXiv preprint arXiv:2003.07820*, 2020.

[84] Fabio Crestani, Mounia Lalmas, Cornelis J Van Rijsbergen, and Iain Campbell. “is this document relevant?“ probably“ a survey of probabilistic models in information retrieval. *ACM Computing Surveys (CSUR)*, 30(4):528–552, 1998.

[85] Ruixiang Cui, Rahul Aralikatte, Heather Lent, and Daniel Hershcovich. Multilingual compositional wikidata questions. *arXiv preprint arXiv:2108.03509*, 2021.

[86] Deborah A. Dahl, Madeleine Bates, Michael Brown, William Fisher, Kate Hunicke-Smith, David Pallett, Christine Pao, Alexander Rudnicky, and Elizabeth Shriberg. Expanding the scope of the ATIS task: The ATIS-3 corpus. In *Human Language Technology: Proceedings of a Workshop held at Plainsboro, New Jersey, March 8-11, 1994*, 1994.

[87] Deborah A. Dahl, Madeleine Bates, Michael Brown, William Fisher, Kate Hunicke-Smith, David Pallett, Christine Pao, Alexander Rudnicky, and Elizabeth Shriberg. Expanding the scope of the atis task: The atis-3 corpus. In *Proceedings of the Workshop on Human Language Technology*, HLT '94, Stroudsburg, PA, USA, 1994. Association for Computational Linguistics.

[88] Zhuyun Dai and Jamie Callan. Context-aware sentence/passage term importance estimation for first stage retrieval. *arXiv preprint arXiv:1910.10687*, 2019.

[89] Zhuyun Dai and Jamie Callan. Deeper text understanding for ir with contextual neural language modeling. In *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 985–988, 2019.

[90] Zhuyun Dai and Jamie Callan. Deeper text understanding for IR with contextual neural language modeling. In *Proceedings of the 42nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 2019)*, pages 985–988, Paris, France, 2019.

[91] Zhuyun Dai, Chenyan Xiong, Jamie Callan, and Zhiyuan Liu. Convolutional neural networks for soft-matching n-grams in ad-hoc search. In *WSDM*, pages 126–134, 2018.

[92] Jeffrey Dalton, Chenyan Xiong, Vaibhav Kumar, and Jamie Callan. Cast-19: A dataset for conversational information seeking. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 1985–1988, 2020.

[93] Marco Damonte and Shay B Cohen. Structural neural encoders for amr-to-text generation. *arXiv preprint arXiv:1903.11410*, 2019.

[94] Kareem Darwish. Named entity recognition using cross-lingual resources: Arabic as an example. In *Proceedings of ACL*, 2013.

[95] Dipanjan Das, Nathan Schneider, Desai Chen, and Noah A. Smith. Probabilistic frame-semantic parsing. In *NAACL*, 2010.

[96] Michael Brown William Fisher Kate Hunicke-Smith David Pallett Christine Pao Alexander Rudnicky Deborah A. Dahl, Madeleine Bates and Elizabeth Shriber. Expanding the scope of the ATIS task: The ATIS-3 corpus. *Proceedings of the workshop on Human Language Technology*, pages 43–48, 1994.

[97] Mostafa Dehghani, Hamed Zamani, Aliaksei 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*, pages 65–74. ACM, 2017.

[98] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. ImageNet: A Large-Scale Hierarchical Image Database. In *CVPR09*, 2009.

[99] Xiang Deng, Ahmed Hassan Awadallah, Christopher Meek, Oleksandr Polozov, Huan Sun, and Matthew Richardson. Structure-grounded pretraining for text-to-sql. *arXiv preprint arXiv:2010.12773*, 2020.

[100] Xiang Deng, Huan Sun, Alyssa Lees, You Wu, and Cong Yu. Turl: Table understanding through representation learning. *arXiv preprint arXiv:2006.14806*, 2020.

[101] Tim Dettmers, Pasquale Minervini, Pontus Stenetorp, and Sebastian Riedel. Convolutional 2d knowledge graph embeddings. In *Proceedings of AAAI*, 2018.

[102] Daniel Deutch, Nave Frost, and Amir Gilad. Provenance for natural language queries. *Proc. VLDB Endow.*, 10(5):577–588, 2017.

[103] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

[104] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, 2019.

[105] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of NAACL*, 2019.

[106] Bhuwan Dhingra, Manaal Faruqui, Ankur Parikh, Ming-Wei Chang, Dipanjan Das, and William W Cohen. Handling divergent reference texts when evaluating table-to-text generation. *arXiv preprint arXiv:1906.01081*, 2019.

[107] Bhuwan Dhingra, Lihong Li, Xiujun Li, Jianfeng Gao, Yun-Nung Chen, Faisal Ahmed, and Li Deng. End-to-end reinforcement learning of dialogue agents for information access. In *ACL*, 2016.

[108] Giorgio M Di Nunzio, Nicola Ferro, Thomas Mandl, and Carol Peters. Clef 2006: Ad hoc track overview. In *Workshop of the Cross-Language Evaluation Forum for European Languages*. Springer, 2006.

[109] Emily Dinan, Stephen Roller, Kurt Shuster, Angela Fan, Michael Auli, and Jason Weston. Wizard of wikipedia: Knowledge-powered conversational agents. *CoRR*, abs/1811.01241, 2018.

[110] Li Dong and Mirella Lapata. Language to logical form with neural attention. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, ACL 2016, August 7-12, 2016, Berlin, Germany, Volume 1: Long Papers*, 2016.

[111] Li Dong and Mirella Lapata. Coarse-to-fine decoding for neural semantic parsing. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 731–742. Association for Computational Linguistics, 2018.

[112] Li Dong and Mirella Lapata. Coarse-to-fine decoding for neural semantic parsing. *arXiv preprint arXiv:1805.04793*, 2018.

[113] Li Dong, Nan Yang, Wenhui Wang, Furu Wei, Xiaodong Liu, Yu Wang, Jianfeng Gao, Ming Zhou, and Hsiao-Wuen Hon. Unified language model pre-training for natural language understanding and generation. In *Advances in Neural Information Processing Systems*, pages 13063–13075, 2019.

[114] Nan Duan, Duyu Tang, Peng Chen, and Ming Zhou. Question generation for question answering. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 866–874, 2017.

[115] Miles Efron, Jimmy Lin, Jiyin He, and Arjen De Vries. Temporal feedback for tweet search with non-parametric density estimation. In *Proceedings of the 37th international ACM SIGIR conference on Research & development in information retrieval*, pages 33–42. ACM, 2014.

[116] Julian Martin Eisenschlos, Syrine Krichene, and Thomas Müller. Understanding tables with intermediate pre-training. *arXiv preprint arXiv:2010.00571*, 2020.

[117] Akiko Eriguchi, Melvin Johnson, Orhan Firat, Hideto Kazawa, and Wolfgang Macherey. Zero-shot cross-lingual classification using multilingual neural machine translation. *arXiv preprint arXiv:1809.04686*, 2018.

[118] Angela Fan, Yacine Jernite, Ethan Perez, David Grangier, Jason Weston, and Michael Auli. Eli5: Long form question answering. *arXiv preprint arXiv:1907.09190*, 2019.

[119] Hui Fang, Tao Tao, and ChengXiang Zhai. A formal study of information retrieval heuristics. In *Proceedings of the 27th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 49–56. ACM, 2004.

[120] Qingkai Fang, Rong Ye, Lei Li, Yang Feng, and Mingxuan Wang. Stemm: Self-learning with speech-text manifold mixup for speech translation. *arXiv preprint arXiv:2203.10426*, 2022.

[121] Yuwei Fang, Shuohang Wang, Zhe Gan, Siqi Sun, and Jingjing Liu. Filter: An enhanced fusion method for cross-lingual language understanding. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 12776–12784, 2021.

[122] Fangxiaoyu Feng, Yinfei Yang, Daniel Cer, Naveen Arivazhagan, and Wei Wang. Language-agnostic bert sentence embedding. *arXiv preprint arXiv:2007.01852*, 2020.

[123] Catherine Finegan-Dollak, Jonathan K Kummerfeld, Li Zhang, Karthik Ramanathan, Sesh Sadasivam, Rui Zhang, and Dragomir Radev. Improving text-to-sql evaluation methodology. *arXiv preprint arXiv:1806.09029*, 2018.

[124] Catherine Finegan-Dollak, Jonathan K. Kummerfeld, Li Zhang, Karthik Ramanathan Dhanalakshmi Ramanathan, Sesh Sadasivam, Rui Zhang, and Dragomir Radev. Improving text-to-sql evaluation methodology. In *ACL 2018*. Association for Computational Linguistics, 2018.

[125] Association for Computing Machinery. *Computing Reviews*, 24(11):503–512, 1983.

[126] Stefan L. Frank. Uncertainty reduction as a measure of cognitive load in sentence comprehension. *Topics in cognitive science*, 5 3:475–94, 2013.

[127] George W. Furnas, Thomas K. Landauer, Louis M. Gomez, and Susan T. Dumais. The vocabulary problem in human-system communication. *Communications of the ACM*, 30(11):964–971, 1987.

[128] Daniel Furrer, Marc van Zee, Nathan Scales, and Nathanael Scharli. Compositional generalization in semantic parsing: Pre-training vs. specialized architectures. *ArXiv*, abs/2007.08970, 2020.

[129] Luyu Gao and Jamie Callan. Is your language model ready for dense representation fine-tuning? *arXiv preprint arXiv:2104.08253*, 2021.

[130] Luyu Gao and Jamie Callan. Unsupervised corpus aware language model pre-training for dense passage retrieval. *arXiv preprint arXiv:2108.05540*, 2021.

[131] Luyu Gao, Zhuyun Dai, and Jamie Callan. Coil: Revisit exact lexical match in information retrieval with contextualized inverted list. *arXiv preprint arXiv:2104.07186*, 2021.

[132] Yifan Gao, Henghui Zhu, Patrick Ng, Cicero Nogueira dos Santos, Zhiguo Wang, Feng Nan, Dejiao Zhang, Ramesh Nallapati, Andrew O. Arnold, and Bing Xiang. Answering ambiguous questions through generative evidence fusion and round-trip prediction. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 3263–3276, Online, August 2021. Association for Computational Linguistics.

[133] Claire Gardent, Anastasia Shimorina, Shashi Narayan, and Laura Perez-Beltrachini. Creating training corpora for nlg micro-planning. In *55th annual meeting of the Association for Computational Linguistics (ACL)*, 2017.

[134] Jonas Gehring, Michael Auli, David Grangier, Denis Yarats, and Yann N. Dauphin. Convolutional sequence to sequence learning. *arXiv:1705.03122*, 2017.

[135] Alessandra Giordani and Alessandro Moschitti. Automatic generation and reranking of sql-derived answers to nl questions. In *Proceedings of the Second International Conference on Trustworthy Eternal Systems via Evolving Software, Data and Knowledge*, pages 59–76, 2012.

[136] Alessandra Giordani and Alessandro Moschitti. Translating questions to sql queries with generative parsers discriminatively reranked. In *COLING (Posters)*, pages 401–410, 2012.

[137] Ofer Givoli and Roi Reichart. Zero-shot semantic parsing for instructions. *arXiv preprint arXiv:1911.08827*, 2019.

[138] Heng Gong, Xiaocheng Feng, Bing Qin, and Ting Liu. Table-to-text generation with effective hierarchical encoder on three dimensions (row, column and time). *arXiv preprint arXiv:1909.02304*, 2019.

[139] Heng Gong, Yawei Sun, Xiaocheng Feng, Bing Qin, Wei Bi, Xiaojiang Liu, and Ting Liu. Tablegpt: Few-shot table-to-text generation with table structure reconstruction and content matching. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 1978–1988, 2020.

[140] James Goodman, Andreas Vlachos, and Jason Naradowsky. Noise reduction and targeted exploration in imitation learning for abstract meaning representation parsing. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1–11. Association for Computational Linguistics, 2016.

[141] Saiping Guan, Xiaolong Jin, Yuanzhuo Wang, Yantao Jia, Huawei Shen, Zixuan Li, and Xueqi Cheng. Self-learning and embedding based entity alignment. *Knowledge and Information Systems*, 2019.

[142] Daya Guo, Yibo Sun, Duyu Tang, Nan Duan, Jian Yin, Hong Chi, James Cao, Peng Chen, and Ming Zhou. Question generation from sql queries improves neural semantic parsing. *arXiv preprint arXiv:1808.06304*, 2018.

[143] Jiafeng Guo, Yinqiong Cai, Yixing Fan, Fei Sun, Ruqing Zhang, and Xueqi Cheng. Semantic models for the first-stage retrieval: A comprehensive review. *ACM Transactions on Information Systems (TOIS)*, 40(4):1–42, 2022.

[144] Jiafeng Guo, Yixing Fan, Qingyao Ai, and W Bruce Croft. A deep relevance matching model for ad-hoc retrieval. In *Proceedings of the 25th ACM international on conference on information and knowledge management*, pages 55–64, 2016.

