

https://dspace.org/

Querying relational database systems in natural language using sequence to sequence learning with neural networks

Khalo, Nomonde

2021
https://hdl.handle.net/10539/33099

Downloaded from a DSpace repository. https://dspace.org/

Querying Relational Database Systems in Natural Language using Sequence to Sequence Learning with Neural Networks



UNIVERSITY OF THE
WITWATERSRAND,
JOHANNESBURG

Author: Nomonde Khalo

317845

Supervisor: Prof. Turgay Celik

A dissertation submitted to the Faculty of Science in fulfillment of the requirements for the degree

Master of Science (Computer Science)

October 2021

Declaration

I, Nomonde Khalo, declare that this research project is my own work.
All information sources used to conduct this research project have been referenced.
This work has never been submitted to any other university for any other degree.

Signed:



Date: 07/10/2021

Abstract

Retrieving information in the database requires a person to know a querying language, such as *Structured Query Language (SQL)*. This becomes challenging for individuals with little or no understanding of database querying language. Querying databases using natural language can thus help individuals retrieve information without knowing SQL or the underlying domain of the databases. This research's main objective is to comparatively investigate deep learning approaches, specifically sequence-to-sequence architectures for the task of translating natural language to SQL.

This is achieved by answering three research questions; *Which sequence-to-sequence architecture is more efficient in translating natural language to SQL?, What are the key factors of failures in the generation of semantically equivalent SQL queries?, How are failures mitigated in the generation semantically equivalent SQL queries?*. The first research question is answered by comparatively investigating the sequence-to-sequence models; LSTM, BiLSTM, Encoder-Decoder, Column Attention, and Pointer Network on the WikiSQL dataset that consists of question-query pairs and their three SQL components, AGGREGATION, SELECT, and WHERE-Clause. The investigation showed that no one model is fit for an end-to-end solution for all the SQL components. The AGGREGATION showed to perform well on the the BiLSTM, the Column Attention showed to specifically handle the column names prediction for SELECT and WHERE-Clause. The Pointer Network is noted to predict more robust WHERE-Clause. The second research question is answered through the SQL output, it is evident from this, that due to lack of information on the question some models were unable to predict the correct information. Furthermore, the error analysis highlighted various types of errors generated, this includes, errors in the ground truth, multiple valid SQL queries, and unexplainable errors. The third question is answered by the error analysis, to mitigate error analysis, the ground truth needs to be standardised for each question that might have multiple SQL query, errors in ground truth needs to first be fixed so that the evaluation can be of quality, training the models according to their question capacity, where they are most successfully and able to predict is also a strategy that can mitigate errors.

This work contributes to the body of work that investigate semantic parsing of natural language to SQL for querying databases, providing extensive study that shows that different SQL components achieves best results on different sequence-to-sequence models, comprehensive error analysis, and direction to mitigate the errors generated. Future work will focus on the generation of standardised ground truth for multiple valid SQL queries and correcting the ground truth.

Contents

Declaration i

Abstract ii

Contents iii

List of Figures vi

List of Tables viii

Dedication xiv

Acknowledgements xv

1 Introduction 1
- 1.1 Introduction . . . 1
- 1.2 Problem Definition . . . 3
- 1.3 Research Aim . . . 3
  - 1.3.1 Research Objective . . . 3
- 1.4 Research Questions . . . 4
- 1.5 Limitations . . . 5
- 1.6 Research Significance . . . 5
- 1.7 Thesis Structure . . . 5

2 Background and Literature Review 6
- 2.1 Introduction . . . 6
- 2.2 Natural Language Processing . . . 6
  - 2.2.1 Neural Natural language processing representation . . . 10
    - 2.2.1.1 Distributed word representation . . . 10
- 2.3 Semantic Parsing . . . 15
  - 2.3.1 Traditional Semantic Parsing . . . 15
    - 2.3.1.1 Traditional Statistical Approaches . . . 16
    - 2.3.1.2 Machine Translation . . . 17
- 2.4 Semantic Parsing of Natural language to SQL . . . 18
  - 2.4.1 Traditional Natural language to SQL . . . 18
  - 2.4.2 Natural language to SQL with deep neural networks . . . 20

2.4.2.1 Recurrent Neural Network . . . 21
2.4.2.2 Bidirectional Recurrent Neural Networks . . . 22
2.4.2.3 Long-Short Term Memory . . . 23
2.4.2.4 Gated Recurrent Unit . . . 26
2.4.2.5 Encoder-Decoder Architectures . . . 27
2.4.2.6 Attention Mechanism . . . 30
2.4.2.7 Pointer Network . . . 35
2.4.2.8 Training and Optimisation for Neural Networks . . . 37
2.4.2.9 Neural Network Hyper-parameters . . . 42
2.4.3 Regularisation . . . 43
2.4.3.1 Dropout . . . 43
2.4.3.2 Performance Measures . . . 44
2.5 Summary . . . 46

3 Methodology **47**
3.1 Introduction . . . 47
3.2 Research Approach . . . 47
3.2.1 Research pipeline . . . 48
3.2.2 The WikiSQL Dataset . . . 49
3.2.3 Pre-processing . . . 53
3.2.3.1 Words Representation . . . 54
3.3 Research methods . . . 55
3.4 Hardware and Software . . . 57
3.4.1 Hardware . . . 57
3.4.2 Software . . . 57
3.5 Experimental Setup . . . 58
3.6 Summary . . . 59

4 Results and Analysis **60**
4.1 Introduction . . . 60
4.2 Training Level Results . . . 61
4.2.1 Effect on batch size . . . 62
4.2.2 Effect on hidden nodes . . . 64
4.2.3 Effects of Word embeddings . . . 65
4.3 Testing level results . . . 66
4.3.1 Evaluation . . . 69
4.3.2 SQL Prediction output . . . 71
4.3.3 Exact Match . . . 71
4.3.4 Paraphrase . . . 71
4.3.5 Partial Clue . . . 72
4.3.6 External Knowledge . . . 73
4.3.7 Ambiguous . . . 75
4.4 Error Analysis . . . 77
4.4.1 Errors in the Aggregation . . . 78
4.4.2 Error in the SELECT and WHERE-Clause column names . . . 80
4.4.3 Error in the Operation . . . 82
4.4.4 Error in the WHERE-Clause condition . . . 83

4.5 Discussion . . . 85
4.5.1 Training level Results . . . 85
4.5.2 Testing level results . . . 87
4.5.3 SQL prediction output . . . 89
4.5.4 Error Analysis . . . 91
4.6 Summary . . . 92

5 Conclusion **95**
5.1 Overview . . . 95
5.2 Research Achievements . . . 95
5.2.1 Which sequence-to-sequence architecture is more efficient in translating natural language to SQL? . . . 96
5.2.2 What are the main factors that cause SQL queries to be incorrectly predicted? . . . 97
5.2.3 How can prediction failures when generating semantically equivalent SQL queries be mitigated? . . . 97
5.3 Future Work . . . 98

Bibliography **100**

List of Figures

2.1 The CBOW architecture where the current word is predicted based on the context (Rong 2014). . . . 10
2.2 An architecture of the FastText Model (Joulin, Grave, Bojanowski & Mikolov 2016). . . . 14
2.3 A building block of neural networks (Saxen 2017). . . . 21
2.4 A structure of Recurrent Neural Network (Krenker et al. 2011). . . . 22
2.5 An architecture of a Bidirectional Recurrent Neural Network (Graves et al. 2013). . . . 23
2.6 The Long-Short Term Memory architecture (Olah 2015). . . . 24
2.7 The taking as input the question and column names (left), three components and their respective methods, and the output SQL query (right) (Zhong et al. 2017). . . . 26
2.8 An architecture of the Gated Recurrent Unit (Drakos 2019). . . . 27
2.9 An architecture of an Encoder-Decoder model for text translation, taking (a, b, c) as input to the Encoder and the Decoder outputs (w, y, z) (Sutskever et al. 2014*a*). . . . 28
2.10 Question input and their logical forms, encoded and decoded using LSTM. An attention layer is used to learn soft alignments (Dong & Lapata 2016). 29
2.11 Given a source sentence $(x_1, x_2, ..., x_T)$ the $t-th$ target word y_t is generated (Bahdanau et al. 2014) . . . 32
2.12 The Global Attention (Luong, Pham & Manning 2015) . . . 33
2.13 The Local Attention (Luong, Pham & Manning 2015) . . . 33
2.14 When a training dataset perfectly fits the model during training. An example of overfitted dataset (Goodfellow et al. 2016). . . . 44
2.15 A deep neural network with dropped neurons represented by the Orange neuron (Srivastava et al. 2014). . . . 44

3.1 The research pipeline overview for the undertaken research, all the models follow same pipeline to achieve comparative results. The pipeline covers Chapter 3 through Chapter 4 . . . 49
3.2 The questions distribution of the WikiSQL dataset (Zhong et al. 2017). . 50
3.3 The distribution of question, query and table from the WikiSQL dataset (Zhong et al. 2017). . . . 50
3.4 Methodology pipeline from data pre-processing to evaluation . . . 56

4.1 Training Loss vs epochs on the LSTM model for 50 epochs . . . 61
4.2 BiLSTM training accuracy on batch sizes 32 and 64 . . . 62
4.3 The Encoder-Decoder model training performance on batch sizes of 32 and 64 respectively . . . 63

4.4 The Column Attention Training accuracy vs Epochs on batch sizes of 32 and 64 respectively 63
4.5 The Pointer Network training performance on batch sizes of 32 and 64 respectively 63
4.6 FastText vs GloVe word embeddings on the Pointer Network model 66

List of Tables

3.1 Examples of WikiSQL question and column name over the distribution of knowledge group (Yavuz et al. 2018) . . . 51
3.2 Question and Query . . . 52
3.3 The SQL query structure . . . 52
3.4 WikiSQL question and table header . . . 53
3.5 Data distribution for Training/Validation/Testing . . . 54
3.6 The created vocabulary using pre-trained GloVe embeddings and FastText pre-trained embeddings, and the unknown vocabulary. . . . 54
3.7 Experimental hardware specifications . . . 57
3.8 Hyper-parameters of experimental models . . . 58

4.1 Performance results on the test dataset, AGGREGATION, SELECT, WHERE-Clause, Logical, Execution accuracies, and time taken for each epoch for different number of hidden nodes . . . 64
4.2 Performance results on the test dataset, AGGREGATION, SELECT, WHERE-Clause, Logical, Execution accuracies, and time taken for each epoch for the FastText and GloVe word embedding on the Pointer Network model . . . 65
4.3 Final Hyper-parameters of experimental models . . . 67
4.4 Performance results on the test dataset, AGGREGATION, SELECT, WHERE-Clause, Logical, Execution accuracies, and time taken for each epoch for each model . . . 67
4.5 The evaluation metrics for all five models, F1-Score, Precision, Recall, and BLEU Score . . . 69
4.6 The predicted SQL statement with its corresponding ground truth SQL statement for exact match questions . . . 72
4.7 The predicted SQL statement with its corresponding ground-truth SQL statement for paraphrase questions . . . 73
4.8 The predicted SQL statement with its corresponding ground-truth SQL statement for partial clue questions . . . 74
4.9 The predicted SQL statement with its corresponding ground-truth SQL statement for external knowledge questions . . . 75
4.10 The predicted SQL statement with its corresponding ground truth SQL statement for ambiguous questions . . . 76
4.11 The predicted SQL statement with its corresponding ground truth SQL statement where the AGGREGATION has incorrect predictions . . . 78
4.12 Incorrect matching between the Column names of both the SELECT and the WHERE-Clause . . . 80
4.13 Incorrect WHERE-Clause Operation Prediction . . . 82

4.14 Incorrect WHERE-Clause Condition prediction 84

1 Generated output results of the SQL query, ground-truth SQL query, questions, and column name 116
1 Generated output results of the SQL query, ground-truth SQL query, questions, and column name 117
1 Generated output results of the SQL query, ground-truth SQL query, questions, and column name 118
1 Generated output results of the SQL query, ground-truth SQL query, questions, and column name 119
1 Generated output results of the SQL query, ground-truth SQL query, questions, and column name 120
1 Generated output results of the SQL query, ground-truth SQL query, questions, and column name 121
1 Generated output results of the SQL query, ground-truth SQL query, questions, and column name 122
2 Randomly selected generated output where incorrect prediction are found of the prediction for the sequence-to-sequence models 123
2 Randomly selected generated output where incorrect prediction are found of the prediction for the sequence-to-sequence models 124
2 Randomly selected generated output where incorrect prediction are found of the prediction for the sequence-to-sequence models 125
2 Randomly selected generated output where incorrect prediction are found of the prediction for the sequence-to-sequence models 126
2 Randomly selected generated output where incorrect prediction are found of the prediction for the sequence-to-sequence models 127
2 Randomly selected generated output where incorrect prediction are found of the prediction for the sequence-to-sequence models 128
2 Randomly selected generated output where incorrect prediction are found of the prediction for the sequence-to-sequence models 129
3 The predicted SQL statement with its corresponding ground-truth SQL statement where no exact matching on the AGGREGATION 129
3 The predicted SQL statement with its corresponding ground-truth SQL statement where no exact matching on the AGGREGATION 130
3 The predicted SQL statement with its corresponding ground-truth SQL statement where no exact matching on the AGGREGATION 131
3 The predicted SQL statement with its corresponding ground-truth SQL statement where no exact matching on the AGGREGATION 132
3 The predicted SQL statement with its corresponding ground-truth SQL statement where no exact matching on the AGGREGATION 133
3 The predicted SQL statement with its corresponding ground-truth SQL statement where no exact matching on the AGGREGATION 134
3 The predicted SQL statement with its corresponding ground-truth SQL statement where no exact matching on the AGGREGATION 135
3 The predicted SQL statement with its corresponding ground-truth SQL statement where no exact matching on the AGGREGATION 136
3 The predicted SQL statement with its corresponding ground-truth SQL statement where no exact matching on the AGGREGATION 137

3 The predicted SQL statement with its corresponding ground-truth SQL statement where no exact matching on the AGGREGATION 138
3 The predicted SQL statement with its corresponding ground-truth SQL statement where no exact matching on the AGGREGATION 139
3 The predicted SQL statement with its corresponding ground-truth SQL statement where no exact matching on the AGGREGATION 140
3 The predicted SQL statement with its corresponding ground-truth SQL statement where no exact matching on the AGGREGATION 141
3 The predicted SQL statement with its corresponding ground-truth SQL statement where no exact matching on the AGGREGATION 142
4 Incorrect matching between the Column names of both the SELECT and the WHERE-Clause 142
4 Incorrect matching between the Column names of both the SELECT and the WHERE-Clause 143
4 Incorrect matching between the Column names of both the SELECT and the WHERE-Clause 144
4 Incorrect matching between the Column names of both the SELECT and the WHERE-Clause 145
4 Incorrect matching between the Column names of both the SELECT and the WHERE-Clause 146
4 Incorrect matching between the Column names of both the SELECT and the WHERE-Clause 147
4 Incorrect matching between the Column names of both the SELECT and the WHERE-Clause 148
4 Incorrect matching between the Column names of both the SELECT and the WHERE-Clause 149
4 Incorrect matching between the Column names of both the SELECT and the WHERE-Clause 150
4 Incorrect matching between the Column names of both the SELECT and the WHERE-Clause 151
4 Incorrect matching between the Column names of both the SELECT and the WHERE-Clause 152
4 Incorrect matching between the Column names of both the SELECT and the WHERE-Clause 153
4 Incorrect matching between the Column names of both the SELECT and the WHERE-Clause 154
4 Incorrect matching between the Column names of both the SELECT and the WHERE-Clause 155
5 Incorrect WHERE-Clause Operation Prediction 155
5 Incorrect WHERE-Clause Operation Prediction 156
5 Incorrect WHERE-Clause Operation Prediction 157
5 Incorrect WHERE-Clause Operation Prediction 158
5 Incorrect WHERE-Clause Operation Prediction 159
5 Incorrect WHERE-Clause Operation Prediction 160
5 Incorrect WHERE-Clause Operation Prediction 161
5 Incorrect WHERE-Clause Operation Prediction 162
5 Incorrect WHERE-Clause Operation Prediction 163
5 Incorrect WHERE-Clause Operation Prediction 164

5 Incorrect WHERE-Clause Operation Prediction 165
5 Incorrect WHERE-Clause Operation Prediction 166
5 Incorrect WHERE-Clause Operation Prediction 167
5 Incorrect WHERE-Clause Operation Prediction 168
6 Incorrect WHERE-Clause Condition prediction 168
6 Incorrect WHERE-Clause Condition prediction 169
6 Incorrect WHERE-Clause Condition prediction 170
6 Incorrect WHERE-Clause Condition prediction 171
6 Incorrect WHERE-Clause Condition prediction 172
6 Incorrect WHERE-Clause Condition prediction 173
6 Incorrect WHERE-Clause Condition prediction 174
6 Incorrect WHERE-Clause Condition prediction 175
6 Incorrect WHERE-Clause Condition prediction 176
6 Incorrect WHERE-Clause Condition prediction 177
6 Incorrect WHERE-Clause Condition prediction 178
6 Incorrect WHERE-Clause Condition prediction 179
6 Incorrect WHERE-Clause Condition prediction 180
6 Incorrect WHERE-Clause Condition prediction 181
6 Incorrect WHERE-Clause Condition prediction 182

List of Acronyms

Adam Adaptive Moment Estimation

BP Brevity Penalty

BRNN Bidirectional Recurrent Neural Networks

CFG Context-free grammar

DBOW Distributed Bag-of-Words

DM Distributed Memory

GD Gradient Descent

GloVe Global Vector Word Representation

GRU Gated Recurrent Unit

KL Kullback-Heibler

LSTM Long-Short Term Memory

ML Machine Learning

MT Machine Translation

NLIDB Natural Language Interface to Database

NLP Natural Language Processing

POS Part of Speech

ReLu Rectified Linear Unit

RL Reinforcement Learning

RNN Recurrent Neural Network

SCFG Synchronous Context-Free Grammar

SGD Stochastic Gradient Descent

SP Stanford Parser

SVM Support Vector Machine

VSM Vector Space Model

Dedication

This work is dedicated to my grandfather John Radebe, late mother Grace Khalo, late father Godfrey Khalo, and late grandmother Francina Radebe. Thank you for instilling all the principals of life that has guided me towards this journey of seeking knowledge. The discipline you instilled has provided me with a persevering heart and the strength to pursue my dreams. Your dedication has not gone unnoticed. May your spirits live in my heart forever and may you continue resting in peace. God my solice, it has been a pleasure to realise my dreams. This work is also dedicated to me, Nomonde Francisca Katlego Khalo, o šumile setlogolo sa Matebele, Gadebe, Bhungane, Mthimkhulu, ageeh Mothokwa wee, ke Mothokwa wa mmaseboko, bare ke bathokwa ga ba loyani ba senya ka go utswa fela. Thank you for your resilience. Thank you for sticking it through when the roads became tough to walk.