[145] Jiaqi Guo, Ziliang Si, Yu Wang, Qian Liu, Ming Fan, Jian-Guang Lou, Zijiang Yang, and Ting Liu. Chase: A large-scale and pragmatic Chinese dataset for cross-database context-dependent text-to-SQL. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 2316–2331, Online, August 2021. Association for Computational Linguistics.

[146] Jiaqi Guo, Zecheng Zhan, Yan Gao, Yan Xiao pei Liu Xiao, Jian-Guang Lou, Ting Liu, and Dongmei Zhang. Towards complex text-to-sql in cross-domain database with intermediate representation. In *ACL*, 2019.

[147] Jiaqi Guo, Zecheng Zhan, Yan Gao, Yan Xiao, Jian-Guang Lou, Ting Liu, and Dongmei Zhang. Towards complex text-to-sql in cross-domain database with intermediate representation. *arXiv preprint arXiv:1905.08205*, 2019.

[148] Jiaqi Guo, Zecheng Zhan, Yan Gao, Yan Xiao, Jian-Guang Lou, Ting Liu, and Dongmei Zhang. Towards complex text-to-sql in cross-domain database with intermediate representation, 2019.

[149] Zhijiang Guo, Yan Zhang, Zhiyang Teng, and Wei Lu. Attention guided graph convolutional networks for relation extraction. In *Proceedings of ACL*, 2019.

[150] Zhijiang Guo, Yan Zhang, Zhiyang Teng, and Wei Lu. Densely connected graph convolutional networks for graph-to-sequence learning. *Transactions of the Association of Computational Linguistics*, 7:297–312, 2019.

[151] Izzeddin Gur, Semih Yavuz, Yu Su, and Xifeng Yan. Dialsql: Dialogue based structured query generation. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1339–1349. Association for Computational Linguistics, 2018.

[152] Dan Gusfield. *Algorithms on Strings, Trees and Sequences.* Cambridge University Press, Cambridge, UK, 1997.

[153] John Hale. Uncertainty about the rest of the sentence. *Cognitive science*, 30 4:643–72, 2006.

[154] Xu Han, Zhiyuan Liu, and Maosong Sun. Joint representation learning of text and knowledge for knowledge graph completion. *arXiv preprint arXiv:1611.04125*, 2016.

[155] Yanchao Hao, Yuanzhe Zhang, Shizhu He, Kang Liu, and Jun Zhao. A joint embedding method for entity alignment of knowledge bases. In *Proceedings of China Conference on Knowledge Graph and Semantic Computing*, 2016.

[156] Pei hao Su, Milica Gasic, Nikola Mrksic, Lina Maria Rojas-Barahona, Stefan Ultes, David Vandyke, Tsung-Hsien Wen, and Steve J. Young. On-line active reward learning for policy optimisation in spoken dialogue systems. *ACL*, 2016.

[157] Donna K Harman. *Overview of the third text retrieval conference (TREC-3)*. Number 500. DIANE Publishing, 1995.

[158] Mary Harper. Learning from 26 languages: Program management and science in the Babel program. In *Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers*, page 1. Dublin City University and Association for Computational Linguistics, 2014.

[159] He He, Anusha Balakrishnan, Mihail Eric, and Percy Liang. Learning symmetric collaborative dialogue agents with dynamic knowledge graph embeddings. In *Proceedings of ACL*, 2017.

[160] Hua He, Kevin Gimpel, and Jimmy Lin. Multi-perspective sentence similarity modeling with convolutional neural networks. In *EMNLP*, pages 1576–1586, 2015.

[161] Hua He, John Wieting, Kevin Gimpel, Jinfeng Rao, and Jimmy Lin. UMD-TTIC-UW at SemEval-2016 task 1: Attention-based multi-perspective convolutional neural networks for textual similarity measurement. In *SemEval-2016*, pages 1103–1108, 2016.

[162] Charles T. Hemphill, John J. Godfrey, and George R. Doddington. The ATIS spoken language systems pilot corpus. In *Speech and Natural Language: Proceedings of a Workshop Held at Hidden Valley, Pennsylvania, June 24-27,1990*, 1990.

[163] Charles T. Hemphill, John J. Godfrey, and George R. Doddington. The atis spoken language systems pilot corpus. In *Speech and Natural Language: Proceedings of a Workshop Held at Hidden Valley, Pennsylvania, June 24-27,1990*, 1990.

[164] Matthew Henderson, Blaise Thomson, and Jason D. Williams. The second dialog state tracking challenge. In *SIGDIAL Conference*, 2014.

[165] Matthew Henderson, Blaise Thomson, and Steve Young. Deep neural network approach for the dialog state tracking challenge. In *Proceedings of the SIGDIAL 2013 Conference*, pages 467–471, Metz, France, August 2013. Association for Computational Linguistics.

[166] Karl Moritz Hermann and Phil Blunsom. Multilingual models for compositional distributed semantics. In *Proceedings of ACL*, 2014.

[167] Jonathan Herzig, Thomas Müller, Syrine Krichene, and Julian Martin Eisenschlos. Open domain question answering over tables via dense retrieval. *arXiv preprint arXiv:2103.12011*, 2021.

[168] Jonathan Herzig, Paweł Krzysztof Nowak, Thomas Müller, Francesco Piccinno, and Julian Martin Eisenschlos. Tapas: Weakly supervised table parsing via pre-training. *arXiv preprint arXiv:2004.02349*, 2020.

[169] John Hewitt and Percy Liang. Designing and interpreting probes with control tasks. *arXiv preprint arXiv:1909.03368*, 2019.

[170] John Hewitt and Christopher D Manning. A structural probe for finding syntax in word representations. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 4129–4138, 2019.

[171] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2015.

[172] Vu Cong Duy Hoang, Philipp Koehn, Gholamreza Haffari, and Trevor Cohn. Iterative back-translation for neural machine translation. In *Proceedings of the 2nd Workshop on Neural Machine Translation and Generation*, pages 18–24, 2018.

[173] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.

[174] Yushi Hu, Chia-Hsuan Lee, Tianbao Xie, Tao Yu, Noah A Smith, and Mari Ostendorf. In-context learning for few-shot dialogue state tracking. *arXiv preprint arXiv:2203.08568*, 2022.

[175] Haoyang Huang, Yaobo Liang, Nan Duan, Ming Gong, Linjun Shou, Daxin Jiang, and Ming Zhou. Unicoder: A universal language encoder by pre-training with multiple cross-lingual tasks. *arXiv preprint arXiv:1909.00964*, 2019.

[176] Po-Sen Huang, Xiaodong He, Jianfeng Gao, Li Deng, Alex Acero, and Larry Heck. Learning deep structured semantic models for web search using clickthrough data. In *Proceedings of the 22nd ACM international conference on Information & Knowledge Management*, pages 2333–2338, 2013.

[177] Po-Sen Huang, Chenglong Wang, Rishabh Singh, Wen tau Yih, and Xiaodong He. Natural language to structured query generation via meta-learning. In *NAACL*, 2018.

[178] Xiaolei Huang, Jonathan May, and Nanyun Peng. What matters for neural cross-lingual named entity recognition: An empirical analysis. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 6395–6401, Hong Kong, China, 2019.

[179] Binyuan Hui, Ruiying Geng, Lihan Wang, Bowen Qin, Yanyang Li, Bowen Li, Jian Sun, and Yongbin Li. S^{2}SQL: Injecting syntax to question-schema interaction graph encoder for text-to-SQL parsers. In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 1254–1262, Dublin, Ireland, May 2022.

[180] Kai Hui, Andrew Yates, Klaus Berberich, and Gerard de Melo. Pacrr: A position-aware neural ir model for relevance matching. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 1049–1058, 2017.

[181] Kai Hui, Andrew Yates, Klaus Berberich, and Gerard de Melo. Re-pacrr: A context and density-aware neural information retrieval model. *arXiv:1706.10192*, 2017.

[182] Wonseok Hwang, Jinyeong Yim, Seunghyun Park, and Minjoon Seo. A comprehensive exploration on wikisql with table-aware word contextualization. *arXiv preprint arXiv:1902.01069*, 2019.

[183] Hiroshi Iida, Dung Thai, Varun Manjunatha, and Mohit Iyyer. Tabbie: Pretrained representations of tabular data. *arXiv preprint arXiv:2105.02584*, 2021.

[184] Ryo Ishida, Kentaro Torisawa, Jong-Hoon Oh, Ryu Iida, Canasai Kruengkrai, and Julien Kloetzer. Semi-distantly supervised neural model for generating compact answers to open-domain why questions. In *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018.

[185] Srinivasan Iyer, Ioannis Konstas, Alvin Cheung, Jayant Krishnamurthy, and Luke Zettlemoyer. Learning a neural semantic parser from user feedback. *CoRR*, abs/1704.08760, 2017.

[186] Srinivasan Iyer, Ioannis Konstas, Alvin Cheung, Jayant Krishnamurthy, and Luke Zettlemoyer. Learning a neural semantic parser from user feedback. *arXiv preprint arXiv:1704.08760*, 2017.

[187] Srinivasan Iyer, Ioannis Konstas, Alvin Cheung, and Luke Zettlemoyer. Summarizing source code using a neural attention model. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2073–2083, Berlin, Germany, August 2016. Association for Computational Linguistics.

[188] Srinivasan Iyer, Ioannis Konstas, Alvin Cheung, and Luke Zettlemoyer. Mapping language to code in programmatic context. *arXiv preprint arXiv:1808.09588*, 2018.

[189] Mohit Iyyer, Wen-tau Yih, and Ming-Wei Chang. Search-based neural structured learning for sequential question answering. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1821–1831, 2017.

[190] Mohit Iyyer, Wen-tau Yih, and Ming-Wei Chang. Search-based neural structured learning for sequential question answering. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1821–1831. Association for Computational Linguistics, 2017.

[191] Gautier Izacard and Edouard Grave. Leveraging passage retrieval with generative models for open domain question answering. *arXiv preprint arXiv:2007.01282*, 2020.

[192] Gautier Izacard and Edouard Grave. Leveraging passage retrieval with generative models for open domain question answering, 2021.

[193] Alankar Jain, Bhargavi Paranjape, and Zachary C Lipton. Entity projection via machine translation for cross-lingual ner. *arXiv preprint arXiv:1909.05356*, 2019.

[194] Guoliang Ji, Shizhu He, Liheng Xu, Kang Liu, and Jun Zhao. Knowledge graph embedding via dynamic mapping matrix. In *Proceedings of IJCNLP*, 2015.

[195] Robin Jia and Percy Liang. Data recombination for neural semantic parsing. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 2016.

[196] Robin Jia and Percy Liang. Data recombination for neural semantic parsing. *arXiv preprint arXiv:1606.03622*, 2016.

[197] Zhuolin Jiang, Amro El-Jaroudi, William Hartmann, Damianos Karakos, and Lingjun Zhao. Cross-lingual information retrieval with bert. *arXiv preprint arXiv:2004.13005*, 2020.

[198] Melvin Johnson, Mike Schuster, Quoc V. Le, Maxim Krikun, Yonghui Wu, Zhifeng Chen, Nikhil Thorat, Fernanda Viégas, Martin Wattenberg, Greg Corrado, et al. Google‘‘s multilingual neural machine translation system: Enabling zero-shot translation. *Transactions of the Association of Computational Linguistics*, 5:339–351, 2017.

[199] Bevan Jones, Mark Johnson, and Sharon Goldwater. Semantic parsing with bayesian tree transducers. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 488–496, 2012.

[200] Mandar Joshi, Eunsol Choi, Daniel S Weld, and Luke Zettlemoyer. Triviaqa: A large scale distantly supervised challenge dataset for reading comprehension. *arXiv preprint arXiv:1705.03551*, 2017.

[201] Pratik Joshi, Sebastin Santy, Amar Budhiraja, Kalika Bali, and Monojit Choudhury. The state and fate of linguistic diversity and inclusion in the NLP world. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, Online, July 2020.

[202] Armand Joulin, Piotr Bojanowski, Tomas Mikolov, Hervé Jégou, and Edouard Grave. Loss in translation: Learning bilingual word mapping with a retrieval criterion. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2979–2984, Brussels, Belgium, 2018.

[203] Aishwarya Kamath and Rajarshi Das. A survey on semantic parsing. *arXiv preprint arXiv:1812.00978*, 2018.

[204] Giannis Karamanolakis, Daniel Hsu, and Luis Gravano. Cross-lingual text classification with minimal resources by transferring a sparse teacher. *arXiv preprint arXiv:2010.02562*, 2020.

[205] Vladimir Karpukhin, Barlas Oguz, Sewon Min, Patrick Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. Dense passage retrieval for open-domain question answering. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6769–6781, Online, November 2020. Association for Computational Linguistics.

[206] Amol Kelkar, Rohan Relan, Vaishali Bhardwaj, Saurabh Vaichal, and Peter Relan. Bertrand-dr: Improving text-to-sql using a discriminative re-ranker. *arXiv preprint arXiv:2002.00557*, 2020.

[207] Daniel Khashabi, Sewon Min, Tushar Khot, Ashish Sabharwal, Oyvind Tafjord, Peter Clark, and Hannaneh Hajishirzi. Unifiedqa: Crossing format boundaries with a single qa system. *arXiv preprint arXiv:2005.00700*, 2020.

[208] Omar Khattab and Matei Zaharia. Colbert: Efficient and effective passage search via contextualized late interaction over bert. In *Proceedings of the 43rd International ACM SIGIR conference on research and development in Information Retrieval*, pages 39–48, 2020.

[209] Douwe Kiela and Léon Bottou. Learning image embeddings using convolutional neural networks for improved multi-modal semantics. In *Proceedings of EMNLP*, 2014.

[210] Yoon Kim. Convolutional neural networks for sentence classification. In *EMNLP*, pages 1746–1751, 2014.

[211] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[212] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *The 3rd International Conference for Learning Representations, San Diego*, 2015.

[213] Thomas N. Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. In *Proceedings of ICLR*, 2017.

[214] Kevin Knight, Bianca Badarau, Laura Baranescu, Claire Bonial, Madalina Bardocz, Kira Griffitt, Ulf Hermjakob, Daniel Marcu, Martha Palmer, Tim O'Gorman, et al. Abstract meaning representation (amr) annotation release 3.0. 2021.

[215] Tomáš Kočiskỳ, Jonathan Schwarz, Phil Blunsom, Chris Dyer, Karl Moritz Hermann, Gábor Melis, and Edward Grefenstette. The narrativeqa reading comprehension challenge. *Transactions of the Association for Computational Linguistics*, 6:317–328, 2018.

[216] Rik Koncel-Kedziorski, Dhanush Bekal, Yi Luan, Mirella Lapata, and Hannaneh Hajishirzi. Text generation from knowledge graphs with graph transformers. In *Proceedings of NAACL*, 2019.

[217] Kalpesh Krishna, Aurko Roy, and Mohit Iyyer. Hurdles to progress in long-form question answering. *arXiv preprint arXiv:2103.06332*, 2021.

[218] Jayant Krishnamurthy, Pradeep Dasigi, and Matt Gardner. Neural semantic parsing with type constraints for semi-structured tables.

[219] Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur Parikh, Chris Alberti, Danielle Epstein, Illia Polosukhin, Matthew Kelcey, Jacob Devlin, Kenton Lee, Kristina N. Toutanova, Llion Jones, Ming-Wei Chang, Andrew Dai, Jakob Uszkoreit, Quoc Le, and Slav Petrov. Natural questions: a benchmark for question answering research. *Transactions of the Association of Computational Linguistics*, 2019.

[220] Guillaume Lample, Alexis Conneau, Ludovic Denoyer, and Marc'Aurelio Ranzato. Unsupervised machine translation using monolingual corpora only. *arXiv preprint arXiv:1711.00043*, 2017.

[221] Guillaume Lample, Alexis Conneau, Marc'Aurelio Ranzato, Ludovic Denoyer, and Hervé Jégou. Word translation without parallel data. In *Proceedings of the 6th International Conference on Learning Representations (ICLR 2018)*, 2018.

[222] Walter Lasecki, Ece Kamar, and Dan Bohus. Conversations in the crowd: Collecting data for task-oriented dialog learning. In *In Proceedings of the Human Computation Workshop on Scaling Speech and Language Understanding and Dialog through Crowdsourcing at HCOMP 2013.*, January 2013.

[223] Walter S. Lasecki, Mitchell Gordon, Danai Koutra, Malte F. Jung, Steven P. Dow, and Jeffrey P. Bigham. Glance: Rapidly coding behavioral video with the crowd.

In *Proceedings of the 27th Annual ACM Symposium on User Interface Software and Technology*, New York, NY, USA, 2014. ACM.

[224] Victor Lavrenko and W. Bruce Croft. Relevance-based language models. In *SIGIR*, pages 120–127, 2001.

[225] Rémi Lebret, David Grangier, and Michael Auli. Neural text generation from structured data with application to the biography domain. *arXiv preprint arXiv:1603.07771*, 2016.

[226] Rémi Lebret, David Grangier, and Michael Auli. Neural text generation from structured data with application to the biography domain. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1203–1213. Association for Computational Linguistics, 2016.

[227] Chia-Hsuan Lee, Oleksandr Polozov, and Matthew Richardson. Kaggledbqa: Realistic evaluation of text-to-sql parsers. *arXiv preprint arXiv:2106.11455*, 2021.

[228] Ji Young Lee and Franck Dernoncourt. Sequential short-text classification with recurrent and convolutional neural networks. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 515–520, San Diego, California, June 2016. Association for Computational Linguistics.

[229] Jinhyuk Lee, Mujeen Sung, Jaewoo Kang, and Danqi Chen. Learning dense representations of phrases at scale. *arXiv preprint arXiv:2012.12624*, 2020.

[230] Kyungjae Lee, Sunghyun Park, Hojae Han, Jinyoung Yeo, Seung-won Hwang, and Juho Lee. Learning with limited data for multilingual reading comprehension. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 2840–2850, 2019.

[231] Roger P. Levy. Expectation-based syntactic comprehension. *Cognition*, 106:1126–1177, 2008.

[232] Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Ves Stoyanov, and Luke Zettlemoyer. Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. *arXiv preprint arXiv:1910.13461*, 2019.

[233] Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7871–7880, Online, July 2020.

[234] Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7871–7880, 2020.

[235] Patrick Lewis, Barlas Oğuz, Ruty Rinott, Sebastian Riedel, and Holger Schwenk. Mlqa: Evaluating cross-lingual extractive question answering. *arXiv preprint arXiv:1910.07475*, 2019.

[236] Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich K'ttler, Mike Lewis, Wen tau Yih, Tim Rockt'schel, Sebastian Riedel, and Douwe Kiela. Retrieval-augmented generation for knowledge-intensive nlp tasks, 2021.

[237] Belinda Z Li, Sewon Min, Srinivasan Iyer, Yashar Mehdad, and Wen-tau Yih. Efficient one-pass end-to-end entity linking for questions. *arXiv preprint arXiv:2010.02413*, 2020.

[238] Canjia Li, Andrew Yates, Sean MacAvaney, Ben He, and Yingfei Sun. PARADE: Passage representation aggregation for document reranking. *arXiv:2008.09093*, 2020.

[239] Fei Li and H. V. Jagadish. Constructing an interactive natural language interface for relational databases. *Proceedings of the VLDB Endowment*, 8(1):73–84, September 2014.

[240] Fei Li and HV Jagadish. Constructing an interactive natural language interface for relational databases. *VLDB*, 2014.

[241] Haoran Li, Abhinav Arora, Shuohui Chen, Anchit Gupta, Sonal Gupta, and Yashar Mehdad. Mtop: A comprehensive multilingual task-oriented semantic parsing benchmark. *arXiv preprint arXiv:2008.09335*, 2020.

[242] Jicheng Li, Pengzhi Gao, Xuanfu Wu, Yang Feng, Zhongjun He, Hua Wu, and Haifeng Wang. Mixup decoding for diverse machine translation. In *Findings of*

the Association for Computational Linguistics: EMNLP 2021, pages 312–320, Punta Cana, Dominican Republic, November 2021. Association for Computational Linguistics.

[243] Jiwei Li, Will Monroe, Alan Ritter, Daniel Jurafsky, Michel Galley, and Jianfeng Gao. Deep reinforcement learning for dialogue generation. In *EMNLP*, 2016.

[244] Junyi Li, Tianyi Tang, Wayne Xin Zhao, Zhicheng Wei, Nicholas Jing Yuan, and Ji-Rong Wen. Few-shot knowledge graph-to-text generation with pretrained language models. *arXiv preprint arXiv:2106.01623*, 2021.

[245] Yunyao Li, Huahai Yang, and HV Jagadish. Constructing a generic natural language interface for an xml database. In *EDBT*, volume 3896, pages 737–754. Springer, 2006.

[246] Zechang Li, Yuxuan Lai, Yansong Feng, and Dongyan Zhao. Domain adaptation for semantic parsing, 2020.

[247] P. Liang, M. I. Jordan, and D. Klein. Learning dependency-based compositional semantics. In *Association for Computational Linguistics (ACL)*, pages 590–599, 2011.

[248] Percy Liang. Lambda dependency-based compositional semantics. *arXiv preprint arXiv:1309.4408*, 2013.

[249] Jimmy Lin. The neural hype and comparisons against weak baselines. *SIGIR Forum*, 52(2):40–51, 2018.

[250] Jimmy Lin and Miles Efron. Overview of the TREC-2013 Microblog Track. In *TREC*, 2013.

[251] Jimmy Lin, Miles Efron, Yulu Wang, and Garrick Sherman. Overview of the TREC-2014 Microblog Track. In *Proceedings of the Twenty-Third Text REtrieval Conference (TREC 2014)*, Gaithersburg, Maryland, 2014.

[252] Jimmy Lin, Miles Efron, Yulu Wang, and Garrick Sherman. Overview of the TREC-2014 Microblog Track. In *TREC*, 2014.

[253] Jimmy Lin, Rodrigo Nogueira, and Andrew Yates. Pretrained transformers for text ranking: Bert and beyond. *arXiv preprint arXiv:2010.06467*, 2020.

[254] Jimmy Lin, Rodrigo Nogueira, and Andrew Yates. *Pretrained Transformers for Text Ranking: BERT and Beyond*. Morgan & Claypool Publishers, 2021.

[255] Sheng-Chieh Lin, Jheng-Hong Yang, and Jimmy Lin. Distilling dense representations for ranking using tightly-coupled teachers. *arXiv preprint arXiv:2010.11386*, 2020.

[256] Sheng-Chieh Lin, Jheng-Hong Yang, and Jimmy Lin. Contextualized query embeddings for conversational search. *arXiv preprint arXiv:2104.08707*, 2021.

[257] Xi Victoria Lin, Richard Socher, and Caiming Xiong. Bridging textual and tabular data for cross-domain text-to-sql semantic parsing. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 4870–4888, 2020.

[258] Xi Victoria Lin, Chenglong Wang, Luke Zettlemoyer, and Michael D Ernst. Nl2bash: A corpus and semantic parser for natural language interface to the linux operating system. *arXiv preprint arXiv:1802.08979*, 2018.

[259] Xi Victoria Lin, Chenglong Wang, Luke Zettlemoyer, and Michael D. Ernst. Nl2bash: A corpus and semantic parser for natural language interface to the linux operating system. In *LREC*, 2018.

[260] Yankai Lin, Zhiyuan Liu, Huanbo Luan, Maosong Sun, Siwei Rao, and Song Liu. Modeling relation paths for representation learning of knowledge bases. *arXiv preprint arXiv:1506.00379*, 2015.

[261] Yankai Lin, Zhiyuan Liu, Maosong Sun, Yang Liu, and Xuan Zhu. Learning entity and relation embeddings for knowledge graph completion. In *Proceedings of AAAI*, 2015.

[262] Ying Lin, Heng Ji, Fei Huang, and Lingfei Wu. A joint neural model for information extraction with global features. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7999–8009, 2020.

[263] Wang Ling, Phil Blunsom, Edward Grefenstette, Karl Moritz Hermann, Tomás Ko-ciský, Fumin Wang, and Andrew Senior. Latent predictor networks for code generation. In *ACL (1)*. The Association for Computer Linguistics, 2016.

[264] Wang Ling, Edward Grefenstette, Karl Moritz Hermann, Tomáš Kočiskỳ, Andrew Senior, Fumin Wang, and Phil Blunsom. Latent predictor networks for code generation. *arXiv preprint arXiv:1603.06744*, 2016.

[265] Robert Litschko, Ivan Vulić, Simone Paolo Ponzetto, and Goran Glavaš. Evaluating multilingual text encoders for unsupervised cross-lingual retrieval. *arXiv preprint arXiv:2101.08370*, 2021.

[266] Robert Litschko, Ivan Vulić, Simone Paolo Ponzetto, and Goran Glavaš. On cross-lingual retrieval with multilingual text encoders. *Information Retrieval Journal*, 25(2):149–183, 2022.

[267] Jiachang Liu, Dinghan Shen, Yizhe Zhang, Bill Dolan, Lawrence Carin, and Weizhu Chen. What makes good in-context examples for gpt-3? *arXiv preprint arXiv:2101.06804*, 2021.

[268] Nelson F Liu, Matt Gardner, Yonatan Belinkov, Matthew E Peters, and Noah A Smith. Linguistic knowledge and transferability of contextual representations. *arXiv preprint arXiv:1903.08855*, 2019.

[269] Peter J Liu, Mohammad Saleh, Etienne Pot, Ben Goodrich, Ryan Sepassi, Lukasz Kaiser, and Noam Shazeer. Generating wikipedia by summarizing long sequences. *arXiv preprint arXiv:1801.10198*, 2018.

[270] Qian Liu, Bei Chen, Jiaqi Guo, Zeqi Lin, and Jian-guang Lou. Tapex: Table pre-training via learning a neural sql executor. *arXiv preprint arXiv:2107.07653*, 2021.

[271] Tianyu Liu, Kexiang Wang, Lei Sha, Baobao Chang, and Zhifang Sui. Table-to-text generation by structure-aware seq2seq learning. *arXiv preprint arXiv:1711.09724*, 2017.