Acknowledgements

I would like to graciously thank my supervisor Professor Turgay Celik for his encouragement to pursue this degree. This journey has taken me from strength to strength, I would have not realised the importance of post graduate education if you did not create an environment that promotes it. I have learnt a lot about myself, my capacity, and that everything is possible. Thank you for the funding provided throughout my journey. Lastly, thank you for the support provided throughout my studies.

I would like to thank my sponsors BankSeta and IBM to whom without I would have not been able to sustain this journey, I thank you. A special thank you to Sibongile Mkwanazi who has been instrumental in this journey.

A special thank you to my mentor Dr. Joan Byamugisha your love and support took me from strength to strength. Your advises when it was difficult has given me all the hope when it seemed there was no hope. Thank you for believing in me at times when i did not believe in myself, your words of encouragement has sustained me.

Thank you to my aunts who took it upon themselves to take care of me, my health, and well being. I will forever be indebted to you. My family and friends thank you for the support that you have given me throughout this journey.

Thank you to the following friends Dr. Aruleba, Mr. Pandelani, Ms. Risuna, Ms. Thompho, Ms. Nkateko, Mrs. Kai, Ms. Naledi, Ms. Thoko, and Ms. Mahlogonolo it can only be God who blessed me, your continues support is appreciated. To my Lab mates, the journey has just began thank you for being the pillars of strength when the journey seemed hard, we did it.

Chapter 1

Introduction

1.1 Introduction

Querying databases using a natural language is a long-standing research problem that has gained momentum in both natural language processing and the database research community, benefiting from the evolution of technology and the generation of data through various channels (Aditya et al. 2002). Natural languages are a diverse medium of communication for humans, learnt from the environment through text or speech such as English (Goldberg 2017). To query a database, a domain-specific language is required. One of such language is the *Structured Query Language* (SQL), designed to handle information organised in a relational database (Sathick & Jaya 2015). SQL is an expressive language that can make data stored in relational databases inaccessible to users with no technical knowledge (Li & Jagadish 2014*a*). This is because, in order to efficiently manipulate their datasets, users are required to have prior knowledge of the schema of their databases, such as entities and relations, as well as a working understanding of SQL syntax and semantics (Li & Jagadish 2014*a*). To find a solution that can facilitate for ordinary people who are not familiar with SQL to interact with databases, there arises a need to bridge the two languages (natural language and SQL) to have an ability to use natural language as the central communication language with databases (Li & Jagadish 2014*a*).

Natural Language Interface to Database (NLIDB) aims to create an environment in which everyday users can communicate with databases using natural language, making it easier for everyone to access data (Androutsopoulos et al. 1995, Damljanovic et al. 2008). The idea of NLIDB is to ask a question in natural language without knowing the database table information. To achieve this the process of mapping natural language to SQL known as *semantic parsing*, is used (Li & Jagadish 2014*a*). The aim of semantic

parsing is to translate a natural language expression into a machine-understandable representation of its context (Giordani & Moschitti 2009, Popescu et al. 2003, Quirk et al. 2015, Giordani & Moschitti 2012). Several approaches to translating natural languages into machine-readable instructions have been proposed in the last decade, including natural language to SQL translation (Dale et al. 2000, Popescu et al. 2004, Ge & Mooney 2005, Wong 2005, Kate & Mooney 2006, Mooney 2007), exploring datasets such as Geoquery (Zelle & Mooney 1996), ATIS (Aditya et al. 2002), Overnight (Wang & Yang 2015), WebQuestions (Berant et al. 2013), and Freebase917 (Cai & Yates 2013). The most significant benefit of using natural language for querying databases is that users can query complex databases without understanding SQL or the underlying database domain.

The dawn of artificial intelligence has significantly transformed the landscape of natural language processing research by developing systems that are less feature reliant with techniques that aim to 'understand' the language used by humans (Bengio 2012, Goodfellow et al. 2016). Deep learning; a subset of artificial intelligence attempts to learn multiple representation levels, through acquiring and representing knowledge, and making use of context for interpretation (Goodfellow et al. 2016, Goldberg 2017). Semantic parsing, in particular, has successfully used deep learning in developing tasks, using multi-layered neural network models in an input-output fashion (Bengio 2012). Deep learning architectures such as sequence-to-sequence learning have been successfully used to translate natural language to SQL in recent work (Cho et al. 2014, Pasupat & Liang 2015, Dong & Lapata 2016, Zhong et al. 2017), attention mechanisms (Xu et al. 2017, Vaswani et al. 2017), pointer network (Vinyals et al. 2015, Wang, Brockschmidt & Singh 2018) on an end-to-end training approach.

The challenge with natural languages is their inherent ambiguity, ever-changing, and not well defined (Goldberg 2017). For example, a question may be asked in two different forms, requiring the same interpretation by the database to obtain the same information: *Tell me the result for green bay packers?* and *What are the results for green bay packers?*. Though the two questions have the same context, however, they are posed in two different ways. To create an understanding between natural language and SQL a good context capturing approach is needed when implementing natural language translation to SQL (Affolter et al. 2019). Complex questions may be hard to interpret, making it difficult to predict correct or exact answers from the database. These challenges remain even with deep learning architectures' success in this task (Mou et al. 2017, Sun et al. 2018, Yavuz et al. 2018). This research focuses on comparatively investigating the performance of sequence-to-sequence architectures for translating natural language to SQL to query a database and to further analyse where the architectures fail to generate equivalent SQL queries to their respective ground truth.

This research contributes to the body of work that investigates the semantic parsing of natural language to SQL for querying databases. Specifically identifying which sequence-to-sequence models achieves the best performance for the correct prediction of SQL query from natural language. To further identify the key factors of the prediction errors on various levels of each sequence-to-sequence model and finally identify how the prediction errors can be mitigated.

1.2 Problem Definition

The gap between the natural language used for communication between humans and the programming languages used by machines is a significant obstacle for accessing information in the databases for ordinary users. A lot of effort has gone into translating natural language to SQL, with the recent approaches using deep learning architectures, implementing the translation without overly engineering the features. The WikiSQL dataset has become one of the widely used dataset to implement natural language to SQL translation, consisting of the three main SQL query components; SELECT, AGGREGATION, and WHERE-Clause. Deep learning architectures, notably sequence-to-sequence architectures, are used for this task because of their none reliance on heavy feature engineering. Though the advancement in the deep learning field has lead to the use of sequence-to-sequence models for solving this problem, there has not been conclusive research on the strength and weaknesses of various sequence-to-sequence architectures for the translation of natural language to SQL in order to improve upon them. Additionally, the limitation in the form of errors has not been effectively investigated.

1.3 Research Aim

This research aims to comparatively investigate the performance of sequence-to-sequence architectures for the translation of natural language to SQL, to identify key factors of generating incorrect SQL prediction and how to mitigate the generation of incorrect predictions.

1.3.1 Research Objective

To achieve the aim of this research, the objectives are:

- To research and study natural language processing in depth, with a particular emphasis on semantic parsing, distributional representation, deep learning, and natural language to SQL translation.
- To translate natural language to SQL using the WikiSQL question-query dataset.
- To find an architecture that can generate the most correct translation of natural language to SQL to its corresponding ground truth.
- To evaluate the developed architectures and analyse the error generated by the prediction of each architecture in an effort to understand where the sequence-to-sequence models fail to generate correct SQL queries.

1.4 Research Questions

The following questions are answered in order to achieve the research's aim:

1. Which is the most effective sequence-to-sequence architecture for translating natural language to SQL?.
    - Given the different sets of parameters that govern the training process, the sequence-to-sequence are comparatively investigated to translate natural language to SQL.
2. What are the main factors that cause SQL queries to be incorrectly predicted?
    - A model may fail to execute correct predictions for various reasons, an investigation of the model's output on the different errors generated can provide an understanding of where the models fail. An error analysis tries to bring forth errors to understand where the architectures fail to generate a correct prediction.
3. How can prediction failures when generating semantically equivalent SQL queries be mitigated?
    - Given the generated errors where each of the models have failed to predict the correct SQL component, from error analysis, learning how the errors can be mitigated may help generate correct predictions.

1.5 Limitations

The WikiSQL dataset is currently one of the largest collections of question-query pairs often used to translate natural language into SQL. However, it consists of three main SQL components; SELECT, AGGREGATION, and WHERE-Clause. This limits the scope of the investigation as the SQL structure consists of various components such as JOIN, GROUP BY, HAVING, and ORDER BY amongst others. Another limitation is that only one table can be used to query at a time in the WikiSQL dataset. These dataset limitations are noted. However, because of the challenges of this task, the problem is still worth investigating.

1.6 Research Significance

As already stated, one of the most important tasks in this information age is the retrieval of information in places such as databases. Translating natural language to SQL seeks to bridge the gap so that ordinary people can access information. The use of sequence-to-sequence models has shown significant success in developing various semantic parsing methods. However, there are still some drawbacks for the models to learn and understand natural languages' semantics. Investigating deep learning architecture can help determine the effectiveness of neural networks. Furthermore, neural network models often suffer from various factors when modeling sequence problems. One example out-of-vocabulary (OOV). Understanding where the neural network architecture fails, gives significant information as to whether the architectures are the problem or the models need significant pre-processing.

1.7 Thesis Structure

The remaining sections of the thesis are organised as follows, Chapter 2, provides a comprehensive background and literature review, Chapter 3, discusses the methodology of the thesis, architectural setup, the work flow, details of the implementation, as well as the experimental setup Chapter 4 provides the results, analysis, and error analysis discussion, and Chapter 5, concludes the research and provides the future work.

Chapter 2

Background and Literature Review

2.1 Introduction

The previous chapter introduced the work on natural language as the main medium of communicating with databases through the parsing to SQL queries. Semantic parsing tasks such as translating natural language to SQL has attracted more research over the years because of the richness of data and the development of technology. This research progress has led to designing translation systems from rule-based methods to more trainable with less feature engineering methods. This chapter sets the context of the thesis, with an overview of natural language processing, which is the foundation of this project. Following that, a review of semantic parsing from conventional to recent work, natural language to SQL, and finally, neural network training is discussed.

2.2 Natural Language Processing

Natural Language Processing (NLP) is a branch of artificial intelligence and linguistics concerned with teaching computers to comprehend and process human languages through text and speech (Young et al. 2017). Human languages often described as natural language in linguistics, is any language learnt from the environment through experience, built on grammar and vocabulary (Stevenson 2010). They differ from constructed and formal languages used for computer programming or to study logic. Since its onset in 1950s, NLP research focused on tasks such as shallow parsing (part-of-speech tagging, chunking) (Collobert et al. 2011), named-entity (Lample et al. 2016),

spell checking (Singh et al. 2016), language identification (Heinecke 2010), speech synthesis (Taylor 2009), sentiment analysis (Feldman 2013), machine translation (Luong, Le, Sutskever, Vinyals & Kaiser 2015), semantic parsing (Jia & Liang 2016, Cheng et al. 2017), text summarisation (Nallapati et al. 2016), question answering (Kumar et al. 2016), information extraction (Cambria & White 2014), chatbots (Dale 2016), speech recognition (Recognition-Fifth 2013), and opinion mining (Bakshi et al. 2016, Hirschberg & Manning 2015), grammar checkers (Bhirud et al. 2017), and email filtering (Owen & Steiner 2009). The use of NLP is important in building these applications from highly unstructured data to derive understanding and meaning from natural languages (Young et al. 2017). Several components are the building block or are fundamental in developing NLP tasks namely; morphological analysis (examines the arrangement of terms and their constituent parts, such as roots, root words, prefixes, and suffixes); lexical analysis (includes the division of the text into paragraphs, phrases, tokens, and sentences, as well as the identification and definition of word structure), pragmatic analysis (deals with analysing the meaning of any single sentence that depends on that sentence) and syntax analysis (focuses on the ordering of words that may have an impact on their meaning) (Sathick & Jaya 2015, Chowdhary 2020). These components form the fundamental of understanding words, the building blocks of natural language texts (Sathick & Jaya 2015, Chowdhary 2020).

Early NLP research focused on building tasks based on syntax-driven processing, logic, and ontologies, and this requires rules derived from handcrafted feature engineering or statistical analysis performed on labelled data (Cambria & White 2014). In the context of semantic parsing, a syntactic parser creates a parser tree that is mapped to a database query language using rules (Woods 1979). Zelle & Mooney (1996) created a system that converts a natural language sentence to a database query, allowing for an evaluation to see whether the response returned is satisfactory on the database using inductive logic methods. These are word-level approaches that are created in a rule-based manner with a small dataset. However, even with a small dataset, rule-based methods are hard and manually taxing because of the need to represent the whole set of grammatical rules (how words are formed) for the whole dataset (Zelle & Mooney 1996).

One of NLP research's primary goals has been to develop scalable algorithms that learns the necessary words in various stages of representation (Socher 2014). However, the challenge is that natural languages are often complex and ambiguous (Goldberg 2017). A sentence can contain different grammatical structures or more than one interpretation at any level of word formation (Socher 2014, Patil et al. 2014). For instance, the ambiguity can be shown in sentences where the meaning differ although containing similar words (Patil et al. 2014), for instance:

- She bagged two silver medals, and She made a silver speech.
- The tank was full of water, and I saw a military tank.

Another challenge in NLP is feature representation, data sparsity, and interpretability, these complexities often make it difficult to develop NLP tasks (Goldberg 2017).
To overcome some of these challenges an alternative to rule-based methods for modeling natural language tasks emerged in statistical learning and machine learning (ML) (Marquez & Salgado 2000). The alternative method has domination in linear modeling and learning approaches which include; supervised learning (Marquez & Salgado 2000) (which learns from labeled data with models such as perceptrons (Rosenblatt 1961), linear support vector machines (SVM), logistic regression, and decision tree (Marquez & Salgado 2000)), semi-supervised learning (During training, it learns on a small amount of labeled data and a large amount of unlabeled data), and unsupervised learning (which trains over unlabeled data) (Shawe-Taylor & Cristianini 2000). This learning approaches train over very high dimensional and very sparse feature vectors. Their performance depends crucially on the feature representations of the input (Socher 2014). The alternative methods require one to force the dataset into a format compatible with the training algorithm (Socher 2014, Young et al. 2017).

Forcing data into a compatible format is a typical first step in text classification by ignoring grammatical structure and word order, representing texts as unordered lists of words (Socher 2014, Goldberg 2017). Data can be expressed in various ways, a "one-hot" encoding a vector of binary representation is one such way. Given a dataset with three different words: 'people', 'love' and 'food', the vocabulary size for this is three, and the one-hot representation just described, might be that 'people' is encoded by (1, 0, 0), 'love' by (0, 1, 0) and 'food' by (0, 0, 1) (Socher 2014, Goldberg 2017). One-hot encoding maps each word to an index of the vocabulary, which can be very inefficient for storage and computation (Socher 2014). Another drawback with one-hot encoding is that data representation is in a high dimension and is sparse, therefore not capturing any similarity between words (Socher 2014).
The Vector space model (VSM) is another approach to represent data in a document (sentence) that was introduced to overcome the one-hot representation by using a fixed-length vector representation of the document (Pandit 2008). The most common document vectors method are the bag-of-words (BOW) (represents a sentence as a bag of words vector) (Salton et al. 1975), term frequency-inverse document frequency (TFIDF) approach treats words and phrases as unique and discrete symbols (Pandit 2008). These fails to capture the similarity between words or phrases and suffers from sparsity and high dimensionality; and the N-gram (are sequences of elements as they appear in texts,

the values of these features are frequencies of the n-grams) (Robertson & Willett 1998, Pandit 2008). The key concept behind VSM is to describe terms in the vector space that have a similar meaning (Pandit 2008). VSM allows for continuous computation between similar queries and documents without portraying words as binary, as well as rating documents based on their potential relevance (Melucci 2005). However, its drawback is that long documents can be poorly represented because of poor similarity values (Melucci 2005). In the vector space representation, the order in which the terms appear in the text is lost; different sentences may have exactly the same representation as long as the same words are used (Melucci 2005). Documents with similar meaning but different term terminology are not connected, resulting in a false positive (Turney & Pantel 2010). The growing number of noisy words expands the feature size; as a result, the produced feature representation loses some of its key material, and dimensionality issues emerge (Turney & Pantel 2010).
Most rule-based, statistical, and machine learning algorithms look at very sparse vector representation of data and consists of heavy processing which contributes to losing important information from the data (Socher 2014). Although machine learning is a good solution from pure rule-based methods, it is not enough to use as a single type method (Socher 2014).

In recent years, developments in computer vision and natural language processing have been made thanks to a breakthrough in deep learning, a sub-field of artificial intelligence inspired by the way computing functions in the brain works (Krizhevsky et al. 2012). Multi-layer neural networks are often used in conjunction with low-dimensional feature representation in deep learning algorithms, which is described as the learning of parameterised differential mathematical functions (Goodfellow et al. 2016). The feature representations are automatically learned by the learning algorithms from the raw data. These representations can then be used to solve the problem of feature representation in prediction tasks (Goodfellow et al. 2016, Goldberg 2017). Deep learning works by correctly representing the data so that it can be predicted (Goldberg 2017). This is accomplished by an input-output mapping, which involves feeding data into a network that generates successive transformations of the input data before the output is predicted by a final transformation (Goodfellow et al. 2016, Goldberg 2017). The concept behind neural language algorithms is to use context-capturing vector representations of words to predict the probability of a word occurring given its context (Bengio et al. 2003). This has set natural language processing forward, particularly concerning semantics in natural languages (Goldberg 2017).

2.2.1 Neural Natural language processing representation

2.2.1.1 Distributed word representation

The use of embedding layers, which depict the input sequence as a low-dimensional feature representation by mapping discrete symbols to continuous vectors, has been a key component in multi-layered neural networks (Goldberg 2017, Shi et al. 2017). Although there exist many different methods for constructing vector representations for words, a highlight in vector representations came with the use of distributed representation for words in Word2Vec (Mikolov, Chen, Corrado & Dean 2013, Mikolov, Sutskever, Chen, Corrado & Dean 2013, Mikolov, Yih & Zweig 2013). Word embedding follows a distributional hypothesis where words with similar meanings occur in similar contexts (Mikolov, Chen, Corrado & Dean 2013, Mikolov, Sutskever, Chen, Corrado & Dean 2013, Pennington et al. 2014). Word2Vec leverages neural networks, learning low-dimensional word representations using prediction directly. Two configurations of the Word2Vec model are the *continuous bag of words* (CBOW) and the *skip-gram* (SG) models.
The **CBOW** learns to predict the middle word in a symmetric window based on the number of the vector representations of the terms in the window collection of word pairs, the same word when observed as a background word (Mikolov, Chen, Corrado & Dean 2013). Given a context word surrounding a target word, CBOW calculates the target word's conditional probability (Young et al. 2017). Figure 2.1 depicts a straightforward CBOW architecture. A fully connected neural network takes input of a one-hot vector

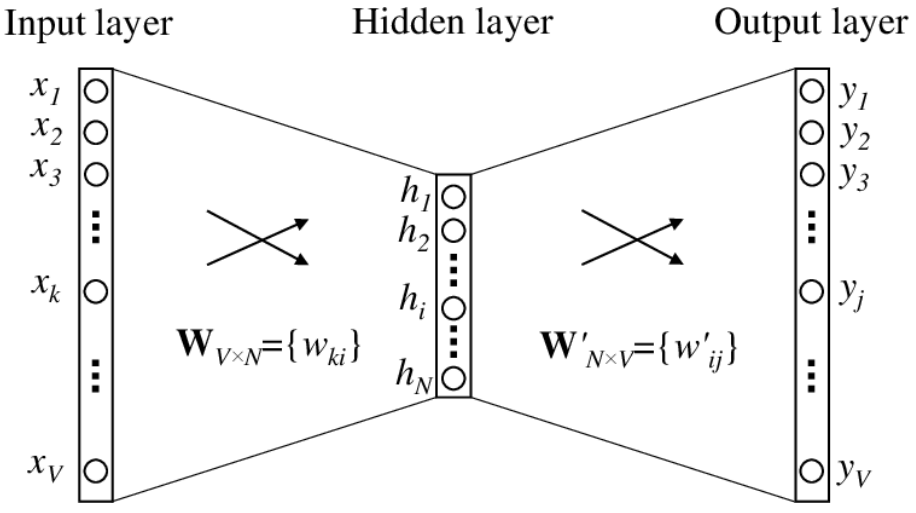

Figure 2.1: The CBOW architecture where the current word is predicted based on the context (Rong 2014).

$x_1, ..., x_V$ of context words with V neurons which are vocabulary size, has one hidden layer $h_1, ...h_N$, the hidden nodes has N neurons, and output layer $y_1, ...y_V$ with V neurons which are vocabulary size (Mikolov, Chen, Corrado & Dean 2013). The layers

connected by $W \in R^{V \times N}$ and $W^{'} \in R^{N \times V}$ are $V \times N$ matrix and $N \times V$ trainable matrix respectively (Mikolov, Chen, Corrado & Dean 2013, Young et al. 2017). Each row of W represent each row with N-dimension vector representation, v_w is an associated word of the input layer. Given a context (a word) assuming $x_k = 1$ and $x_{k^{'}=0}$ for $k^{'} \neq k$,

$$h = W^T x = W^T_{(k,.)} := v^T_{w_I} \tag{2.1}$$

copying the $k-th$ row of W to h. v_{w_I} is the vector representation of the input word w_I, from the hidden layer to the output layer there is a different weight matrix $W^{'} = w^i_{ij}$ which is an $N \times V$ matrix (Rong 2014). The weights are used to calculate a score for each word in the vocabulary, and the score is then subjected to a softmax of log-linear classification to obtain the posterior distribution of the terms (Mikolov, Chen, Corrado & Dean 2013, Rong 2014).

The **Skip-Gram** takes a given central target word and predicts the surrounding context words (Mikolov, Sutskever, Chen, Corrado & Dean 2013, Young et al. 2017). It efficiently learns word vectors from a large corpus used to assume the word's meaning could be learned from its context. The data can be fed into the model in an online way and with little pre-processing and requiring little memory (Mikolov, Sutskever, Chen, Corrado & Dean 2013). The skip-gram model maximizes the average log likelihood by the given equation given a sequence of training terms $w_1, w_2, w_3, ..., w_T$:

$$\frac{1}{T} \sum_{t=1}^{T} = \sum_{-c \leqslant j \leqslant c, j \neq 0} \log P(w_{t+j} | w_t) \tag{2.2}$$

T is the size of the training set, w_t is the input term, and w_{t+j} is the output target word, where c is the size of the training context (which can be a function of the centre word w_t), T is the size of the training set, wt is the input word, and w_{t+j} is the output target word (Mikolov, Sutskever, Chen, Corrado & Dean 2013). The basic skip-gram formulation from equation 2.2 defines $P(w_{t+j}|w_t)$ using the softmax function:

$$P(w_O | w_I) = \frac{\exp({v^{'}_{w_o}}^T v_{w_i})}{\sum_{w=1}^{W} \exp({v^{'}_{w}}^T v_{w_i})} \tag{2.3}$$

where v_w and $v^{'}_w$ are the "*input*" and "*output*" of the vector representations of w respectively, w_o is the target word, The input term is w_i, and the number of terms in the vocabulary is W (Rong 2014, Mikolov, Sutskever, Chen, Corrado & Dean 2013).

Recently, Pennington et al. (2014) demonstrated that the word's meaning could be learned from the co-occurrence relationship between words in the learned vector space,

with a log-linear model trained to encode semantic relationships between words called *GloVe* (Pennington et al. 2014, Jun-Li et al. 2017, Liu et al. 2020). The **GloVe** model is based on the idea that terms with a higher association have a higher co-occurrence count when considering word pair relationships rather than word to word relationships. The word vector captures sub-linear relationships in the vector space (Pennington et al. 2014, Jun-Li et al. 2017). A fixed-size window is set to divide the collection of words into fixed length context. Given words w_1, w_2, and w_3, assuming that w_1 and w_2 are related closer than w_2 is related to w_3, $P(w_1|w_2) > P(w_3|w_2)$ assumes w_1 and w_3 are substantially the same semantic. Therefore $P(w_1, w_2) = P(w_3|w_2)$, where $P(w_i|w_k)$ represents the conditional probability of w_i under w_j condition, and is calculated by co-occurrence count between w_i and w_k given by:

$$P(w_j|w_k) = \frac{X_{jk}}{X_k} \tag{2.4}$$

where X_{jk} is the co-occurrence relation between w_j and w_k which appear in the same context, and X_k represents the count of contexts that contain w_k (Pennington et al. 2014, Jun-Li et al. 2017). The word embedding and $P(w_j|w_k)$ have a relationship where each word has two vectors, a word embedding and a context-word embedding which represent. These two vectors are shown to have the same ability of expression.

$$w_i^T \tilde{w}_k + b_i + \tilde{b}_k = \log(X_{ik}) \tag{2.5}$$

Where X_{ik} is the co-occurrence relation between the w_i, and w_k, while b_i and $\tilde{b}_k$ represent the offset term for w_i, and word w_k respectively and w_i^T and $\tilde{w}_k$ denotes the word embedding of w_i and the context-word embedding of w_k respectively.

$$J = \sum_{i,j=1}^{v} f(X_{ij})(w_i^T \tilde{w}_j + b_i + \tilde{b}_j - \log(X_{ij}))^2 \tag{2.6}$$

$$f(x) = \begin{cases} (\frac{x}{x_{max}})\ \hat{a} & x < x_{max} \\ 1 & \text{other} \end{cases} \tag{2.7}$$

Equation 2.4 calculates the co-occurrence relationship between words where the word pair count appears in the same context. Equation 2.5 train the GloVe model based on the co-occurrence relationship (Jun-Li et al. 2017). Equation 2.6 is the objective function of the GloVe model where f is the weighted function given by equation 2.7 (Pennington et al. 2014, Jun-Li et al. 2017). Word2Vec and GloVe have something in common: they both depend on the premise that words in similar contexts mean the same thing. Both Word2Vec and GloVe have the disadvantage of not solving out-of-vocabulary terms and not separating any opposite word pairs, for example, “good” and “bad” are typically

placed very close to each other in the vector space, limiting the output of word vectors in NLP tasks like sentiment analysis (Filling 2017). Zhong et al. (2017) showed the effectiveness of using GloVe on the Seq2SQL model that translate natural language to SQL. (Bordawekar & Shmueli 2017) uses Word2Vec to query the database, the results were not conclusive making it hard to effectively compare it to the Seq2SQL model (Zhong et al. 2017).

A similar representation of word embeddings, (Joulin, Grave, Bojanowski, Douze, Jégou & Mikolov 2016) proposed a text classification method *FastText*. This is based on a linear combination of word embeddings based on previous work of Word2Vec (Joulin, Grave, Bojanowski, Douze, Jégou & Mikolov 2016). The use of n-grams distinguishes FastText from Word2Vec; Word2Vec learns vectors only for full words contained in the training dataset. FastText, on the other hand, learns vectors for each word's n-grams as well as each full word. The mean of the target word vector and its component n-gram vectors are used in each training phase (Joulin, Grave, Bojanowski, Douze, Jégou & Mikolov 2016). The advantage of FastText over Word2Vec is the representation of words using their sub-words potentially resolve the out-of-vocabulary problem by capturing the meaning of shorter words and allowing the embeddings to understand suffixes and prefixes (Joulin, Grave, Bojanowski & Mikolov 2016, Bojanowski et al. 2017).

A skip-gram model is trained to learn the embeddings after the word has been represented using character n-grams (Joulin, Grave, Bojanowski & Mikolov 2016). For example, using individual representations for "circum" and "navigation," an insightful representation for "circumnavigation" can be created, which would otherwise appear to be too rare to learn dictionary-level embedding for embedding such as GloVe and Word2Vec (Athiwaratkun et al. 2018). The power of the training process can then be shared through terms with common origins (Athiwaratkun et al. 2018). Word2vec (Mikolov, Sutskever, Chen, Corrado & Dean 2013) and GloVe (Pennington et al. 2014) both fail to provide any vector representation for words that are not in the model dictionary (Athiwaratkun et al. 2018). Figure 2.2 shows an architecture of the FastText model. FastText architecture presents a sentence with N-ngram features $x_1, ..., x_N$. The features are embedded and averaged to form the hidden variable. A softmax function (discussed in section) is used to compute the probability distribution over the predefined classes, this minimises the negative loglikelihood over the classes (Joulin, Grave, Bojanowski & Mikolov 2016).

Word embeddings are good in capturing word semantics, but the meaning and deep semantics of text do not depend on the meaning of words alone, the constructed sentence can bear more meaning than word level (Le & Mikolov 2014). Beyond word embeddings, it is also possible to embed text at paragraph or document level. In most

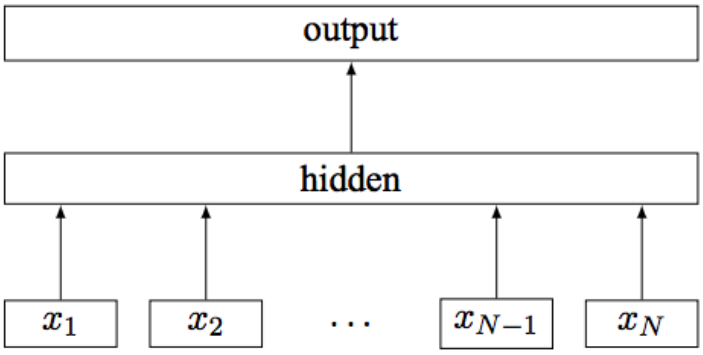

Figure 2.2: An architecture of the FastText Model (Joulin, Grave, Bojanowski & Mikolov 2016).

recent developments, document embedding (Le & Mikolov 2014), an unsupervised algorithm for learning fixed-length feature representations from variable-length text fragments like sentences, paragraphs, and documents. It follows the work on Word2Vec by using continuous distributed vector representations applied to variable-length pieces of texts. The paragraph analogue of the skip-gram model called **Distributed Bag-of-Words (DBOW)** and the analogue of CBOW called **Distributed Memory (DM)** . Instead of attempting to predict a context given a phrase, as the skip-gram model does, the DBOW model attempts to predict a context given a passage (Le & Mikolov 2014, Dai et al. 2015). The meaning is a list of words randomly selected from a text window from the passage (Le & Mikolov 2014, Dai et al. 2015). The DM model acts as a memory that remembers what is missing from the current context (Le & Mikolov 2014, Dai et al. 2015). An important advantage of paragraph vector is that it considers the word order (Dai et al. 2015).

Paragraph Vectors also address the most important property of capturing the words' semantics, carries as much semantic information as possible (Audet 2008, Le & Mikolov 2014, Dai et al. 2015). In contexts sampled from the paragraph, the vector representations are learned to predict the surrounding terms (Collobert & Weston 2008, Le & Mikolov 2014, Goldberg 2017).
Other types of embeddings are the hierarchical (Sanh et al. 2019), graph (Cai et al. 2018), concept (Ma & Cambria 2018), dependency-based (Levy & Goldberg 2014), and multi-sense embeddings (Li & Jurafsky 2015) to name a few. These embeddings are usually used to feed into the training network such as deep learning algorithms, learning to combine these word vectors in a way that is useful for prediction while eliminating the discreteness, data sparsity, and high dimensionality effects to some degree (Le & Mikolov 2014, Turian et al. 2010, Huang et al. 2012, Liu et al. 2020). They represent features better than traditional high-dimensional vectors such as TF-IDF and BOW (Le & Mikolov 2014).