[272] Yinhan Liu, Jiatao Gu, Naman Goyal, Xian Li, Sergey Edunov, Marjan Ghazvininejad, Mike Lewis, and Luke Zettlemoyer. Multilingual denoising pre-training for neural machine translation. *Transactions of the Association for Computational Linguistics*, 8:726–742, 2020.

[273] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*, 2019.

[274] Reginald Long, Panupong Pasupat, and Percy Liang. Simpler context-dependent logical forms via model projections. *arXiv preprint arXiv:1606.05378*, 2016.

[275] Shayne Longpre, Yi Lu, and Joachim Daiber. Mkqa: A linguistically diverse benchmark for multilingual open domain question answering. *arXiv preprint arXiv:2007.15207*, 2020.

[276] Shuqi Lu, Chenyan Xiong, Di He, Guolin Ke, Waleed Malik, Zhicheng Dou, Paul Bennett, Tieyan Liu, and Arnold Overwijk. Less is more: Pre-training a strong siamese encoder using a weak decoder. *arXiv preprint arXiv:2102.09206*, 2021.

[277] Wei Lu and Hwee Tou Ng. A probabilistic forest-to-string model for language generation from typed lambda calculus expressions. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*, pages 1611–1622, 2011.

[278] Yao Lu, Max Bartolo, Alastair Moore, Sebastian Riedel, and Pontus Stenetorp. Fantastically ordered prompts and where to find them: Overcoming few-shot prompt order sensitivity. *arXiv preprint arXiv:2104.08786*, 2021.

[279] 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*, 2015.

[280] Jianqiang Ma, Zeyu Yan, Shuai Pang, Yang Zhang, and Jianping Shen. Mention extraction and linking for sql query generation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6936–6942, 2020.

[281] Kaixin Ma, Hao Cheng, Xiaodong Liu, Eric Nyberg, and Jianfeng Gao. Open domain question answering with a unified knowledge interface. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1605–1620, 2022.

[282] Laurens van der Maaten and Geoffrey Hinton. Visualizing data using t-SNE. *Journal of machine learning research*, 9:2579–2605, 2008.

[283] Sean MacAvaney, Franco Maria Nardini, Raffaele Perego, Nicola Tonellotto, Nazli Goharian, and Ophir Frieder. Efficient document re-ranking for transformers by precomputing term representations. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 49–58, 2020.

[284] Sean MacAvaney, Luca Soldaini, and Nazli Goharian. Teaching a new dog old tricks: Resurrecting multilingual retrieval using zero-shot learning. In *European Conference on Information Retrieval*, pages 246–254. Springer, 2020.

[285] Sean MacAvaney, Luca Soldaini, and Nazli Goharian. Teaching a new dog old tricks: Resurrecting multilingual retrieval using zero-shot learning. In *Proceedings of the 42nd European Conference on Information Retrieval, Part II (ECIR 2020)*, pages 246–254, 2020.

[286] Sean MacAvaney, Luca Soldaini, and Nazli Goharian. Teaching a new dog old tricks: Resurrecting multilingual retrieval using zero-shot learning. In *Proceedings of the 42nd European Conference on Information Retrieval, Part II (ECIR 2020)*, pages 246–254, 2020.

[287] Sean MacAvaney, Andrew Yates, Arman Cohan, and Nazli Goharian. CEDR: Contextualized embeddings for document ranking. In *Proceedings of the 42nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 2019)*, pages 1101–1104, Paris, France, 2019.

[288] Mitchell P. Marcus, Mary Ann Marcinkiewicz, and Beatrice Santorini. Building a large annotated corpus of english: The penn treebank. *Comput. Linguist.*, 19(2):313–330, June 1993.

[289] Bryan McCann, Nitish Shirish Keskar, Caiming Xiong, and Richard Socher. The natural language decathlon: Multitask learning as question answering. *arXiv preprint arXiv:1806.08730*, 2018.

[290] Ryan McDonald, George Brokos, and Ion Androutsopoulos. Deep relevance ranking using enhanced document-query interactions. In *EMNLP*, pages 1849–1860, 2018.

[291] Kathleen R. McKeown. *Text Generation: Using Discourse Strategies and Focus Constraints to Generate Natural Language Text*. Cambridge University Press, New York, NY, USA, 1985.

[292] Tomas Mikolov, Quoc V Le, and Ilya Sutskever. Exploiting similarities among languages for machine translation. *arXiv preprint arXiv:1309.4168*, 2013.

[293] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. Distributed representations of words and phrases and their compositionality. In C. J. C. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 26*, pages 3111–3119. Curran Associates, Inc., 2013.

[294] Qingkai Min, Yuefeng Shi, and Yue Zhang. A pilot study for chinese sql semantic parsing. *arXiv preprint arXiv:1909.13293*, 2019.

[295] Sewon Min, Julian Michael, Hannaneh Hajishirzi, and Luke Zettlemoyer. AmbigQA: Answering ambiguous open-domain questions. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 5783–5797, Online, November 2020. Association for Computational Linguistics.

[296] Teruko Mitamura, Eric Nyberg, Hideki Shima, Tsuneaki Kato, Tatsunori Mori, Chin-Yew Lin, Ruihua Song, Chuan-Jie Lin, Tetsuya Sakai, Donghong Ji, et al. Overview of the ntcir-7 aclia tasks: Advanced cross-lingual information access. In *NTCIR*, 2008.

[297] Bhaskar Mitra, Fernando Diaz, and Nick Craswell. Learning to match using local and distributed representations of text for web search. In *WWW*, pages 1291–1299, 2017.

[298] Salman Mohammed, Peng Shi, and Jimmy Lin. Strong baselines for simple question answering over knowledge graphs with and without neural networks. In *NAACL*, pages 291–296, 2018.

[299] Salman Mohammed, Peng Shi, and Jimmy Lin. Strong baselines for simple question answering over knowledge graphs with and without neural networks. In *Proceedings of NAACL*, 2018.

[300] Lili Mou, Zhengdong Lu, Hang Li, and Zhi Jin. Coupling distributed and symbolic execution for natural language queries. *ICML*, 2017.

[301] Lili Mou, Rui Men, Ge Li, Yan Xu, Lu Zhang, Rui Yan, and Zhi Jin. Natural language inference by tree-based convolution and heuristic matching. In *ACL*, 2016.

[302] Diego Moussallem, Matthias Wauer, and Axel-Cyrille Ngonga Ngomo. Machine translation using semantic web technologies: A survey. *Journal of Web Semantics*, 51:1–19, 2018.

[303] Nikola Mrkšić, Diarmuid Ó Séaghdha, Tsung-Hsien Wen, Blaise Thomson, and Steve Young. Neural belief tracker: Data-driven dialogue state tracking. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1777–1788. Association for Computational Linguistics, 2017.

[304] Nikola Mrksic, Diarmuid Ó Séaghdha, Blaise Thomson, Milica Gasic, Pei hao Su, David Vandyke, Tsung-Hsien Wen, and Steve J. Young. Multi-domain dialog state tracking using recurrent neural networks. In *ACL*, 2015.

[305] Nikola Mrksic, Diarmuid Ó Séaghdha, Tsung-Hsien Wen, Blaise Thomson, and Steve J. Young. Neural belief tracker: Data-driven dialogue state tracking. 2017.

[306] Sidharth Mudgal, Han Li, Theodoros Rekatsinas, AnHai Doan, Youngchoon Park, Ganesh Krishnan, Rohit Deep, Esteban Arcaute, and Vijay Raghavendra. Deep

learning for entity matching: A design space exploration. In *Proceedings of SIGMOD*, 2018.

[307] Phoebe Mulcaire, Swabha Swayamdipta, and Noah Smith. Polyglot semantic role labeling. *arXiv preprint arXiv:1805.11598*, 2018.

[308] Linyong Nan, Chiachun Hsieh, Ziming Mao, Xi Victoria Lin, Neha Verma, Rui Zhang, Wojciech Kryściński, Nick Schoelkopf, Riley Kong, Xiangru Tang, et al. Fetaqa: Free-form table question answering. *arXiv preprint arXiv:2104.00369*, 2021.

[309] Linyong Nan, Dragomir Radev, Rui Zhang, Amrit Rau, Abhinand Sivaprasad, Chiachun Hsieh, Xiangru Tang, Aadit Vyas, Neha Verma, Pranav Krishna, et al. Dart: Open-domain structured data record to text generation. *arXiv preprint arXiv:2007.02871*, 2020.

[310] Isil Dillig Navid Yaghmazadeh, Yuepeng Wang and Thomas Dillig. Sqlizer: Query synthesis from natural language. In *International Conference on Object-Oriented Programming, Systems, Languages, and Applications, ACM*, pages 63:1–63:26, October 2017.

[311] Roberto Navigli and Simone Paolo Ponzetto. BabelNet: The automatic construction, evaluation and application of a wide-coverage multilingual semantic network. *Artificial Intelligence*, 193:217–250, 2012.

[312] Axel-Cyrille Ngonga Ngomo and Soren Auer. Limes: a time-efficient approach for large-scale link discovery on the web of data. In *Proceedings of IJCAI*, 2011.

[313] Anh Tuan Nguyen, Mai Hoang Dao, and Dat Quoc Nguyen. A pilot study of text-to-sql semantic parsing for vietnamese. *arXiv preprint arXiv:2010.01891*, 2020.

[314] Dai Quoc Nguyen, Tu Dinh Nguyen, Dat Quoc Nguyen, and Dinh Phung. A novel embedding model for knowledge base completion based on convolutional neural network. In *Proceedings of ENMLP*, 2018.

[315] Dai Quoc Nguyen, Thanh Vu, Tu Dinh Nguyen, Dat Quoc Nguyen, and Dinh Q. Phung. A capsule network-based embedding model for knowledge graph completion and search personalization. In *Proceedings of NAACL*, 2019.

[316] Dat Quoc Nguyen. An overview of embedding models of entities and relationships for knowledge base completion. *arXiv preprint arXiv:1703.08098*, 2017.

[317] Dat Quoc Nguyen, Kairit Sirts, Lizhen Qu, and Mark Johnson. Stranse: a novel embedding model of entities and relationships in knowledge bases. In *Proceedings of NAACL*, 2016.

[318] Tri Nguyen, Mir Rosenberg, Xia Song, Jianfeng Gao, Saurabh Tiwary, Rangan Majumder, and Li Deng. Ms marco: A human generated machine reading comprehension dataset. In *CoCo@ NIPS*, 2016.

[319] Maximilian Nickel, Kevin Murphy, Volker Tresp, and Evgeniy Gabrilovich. A review of relational machine learning for knowledge graphs. *Proceedings of the IEEE*, 104(1):11–33, 2016.

[320] Jian-Yun Nie. *Cross-Language Information Retrieval.* Morgan & Claypool Publishers, 2010.

[321] Erik Nijkamp, Bo Pang, Hiroaki Hayashi, Lifu Tu, Huan Wang, Yingbo Zhou, Silvio Savarese, and Caiming Xiong. A conversational paradigm for program synthesis. *arXiv preprint arXiv:2203.13474*, 2022.

[322] Joakim Nivre, Marie-Catherine De Marneffe, Filip Ginter, Yoav Goldberg, Jan Hajic, Christopher D Manning, Ryan McDonald, Slav Petrov, Sampo Pyysalo, Natalia Silveira, et al. Universal dependencies v1: A multilingual treebank collection. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*, pages 1659–1666, 2016.

[323] Rodrigo Nogueira and Kyunghyun Cho. Passage re-ranking with bert. *arXiv preprint arXiv:1901.04085*, 2019.

[324] Rodrigo Nogueira, Wei Yang, Kyunghyun Cho, and Jimmy Lin. Multi-stage document ranking with bert. *arXiv preprint arXiv:1910.14424*, 2019.

[325] Rodrigo Nogueira, Wei Yang, Jimmy Lin, and Kyunghyun Cho. Document expansion by query prediction. *arXiv preprint arXiv:1904.08375*, 2019.

[326] Jekaterina Novikova, Ondřej Dušek, and Verena Rieser. The e2e dataset: New challenges for end-to-end generation. *arXiv preprint arXiv:1706.09254*, 2017.

[327] Douglas W Oard and Fredric C Gey. The trec 2002 arabic/english clir track. In *TREC*, 2002.

[328] Yusuke Oda, Hiroyuki Fudaba, Graham Neubig, Hideaki Hata, Sakriani Sakti, Tomoki Toda, and Satoshi Nakamura. Learning to generate pseudo-code from source code using statistical machine translation (t). In *Proceedings of the 2015 30th IEEE/ACM International Conference on Automated Software Engineering (ASE)*, ASE '15, 2015.

[329] Yusuke Oda, Hiroyuki Fudaba, Graham Neubig, Hideaki Hata, Sakriani Sakti, Tomoki Toda, and Satoshi Nakamura. Learning to generate pseudo-code from source code using statistical machine translation (t). In *2015 30th IEEE/ACM International Conference on Automated Software Engineering (ASE)*, pages 574–584. IEEE, 2015.