2.3 Semantic Parsing

In natural language processing, semantic parsing is a task that maps natural languages into their machine translatable representation (Kate & Mooney 2006, Zettlemoyer & Collins 2009). Semantic refers to " Meaning", while parsing refers to resolving sentences into their component (Goddard 2011). This task is distinct from other sequential prediction tasks such as machine translation and natural language generation in that it entails translating from unstructured languages to structured languages that follow their own set of laws, thus increasing reliance on the task for which inference is necessary (Kamath & Das 2018). The development of systems that understand natural language is an essential activity in natural language processing, with applications like code generation (Popescu et al. 2003, Giordani & Moschitti 2012, Poon 2013, Quirk et al. 2015, Ling et al. 2016) which maps natural language query directly to high-level, general-purpose programming language such as Python, SQL. Other applications includes question answering (Yih et al. 2014), automated reasoning (Furbach et al. 2010), semantic role labeling (Palmer et al. 2010) to name a few. Semantic parsing is inherently more challenging to other parsing tasks because it requires understanding of concepts from different word phrases (Suliman & Zhang 2015).

2.3.1 Traditional Semantic Parsing

Earlier research developed domain specific representations with a complex lexicon, grammar, syntactic structure, and lexical matching between question making deployment on new environments challenging (Woods 1979, Popescu et al. 2003, Giordani & Moschitti 2012, Poon 2013). The grammatical structure can be used to define the trigger word's dependencies from a question (Popescu et al. 2003, Giordani & Moschitti 2012, Poon 2013). This is needed for long-term dependencies that are difficult to detect using simple natural language patterns (Saha et al. 2016). Early semantic parsers were built for natural language interfaces to databases including (Woods 1972), Chat-80 (Warren & Pereira 1982), TEAM (Grosz 1983), and ATHENA (Saha et al. 2016). However, grammar-based systems are powerful; however, they are highly dependent on handcrafted feature extraction. Which is often manually taxing. Several earlier approach built semantic parsing tasks using a more data-driven approach to semantic parsing using statistics, and machine learning. These algorithms generate semantic parsers that are more robust, accurate, and less application-specific than hand-crafted parsers (Popescu et al. 2003, Giordani & Moschitti 2012, Poon 2013).

2.3.1.1 Traditional Statistical Approaches

Kate & Mooney (2006) uses context-free grammar rules (capturing the "block structure" of sentences naturally) to create logical forms from an annotated corpus of augmented parse trees as a bag of linguistic features, and to use kernel Support Vector Machines (SVM) in the development of KRISP (Shawe-Taylor & Cristianini 2000, Lodhi et al. 2002). Meaning representations (MR) string kernels are used to find the most likely semantic parse for natural language sentences, learning semantic parsers using SVM classifiers based on string subsequence kernels. KRISP is particularly robust to noise (Kate & Mooney 2006). Kate & Mooney (2006) found KRISP to be comparable to other existing systems, WASP (McDonald 1992, Wong & Mooney 2006), a framework that uses predictive machine translation techniques to learn transformation rules; SCISSOR (Ge & Mooney 2005), a framework that learns a syntactic-semantic parser that is incorporated; and CHILL (Tang & Mooney 2001). The main drawback of context-free grammars (CFG) is that they require augmented parse trees for training, which can be difficult to come by Wong & Mooney (2006).

Giordani & Moschitti (2009) addresses this, by defining semantically equivalent NL questions and SQL queries with a dataset containing syntactic trees of questions and queries and encode them in SVM using kernel functions. This is accomplished by adding a linear kernel to the union of the question and query bag-of-words. The product of the two kernels representing questions and queries yields feature pairs that express the relational characteristics of the two languages' syntactic/semantic representations. The grammar determines the space of derivations from utterances to logical forms, and the model and parsing algorithm determine which derivation has the highest score (Giordani & Moschitti 2009). Giordani & Moschitti (2009) show the effectiveness of using kernel products to develop natural language to SQL translation. The drawback of (Giordani & Moschitti 2009) is that the trained data is manually paired, based on the initial annotation available, annotated using a semi-supervised algorithm. Another drawback is that the dataset used a small dataset with semantically equivalent. (Giordani & Moschitti 2009) is comparative to (Zhang & Lee 2003, Cumby & Roth 2003).

Zettlemoyer & Collins (2007) centers on a non-standard probabilistic Combinatory Category Grammar, learning to parse sentences into lambda-calculus representations of their underlying semantics (CCG) (Steedman 2000). A PCCG (Probabilistic CCG) is a tool for rating possible parses for a given sentence in terms of the likelihood of the various parses that might occur for that sentence, so that variations can be treated. Each lexicon entry is made up of a word and its corresponding category. CCG helps a framework to manage complex semantic effects such as synchronization or scoping phenomena; however, CCG can be static, which can trigger issues when dealing with

natural language information that is random and unedited. The drawback of using this approach is that it requires manually specification of combination rules inducing the rules from annotated logical forms as supervision and they operate in limited domains with too small number of logical predicates (Kwiatkowksi et al. 2010, Krishnamurthy & Mitchell 2012, Cai & Yates 2013). Zettlemoyer & Collins (2007) found performance comparable with (He & Young 2006) which uses semantic constructs with a semantic non-terminal mark are similar to context-free parses. The ATIS results (Zettlemoyer & Collins 2007) outperform previous methods in two benchmark database domains (He & Young 2006). Zettlemoyer & Collins (2007), had deficiencies in the form of data sparsity caused by not sharing weights for related classes of words.
Kwiatkowski et al. (2011) introduces factored lexicons to boost this. These systems necessitate extensive annotation and are sometimes difficult to scale. Another disadvantage is that they only function in a limited range of domains. The process of manually annotating data sets remains difficult, and as a result, numerous types of oversight have been investigated Kwiatkowski et al. (2011).
Cai & Yates (2013) solves the annotation problem by using supervised training of a semantic parser, schema matching, and pattern learning without annotated logical types that scale to a large number of predicates on the Freebase database (Zettlemoyer & Collins 2007). This employs a completely automatic method of constructing Combinatory Category Grammar (CCG) lexical entries for the semantic parser by turning it into a prediction task that makes use of synonym-matching. This is based on convergence of second-order logic, learning a lexicon from the training data using linear regression with least-squares parameter estimation with support vector regression models with non-linear kernels, and support vector regression models with non-linear kernels (Cai & Yates 2013). The belief that function terms have no domain-specific meaning, which prepositions, in particular, can breach, is a flaw in this model (Cai & Yates 2013). When there are few valid findings within the set of extracted relations, the accuracy is poor (Kwiatkowksi et al. 2010, Krishnamurthy & Mitchell 2012, Artzi et al. 2014, Krishnamurthy & Mitchell 2015, Krishnamurthy 2016).

2.3.1.2 Machine Translation

Wong & Mooney (2006) uses predictive machine translation (MT) techniques for parsing a syntax-based translation paradigm with a recursive framework for semantic parsing. The algorithm, which requires no previous knowledge of NL syntax, learns a semantic parser from a series of natural language sentences annotated with their proper meaning representation. However, it assumes that the grammar which is unambiguous and context-free of the target MRL is available. This uses synchronous context-free grammar

(SCFG) (Capturing grammatical structures on each end of the paired dataset) for generating the pairs in a translation. Each SCFG rule is made up of only one non-terminal. For lexical literacy, a word-based alignment model is used, and the parsing model may be thought of as a syntax-based translation model. Wong & Mooney (2006) compared this approach with existing learning methods requiring similar (Tang & Mooney 2001), and SCISSOR (Ge & Mooney 2005, Wong & Mooney 2006). COCKTAIL (Tang & Mooney 2001); a shift-reduce parser built on inductive logic programming that is deterministic. Wong & Mooney (2006) outperformed SILT (Kate et al. 2005) in terms of recall, a local bottom-up search is used for lexical learning, which is much less successful than WASP's word alignment-based algorithm. The model was more resistant to changes in mission difficulty and word order. The disadvantage suggests that the goal MRL is unambiguous.

2.4 Semantic Parsing of Natural language to SQL

In section 2.2 natural language processing was introduced generally and section 2.3 introduced semantic parsing generally. This section presents the parsing of natural language to SQL, where a natural language is converted to its corresponding SQL query. Section 2.4.1 presents the traditional methods and the rest of the section presents recent literature that uses deep learning to translate natural language to SQL.

2.4.1 Traditional Natural language to SQL

Berant et al. (2013) uses a knowledgebase with a huge corpus on question-answer pairs by defining over-generate composition rules and then using model features to simulate soft rules and categories using Part-of-Speech (POS) for features and on the denotations of the projected logical types without annotation corpus to Freebase, reminiscent of (Cai & Yates 2013). It also employs a bridging operation to create additional predicates dependent on neighboring predicates where annotated logical types are not accessible. Lexicon has an effect on individual expressions, resulting in truth aggregation. This alignment, on the other hand, only allows for the generation of a subset of the desired predicates, minimising the amount of oversight and allowing for a more thorough assessment of a new dataset. The disadvantage of this approach is that the noise and difficulties in canonicalisation make it impossible to perform accurate composition, thereby invalidating one of semantic parsing's main benefits. When compared to the previous year, the findings showed an increase in efficiency (Cai & Yates 2013).

Pasupat & Liang (2015) utilises question-answer pairs as supervision to tackle the task of answering questions on semi-structured tables. To deal with the problem of hidden

tables, an information graph from the table in question is used, resulting in the encoding of all related relations. For a high-coverage grammar, the method parses the query into candidate logical types. A log-linear model is used to rerank the contestants, and then the highest-scoring logical form is used to generate the response denotation. It also employs beam scan in conjunction with pruning techniques (Steinbiss et al. 1994), to monitor the combinatorial explosion, based on form and denotation constraints. The HTML tables are semi-structured tables that contain open-ended data from the network that has not been normalized. The Freebase, on the other hand, has a global fixed relation schema of normalised entities and relations (Liang et al. 2013). Pasupat & Liang (2015) compared results with IR (Cai & Yates 2013) and WR (Berant et al. 2013) outperforming both systems.

Waltinger et al. (2013), uses numerous NLP methods, lemmatisation, POS tagging, named object identification, and dependency parsing to provide natural language access to massive data bodies with multiple databases. Following the application of NLP methods, syntax principles are used to re-validate object information by handling domain-specific input words. Syntax laws are often used to determine the question's emphasis. The next step is to identify and address any possible concepts in the input query. Question meanings, including how terms and instances are related to one another, are created, then validated and graded. The system's ability to query from many database domains is one of its benefits. Furthermore, users have the option of selecting the kind of feedback questions they choose.

NaLIR Li & Jagadish (2014*a*,*b*) parses the input question using the Stanford Parser (a program that works out the grammatical structure of natural language) and convert the parse tree into SQL. If there are some ambiguous queries, the users are asked for clarity in the mapping phrases of the input query parse tree to SQL components. The users are presented with the candidate interpretations provided by a legitimate parse tree, and they must choose the right interpretation. The final stage converts the user-selected, modified, and correct parse tree into SQL. This gives the user the answer as well as a rewritten feedback query depending on the translation. The connection with the customer is the power of NaLIR. It is, however, heavily reliant on the parse tree (Affolter et al. 2019).

BioSmart Jamil (2017) translates a natural language query into a programming language like SQL using a syntactic classification of the input question. Using the parser tree, the method splits the input questions into three categories of queries: iterative, conditional, and imperative. A verb is preceded by an object in an imperative question, for example. Through nesting basic query types randomly, a more articulate and therefore complex input question can be created. The Stanford parser is also used by BioSmart to decode the input question. The method then maps the parse tree to each of the query templates

or predefined queries. A query in their scheme is made up of some of these templates, each of which captures the essence of the question. The tables and joins used to compute the query are then determined. The knowledge about the tables and joins is used to convert the models into a logical query. BioSmart has the ability to query any database. However, if the machine is unable to fit the input question to certain query types, it would be unable to respond, and the structure is application-specific and manual.
Other traditional system querying database using natural language by translating natural language to SQL includes (Walter et al. 2012, Song et al. 2015).

2.4.2 Natural language to SQL with deep neural networks

Artificial neural networks (ANNs) built of interconnected neurons, are the bases of deep learning algorithms (McCulloch & Pitts 1943, Mitchell 1997). ANNs are mathematical models inspired by the functions of the human brain, modelled as a system of neurons connected by multiple layers (Mitchell 1997). These layers consist of the *input layer*, an arbitrary number of intermediate layers called *hidden layers*, then finally the *output layer*. They can represent an arbitrary non-linear mapping from inputs to outputs and characterised by simple neurons processing elements, weighted connections, and distributed representation of knowledge over the connections (Mitchell 1997). These networks, therefore, provides a robust method of approximating vector-valued, real-valued, or discrete-valued target functions from input functions (Mitchell 1997). ANNs have successfully been implemented in areas such as text classification (Minaee et al. 2020), pattern matching (Lu & Li 2013), semantic parsing (Grefenstette et al. 2014, Dong & Lapata 2016), anomaly detection (Ryan et al. 1998), image processing (Schalkoff 1989), signal processing (Cichocki et al. 1993), pattern recognition (Bishop et al. 1995) and natural language processing (Goldberg 2017). Current semantic parsing has shown significant success using NNs over traditional methods (Grefenstette et al. 2014, Dong & Lapata 2016). Figure 2.3 shows a typical neural network structure. The neural network consists of three parts:

- x_i input to the neurons and input weight w_i corresponding to the neurons.
- Adding the weighted input.
- An activation function.

where x_i for $i = 1, .., n$ is the input to the neurons and where n is the number of input signals connected to the output node by connection weights, w_{ij}, y_i.

$$y_i(x) = \sum_{i=1}^{n} x_i \cdot w_i = \mathbf{x} \cdot \mathbf{w} \tag{2.8}$$

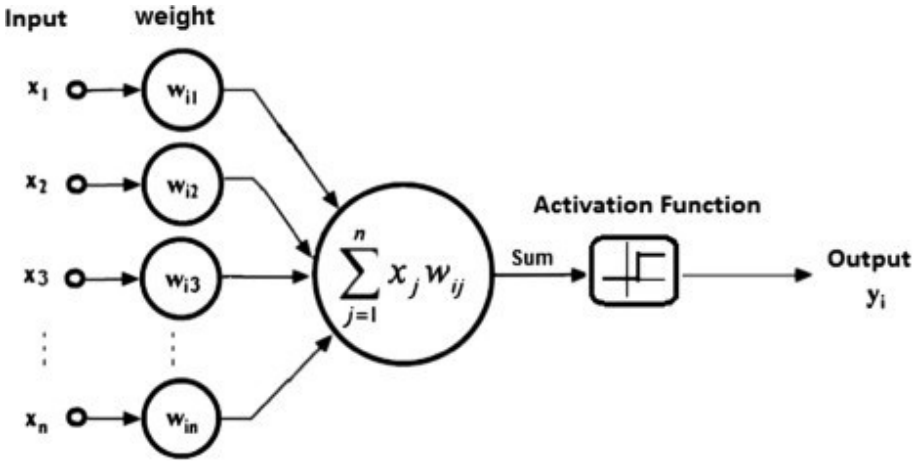

Figure 2.3: A building block of neural networks (Saxen 2017).

Equation 2.8 represents the weighted sum, where w is the weights and x is the inputs into the neuron. To introduce non-linearity activation function (f) which decides whether or not to fire in the neurons. An activation function (discussed in Chapter 2.5) can be used.

$$f(y_i) = \frac{1}{1 + e^{-\alpha y_i}} \tag{2.9}$$

where α is the slope parameter. A bias b can be introduced as below:

$$y_i(x) = \sum_{i=1}^{n} x_i \cdot w_i + b \tag{2.10}$$

The addition of more layers to the NN can become complex, needing more resources to train. This makes the network deeper bringing forth other NNs such as Recurrent Neural Network (Mikolov et al. 2010) and Convolutional Neural Network (Ciresan et al. 2011).

2.4.2.1 Recurrent Neural Network

Recurrent Neural Networks (RNNs) are a variant of NNs, regarded as a deep network that makes it possible to model long dependencies (Neubig 2017, Sutskever et al. 2014*a*, Schaefer et al. 2008). When processing a variable-length sequence $x = (x_1, x_2, ..., x_n)$, the RNNs computes the sequence-to-sequence problem by referring from the previous state when calculating the current state, giving it the ability to keep some information from the previous input, this is given by:

$$h_t = \begin{cases} \tanh(W_{xh}x_t + W_{hh}h_{t-1} + b_h) & \text{t} \geq 1, \\ 0 & \text{otherwise} \end{cases} \tag{2.11}$$

where x_t are the inputs at time step t, h_{t-1} is the previous hidden layer, $W_{xh}x_t$, and W_{hh} are the trainable weights and b_h is the bias for time steps $t \geq 1$ (Goodfellow et al. 2016, Neubig 2017). Figure 2.4 is an example of a RNN structure flow to the previous state; learning from it while its current state information is kept in the loop. Basik et al.