[330] Barlas Oguz, Xilun Chen, Vladimir Karpukhin, Stan Peshterliev, Dmytro Okhonko, Michael Schlichtkrull, Sonal Gupta, Yashar Mehdad, and Scott Yih. Unik-qa: Unified representations of structured and unstructured knowledge for open-domain question answering. *arXiv preprint arXiv:2012.14610*, 2020.

[331] Jong-Hoon Oh, Kentaro Torisawa, Canasai Kruengkrai, Ryu Iida, and Julien Kloetzer. Multi-column convolutional neural networks with causality-attention for why-question answering. In *Proceedings of the Tenth ACM international conference on web search and data mining*, pages 415–424, 2017.

[332] Iadh Ounis, Craig Macdonald, Jimmy Lin, and Ian Soboroff. Overview of the TREC-2011 Microblog Track. In *TREC*, 2011.

[333] Sebastian Padó and Mirella Lapata. Cross-lingual annotation projection for semantic roles. *Journal of Artificial Intelligence Research*, 36:307–340, 2009.

[334] Liang Pang, Yanyan Lan, Jiafeng Guo, Jun Xu, Shengxian Wan, and Xueqi Cheng. Text matching as image recognition. In *AAAI*, pages 2793–2799, 2016.

[335] Liang Pang, Yanyan Lan, Jiafeng Guo, Jun Xu, Jingfang Xu, and Xueqi Cheng. DeepRank: A new deep architecture for relevance ranking in information retrieval. In *CIKM*, pages 257–266, 2017.

[336] Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. Bleu: A method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*, ACL '02, pages 311–318. Association for Computational Linguistics, 2002.

[337] Ankur P Parikh, Xuezhi Wang, Sebastian Gehrmann, Manaal Faruqui, Bhuwan Dhingra, Diyi Yang, and Dipanjan Das. Totto: A controlled table-to-text generation dataset. *arXiv preprint arXiv:2004.14373*, 2020.

[338] Panupong Pasupat and Percy Liang. Compositional semantic parsing on semi-structured tables. *arXiv preprint arXiv:1508.00305*, 2015.

[339] Panupong Pasupat, Yuan Zhang, and Kelvin Guu. Controllable semantic parsing via retrieval augmentation. *arXiv preprint arXiv:2110.08458*, 2021.

[340] Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. Automatic differentiation in pytorch. *NIPS 2017 Workshop*, 2017.

[341] Shichao Pei, Lu Yu, Robert Hoehndorf, and Xiangliang Zhang. Semi-supervised entity alignment via knowledge graph embedding with awareness of degree difference. In *The World Wide Web Conference*, pages 3130–3136, 2019.

[342] Jeffrey Pennington, Richard Socher, and Christopher Manning. GloVe: Global vectors for word representation. In *EMNLP*, pages 1532–1543, 2014.

[343] Jeffrey Pennington, Richard Socher, and Christopher D. Manning. Glove: Global vectors for word representation. In *EMNLP*, pages 1532–1543. ACL, 2014.

[344] Maria Pershina, Mohamed Yakout, and Kaushik Chakrabarti. Holistic entity matching across knowledge graphs. In *Proceedings of IEEE International Conference on Big Data*, 2015.

[345] Matthew E Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. Deep contextualized word representations. In *Proceedings of NAACL*, 2018.

[346] Matthew E Peters, Mark Neumann, Robert L Logan IV, Roy Schwartz, Vidur Joshi, Sameer Singh, and Noah A Smith. Knowledge enhanced contextual word representations. *arXiv preprint arXiv:1909.04164*, 2019.

[347] Michael Petrochuk and Luke Zettlemoyer. SimpleQuestions nearly solved: A new upperbound and baseline approach. *arXiv:1804.08798*, 2018.

[348] Matúš Pikuliak, Marián Šimko, and Maria Bielikova. Cross-lingual learning for text processing: A survey. *Expert Systems with Applications*, 165:113765, 2021.

[349] Gabriel Poesia, Alex Polozov, Vu Le, Ashish Tiwari, Gustavo Soares, Chris Meek, and Sumit Gulwani. Synchromesh: Reliable code generation from pre-trained language models. 2021.

[350] Gabriel Poesia, Oleksandr Polozov, Vu Le, Ashish Tiwari, Gustavo Soares, Christopher Meek, and Sumit Gulwani. Synchromesh: Reliable code generation from pre-trained language models. *arXiv preprint arXiv:2201.11227*, 2022.

[351] Jay M. Ponte and W. Bruce Croft. A language modeling approach to information retrieval. In *SIGIR*, pages 275–281, 1998.

[352] Hoifung Poon. Grounded unsupervised semantic parsing. In *ACL*, 2013.

[353] Hoifung Poon. Grounded unsupervised semantic parsing. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 933–943, 2013.

[354] Ana-Maria Popescu, Alex Armanasu, Oren Etzioni, David Ko, and Alexander Yates. Modern natural language interfaces to databases: Composing statistical parsing with semantic tractability. In *Proceedings of the 20th international conference on Computational Linguistics*, page 141. Association for Computational Linguistics, 2004.

[355] Ana-Maria Popescu, Oren Etzioni, and Henry Kautz. Towards a theory of natural language interfaces to databases. In *Proceedings of the 8th international conference on Intelligent user interfaces*, pages 149–157. ACM, 2003.

[356] Ana-Maria Popescu, Oren Etzioni, and Henry Kautz. Towards a theory of natural language interfaces to databases. In *Proceedings of the 8th International Conference on Intelligent User Interfaces*, 2003.

[357] P. J. Price. Evaluation of spoken language systems: the atis domain. In *Speech and Natural Language: Proceedings of a Workshop Held at Hidden Valley, Pennsylvania, June 24-27,1990*, pages 91–95, 1990.

[358] Jiexing Qi, Jingyao Tang, Ziwei He, Xiangpeng Wan, Chenghu Zhou, Xinbing Wang, Quanshi Zhang, and Zhouhan Lin. RASAT: Integrating relational structures into pretrained seq2seq model for text-to-SQL. *arXiv preprint arXiv:2205.06983*, 2022.

[359] Libo Qin, Minheng Ni, Yue Zhang, and Wanxiang Che. Cosda-ml: Multi-lingual code-switching data augmentation for zero-shot cross-lingual nlp. *arXiv preprint arXiv:2006.06402*, 2020.

[360] Chris Quirk, Raymond Mooney, and Michel Galley. Language to code: Learning semantic parsers for if-this-then-that recipes. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, volume 1, pages 878–888, 2015.

[361] Maxim Rabinovich, Mitchell Stern, and Dan Klein. Abstract syntax networks for code generation and semantic parsing. In *ACL (1)*, pages 1139–1149. Association for Computational Linguistics, 2017.

[362] Dragomir Radev, Rui Zhang, Amrit Rau, Abhinand Sivaprasad, Chiachun Hsieh, Nazneen Fatema Rajani, Xiangru Tang, Aadit Vyas, Neha Verma, Pranav Krishna, et al. Dart: Open-domain structured data record to text generation. *arXiv preprint arXiv:2007.02871*, 2020.

[363] Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. Improving language understanding by generative pre-training. 2018.

[364] Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9, 2019.

[365] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *arXiv preprint arXiv:1910.10683*, 2019.

[366] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, Peter J Liu, et al. Exploring the limits of transfer learning with a unified text-to-text transformer. *J. Mach. Learn. Res.*, 21(140):1–67, 2020.

[367] Yves Raimond, Christopher Sutton, and Mark B. Sandler. Automatic interlinking of music datasets on the semantic web. In *Proceedings of WWW workshop on Linked Data on the Web*, 2008.

[368] Nitarshan Rajkumar, Raymond Li, and Dzmitry Bahdanau. Evaluating the text-to-sql capabilities of large language models. *arXiv preprint arXiv:2204.00498*, 2022.

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

[370] Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. Squad: 100,000+ questions for machine comprehension of text. *arXiv preprint arXiv:1606.05250*, 2016.

[371] Jinfeng Rao, Hua He, and Jimmy Lin. Noise-contrastive estimation for answer selection with deep neural networks. In *Proceedings of CIKM*, 2016.

[372] Jinfeng Rao, Hua He, and Jimmy Lin. Experiments with convolutional neural network models for answer selection. In *Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 1217–1220. ACM, 2017.

[373] Jinfeng Rao, Hua He, Haotian Zhang, Ferhan Ture, Royal Sequiera, Salman Mohammed, and Jimmy Lin. Integrating lexical and temporal signals in neural ranking models for searching social media streams. *arXiv:1707.07792*, 2017.

[374] Jinfeng Rao, Jimmy Lin, and Miles Efron. Reproducible experiments on lexical and temporal feedback for tweet search. In *European Conference on Information Retrieval*, pages 755–767. Springer, 2015.

[375] Jinfeng Rao, Wei Yang, Yuhao Zhang, Ferhan Ture, and Jimmy Lin. Multi-perspective relevance matching with hierarchical ConvNets for social media search. In *AAAI*, 2019.

[376] Mohammad Sadegh Rasooli and Joel R. Tetreault. Yara parser: A fast and accurate dependency parser. *Computing Research Repository*, arXiv:1503.06733, 2015. version 2.

[377] Antoine Raux, Brian Langner, Dan Bohus, Alan W. Black, and Maxine Eskénazi. Let's go public! taking a spoken dialog system to the real world. In *INTERSPEECH*, 2005.

[378] Thomas Rebele, Fabian Suchanek, Johannes Hoffart, Joanna Biega, Erdal Kuzey, and Gerhard Weikum. YAGO: A multilingual knowledge base from Wikipedia, WordNet, and GeoNames. In *Proceedings of International Semantic Web Conference*, 2016.

[379] Siva Reddy, Mirella Lapata, and Mark Steedman. Large-scale semantic parsing without question-answer pairs. *Transactions of the Association for Computational Linguistics*, 2:377–392, 2014.

[380] Machel Reid and Mikel Artetxe. Paradise: Exploiting parallel data for multilingual sequence-to-sequence pretraining. *arXiv preprint arXiv:2108.01887*, 2021.

[381] Nils Reimers and Iryna Gurevych. Reporting score distributions makes a difference: Performance study of lstm-networks for sequence tagging. *arXiv:1707.09861*, 2017.

[382] Nils Reimers and Iryna Gurevych. Sentence-bert: Sentence embeddings using siamese bert-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, 11 2019.

[383] Nils Reimers and Iryna Gurevych. Sentence-bert: Sentence embeddings using siamese bert-networks. *arXiv preprint arXiv:1908.10084*, 2019.

[384] Ruiyang Ren, Yingqi Qu, Jing Liu, Wayne Xin Zhao, Qiaoqiao She, Hua Wu, Haifeng Wang, and Ji-Rong Wen. Rocketqav2: A joint training method for dense passage retrieval and passage re-ranking. *arXiv preprint arXiv:2110.07367*, 2021.

[385] Leonardo FR Ribeiro, Martin Schmitt, Hinrich Schütze, and Iryna Gurevych. Investigating pretrained language models for graph-to-text generation. *arXiv preprint arXiv:2007.08426*, 2020.

[386] Stephen Robertson and Hugo Zaragoza. *The probabilistic relevance framework: BM25 and beyond*. Now Publishers Inc, 2009.

[387] Stephen E Robertson, Steve Walker, Susan Jones, Micheline M Hancock-Beaulieu, Mike Gatford, et al. Okapi at trec-3. *Nist Special Publication Sp*, 109:109, 1995.

[388] Corby Rosset, Chenyan Xiong, Minh Phan, Xia Song, Paul Bennett, and Saurabh Tiwary. Knowledge-aware language model pretraining. *arXiv preprint arXiv:2007.00655*, 2020.

[389] Uma Roy, Noah Constant, Rami Al-Rfou, Aditya Barua, Aaron Phillips, and Yinfei Yang. Lareqa: Language-agnostic answer retrieval from a multilingual pool. *arXiv preprint arXiv:2004.05484*, 2020.

[390] Ohad Rubin and Jonathan Berant. Smbop: Semi-autoregressive bottom-up semantic parsing. *arXiv preprint arXiv:2010.12412*, 2020.

[391] Ohad Rubin, Jonathan Herzig, and Jonathan Berant. Learning to retrieve prompts for in-context learning. *arXiv preprint arXiv:2112.08633*, 2021.

[392] Andreas Rücklé, Jonas Pfeiffer, and Iryna Gurevych. MultiCQA: Zero-shot transfer of self-supervised text matching models on a massive scale. *arXiv:2010.00980*, 2020.

[393] Cicero dos Santos, Ming Tan, Bing Xiang, and Bowen Zhou. Attentive pooling networks. *arXiv:1602.03609*, 2016.

[394] François Scharffe, Yanbin Liu, and Chuguang Zhou. RDF-AI: an architecture for RDF datasets matching, fusion and interlink. In *Proceedings of IJCAI workshop on Identity and Reference in Web-based Knowledge Representation*, 2009.

[395] Michael Schlichtkrull, Thomas N. Kipf, Peter Bloem, Rianne Van Den Berg, Ivan Titov, and Max Welling. Modeling relational data with graph convolutional networks. In *Proceedings of European Semantic Web Conference*, 2018.

[396] Torsten Scholak, Raymond Li, Dzmitry Bahdanau, Harm de Vries, and Chris Pal. Duorat: Towards simpler text-to-sql models. *arXiv preprint arXiv:2010.11119*, 2020.

[397] Torsten Scholak, Nathan Schucher, and Dzmitry Bahdanau. Picard: Parsing incrementally for constrained auto-regressive decoding from language models. *arXiv preprint arXiv:2109.05093*, 2021.

[398] Torsten Scholak, Nathan Schucher, and Dzmitry Bahdanau. PICARD: Parsing incrementally for constrained auto-regressive decoding from language models. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 9895–9901. Association for Computational Linguistics, November 2021.

[399] Abigail See, Peter J. Liu, and Christopher D. Manning. Get to the point: Summarization with pointer-generator networks. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics, ACL 2017, Vancouver, Canada, July 30 - August 4, Volume 1: Long Papers*, pages 1073–1083, 2017.

[400] Stephanie Seneff and Joseph Polifroni. Dialogue management in the mercury flight reservation system. In *Proceedings of the 2000 ANLP/NAACL Workshop on Conversational Systems - Volume 3*, ANLP/NAACL-ConvSyst '00, pages 11–16, 2000.

[401] Minjoon Seo, Aniruddha Kembhavi, Ali Farhadi, and Hannaneh Hajishirzi. Bidirectional attention flow for machine comprehension. *arXiv:1611.01603*, 2016.

[402] Aliaksei Severyn and Alessandro Moschitti. Learning to rank short text pairs with convolutional deep neural networks. In *SIGIR*, pages 373–382, 2015.

[403] Siamak Shakeri, Noah Constant, Mihir Kale, and Linting Xue. Towards zero-shot multilingual synthetic question and answer generation for cross-lingual reading comprehension. In *Proceedings of the 14th International Conference on Natural Language Generation*, pages 35–45, 2021.

[404] Siamak Shakeri, Cicero Nogueira dos Santos, Henghui Zhu, Patrick Ng, Feng Nan, Zhiguo Wang, Ramesh Nallapati, and Bing Xiang. End-to-end synthetic data generation for domain adaptation of question answering systems. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 5445–5460, Online, November 2020. Association for Computational Linguistics.

[405] Bo Shao, Yeyun Gong, Weizhen Qi, Nan Duan, and Xiaola Lin. Multi-level alignment pretraining for multi-lingual semantic parsing. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 3246–3256, 2020.

[406] Yelong Shen, Xiaodong He, Jianfeng Gao, Li Deng, and Grégoire Mesnil. Learning semantic representations using convolutional neural networks for web search. In *WWW*, pages 373–374, 2014.

[407] Tom Sherborne and Mirella Lapata. Zero-shot cross-lingual semantic parsing. *arXiv preprint arXiv:2104.07554*, 2021.

[408] Tom Sherborne and Mirella Lapata. Meta-learning a cross-lingual manifold for semantic parsing. *arXiv preprint arXiv:2209.12577*, 2022.

[409] Tom Sherborne, Yumo Xu, and Mirella Lapata. Bootstrapping a crosslingual semantic parser. *arXiv preprint arXiv:2004.02585*, 2020.

[410] Freda Shi, Mirac Suzgun, Markus Freitag, Xuezhi Wang, Suraj Srivats, Soroush Vosoughi, Hyung Won Chung, Yi Tay, Sebastian Ruder, Denny Zhou, et al. Language models are multilingual chain-of-thought reasoners. *arXiv preprint arXiv:2210.03057*, 2022.

[411] Peng Shi, He Bai, and Jimmy Lin. Cross-lingual training of neural models for document ranking. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 2768–2773, 2020.

[412] Peng Shi, He Bai, and Jimmy Lin. Cross-lingual training of neural models for document ranking. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: Findings*, pages 2768–2773, 2020.

[413] Peng Shi and Jimmy Lin. Cross-lingual relevance transfer for document retrieval. *arXiv:1911.02989*, 2019.

[414] Peng Shi and Jimmy Lin. Cross-lingual relevance transfer for document retrieval. *arXiv preprint arXiv:1911.02989*, 2019.

[415] Peng Shi, Patrick Ng, Zhiguo Wang, Henghui Zhu, Alexander Hanbo Li, Jun Wang, Cicero Nogueira dos Santos, and Bing Xiang. Learning contextual representations for semantic parsing with generation-augmented pre-training. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 13806–13814, 2021.

[416] Peng Shi, Patrick Ng, Zhiguo Wang, Henghui Zhu, Alexander Hanbo Li, Jun Wang, Cicero Nogueira dos Santos, and Bing Xiang. Learning contextual representations for semantic parsing with generation-augmented pre-training. *arXiv preprint arXiv:2012.10309*, 2020.

[417] Peng Shi, Jinfeng Rao, and Jimmy Lin. Simple attention-based representation learning for ranking short social media posts. In *Proceedings of NAACL*, 2019.

[418] Peng Shi, Tao Yu, Patrick Ng, and Zhiguo Wang. End-to-end cross-domain text-to-sql semantic parsing with auxiliary task. *arXiv preprint arXiv:2106.09588*, 2021.

[419] Peng Shi, Rui Zhang, He Bai, and Jimmy Lin. Cross-lingual training with dense retrieval for document retrieval. *arXiv preprint arXiv:2109.01628*, 2021.

[420] Tianze Shi, Kedar Tatwawadi, Kaushik Chakrabarti, Yi Mao, Oleksandr Polozov, and Weizhu Chen. Incsql: Training incremental text-to-sql parsers with nondeterministic oracles. *arXiv preprint arXiv:1809.05054*, 2018.

[421] Richard Shin and Benjamin Van Durme. Few-shot semantic parsing with language models trained on code. *arXiv preprint arXiv:2112.08696*, 2021.

[422] Chang Shu, Yusen Zhang, Xiangyu Dong, Peng Shi, Tao Yu, and Rui Zhang. Logic-consistency text generation from semantic parses. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 4414–4426, Online, August 2021. Association for Computational Linguistics.

[423] Devendra Singh, Siva Reddy, Will Hamilton, Chris Dyer, and Dani Yogatama. End-to-end training of multi-document reader and retriever for open-domain question answering. *Advances in Neural Information Processing Systems*, 34:25968–25981, 2021.

[424] Samuel L Smith, David HP Turban, Steven Hamblin, and Nils Y Hammerla. Offline bilingual word vectors, orthogonal transformations and the inverted softmax. *arXiv preprint arXiv:1702.03859*, 2017.

[425] Mark D. Smucker, James Allan, and Ben Carterette. A comparison of statistical significance tests for information retrieval evaluation. In *CIKM*, pages 623–632, 2007.

[426] Ian Soboroff, Iadh Ounis, Craig Macdonald, and Jimmy Lin. Overview of the TREC-2012 Microblog Track. In *TREC*, 2012.

[427] Linfeng Song, Yue Zhang, Zhiguo Wang, and Daniel Gildea. A graph-to-sequence model for amr-to-text generation. In *Proceedings of ACL*, 2018.

[428] Dennis Spohr, Laura Hollink, and Philipp Cimiano. A machine learning approach to multilingual and cross-lingual ontology matching. In *Proceedings of International Semantic Web Conference*, 2011.

[429] Alvin Cheung Jayant Krishnamurthy Srinivasan Iyer, Ioannis Konstas and Luke Zettlemoyer. Learning a neural semantic parser from user feedback. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 963–973, 2017.

[430] Rupesh K. Srivastava, Klaus Greff, and Jürgen Schmidhuber. Training very deep networks. In *Proceedings of NIPS*, 2015.

[431] Fabian M. Suchanek, Gjergji Kasneci, and Gerhard Weikum. YAGO: A large ontology from Wikipedia and WordNet. *Web Semantics: Science, Services and Agents on the World Wide Web*, 6(3):203–217, 2008.

[432] Alane Suhr, Ming-Wei Chang, Peter Shaw, and Kenton Lee. Exploring unexplored generalization challenges for cross-database semantic parsing. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8372–8388, Online, July 2020. Association for Computational Linguistics.

[433] Alane Suhr, Srinivasan Iyer, and Yoav Artzi. Learning to map context-dependent sentences to executable formal queries. In *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 2238–2249. Association for Computational Linguistics, 2018.

[434] Shuo Sun and Kevin Duh. Clirmatrix: A massively large collection of bilingual and multilingual datasets for cross-lingual information retrieval. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 4160–4170, 2020.

[435] Yibo Sun, Duyu Tang, Nan Duan, Jianshu Ji, Guihong Cao, Xiaocheng Feng, Bing Qin, Ting Liu, and Ming Zhou. Semantic parsing with syntax-and table-aware sql generation. In *ACL 2018*, 2018.

[436] Yibo Sun, Zhao Yan, Duyu Tang, Nan Duan, and Bing Qin. Content-based table retrieval for web queries. *Neurocomputing*, 349:183–189, 2019.

[437] Zequn Sun, Wei Hu, and Chengkai Li. Cross-lingual entity alignment via joint attribute-preserving embedding. In *Proceedings of International Semantic Web Conference*, 2017.

[438] Zequn Sun, Wei Hu, Qingheng Zhang, and Yuzhong Qu. Bootstrapping entity alignment with knowledge graph embedding. In *IJCAI*, volume 18, pages 4396–4402, 2018.

[439] Zequn Sun, Chengming Wang, Wei Hu, Muhao Chen, Jian Dai, Wei Zhang, and Yuzhong Qu. Knowledge graph alignment network with gated multi-hop neighborhood aggregation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 222–229, 2020.

[440] Raymond Hendy Susanto and Wei Lu. Neural architectures for multilingual semantic parsing. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 38–44, 2017.

[441] Raymond Hendy Susanto and Wei Lu. Semantic parsing with neural hybrid trees. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 31, 2017.

[442] Ilya Sutskever, Oriol Vinyals, and Quoc V Le. Sequence to sequence learning with neural networks. In *Advances in neural information processing systems*, pages 3104–3112, 2014.

[443] Hongyin Tang, Xingwu Sun, Beihong Jin, Jingang Wang, Fuzheng Zhang, and Wei Wu. Improving document representations by generating pseudo query embeddings for dense retrieval. *arXiv preprint arXiv:2105.03599*, 2021.

[444] Lappoon R. Tang and Raymond J. Mooney. Automated construction of database interfaces: Intergrating statistical and relational learning for semantic parsing. In *2000 Joint SIGDAT Conference on Empirical Methods in Natural Language Processing and Very Large Corpora*, pages 133–141, 2000.

[445] Lappoon R. Tang and Raymond J. Mooney. Using multiple clause constructors in inductive logic programming for semantic parsing. In *Proceedings of the 12th European Conference on Machine Learning*, pages 466–477, Freiburg, Germany, 2001.

[446] Lappoon R Tang and Raymond J Mooney. Using multiple clause constructors in inductive logic programming for semantic parsing. In *ECML*, volume 1, pages 466–477. Springer, 2001.

[447] Xiaobin Tang, Jing Zhang, Bo Chen, Yang Yang, Hong Chen, and Cuiping Li. Bert-int: a bert-based interaction model for knowledge graph alignment. *interactions*, 100:e1, 2020.

[448] James Thorne, Andreas Vlachos, Christos Christodoulopoulos, and Arpit Mittal. Fever: a large-scale dataset for fact extraction and verification. *arXiv preprint arXiv:1803.05355*, 2018.

[449] Jörg Tiedemann and Santhosh Thottingal. OPUS-MT – building open translation services for the world. In *Proceedings of the 22nd Annual Conference of the European Association for Machine Translation*, pages 479–480, Lisboa, Portugal, November 2020. European Association for Machine Translation.

[450] Kristina Toutanova, Victoria Lin, Wen-tau Yih, Hoifung Poon, and Chris Quirk. Compositional learning of embeddings for relation paths in knowledge base and text. In *Proceedings of ACL*, 2016.

[451] Théo Trouillon, Johannes Welbl, Sebastian Riedel, Éric Gaussier, and Guillaume Bouchard. Complex embeddings for simple link prediction. In *International Conference on Machine Learning*, 2016.

[452] Bayu Distiawan Trsedya, Jianzhong Qi, and Rui Zhang. Entity alignment between knowledge graphs using attribute embeddings. In *Proceedings of AAAI*, 2019.

[453] Shyam Upadhyay, Manaal Faruqui, Gokhan Tür, Hakkani-Tür Dilek, and Larry Heck. (almost) zero-shot cross-lingual spoken language understanding. In *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 6034–6038. IEEE, 2018.

[454] Jos van der Westhuizen and Joan Lasenby. The unreasonable effectiveness of the forget gate. *arXiv:1702.03814*, 2018.

[455] Vikas Verma, Alex Lamb, Christopher Beckham, Amir Najafi, Ioannis Mitliagkas, David Lopez-Paz, and Yoshua Bengio. Manifold mixup: Better representations by interpolating hidden states. In *International Conference on Machine Learning*, pages 6438–6447. PMLR, 2019.

[456] Caiming Xiong Victor Zhong and Richard Socher. Seq2sql: Generating structured queries from natural language using reinforcement learning. *CoRR*, abs/1709.00103, 2017.