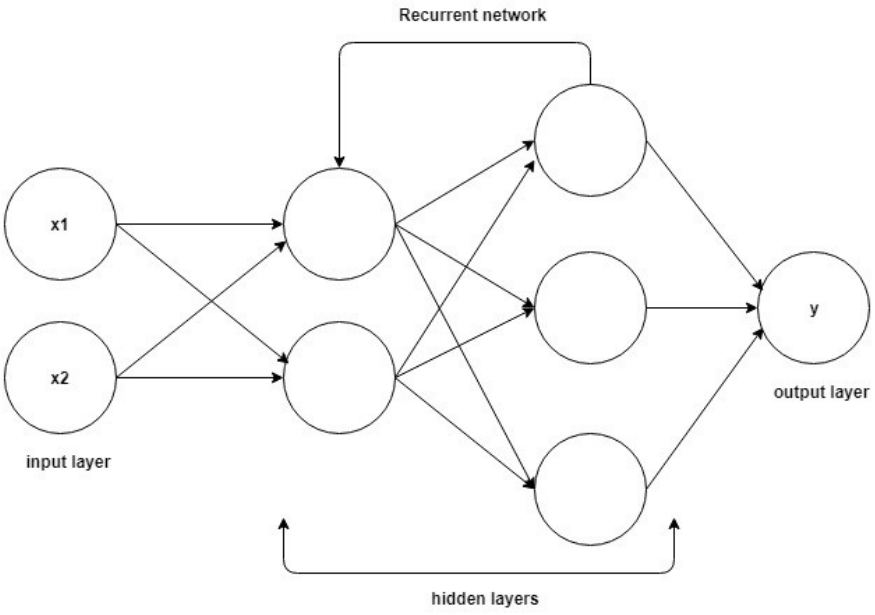

Figure 2.4: A structure of Recurrent Neural Network (Krenker et al. 2011).

(2018) developed a system called **DBPal**, centered on the RNN model, which provides an auto-completion model that proposes partial query extensions to users during query formulation, thus assisting in the writing of complex queries. DBPal describes natural language and SQL pairs using database schema and query models, but it does worse on multi-table queries, and the small training set of 600 examples does not appear to generalize well for join queries. Its biggest flaw is that it lacks support for more complex requests, such as various types of SQL nesting.

RNNs have architectural pitfalls with long-term dependencies, including their computational simplicity. In reality, training them becomes problematic because squeezing non-linearity at each step causes an exponential decay in the error signal over time, resulting in vanishing or bursting gradients (Neubig 2017). The gradient vanishes when adding more layers over a sequence training of the RNN; the gradient of the loss function turns to zero, making it harder to train the RNN network (Neubig 2017).

2.4.2.2 Bidirectional Recurrent Neural Networks

Bidirectional Recurrent Neural Network (Schuster & Paliwal 1997) extends the RNN by predicting the next word in the sequence based on previous data. Predictions are made with future inputs by having the RNN model read through the corpus backward

(Schuster & Paliwal 1997). BRNN has two hidden layers at each time-step, one hidden layer propagates from left to right and another hidden layer propagates from right to left (Goodfellow et al. 2016). The predicted output is generated by combining the scores produced by the forward and backward RNN hidden layers (Socher et al. 2015, Goodfellow et al. 2016). In Figure 2.5, a BRNN computes the forward hidden layer $\overrightarrow{h}_t$, the backward hidden layer $\overleftarrow{h}_t$, and the output sequence y by iterating the backward layer from $t = T$ to 1, the forward layer from $t = 1$ to T and then updating the output layer, given by the below equation:

$$\overrightarrow{h}_t = f(\overrightarrow{W}x_t + \overrightarrow{V}\overrightarrow{h}_{t-1} + b) \tag{2.12}$$

$$\overleftarrow{h}_t = f(\overleftarrow{W}x_t + \overleftarrow{V}\overleftarrow{h}_{t+1} + b) \tag{2.13}$$

$$\hat{y}_t = g(Uh_t + c) = g(U[\overrightarrow{h}_t; \overleftarrow{h}_t] + c) \tag{2.14}$$

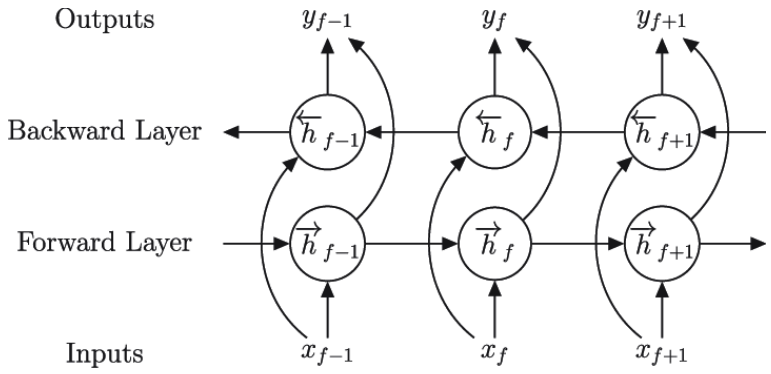

Figure 2.5: An architecture of a Bidirectional Recurrent Neural Network (Graves et al. 2013).

where g is the activation function, U and V are trainable matrices, and h_t is the hidden states. Consider the task of sequence tagging over a sentence $x_1, ..., x_n$. An RNN allows us to compute the ith word x_i function based on the past-the words x_1W_i up to and including it (Goodfellow et al. 2016).The BiRNN relaxes the set window size assumption, enabling users to peer selectively deep into the sequence's history and potential (Goodfellow et al. 2016).

2.4.2.3 Long-Short Term Memory

Long-Short Term Memory (LSTMs) (Hochreiter & Schmidhuber 1997) are a special kind of RNNs that has shown to model long-term dependencies with much greater reliability

and more resilient to the problem of vanishing gradients (Neubig 2017). LSTM learns long-term dependencies by re-parameterisation. In addition to computing the standard hidden state h it has a memory cell c (equation 2.18) for which its gradient is exactly one, removing the gradient descent problem by introducing a gating mechanism (Neubig 2017, Jozefowicz et al. 2015). The sigmoid layer called the "forget gate layer" in equation 2.15 decides what information is going in or discarded from the cell state. The "gates" are meant to control information to either pass through or blocked from passing. From an input sequence $x = (x_1, ..., x_t)$ a mapping is generated to an output sequence $y = (y_1, ..., y_t)$ which calculates the network unit activation from equations from t = 1 to T:

$$f_t = \sigma(W_f x_t + W_f h_{t-1} + b_f) \tag{2.15}$$

$$i_t = \sigma(W_i x_t + W_i h_{t-1} + b_i) \tag{2.16}$$

$$\widehat{C}_t = \tanh(W_c x_t + W_c h_{t-1} + b_c) \tag{2.17}$$

$$c_t = f_t * c_{t-1} + i_t * \widehat{C}_t \tag{2.18}$$

$$o_t = \sigma(W_o x_t + W_o h_{t-1} + b_o) \tag{2.19}$$

$$h_t = o_t * \tanh(c_t) \tag{2.20}$$

Figure 2.6 shows the LSTM and its interactive layers. Equation 2.16 decides which values

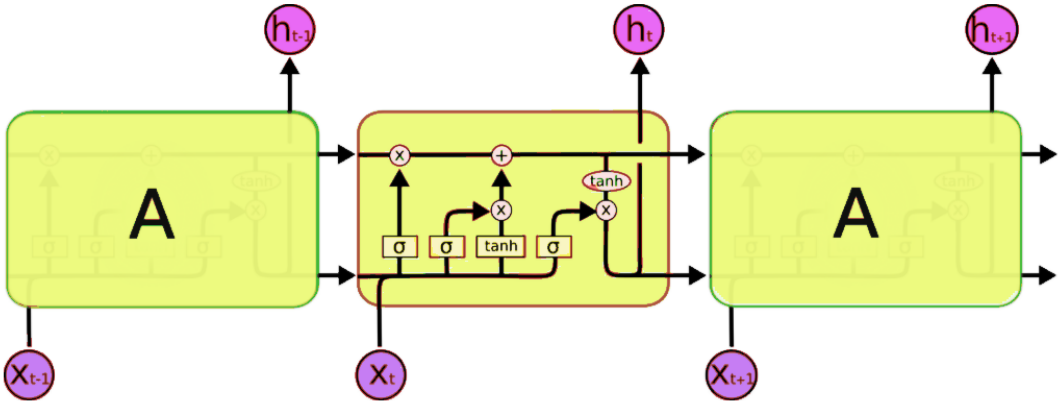

Figure 2.6: The Long-Short Term Memory architecture (Olah 2015).

will be updated, equation 2.17 performs an affine transformation via a non-linearity activation mechanism to generate a vector of new candidate values tanh (Neubig 2017), equation 2.18 computes the memory cell; the equation implements the gradient $\frac{dc_t}{dc_{t-1}}$ equals one, solving the vanishing gradient problem. Equation 2.19 is a sigmoid layer that decides which part of the cell states will output (Sak et al. 2014). Equation 2.20 the memory cell state goes through a non-linear activation tanh and multiply it by the sigmoid gate's output. Where W is the weight matrix, b is the bias vector, σ is

the logical sigmoid function, i_t, f_t, o_t, and c_t are input gate, forget gate, output gate, and memory cell respectively (Sak et al. 2014). An architecture of LSTM can be a variation of BRNN where the RNN architecture is replaced with LSTM and is then called BiLSTM (Goodfellow et al. 2016). BiLSTM has shown the ability to translate natural language into SQL. In recent literature the BiLSTM is used to compute the semantic parsing task of natural language to SQL (Zhong et al. 2017, Xu et al. 2017, Wang, Brockschmidt & Singh 2018, Yu et al. 2018). The input to the BiLSTM is the pretrained GloVe (Pennington et al. 2014) to generate embeddings. (Zhong et al. 2017) uses an LSTM model to translate natural language questions into corresponding SQL queries, building a state-of-the-art called Seq2SQL from natural language question and table schema (Zhong et al. 2017). The baseline of the model developed is the sequence-to-sequence model (Cho et al. 2014). The task is divided into components which are trained separately and concatenated at the end. A BiLSTM is used to encode the input data and a unidirectional two-layer LSTM used as a decoder. The input sentence is a concatenation of the column names required for the selected column and the condition column, the question required for the query condition and the SQL vocabulary such as SELECT, COUNT, MIN, MAX etc. Given an input sequence as x denoted as a sequence.

$$x_J^c = [x_{j,1}^c, x_{j,2}^c, ..., x_{j,T_J}^c] \tag{2.21}$$

the input sequence x is given by

$$x = [< col >; x_1^c; x_2^c; .....; x_N^c; < SQL >; x^s; < question >; x^q] \tag{2.22}$$

where x^q and x^s denote a sequence of words in the question and the set of unique words in the SQL vocabulary respectively. The input sentence is augmented using a pointer network (Vinyals et al. 2015) to generate tokens (Zhong et al. 2017). The SQL query is generated by the pointer network by selecting only from the input. A two-layer BiLSTM decoder network is used in the decoder network. The decoder step s takes an input y_{s-1} the query token generated during the previous decoding step and outputs the state g_s. The decoder then generates a scalar attention score $\alpha_{s,t}^{ptr}$ for each position t of the input sequence:

$$\alpha_{s,t}^{ptr} = W^{prt} tanh(U^{prt} g_s + V^{prt} h_t) \tag{2.23}$$

Seq2SQL further uses Reinforcement Learning (RL) to create the query's conditions; this improved the model's efficiency on the WHERE argument. Figure 2.7 shows the three components of the Seq2SQL model (Zhong et al. 2017). Compare the findings to the attentional sequence-to-sequence neural semantic parser by (Dong & Lapata 2016). Despite not utilising hand-engineered grammars, the seq2SQL obtained state-of-the-art

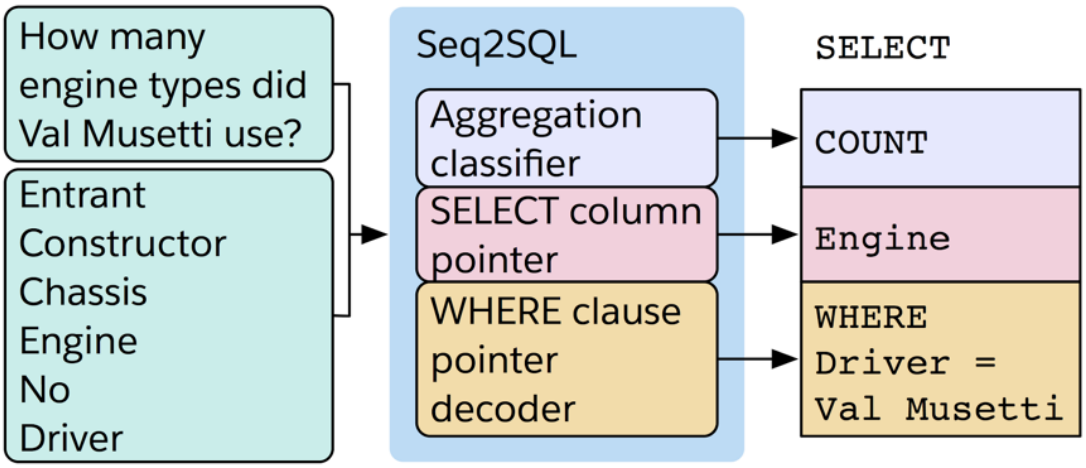

Figure 2.7: The taking as input the question and column names (left), three components and their respective methods, and the output SQL query (right) (Zhong et al. 2017).

outcomes on a variety of semantic parsing datasets, outperforming a number of non-neural semantic parsers (Dong & Lapata 2016, Zhong et al. 2017). Many invalid queries were discovered as a result of column names referring to collection columns that were not included in the table (Zhong et al. 2017).

2.4.2.4 Gated Recurrent Unit

Gated Recurrent Unit (GRU)is an LSTM derivative that modulates information flow within the device without the need for separate memory cells (Chung et al. 2014). They are based on two multiplicative gates expressed in the following equations:

$$r_t = \sigma(W_{xr}x_t + W_{hr}h_{t-1} + b_r) \tag{2.24}$$

$$z_t = \sigma(W_{xz}x_t + W_{hz}h_{t-1} + b_z) \tag{2.25}$$

$$\tilde{h}_t = \tanh(Wx_t + W_{hh}(r_t \odot h_{t-1} + b_h)) \tag{2.26}$$

$$h_t = (1 - z_t)ht - 1 + z_t\tilde{h}_t \tag{2.27}$$

where r_t is a set of reset gate; the reset gate controls what part of the input state in the relationship between past and future states, is used to compute the next target state, adding an additional nonlinear effect (Cho et al. 2014). $\tilde{h}_t$ is a candidate activation, h_t^j is a time t activation function; linear interpolation of previous values activation h_{t-1}^j and the candidate activation $\tilde{h}_t$, z_t^j an update gate that decides the level to which the unit updates its activation and $\odot$ is an element-wise multiplication (Chung et al. 2014). Figure 2.8 presents a graphical representation of the Gated Recurrent Unit architecture.

The most prominent feature shared between the LSTM and the GRU is their update

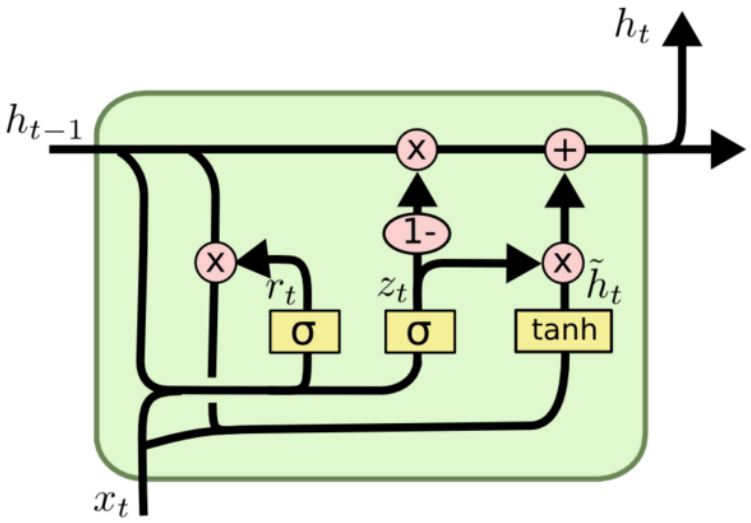

Figure 2.8: An architecture of the Gated Recurrent Unit (Drakos 2019).

additive component from t to $t+1$, which is missing in RNNs. The activation function is often replaced by a new value determined by the current input and the previous hidden state by the RNN unit. At the same time LSTM and GRU both retain the existing content and the new content (Chung et al. 2014).

2.4.2.5 Encoder-Decoder Architectures

The recurrent neural network Encoder-Decoder architecture was created to solve the issue of vanishing gradients in RNNs (Cho et al. 2014). Two RNNs are used in sequence-to-sequence; the first encodes a sequence into a fixed-length vector representation, and the second decodes the representation into a new sequence of symbols (Cho et al. 2014, Sutskever et al. 2014*b*). The conditional probability of the goal sequence is maximized by training the two networks together. Figure 2.9 shows a representation of the Encoder-Decoder architecture, where the encoder takes as input (a, b, c) and the decoder decodes output (w, y, z). The Encoder-Decoder networks are used for sequence of variable length; the encoder reads input of vectors $x = (x_1, ..., x_{T_x})$, into a vector c such that:

$$h_t = f(x_t, h_{t-1}) \; and \; c = q(h_1, \ldots, h_{T_x}) \tag{2.28}$$

where $h_t \in \mathbb{R}^n$ is a hidden state at time t and c is a context vector generated from the sequence of the hidden state of the encoder f and q are some non-linear functions (Bahdanau et al. 2016). To predict the next word, the decoder is used $y_{t'}$, given the context vector c and all the previous predicted words, $y_1, ..., y_{t'-1}$. By decomposing the joint probability into the ordered conditional probability, the decoder determines a

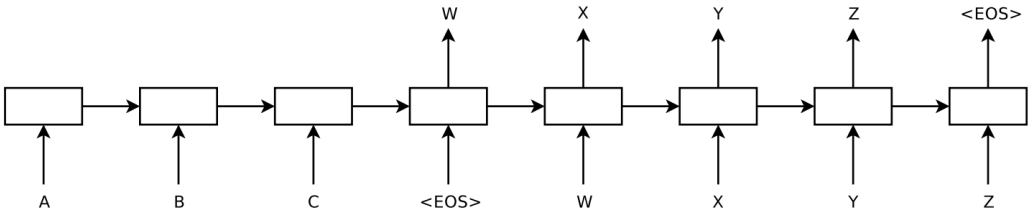

Figure 2.9: An architecture of an Encoder-Decoder model for text translation, taking (a, b, c) as input to the Encoder and the Decoder outputs (w, y, z) (Sutskever et al. 2014*a*).

probability over the translation $\mathbf{y}$:

$$p(\mathbf{y}) = \prod_{t=1}^{T} P(y_t|y_1, ..., y_{t-1}, c) \tag{2.29}$$

where $\mathbf{y} = (y_1, ..., y_{t_y})$ is the output sequence. In an RNN, each conditional probability is modeled as

$$p(y_t|y_1, \ldots, y_{t-1}, c) = g(y_{t-1}, h_t, c) \tag{2.30}$$

where g is a non-linear function that outputs the probability of the next target word, y_t, and h_t is the hidden state of the RNN (Bahdanau et al. 2014).

Dong & Lapata (2016), uses an Encoder-Decoder model to map natural language to their corresponding logical form representation (Cho et al. 2014). Natural languages are encoded into a vector to generate corresponding logical form as sequences. The model uses two different LSTM which processes tokens one by one in a recursive manner. Given natural language input $q = x_1, ..., x_{|q|}$ to a logical form representation of its meaning $a = y_1, ..., y_{|a|}$. The conditional probability is given by $P(a|q)$ decomposed as:

$$P(a|q) = \prod_{t=1}^{|a|} p(y_t|y_{<t}, q) \tag{2.31}$$

where $y_{<t} = y_1, ..., y_{t-1}$. The encoder is used to encode natural language input q into a vector representation and the decoder learns to generate logical form $y_1, ..., y_{|a|}$. Input utterances into vector representations, then condition output sequences or trees on the encoding vectors to produce their logical forms. The new input token's term vector $h_t^0 = W_q e(x_t)$ for the encoder; for the decoder, $h_t^0 = W_a e(y_{t-1})$ is the word vector of the previously predicted word. Where $W_q \in \mathbb{R}^{n \times |V_q|}$ being a parameter matrix, $e(\cdot)$ the index of the corresponding token and $W_a \in \mathbb{R}^{n \times |V_a|}$. Figure 2.10 shows an example of a question translated to its logical form.

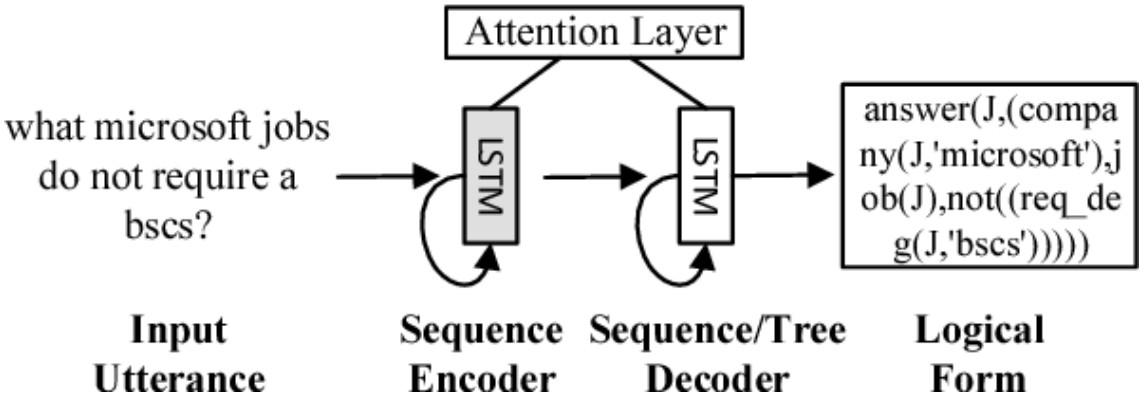

Figure 2.10: Question input and their logical forms, encoded and decoded using LSTM. An attention layer is used to learn soft alignments (Dong & Lapata 2016).

Cai et al. (2017) uses Encoder-Decoder injecting grammar-aware into the memory of decoder and recursive state management for sub-queries. This uses NLTK to implement Conditional Random Fields (CRF) (are a class of statistical modeling method) to annotate the natural language queries. The central concept is to fully incorporate the established SQL grammar structure into the encoder and decoder neural network structures.

Iyer et al. (2017) uses Encoder-Decoder model with global attention (Luong, Pham & Manning 2015) to create SQL queries directly from a natural language text (Cho et al. 2014). The anonymised utterance with the BiLSTM network handles the small number of data points in the datasets by encoding pre-trained word embeddings from Word2vec (Mikolov, Sutskever, Chen, Corrado & Dean 2013). Using a variation of the previous SQL token embedding, the decoder predicts a conditional probability distribution over potential values for the next SQL token provided the previous tokens. The attention is paid to the hidden states by:

$$P(q_i|q_1, ..., q_{i-1}) \propto exp(\mathbf{W} \tanh(\widehat{W}[h_i : c_i])) \tag{2.32}$$

where q_i is an embedding for the ith SQL token, h_i is the hidden state output of the BiLSTM at the ith time-step, c_i is the context vector generated using the attention weighted sum of the encoder hidden state. W and $\widehat{W}$ are linear transformations. The context vector is defined by:

$$c_i = \sum_{j=1}^{k} \alpha_{i,j} \cdot s_j \tag{2.33}$$

where s_j is the hidden state representation by the encoder, attention weight $\alpha_{i,j}$ is computed using the inner product between the decoder hidden state for the current

time-step h_i and the hidden representation of the j_{th} source token s_j :

$$\alpha_{i,j} = \frac{exp(h_i^T F s_j)}{\sum_{j=1}^{k} exp(h_i^T F s_j)} \tag{2.34}$$

where F is a linear transformation, f is the decoder LSTM cell that uses previous hidden and cell states, h_{i-1}, m_{i-1}, the embedding of the previous SQL token q_{i-1} and the context vector of the previous timestep, c_{i-1} to compute the next hidden state h_i and cell state, m_i. Given by $h_i, m_i = f(h_{i-1}, m_{i-1}, q_{i-1}, c_{i-1})$ (Iyer et al. 2017). Sections of the query are replaced with synonyms or paraphrases of the response, and the training range is expanded by incorporating linguistic variants of the input queries. This method has the bonus of being query language agnostic. The disadvantage is that a massive, manually designed training set is required. The feedback learning technique uses semantic parsing to construct a new domain and uses feedback and selective annotation to iteratively refine the parser.

On two benchmark datasets, GEO880 and ATIS, the neural sequence-based approaches that either map from utterances to logical forms or produce SQL are most important to this job (Jia & Liang 2016, Dong & Lapata 2016). A key advantage of Iyer et al. (2017)'s approach is that they are not language-specific and can easily be ported to other commonly used query languages, such as SPARQL or ElasticSearch (Iyer et al. 2017).

2.4.2.6 Attention Mechanism

The traditional sequence-to-sequence paradigm encodes the entire input sentence into a fixed-length vector from which a translation is decoded (Cho et al. 2014). The encoder-decoder solution has the disadvantage that all of the requisite source data is compressed into a fixed-length vector (Cho et al. 2014). Bahdanau et al. (2014) proposed the attention model to counter the challenge by focusing on different words during prediction, retaining and using all hidden states of the input sequence during the decoding process. This method searches for sections of a source sentence that are important to predicting a target word without having to explicitly structure these parts as a hard fragment (Vaswani et al. 2017). The result is a weighted sum of the values, with the consistency function of the question determining the weight assigned to each value with the corresponding key (Vaswani et al. 2017, Cho et al. 2014, Neubig 2017, Luong, Pham & Manning 2015, Yu & Li 2017). Each output that the decoder produces has access to the entire sequence and can select specific elements from the sequence to produce the output (Neubig 2017). Using a series of attention weights, the decoder is told at each decoding stage how much "attention" needs to be paid to each input expression.

The attention weights give the decoder translation background detail. To generate the background vector, all hidden states of the encoder and decoder are used (Neubig 2017). Three factors that calculate the attention layer:

- Alignment score - Is dependent on the hidden state of the previous decoder before predicting the target word, as well as the hidden state of the input word, given by $e_{ij} = a(s_{i-1}, h_j)$. This scores the inputs around position j and the output at position i based on the RNN hidden state s_{i-1}.
- Attention weight - The attention weight is calculated by multiplying the alignment score by a softmax.
- Context vector - Is the weighted sum of the encoder's attention weights and hidden states mapping to the input sequence used to calculate the decoder's final output given by:

$$c_i = \sum_{j=1}^{T_x} \alpha ij h_j \tag{2.35}$$

where αij is the weight of each alignment score and computed by the following equation:

$$\alpha ij = \frac{\exp(e_{ij})}{\sum_{k=1}^{T_x} \exp(e_{ik})} \tag{2.36}$$

Figure 2.11 illustrates an attention model proposed by (Bahdanau et al. 2014). Luong, Pham & Manning (2015) through multiplying the encoder and decoder states, multiplicative attention restricts the encoder and decoder states to attention ratings. Categorising them into two; *Global attention* and *Local attention* models. The two models differ based on where the "attention" is based. The attention can be placed on the entire sequence or part of the sequence (Luong, Pham & Manning 2015).

- **Global Attention**
  - When computing the context vector, consider all hidden states of the encoder.
  - Pay attention to all the location of the source.
  - Through comparing the current target hidden state h_t with the source hidden state, the variable-length alignment vector a_t is extracted, the size of which equals the number of time steps on the source side $\bar{h_s}$:

$$a_t(s) = align(h_t, \bar{h}_s) = \frac{\exp(score(h_t, \bar{h}_s))}{\sum_{s'} \exp(score(h_t, \bar{h}_s))} \tag{2.37}$$

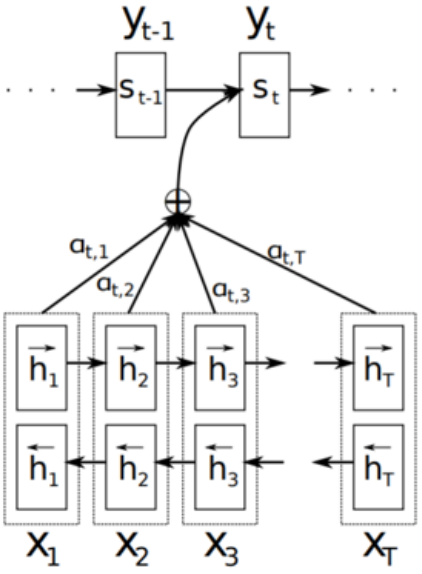

Figure 2.11: Given a source sentence $(x_1, x_2, ..., x_T)$ the $t-th$ target word y_t is generated (Bahdanau et al. 2014)

$$score(h_t, h_s) = \begin{cases} h_t^T h_s & \text{dot} \\ h_t^T h_s & \text{general} \\ v^T \tanh(W_a[h_t; h_s]) & \text{concat} \end{cases} \tag{2.38}$$

Local Attention

- Per target work, local emphasis is attention on a subset of the source series (Luong, Pham & Manning 2015).
- For the current target term, the model first predicts a single matched location p_t. The context vector c_t, a weighted average of the source hidden states in the window, is computed using a window centered on the source location p_t. The actual target state h_t and source state $\bar{h}_s$ in the window are used to calculate the weights at.

The common factor between global and local attention is that both models first take a hidden state h_t at the top layer of a stacking RNN to derive a context vector c_t which captures information to derive the target word y_t. Given a target hidden state h_t and the context vector c_t a concatenation layer combines information from both vectors to produce attention hidden state as follows:

$$\tilde{h}_t = \tanh(W_c[c_t : h_t]) \tag{2.39}$$

The attention vector $\tilde{h}_t$ is fed through the softmax layer to produce the predictive distribution formulated as:

$$P(y_t|y_{<t}, x) = softmax(W_s\tilde{h}_t) \tag{2.40}$$

Figure 2.12 and Figure 2.13 illustrates the Global and Local attention respectively.

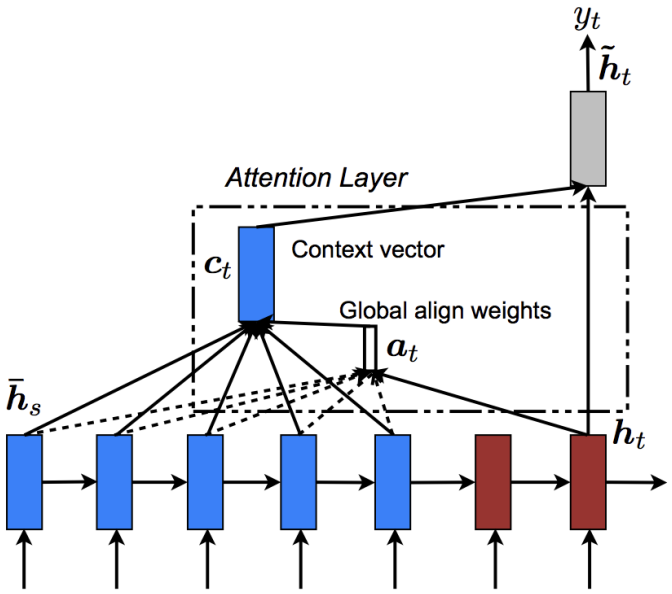

Figure 2.12: The Global Attention (Luong, Pham & Manning 2015)

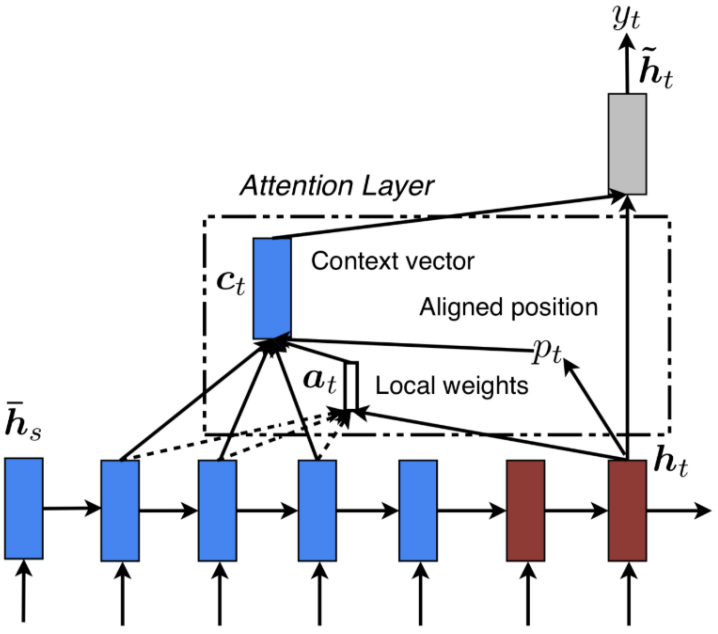

Figure 2.13: The Local Attention (Luong, Pham & Manning 2015)

The goal of both the global and local attention is to derive a context vector c_t from capturing relevant information to help predict the current target word y_t. To improve the performance of translating natural language to SQL (Dong & Lapata 2016) added attention to the encoder-decoder model.
Xu et al. (2017) propose a model called SQLNet that uses sketches to captures the dependency of the prediction. To achieve this, two constraints are proposed: *Sequence & Sets* and *Column Attentions* synthesize the query based on the sketch.