[457] Oriol Vinyals, Meire Fortunato, and Navdeep Jaitly. Pointer networks. In C. Cortes, N. D. Lawrence, D. D. Lee, M. Sugiyama, and R. Garnett, editors, *Advances in Neural Information Processing Systems 28*, pages 2692–2700. Curran Associates, Inc., 2015.

[458] Julius Volz, Christian Bizer, Martin Gaedke, and Georgi Kobilarov. Discovering and maintaining links on the web of data. In *Proceedings of International Semantic Web Conference*, 2009.

[459] Marilyn A. Walker, Alexander I. Rudnicky, Rashmi Prasad, John S. Aberdeen, Elizabeth Owen Bratt, John S. Garofolo, Helen F. Hastie, Audrey N. Le, Bryan L. Pellom, Alexandros Potamianos, Rebecca J. Passonneau, Salim Roukos, Gregory A. Sanders, Stephanie Seneff, and David Stallard. Darpa communicator: cross-system results for the 2001 evaluation. In *INTERSPEECH*, 2002.

[460] Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R Bowman. Glue: A multi-task benchmark and analysis platform for natural language understanding. *arXiv preprint arXiv:1804.07461*, 2018.

[461] Bailin Wang, Mirella Lapata, and Ivan Titov. Meta-learning for domain generalization in semantic parsing. *arXiv preprint arXiv:2010.11988*, 2020.

[462] Bailin Wang, Richard Shin, Xiaodong Liu, Oleksandr Polozov, and Matthew Richardson. Rat-sql: Relation-aware schema encoding and linking for text-to-sql parsers. *arXiv preprint arXiv:1911.04942*, 2019.

[463] Bailin Wang, Richard Shin, Xiaodong Liu, Oleksandr Polozov, and Matthew Richardson. RAT-SQL: Relation-aware schema encoding and linking for text-to-SQL parsers. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7567–7578, Online, July 2020.

[464] Bailin Wang, Ivan Titov, and Mirella Lapata. Learning semantic parsers from denotations with latent structured alignments and abstract programs. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3774–3785, Hong Kong, China, November 2019.

[465] Chenglong Wang, Marc Brockschmidt, and Rishabh Singh. Pointing out sql queries from text. *Technical Report*, 2017.

[466] Chenglong Wang, Alvin Cheung, and Rastislav Bodik. Synthesizing highly expressive sql queries from input-output examples. In *Proceedings of the 38th ACM SIGPLAN Conference on Programming Language Design and Implementation*, pages 452–466. ACM, 2017.

[467] Chenglong Wang, Po-Sen Huang, Alex Polozov, Marc Brockschmidt, and Rishabh Singh. Execution-guided neural program decoding. In *ICML workshop on Neural Abstract Machines and Program Induction v2 (NAMPI)*, 2018.

[468] Lijie Wang, Ao Zhang, Kun Wu, Ke Sun, Zhenghua Li, Hua Wu, Min Zhang, and Haifeng Wang. Chitesql: A large-scale and pragmatic chinese text-to-sql dataset. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6923–6935, 2020.

[469] Quan Wang, Zhendong Mao, Bin Wang, and Li Guo. Knowledge graph embedding: A survey of approaches and applications. *IEEE Transactions on Knowledge and Data Engineering*, 29(12):2724–2743, 2017.

[470] Ruize Wang, Duyu Tang, Nan Duan, Zhongyu Wei, Xuanjing Huang, Cuihong Cao, Daxin Jiang, Ming Zhou, et al. K-adapter: Infusing knowledge into pre-trained models with adapters. *arXiv preprint arXiv:2002.01808*, 2020.

[471] Xinyi Wang, Yulia Tsvetkov, and Graham Neubig. Balancing training for multilingual neural machine translation. *arXiv preprint arXiv:2004.06748*, 2020.

[472] Yushi Wang, Jonathan Berant, Percy Liang, et al. Building a semantic parser overnight. In *ACL (1)*, pages 1332–1342, 2015.

[473] Zhen Wang, Jianwen Zhang, Jianlin Feng, and Zheng Chen. Knowledge graph embedding by translating on hyperplanes. In *Proceedings of AAAI*, 2014.

[474] Zhichun Wang, Juanzi Li, Zhigang Wang, and Jie Tang. Cross-lingual knowledge linking across wiki knowledge bases. In *Proceedings of WWW*, 2012.

[475] Zhichun Wang, Qingsong Lv, Xiaohan Lan, and Yu Zhang. Cross-lingual knowledge graph alignment via graph convolutional networks. In *Proceedings of EMNLP*, 2018.

[476] Zhiguo Wang, Wael Hamza, and Radu Florian. Bilateral multi-perspective matching for natural language sentences. *arXiv:1702.03814*, 2017.

[477] David HD Warren and Fernando CN Pereira. An efficient easily adaptable system for interpreting natural language queries. *Computational Linguistics*, 8(3-4):110–122, 1982.

[478] Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Ed Chi, Quoc Le, and Denny Zhou. Chain of thought prompting elicits reasoning in large language models. *arXiv preprint arXiv:2201.11903*, 2022.

[479] Dirk Weissenborn, Georg Wiese, and Laura Seiffe. Making neural QA as simple as possible but not simpler. In *CoNLL*, pages 271–280, 2017.

[480] Tsung-Hsien Wen, Milica Gasic, Nikola Mrksic, Lina Maria Rojas-Barahona, Pei-Hao Su, Stefan Ultes, David Vandyke, and Steve J. Young. A network-based end-to-end trainable task-oriented dialogue system. *CoRR*, 2016.

[481] John Wieting and Kevin Gimpel. Pushing the limits of paraphrastic sentence embeddings with millions of machine translations. *arXiv preprint arXiv:1711.05732*, 2017.

[482] Genta Indra Winata, Andrea Madotto, Zhaojiang Lin, Rosanne Liu, Jason Yosinski, and Pascale Fung. Language models are few-shot multilingual learners. *arXiv preprint arXiv:2109.07684*, 2021.

[483] Sam Wiseman, Stuart Shieber, and Alexander Rush. Challenges in data-to-document generation. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2253–2263. Association for Computational Linguistics, 2017.

[484] Sam Wiseman, Stuart Shieber, and Alexander Rush. Learning neural templates for text generation. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, Brussels, Belgium, 2018. Association for Computational Linguistics.

[485] Yuk Wah Wong and Raymond J. Mooney. Learning synchronous grammars for semantic parsing with lambda calculus. In *Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics (ACL-2007)*, Prague, Czech Republic, June 2007.

[486] Chien-Sheng Wu, Andrea Madotto, Ehsan Hosseini-Asl, Caiming Xiong, Richard Socher, and Pascale Fung. Transferable multi-domain state generator for task-oriented dialogue systems. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Association for Computational Linguistics, 2019.

[487] Ho Chung Wu, Robert WP Luk, Kam-Fai Wong, and KL Kwok. A retrospective study of a hybrid document-context based retrieval model. *Information processing & management*, 43(5):1308–1331, 2007.

[488] Qianhui Wu, Zijia Lin, Börje F Karlsson, Jian-Guang Lou, and Biqing Huang. Single-/multi-source cross-lingual ner via teacher-student learning on unlabeled data in target language. *arXiv preprint arXiv:2004.12440*, 2020.

[489] Shijie Wu and Mark Dredze. Beto, bentz, becas: The surprising cross-lingual effectiveness of BERT. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 833–844, Hong Kong, China, 2019.

[490] Shijie Wu and Mark Dredze. Beto, bentz, becas: The surprising cross-lingual effectiveness of bert. *arXiv preprint arXiv:1904.09077*, 2019.

[491] Yuting Wu, Xiao Liu, Yansong Feng, Zheng Wang, Rui Yan, and Dongyan Zhao. Relation-aware entity alignment for heterogeneous knowledge graphs. *arXiv preprint arXiv:1908.08210*, 2019.

[492] Yuting Wu, Xiao Liu, Yansong Feng, Zheng Wang, and Dongyan Zhao. Jointly learning entity and relation representations for entity alignment. *arXiv preprint arXiv:1909.09317*, 2019.

[493] Mengzhou Xia, Xiang Kong, Antonios Anastasopoulos, and Graham Neubig. Generalized data augmentation for low-resource translation. *arXiv preprint arXiv:1906.03785*, 2019.

[494] Mengzhou Xia, Guoqing Zheng, Subhabrata Mukherjee, Milad Shokouhi, Graham Neubig, and Ahmed Hassan Awadallah. Metaxl: Meta representation transformation for low-resource cross-lingual learning. *arXiv preprint arXiv:2104.07908*, 2021.

[495] Qizhe Xie, Zihang Dai, Yulun Du, Eduard Hovy, and Graham Neubig. Controllable invariance through adversarial feature learning. *arXiv preprint arXiv:1705.11122*, 2017.

[496] Ruobing Xie, Zhiyuan Liu, Jia Jia, Huanbo Luan, and Maosong Sun. Representation learning of knowledge graphs with entity descriptions. In *Proceedings of AAAI*, 2016.

[497] Tianbao Xie, Chen Henry Wu, Peng Shi, Ruiqi Zhong, Torsten Scholak, Michihiro Yasunaga, Chien-Sheng Wu, Ming Zhong, Pengcheng Yin, Sida I Wang, et al. Unifiedskg: Unifying and multi-tasking structured knowledge grounding with text-to-text language models. *arXiv preprint arXiv:2201.05966*, 2022.

[498] Chenyan Xiong, Zhuyun Dai, Jamie Callan, Zhiyuan Liu, and Russell Power. End-to-end neural ad-hoc ranking with kernel pooling. In *SIGIR*, pages 55–64, 2017.

[499] Wenhan Xiong, Jingfei Du, William Yang Wang, and Veselin Stoyanov. Pretrained encyclopedia: Weakly supervised knowledge-pretrained language model. *arXiv preprint arXiv:1912.09637*, 2019.

[500] Kun Xu, Liwei Wang, Mo Yu, Yansong Feng, Yan Song, Zhiguo Wang, and Dong Yu. Cross-lingual knowledge graph alignment via graph matching neural network. *arXiv preprint arXiv:1905.11605*, 2019.

[501] Kun Xu, Lingfei Wu, Zhiguo Wang, Yansong Feng, and Vadim Sheinin. Sql-to-text generation with graph-to-sequence model. *arXiv preprint arXiv:1809.05255*, 2018.

[502] Kun Xu, Lingfei Wu, Zhiguo Wang, Yansong Feng, and Vadim Sheinin. SQL-to-text generation with graph-to-sequence model. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, Brussels, Belgium, 2018. Association for Computational Linguistics.

[503] Kun Xu, Lingfei Wu, Zhiguo Wang, Mo Yu, Liwei Chen, and Vadim Sheinin. Sql-to-text generation with graph-to-sequence model. *EMNLP*, 2018.

[504] Liyan Xu, Xuchao Zhang, Xujiang Zhao, Haifeng Chen, Feng Chen, and Jinho D Choi. Boosting cross-lingual transfer via self-learning with uncertainty estimation. *arXiv preprint arXiv:2109.00194*, 2021.

[505] Ruochen Xu and Yiming Yang. Cross-lingual distillation for text classification. *arXiv preprint arXiv:1705.02073*, 2017.

[506] Weijia Xu, Batool Haider, and Saab Mansour. End-to-end slot alignment and recognition for cross-lingual nlu. *arXiv preprint arXiv:2004.14353*, 2020.

[507] Xiaojun Xu, Chang Liu, and Dawn Song. Sqlnet: Generating structured queries from natural language without reinforcement learning. *arXiv preprint arXiv:1711.04436*, 2017.

[508] Xiaojun Xu, Chang Liu, and Dawn Song. Sqlnet: Generating structured queries from natural language without reinforcement learning. *arXiv preprint arXiv:1711.04436*, 2017.

[509] Linting Xue, Noah Constant, Adam Roberts, Mihir Kale, Rami Al-Rfou, Aditya Siddhant, Aditya Barua, and Colin Raffel. mt5: A massively multilingual pre-trained text-to-text transformer. *arXiv preprint arXiv:2010.11934*, 2020.

[510] Avinash Yadav and Sukomal Pal Robins Yadav. Ism@ fire-2012 adhoc retrieval task and morpheme extraction task.

[511] Navid Yaghmazadeh, Yuepeng Wang, Isil Dillig, and Thomas Dillig. Sqlizer: Query synthesis from natural language. *Proc. ACM Program. Lang.*, 1(OOPSLA):63:1–63:26, October 2017.

[512] Navid Yaghmazadeh, Yuepeng Wang, Isil Dillig, and Thomas Dillig. Sqlizer: query synthesis from natural language. *Proceedings of the ACM on Programming Languages*, 2017.

[513] Bishan Yang, Wen-tau Yih, Xiaodong He, Jianfeng Gao, and Li Deng. Embedding entities and relations for learning and inference in knowledge bases. In *Proceedings of ICLR*, 2015.

[514] Huiyun Yang, Huadong Chen, Hao Zhou, and Lei Li. Enhancing cross-lingual transfer by manifold mixup. In *International Conference on Learning Representations*, 2021.