- Sequence & Sets predict unordered constraints determines the most probable column in a query given the question and the table structure. Its embedding is trained by two BiLSTMs one for the question and the other for the column names.
- Column attention captures when forecasting, it computes an attention function between tokens in the questions and column names, then generates the judgment over the column using a two-layer multilayer perceptron. To predict the WHERE-Clause three classifiers are used:
  - **Column slot** - uses multilayer perceptron over the probability $P(col|Q)$ (where *col* is a column name and Q is the natural language question) to decide the number of columns to pick.
  - **OP slot** - uses multilayer perceptron to pick the probable operators i.e $(=, <, >)$
  - **Value slot** - predicts the substring from the natural language question by using a BiLSTM and a pointer network with a column attention mechanism.

SQLNet when compared to the Seq2SQL model using the same dataset, reveals a substantial increase in predicting the WHERE-Clause, suggesting that the method used by (Xu et al. 2017) of employing a sequence & set model is capable of dealing with the "order matter" problem. Yu et al. (2018) proposed using a sketch-based approach called TypeSQL, where the generation of SQL queries is regarded as a slot filling challenge, with various slots being grouped together and relationships between attributes being captured. By processing query inputs with type recognition, tokenising each question into n-grams of length 2 to 6, and using them to scan through the table schema and mark each column name that appears in the question as COLUMN, this uses type knowledge to better understand uncommon entities and numbers in natural language queries.This employs two BiLSTMs to encode terms in the query with their forms and column names separately, categorizing numbers and dates into four self-explanatory categories: INTEGER, FLOAT, DATE, and YEAR. Five forms of entities were used to classify named entities in the dataset: PERSON, PLACE, COUNTRY, ORGANIZATION, and SPORT. These categories cover the bulk of the entities in the dataset. If

database content is available words are matched in the query with both the table schema and the columns' content and marks as COLUMN, and match the entry values as the corresponding column names. Although the order of column names does not matter, TypeSQL solves the dilemma that SQLNET often has of choosing the same column in the situation slot as in the select column. The change is due to the fact that the LSTM can catch their occurrences and relationships, and also shows that accessing the content of databases can greatly boost the performance.
The success of deep learning architectures is seen in tasks computer vision (Krizhevsky et al. 2012) in Imagenet classification with deep convolutional neural networks (CNN), in speech recognition (Noda et al. 2015, Yu et al. 2014), and semantic parsing tasks such (Zhong et al. 2017, Dong & Lapata 2016, Xu & Du 2019). Though efficient, neural network methods have a high entry barrier because they demand large datasets for a variety of reasons. The advent of the internet has seen a growth in the amount of data generated, affording deep learning models to be trained.

2.4.2.7 Pointer Network

When using variable performance dictionaries, the attention mechanism does not explicitly answer issues that occur, resulting in a missed forecast when the vocabulary is not present (Vinyals et al. 2015). Vinyals et al. (2015) proposed the **Pointer network** that uses the attention mechanism as a pointer to select a member of the input sequence as the output by calculating the sequence's conditional probability, solving the problem of variable size output dictionaries. The softmax function normalises the vector to be an output distribution over the dictionary of the inputs using attention mechanism given the equation probability $P(C_i|C_1, ..., C_{i-1}, P)$ using an attention mechanism as follows:

$$u_j^i = v^T \tanh(W_1 e_j + W_2 d_j) \qquad j \in (1, ..., n) \tag{2.41}$$

$$P(C_i|C_1, ..., C_{i-1}, P) = softmax(u^i) \tag{2.42}$$

where softmax normalises the vector u_i output distribution over the dictionary of inputs, and v, W_1, and W_2 are learnable metrices of the output model (Vinyals et al. 2015). Pointing aids at accurately reproducing information from the input giving an advantage of solving the problem with variable length and producing out-of-vocabulary words (See et al. 2017). On top of designing the translation of natural language to SQL, Zhong et al. (2017) use a pointer network on the WHERE-Clause to reproduce the WHERE-Clause condition and take from the input. Wang, Brockschmidt & Singh (2018) extended the research done by (Zhong et al. 2017) with an Encoder-Decoder model using the SQL query structure to statistically determine the type of output of a decoding step while

generating the SQL query. A BiLSTM is used as the encoder inputting n-gram and GloVe embeddings (Pennington et al. 2014). The model use types abstracted from the grammar of the target language to guide the decoder to either copy a token from the input using a pointer-based copying mechanism or generate a token from a finite vocabulary. Sum-transfer value-based loss function transforms a distribution over pointer locations into a distribution over token values in the input because multiple columns with similar names exist. Three types are described as follows:

- τv - The output is a token from the terminal $v = \{$Select, From, Where, Id, Max, Min, Count, Sum, Avg, And, $=, \leq, \geq, <, >, < END >, < GO >\}$ of grammar.
- τc - The output is a column name, which is copied from either the table header or the question section of X.
- τq -The output is a constant, which is copied from the query.

Wang, Brockschmidt & Singh (2018) found that cases with wrong predicates are selecting a wrong column where cases are typically caused by the correct column name not being mentioned in the question. Wang, Brockschmidt & Singh (2018) noted errors suggest that the model lacks understanding of the knowledge presented in the table and that embedding an extra decoding layer for constant rewriting could improve the Model's performance. Wang, Brockschmidt & Singh (2018) outperformed the augmented pointer model and Seq2SQL model (with RL) of (Zhong et al. 2017). A variation of pointer network, **Pointer generator network**, allows for the copying of words using a pointer network as well as the creation of words from a set vocabulary. The pointer generator network assists in reproducing knowledge correctly while maintaining the ability to generate new terms through the generator (See et al. 2017). A coverage keeps track of what has been outputted discourages repetition, this calculates the context vector as pointer network and the attention distribution, additionally calculates the generation probability from the context vector for time step t, the decoder state s_t, and s the input x_t decoder:

$$p_{gen} = \sigma(w_h^T \dots c_t + w_s^T s_t + w_x^T x_t + b_{ptr}) \tag{2.43}$$

where w_h^T, w_s^T, w_x^T are learnable metrices, σ is the sigmoid function. p_{gen} is used as a soft switch to choose between generating a word from the vocabulary by sampling from the vocabulary distribution or copying a word from the input sequence by sampling from the attention distribution (See et al. 2017). Pointing generator improves accuracy and the handling of out-of-vocabulary words while retaining the ability to generate new words (See et al. 2017). Lukovnikov et al. (2018) Shows to be effective for eliminating repetition is similar to (Gu et al. 2016) which is often a problem for sequence-to-sequence

models. Lukovnikov et al. (2018) uses pointer generator network to and investigate the order matters problem in semantic parsing for SQL (Lukovnikov et al. 2018).

2.4.2.8 Training and Optimisation for Neural Networks

Loss Function

The loss function in neural networks is used to evaluate how a set of parameters behaves for its actual ground truth in the training data (Goodfellow et al. 2016). The loss function in neural networks is a function key to adjusting the weights of the network, creating a better fitting to the algorithms upon computation. This can be thought of as calculating the total entropy between two probability distributions for a given random dataset. The cross-entropy can be given by entropy and Kullback-Heibler∼(KL) measure of variation between two distributions given by the below equation:

$$\mathbf{H}(p,q) = \mathbf{H}_p(q) - \mathbf{H}(q) = \sum_{i=1}^{N} q(y_n) \cdot [\log p(q(y_n)) - \log p(p(y_n))] \tag{2.44}$$

where p and q are two probability distributions (Goodfellow et al. 2016). Each weight is updated by a sum equal to the partial derivative involving the weights to reduce the error (Nasr et al. 2002). The neural network operates on the testing dataset during forward propagation, and the output probabilities are compared to the target labels (Ho & Wookey 2019). Any divergence from the goal label and the neural network's outputs is penalised by the loss function (Ho & Wookey 2019).

Cross-Entropy Loss

The cross-entropy loss is a log-linear that produces a multi-class classifier distribution over the possible labels by optimising the weights, minimising the negative log probabilities (Nasr et al. 2002, Glorot & Bengio 2010, Nielsen 2015, Goodfellow et al. 2016, Goldberg 2017). The cross-entropy loss builds upon entropy by calculating the difference between two probability distributions (Glorot & Bengio 2010, Nielsen 2015, Goldberg 2017, Ho & Wookey 2019). This is given by:

$$E = -\sum_{i}^{nclasses} t_i \log y_i \tag{2.45}$$

where y_i is the softmax function over the input, t_i is the target output. The error's derivative is computed for each weight connecting the hidden units to the output units using the chain rule (Sadowski 2016). As a result, the error propagating back from each

output unit becomes proportional to the discrepancy between the target and real value, resulting in improved network efficiency and a shorter stagnation time (Nasr et al. 2002, Sadowski 2016).

Gradients

The goal of optimisation in deep learning is to minimise the loss function to achieve the best performance measure of how well a given model can learn to accurately map input examples to its outputs (Bengio 2012). In gradient-based learning, the parameters are updated according to the update rule of minimising the loss function (Bengio 2012, Goldberg 2017). There are two common variants of gradient-based learning: *Gradient Descent* (GD) and *Stochastic Gradient Descent* (SGD).

Gradient Descent (GD) are algorithms used to obtain parameters that minimises the value of loss function in neural networks by computing parameter updates for gradient-based numerical optimisation algorithms (Bengio 2012). The gradient-based techniques work by repeatedly calculating a loss estimate over the training set, calculating the parameter gradients concerning the loss estimate, and moving the parameters in the opposite direction of the gradient set (Goldberg 2017). Given a data pair (x, y), the gradient estimate is computed by taking the average gradient on all pairs of examples in the training set (Goodfellow et al. 2016). The gradient descent is computed by:

$$GD = \frac{1}{n} \sum_{i=1}^{n} \nabla_{\Theta} \mathcal{L}(x_i, y_i, \theta_t) \tag{2.46}$$

where n is total examples in the training set, $\mathcal{L}$ is the loss function to minimise, $\nabla_{\Theta}\mathcal{L}$ is the gradient, and the vector of partial derivatives of $\mathcal{L}$ with respect to θ_t. The gradient estimate is updated by the below equation:

$$\theta_{t+1} = \theta_t - \eta GD \tag{2.47}$$

where η is the learning rate; a value that control how much the parameter are allowed to follow the opposite direction of the gradient estimate (Goodfellow et al. 2016). The average of all n examples is used to compute the gradient estimate, which is computationally expensive. Neural networks have their limited capacity to learn based on the training data size (Goldberg 2017). Intuition suggests that if the dataset size is extremely small, the neural networks will struggle to learn the intricate pattern of the data. To avoid this an extension stochastic gradient algorithm computes gradient estimate using mini-batch (Goodfellow et al. 2016, Goldberg 2017).

Stochastic Gradient Descent (SGD) attempts to reduce computation per iteration at the expense of increasing the amount of iterations required for convergence where

the gradient has reached a point where only minor improvements in the loss function are made (Goodfellow et al. 2016). This is accomplished by sampling a testing example several times and calculating the gradient of the error on the example with respect to the parameters (Goodfellow et al. 2016, Goldberg 2017). The SGD performs a parameter θ update for each training example $x_{(i)}$ and label $y_{(i)}$:

$$\theta = \theta - \eta \cdot \nabla\theta\mathcal{L}(\theta; x_{(i)}; y_{(i)}) \tag{2.48}$$

The advantage of SGD is that the gradient estimate is computed only from the examples in mini-batch (see section 2.4.2.8); the training set does not affect the computation time (Goodfellow et al. 2016). The gradient computation, when evaluated in x, where x is chosen uniformly at any iteration t is given by:

$$SGD(x) = \frac{1}{n}\sum_{i=1}^{n} \nabla_\Theta \mathcal{L}(x, y, \theta_t) \tag{2.49}$$

The expectation $\mathbb{E}$ is given by:

$$\begin{aligned} \mathbb{E}[SGD] &= \frac{1}{n}\sum_{i=1}^{n} SGD(x_i) \\ &= \frac{1}{n}\sum_{i=1}^{n} \nabla_\Theta \mathcal{L}(x_i, y_i, \theta_t) \\ &= \sum_{i=1}^{n} \frac{1}{n} \nabla_\Theta \mathcal{L}(x_i, y_i, \theta_t) \\ &= GD \end{aligned} \tag{2.50}$$

An important feature of SGD and associated mini-batch is that with the number of training examples, the computation time per update does not increase, allowing convergence even when the number of training examples is very high (Goodfellow et al. 2016). SGD may converge to a fixed tolerance of its final test set error before all the training set are processed within a large dataset (Goodfellow et al. 2016, Goldberg 2017).

Adaptive Moment Estimation

Adaptive learning rate approaches have been used successfully in a variety of areas, including deep learning algorithm training (Zhong et al. 2020). Adaptive Moment Estimation (Adam) is a gradient-based first-order optimisation of stochastic objective function that calculates each parameter's individual adaptive learning rates (Kingma & Ba 2014). The method combines the advantages of two recently popular optimisation methods: the ability of AdaGrad (Duchi et al. 2011) to deal with sparse gradients, and the ability of

RMSProp (Tieleman & Hinton 2012) to deal with non-stationary objectives (Kingma & Ba 2014, Goodfellow et al. 2016). The algorithm updates the gradient's exponential average and the squared gradient (Goodfellow et al. 2016). Adam has an essential update rule whereby each step-size is carefully chosen for each parameter. The momentum is created by applying fractions of the previous update to the new update, resulting in a compound effect of repeated changes in the same direction (Goodfellow et al. 2016). The magnitudes of parameter changes are invariant of rescaling the gradient, which is one of the key benefits. It does not need a stationary target since its step-sizes are roughly bounded by the step-size hyperparameter (Kingma & Ba 2014). Adam is the common choice of optimisation in many deep learning, for translation of natural language to SQL uses Adam (Zhong et al. 2017, Xu et al. 2017). These methods can be applied to many training tasks, such as text recognition (Yin et al. 2018), image ranking (Krizhevsky et al. 2012), online education (Su et al. 2018). Especially for tasks with sparse data deep neural networks, multi-layer perceptron, convolutional neural networks (Krizhevsky et al. 2012), and semantic parsing (Zhong et al. 2017).

Activation Function

To convert an input signal into an output signal, which is then fed as input to the next layer in the stack, neural networks require activation functions. This is done by performing a non-linear transformation to the input training dataset (Goldberg 2017). Activation functions determine the output, accuracy, and computational effectiveness of a training model, which can make or break a large-scale neural network. Different types of activation function are used for different set of neural network problems. Four popular activation functions are described below as follows:

- The **Rectified Linear unit (ReLU)** represents a nearly linear function which preserves the properties of linear models (Nair & Hinton 2010, Glorot et al. 2011, Goldberg 2017). ReLU is a neural network activation function that performs a threshold operation for each input variable where the supplied values are less than zero:

$$f(x) = max(0, x) = \begin{cases} x_i & \text{if} \quad x_i \geq 0 \\ 0 & \text{if} \quad x < 0 \end{cases} \tag{2.51}$$

- The **Sigmoid function** (Goldberg 2017) also known as the standard logistic function is defined as:

$$S(x) = \frac{e^x}{1 + e^x} = \frac{1}{1 + e^{-x}}, \quad X \in (-\infty, \infty) \tag{2.52}$$

for all $X \in (-\infty, \infty)$, $S(x)$ is increasing on $(-\infty, \infty)$, acting as an activation function at the output of each neuron.

- The **Tanh function** (Goldberg 2017) is also sigmoidal "s"-shaped, which outputs values that the equation gives range $(-1, 1)$:

$$Tanh(x) = \frac{\sinh(x)}{\cosh(x)} = \frac{e^x - e^{(-x)}}{e^x + e^{-x}} \tag{2.53}$$

- The **Softmax function** is an activation used to compute multi-class classification of neural networks, generating an output ranging from 0 to 1 that normalises the input values into a vector following distribution of probabilities (Goldberg 2017). The softmax function is often used in the final layer of a neural network classifier. Semantic parsing problems are multi-class classification type of tasks that often use softmax function to normalise the input values giving a non-linear variant of multinomial logistic regression (Zhong et al. 2017). The discrete probability distribution for N classes, given by:

$$f(x_i) = \frac{\exp^{(x_i)}}{\sum_{j=1}^{N} \exp^{(x_j)}}, (i = 1, 2, 3, ..., N) \tag{2.54}$$

where $x_1, x_2, x_3, ..., x_n$ are the input values of the softmax layer and $f(x_i)$, the output represents the probability that the sample belongs to the ith category (Wang, Lu, Zhu, Lin & Wang 2018).