[515] Huiyun Yang, Huadong Chen, Hao Zhou, and Lei Li. Enhancing cross-lingual transfer by manifold mixup. *arXiv preprint arXiv:2205.04182*, 2022.

[516] Peilin Yang, Hui Fang, and Jimmy Lin. Anserini: Enabling the use of lucene for information retrieval research. In *Proceedings of the 40th International ACM SIGIR*

Conference on Research and Development in Information Retrieval, pages 1253–1256, 2017.

[517] Peilin Yang, Hui Fang, and Jimmy Lin. Anserini: reproducible ranking baselines using Lucene. *Journal of Data and Information Quality*, 10(4):Article 16, 2018.

[518] Wei Yang, Haotian Zhang, and Jimmy Lin. Simple applications of bert for ad hoc document retrieval. *arXiv preprint arXiv:1903.10972*, 2019.

[519] Yinfei Yang, Daniel Cer, Amin Ahmad, Mandy Guo, Jax Law, Noah Constant, Gustavo Hernandez Abrego, Steve Yuan, Chris Tar, Yun-Hsuan Sung, et al. Multilingual universal sentence encoder for semantic retrieval. *arXiv preprint arXiv:1907.04307*, 2019.

[520] Zeynep Akkalyoncu Yilmaz, Shengjin Wang, Wei Yang, Haotian Zhang, and Jimmy Lin. Applying bert to document retrieval with birch. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP): System Demonstrations*, pages 19–24, 2019.

[521] Zeynep Akkalyoncu Yilmaz, Wei Yang, Haotian Zhang, and Jimmy Lin. Cross-domain modeling of sentence-level evidence for document retrieval. In *Proceedings of the 2019 conference on empirical methods in natural language processing and the 9th international joint conference on natural language processing (EMNLP-IJCNLP)*, pages 3490–3496, 2019.

[522] Pengcheng Yin, Bowen Deng, Edgar Chen, Bogdan Vasilescu, and Graham Neubig. Learning to mine aligned code and natural language pairs from stack overflow. In *International Conference on Mining Software Repositories (MSR)*, 2018.

[523] Pengcheng Yin, Bowen Deng, Edgar Chen, Bogdan Vasilescu, and Graham Neubig. Learning to mine aligned code and natural language pairs from stack overflow. In *2018 IEEE/ACM 15th International Conference on Mining Software Repositories (MSR)*, pages 476–486. IEEE, 2018.

[524] Pengcheng Yin, Zhengdong Lu, Hang Li, and Ben Kao. Neural enquirer: Learning to query tables in natural language. In *Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence, IJCAI 2016, New York, NY, USA, 9-15 July 2016*, pages 2308–2314, 2016.

[525] Pengcheng Yin and Graham Neubig. A syntactic neural model for general-purpose code generation, 2017.

[526] Pengcheng Yin and Graham Neubig. A syntactic neural model for general-purpose code generation. In *ACL (1)*, pages 440–450. Association for Computational Linguistics, 2017.

[527] Pengcheng Yin and Graham Neubig. Reranking for neural semantic parsing. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4553–4559, 2019.

[528] Pengcheng Yin, Graham Neubig, Wen-tau Yih, and Sebastian Riedel. Tabert: Pretraining for joint understanding of textual and tabular data. *arXiv preprint arXiv:2005.08314*, 2020.

[529] Pengcheng Yin, Chunting Zhou, Junxian He, and Graham Neubig. Structvae: Tree-structured latent variable models for semi-supervised semantic parsing. *arXiv preprint arXiv:1806.07832*, 2018.

[530] Wenpeng Yin and Hinrich Schütze. Attentive convolution. *arXiv:1710.00519*, 2017.

[531] Wenpeng Yin, Mo Yu, Bing Xiang, Bowen Zhou, and Hinrich Schütze. Simple question answering by attentive convolutional neural network. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pages 1746–1756, 2016.

[532] Jianxing Yu, Xiaojun Quan, Qinliang Su, and Jian Yin. Generating multi-hop reasoning questions to improve machine reading comprehension. In *Proceedings of The Web Conference 2020*, pages 281–291, 2020.

[533] Puxuan Yu and James Allan. A study of neural matching models for cross-lingual ir. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 1637–1640, 2020.

[534] Puxuan Yu, Hongliang Fei, and Ping Li. Cross-lingual language model pretraining for retrieval. In *Proceedings of the Web Conference 2021*, pages 1029–1039, 2021.

[535] Tao Yu, Zifan Li, Zilin Zhang, Rui Zhang, and Dragomir Radev. Typesql: Knowledge-based type-aware neural text-to-sql generation. In *Proceedings of NAACL*. Association for Computational Linguistics, 2018.

[536] Tao Yu, Zifan Li, Zilin Zhang, Rui Zhang, and Dragomir Radev. Typesql: Knowledge-based type-aware neural text-to-sql generation. *arXiv preprint arXiv:1804.09769*, 2018.

[537] Tao Yu, Chien-Sheng Wu, Xi Victoria Lin, Bailin Wang, Yi Chern Tan, Xinyi Yang, Dragomir Radev, Richard Socher, and Caiming Xiong. Grappa: Grammar-augmented pre-training for table semantic parsing. *arXiv preprint arXiv:2009.13845*, 2020.

[538] 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 *Proceedings of EMNLP*. Association for Computational Linguistics, 2018.

[539] Tao Yu, Rui Zhang, Alex Polozov, Christopher Meek, and Ahmed Hassan Awadallah. Score: Pre-training for context representation in conversational semantic parsing. In *International Conference on Learning Representations*, 2020.

[540] Tao Yu, Rui Zhang, Alex Polozov, Christopher Meek, and Ahmed Hassan Awadallah. Score: Pre-training for context representation in conversational semantic parsing. In *International Conference on Learning Representations*, 2021.

[541] Tao Yu, Rui Zhang, Kai Yang, Michihiro Yasunaga, Dongxu Wang, Zifan Li, James Ma, Irene Li, Qingning Yao, Shanelle Roman, et al. Spider: A large-scale human-labeled dataset for complex and cross-domain semantic parsing and text-to-sql task. *arXiv preprint arXiv:1809.08887*, 2018.

[542] 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 *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 3911–3921, Brussels, Belgium, 2018. Association for Computational Linguistics.

[543] 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 *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, 2018.

[544] 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.

[545] Tao Yu, Rui Zhang, Michihiro Yasunaga, Yi Chern Tan, Xi Victoria Lin, Suyi Li, Irene Li Heyang Er, Bo Pang, Tao Chen, Emily Ji, Shreya Dixit, David Proctor, Sungrok Shim, Vincent Zhang Jonathan Kraft, Caiming Xiong, Richard Socher, and Dragomir Radev. Sparc: Cross-domain semantic parsing in context. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, Florence, Italy, 2019. Association for Computational Linguistics.

[546] Xiaojing Yu, Tianlong Chen, Zhengjie Yu, Huiyu Li, Yang Yang, Xiaoqian Jiang, and Anxiao Jiang. Dataset and enhanced model for eligibility criteria-to-sql semantic parsing. In *Proceedings of The 12th Language Resources and Evaluation Conference*, pages 5829–5837, 2020.

[547] Xinyan Velocity Yu, Akari Asai, Trina Chatterjee, Junjie Hu, and Eunsol Choi. Beyond counting datasets: A survey of multilingual dataset construction and necessary resources. *Findings of EMNLP*, 2022.

[548] Weizhe Yuan, Graham Neubig, and Pengfei Liu. Bartscore: Evaluating generated text as text generation. *arXiv preprint arXiv:2106.11520*, 2021.

[549] Hamed Zamani, Johanne R Trippas, Jeff Dalton, and Filip Radlinski. Conversational information seeking. *arXiv preprint arXiv:2201.08808*, 2022.

[550] John M. Zelle and Raymond J. Mooney. Learning to parse database queries using inductive logic programming. In *Proceedings of the Thirteenth National Conference on Artificial Intelligence - Volume 2*, pages 1050–1055, 1996.

[551] John M Zelle and Raymond J Mooney. Learning to parse database queries using inductive logic programming. In *Proceedings of the national conference on artificial intelligence*, pages 1050–1055, 1996.

[552] John M. Zelle and Raymond J. Mooney. Learning to parse database queries using inductive logic programming. In *AAAI/IAAI*, pages 1050–1055, Portland, OR, August 1996. AAAI Press/MIT Press.

[553] Luke S. Zettlemoyer and Michael Collins. Learning to map sentences to logical form: Structured classification with probabilistic categorial grammars. *UAI*, 2005.

[554] Luke S. Zettlemoyer and Michael Collins. Learning to map sentences to logical form: Structured classification with probabilistic categorial grammars, 2012.

[555] Fuwei Zhang, Zhao Zhang, Xiang Ao, Dehong Gao, Fuzhen Zhuang, Yi Wei, and Qing He. Mind the gap: Cross-lingual information retrieval with hierarchical knowledge enhancement. *arXiv preprint arXiv:2112.13510*, 2021.

[556] Hang Zhang, Yeyun Gong, Yelong Shen, Jiancheng Lv, Nan Duan, and Weizhu Chen. Adversarial retriever-ranker for dense text retrieval. *arXiv preprint arXiv:2110.03611*, 2021.

[557] Hongyi Zhang, Moustapha Cisse, Yann N Dauphin, and David Lopez-Paz. mixup: Beyond empirical risk minimization. *arXiv preprint arXiv:1710.09412*, 2017.

[558] Kaizhong Zhang and Dennis Shasha. Simple fast algorithms for the editing distance between trees and related problems. *SIAM journal on computing*, 18(6):1245–1262, 1989.

[559] Rui Zhang, Caitlin Westerfield, Sungrok Shim, Garrett Bingham, Alexander Fabbri, Neha Verma, William Hu, and Dragomir Radev. Improving low-resource cross-lingual document retrieval by reranking with deep bilingual representations. *arXiv preprint arXiv:1906.03492*, 2019.

[560] 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 *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and 9th International Joint Conference on Natural Language Processing*, 2019.

[561] Sheng Zhang, Xutai Ma, Kevin Duh, and Benjamin Van Durme. Amr parsing as sequence-to-graph transduction. *arXiv preprint arXiv:1905.08704*, 2019.

[562] Shuo Zhang and Krisztian Balog. Ad hoc table retrieval using semantic similarity. In *Proceedings of the 2018 world wide web conference*, pages 1553–1562, 2018.

[563] Xinyu Zhang, Xueguang Ma, Peng Shi, and Jimmy Lin. Mr. tydi: A multi-lingual benchmark for dense retrieval. *arXiv preprint arXiv:2108.08787*, 2021.

[564] Xinyu Zhang, Kelechi Ogueji, Xueguang Ma, and Jimmy Lin. Towards best practices for training multilingual dense retrieval models. *arXiv preprint arXiv:2204.02363*, 2022.

[565] Yuhao Zhang, Peng Qi, and Christopher D. Manning. Graph convolution over pruned dependency trees improves relation extraction. In *Proceedings of EMNLP*, 2018.

[566] Guoqing Zheng and Jamie Callan. Learning to reweight terms with distributed representations. In *Proceedings of the 38th international ACM SIGIR conference on research and development in information retrieval*, pages 575–584, 2015.

[567] Huaping Zhong, Jianwen Zhang, Zhen Wang, Hai Wan, and Zheng Chen. Aligning knowledge and text embeddings by entity descriptions. In *Proceedings of EMNLP*, 2015.

[568] Ruiqi Zhong, Tao Yu, and Dan Klein. Semantic evaluation for text-to-sql with distilled test suites, 2020.

[569] Victor Zhong, Mike Lewis, Sida I Wang, and Luke Zettlemoyer. Grounded adaptation for zero-shot executable semantic parsing. *arXiv preprint arXiv:2009.07396*, 2020.

[570] Victor Zhong, Caiming Xiong, and Richard Socher. Seq2sql: Generating structured queries from natural language using reinforcement learning. *arXiv preprint arXiv:1709.00103*, 2017.

[571] Victor Zhong, Caiming Xiong, and Richard Socher. Seq2sql: Generating structured queries from natural language using reinforcement learning. *CoRR*, abs/1709.00103, 2017.

[572] Victor Zhong, Caiming Xiong, and Richard Socher. Seq2sql: Generating structured queries from natural language using reinforcement learning. *CoRR*, abs/1709.00103, 2017.

[573] Victor Zhong, Caiming Xiong, and Richard Socher. Global-locally self-attentive encoder for dialogue state tracking. In *ACL*, 2018.

[574] Chunting Zhou, Daniel Levy, Xian Li, Marjan Ghazvininejad, and Graham Neubig. Distributionally robust multilingual machine translation. *arXiv preprint arXiv:2109.04020*, 2021.

[575] Hao Zhu, Ruobing Xie, Zhiyuan Liu, and Maosong Sun. Iterative entity alignment via knowledge embeddings. In *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI)*, 2017.

[576] Yanyan Zou and Wei Lu. Learning cross-lingual distributed logical representations for semantic parsing. *arXiv preprint arXiv:1806.05461*, 2018.