Batch and Mini-batch

When training neural network algorithms using the whole set of data it is called *batch* or *deterministic*. All the training examples are processed in a large batch simultaneously which provides less than linear return with a more precise gradient estimate (Goodfellow et al. 2016). A batch often degrades the model's quality as measured by its capacity to generalise (Keskar et al. 2016). Some algorithms are more susceptible than others to noise, either because they use data with few samples that are hard to accurately estimate, or they use data in ways that sample noise (Goodfellow et al. 2016, Keskar et al. 2016). An effective way to deal with the issue of stochastic noise and degradation of the model is to employ a *mini-batch* sampling approach (Yang et al. 2019). Mini batches may have a regularising effect due to the noise they introduce into the learning process, which may necessitate a low rate of learning to preserve stability due to the high variance of gradient calculation (Bengio 2012, Ioffe & Szegedy 2015, Goodfellow et al. 2016, Bottou et al. 2018). The mini-batch iteration are updated based on the average gradient with respect to multiple data points rather than on the gradients calculated on a single data

reducing computational costs by orders of magnitude compared to the batch algorithms, while at the same time yielding significantly better solutions than online SGD (Yang et al. 2019). An independent estimation of the predicted gradient from a series of samples can be determined using randomly chosen samples, giving the gradient samples independence (Goodfellow et al. 2016). Through contrasting training curves after the other hyper-parameters have been chosen, the mini-batch effect training period and not so much test results can be optimised independently from the other hyper-parameters (Bengio 2012, Masters & Luschi 2018).

2.4.2.9 Neural Network Hyper-parameters

Training deep learning models involves different sets of learning parameters which influence the training process and performance called hyperparameters (Pedregosa 2016). Choosing the best hyper-parameters is both a crucial and difficult task as each learning algorithm involves different sets of hyper-parameters and has often shown an influence in model accuracy.

Hyper-parameters optimisation techniques

- Learning rate: The learning rate controls the neural network's weights with respect to the loss gradient by updating the neural network the concepts it has learned. A optimal learning rate is one that is low enough for the network to converge but high enough for the network to be educated in a suitable period of time (Pedregosa 2016). Since smaller adjustments to the weights in each update are required by lower learning rates, further training epochs are required. If the learning rate of the model is too low, it can take a much longer period to reach ideal convergence (Pedregosa 2016). Larger learning rate, on the other hand, result in faster transitions which entail less training epochs (Pedregosa 2016). The ideal state would be overpowered if the learning rate is much higher than the desired value, and the algorithm does not converge (Pedregosa 2016). The choice of the value for learning rate has an impact on how fast the algorithm learns and whether the loss function is minimised or not (Pedregosa 2016).

- Training iteration: Iteration is the number of batches needed to complete one epoch, where a full dataset is passed multiple times to the same neural network to learn from the input. Updating the weights with a single pass or one epoch is not enough because neural networks needs to learn repeatedly, using the input dataset and learned results from the previous trials. Increasing the number of epochs doesn't always mean that the network will give better results (Bengio 2012). The

number of epoch used can be determined by **early stopping** using the principle of early stopping. It can be optimised by keeping track of the out-of-sample error as training progress (Bengio 2012). Early stopping is an inexpensive way to avoid heavy overfitting (Bengio 2012).

Model Hyperparamerisation

- Hidden nodes: The number of neurons in the hidden nodes is a critical component of the overall neural network architecture. A multi-layer neural network's capability is regulated by the scale of each layer. The model's learning ability is primarily measured by the amount of hidden nodes (Panchal & Panchal 2014). The more complex the model, the more capacity the model will need for learning and a large number than necessary may cause overfitting (Bengio 2012). Hidden nodes do not interfere specifically with the outside world, but they do have an effect on the final result (Panchal & Panchal 2014).
- weight decay: Weight regularisation is a technique for reducing overfitting in a deep learning neural network model on training data and improving the model's accuracy on new data, such as the holdout test collection (Bengio 2012).

2.4.3 Regularisation

The goal of training neural network algorithms is to have a good performance on both the training and testing dataset. However, neural networks are prone to overfitting, where algorithms perform too precisely on the training dataset, and performs poorly during testing when presented with a new dataset (Goodfellow et al. 2016). This is because the model learns the dataset including the noise associated with training it (Hinton et al. 2012, Goodfellow et al. 2016). There are several strategies used to reduce overfitting during training collectively known as *regularisation*. Regularisation helps neural network models achieve better generalisation. These include *L1, L2 regularisation, early stopping, and dropout* to name a few (Sutskever et al. 2014*b*, Goodfellow et al. 2016).

2.4.3.1 Dropout

Dropout is one of the several approaches to compact overfitting, that simulates parallel training of a large number of neural networks with various architectures (Goodfellow et al. 2016, Srivastava et al. 2014). In neural network architecture, dropping means momentarily separating the layers from the network, as well as the incoming and outgoing links (Goodfellow et al. 2016). Overfitting can be significantly minimised by randomly

omitting half of the feature detectors in each training event, eliminating co-adaptations in which a feature detector is only useful in the light of many other specific feature detectors (Hinton et al. 2012, Goodfellow et al. 2016). Each unit has a fixed probability of being preserved, regardless of the layers. Given a training data pair (x, y) on Figure 2.14, a representation of an over-fitted model is given. Zhong et al. (2017) uses dropout to regulate overfitting when training their model. Figure 2.15 shows the temporary removal of networks. The orange neurons are temporarily removed.

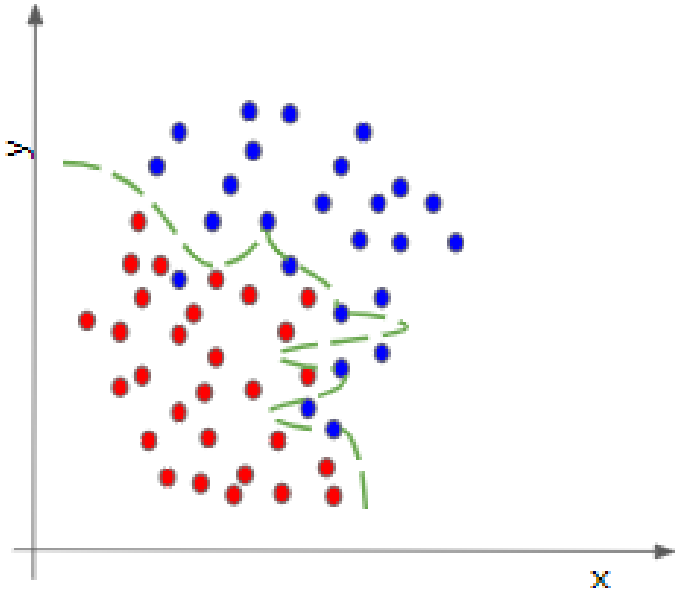

Figure 2.14: When a training dataset perfectly fits the model during training. An example of overfitted dataset (Goodfellow et al. 2016).

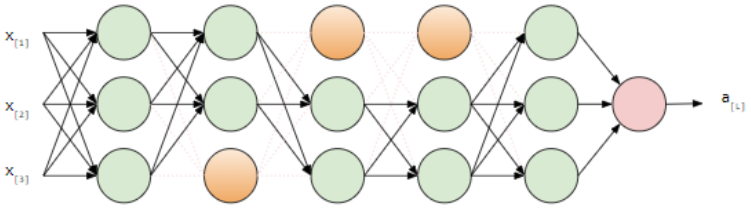

Figure 2.15: A deep neural network with dropped neurons represented by the Orange neuron (Srivastava et al. 2014).

2.4.3.2 Performance Measures

Accuracy Measures and Validation

It is important to consider how a neural network algorithm behaves on new data that was not seen during training. The testing process assesses how well the model performs on unseen data. To validate the correctness of the output predictions several evaluation

measures are used, this is to check the accuracy of the model against the ground truth . The following metrices are used on multi-class classification methods; these metrices are commonly used in many classification problems including translation such as natural language to SQL.

- **Precision**: Calculates a percentage of positive predictions that were correct, to find all proportion of the model that the model says is relevant and are relevant. Given by the equation below:

$$Precision = \frac{TruePositive}{TruePositive + FalsePositive} \tag{2.55}$$

- **Recall**: Calculates the true positive rate of the model performance to find all data points interest in the dataset. Given by the equation below:

$$Recall = \frac{TruePositive}{TruePositive + FalseNegative} \tag{2.56}$$

- **F1-Score**: Calculates the weighted average of Precision and Recall. This score takes both false positives and false negatives into account. Given by the equation below:

$$F1 = 2 \times \frac{precision * recall}{precision + recall} \tag{2.57}$$

- **BLEU-Score**: Bilingual Evaluation Understudy~(BLEU) score which is an algorithm for evaluating the quality of text which has been machine-translated from one natural language to another (Papineni et al. 2002). BLEU score is given by:

$$BLEU = BP.exp(\sum_{n=1}^{N} w_n log p_n) \tag{2.58}$$

where p_n is the n-gram precision, w_n is the weight summing to one and brevity penalty (BP) is given by:

$$\begin{cases} 1 & \text{if} \quad c > r \\ e^{(1-\frac{1}{c})} & \text{if} \quad c \leq r \end{cases} \tag{2.59}$$

where c is the length of the candidate translation and r the effective reference of the corpus length.

- **Logical Form Accuracy**: The exact string match between the SQL queries produced and the ground truth SQL queries is measured as a percentage. Since the same conclusion can be obtained using various logical types (Zhong et al. 2017).

$$ACC_{LF} = \frac{number\ of\ SQL\ with\ correct\ logical\ form}{total\ number\ of\ SQL} \tag{2.60}$$

- **Execution Accuracy**∼(Accex): Is the percentage of SQL queries created that return a correct answer. This is the percentage of examples for which the model produced a query that, when executed, produces the same set of results as the ground truth query (Zhong et al. 2017).

$$ACC_{EX} = \frac{number\ of\ SQL\ with\ correct\ execution\ results}{total\ number\ of\ SQL} \tag{2.61}$$

2.5 Summary

This chapter details literature on traditional and more advanced semantic parsing methods, a highlight of traditional methods that implemented translation using statistical and machine translation is extensively explained. A discussion on new approaches to semantic parsing that is, neural semantic parsing is provided with exhaustive literature on state-of-the-art translation methods of natural language to SQL. A further discussion on neural networks training and optimisation is provided. Lastly, a discussion on the evaluation of the neural network models. The next chapter outlines the methodology carried out for this research.

Chapter 3

Methodology

3.1 Introduction

The previous chapter presented the background on natural language processing, semantic parsing, and the current literature relating to the translation of natural language to SQL using deep learning architectures. This chapter gives details of the methodology set out to address the research objective stated in Chapter 1.3, to comparatively evaluate sequence-to-sequence architectures. Specifically, this chapter presents an overall pipeline of the implementation of the research, comprehensive details of the dataset, pre-processing techniques, architectural implementation, software, and hardware used to achieve the experimental set-up of the research. The results of the research are presented in the next chapter for all the architectures presented.

3.2 Research Approach

An extensive literature on neural semantic parsing highlights the success of deep learning architectures to translate natural language to SQL with techniques that are less reliant techniques, learning on an end-to-end training process with input representation by tokens in vector space (Zhong et al. 2017). As specified, the support of richer linguistic diversity in query expressions is a benefit of neural semantic parsing approaches over conventional semantic parsing approaches. Thus users can formulate queries with greater flexibility. The main important step in neural semantic parsing is the representation of data in a form where the neural architectures can learn and infer. Depending on the task to solve, several approaches are used to transform the dataset to a representation suitable to train deep learning architectures. Part of the pre-processing steps for

these neural semantic parsing is to use natural language processing tools such as tokenisation process where sentences are split into word level without losing any constructive information. This is followed by using techniques such as word embedding, representing individual words as real-valued vectors in a lower-dimensional space (Mikolov, Sutskever, Chen, Corrado & Dean 2013, Pennington et al. 2014). The use of distributional representation has been successful in the processing dataset, representing the data in a word level, dense, and plotting words based on semantic feature values. One of the successful word representation in recent literature for the translation of natural language to SQL is pre-trained word embeddings such as GloVe (Pennington et al. 2014) where words are represented with their co-occurrence to obtain the word vectors. As stated in Chapter 2 there are several embedding approaches from word to document levels that can similarly present words in a form that can be processed by deep learning architectures, boosting generalisation in computation. One of the major challenges of using deep learning approaches is that they require a large training dataset to achieve desired results, the current WikiSQL dataset has shown to be a large enough corpus to conduct training using deep learning architectures. The main aim of this work is to comparatively evaluate the performance of the deep learning architecture specifically the sequence-to-sequence models for the translation of natural language to SQL, this is by addressing the objectives outlined in Chapter 1.

3.2.1 Research pipeline

Figure 3.1 presents the research pipeline that will be followed to achieve the research objective, from dataset acquisition through to evaluation.

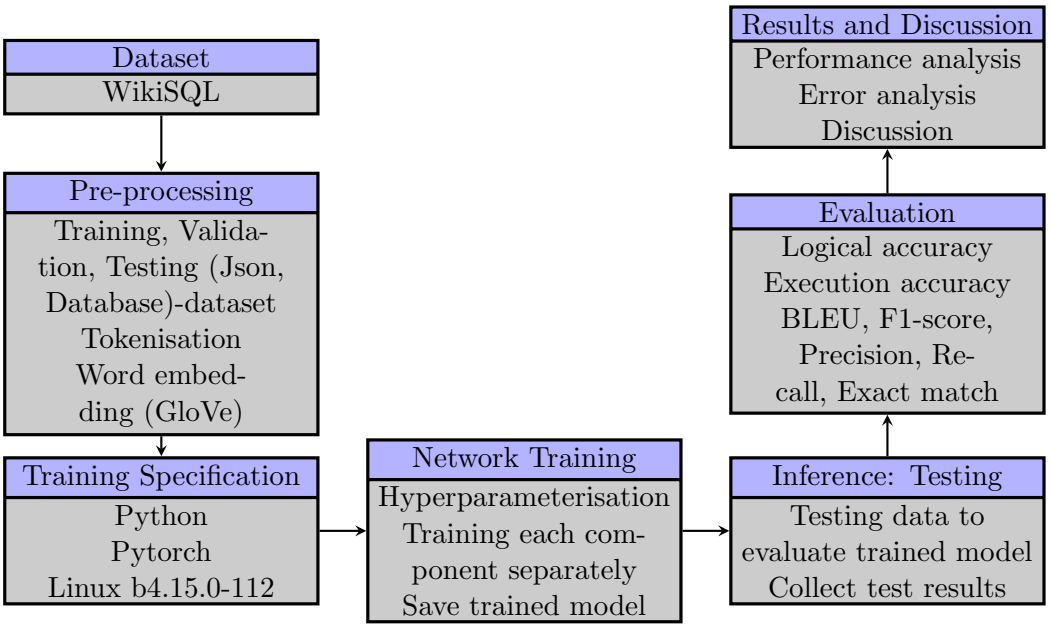

Figure 3.1: The research pipeline overview for the undertaken research, all the models follow same pipeline to achieve comparative results. The pipeline covers Chapter 3 through Chapter 4

Sequence-to-sequence models are investigated using the WikiSQL dataset. The pipeline through the pre-processed, evaluation of sequence-to-sequence models on different sets of hyper-parameters for each model, testing the models on unseen test dataset dataset, evaluating the prediction on different sets of evaluation metrics, and finally performing an error analysis on the predicted output.

3.2.2 The WikiSQL Dataset

The training process of deep learning architectures requires a very large and well annotated dataset. The WikiSQL dataset (Zhong et al. 2017) achieves that, it is currently the largest hand-annotated dataset consisting of 80654 questions and SQL pairs examples distributed across 24241 tables. This dataset includes natural language questions, table_ids, annotated SQL logical form and data for each table. It is a collection of Wikipedia information by crowd-sourcing on Amazon Mechanical Turk. In its current state, it consists of only simple SQL components, i.e SELECT, AGGREGATION, and WHERE-Clause represented as follows:

- Phase: Collection and splitting of the dataset.
- Question: The natural language questions.
- Table_id: The descriptor of the table from which the question is derived.

- SQL: The SQL query with subfields that corresponds to the topic.
  - Sel: The column's numerical index that is being chosen.
  - Agg: The numerical index of the currently used aggregation operator.
  - Conds: A list of triplets (column_index, operator_index, condition).
  - Column_index: The numerical index of the condition column that is being used.
  - Operator_index: The numerical index of the condition operator that is being used.
  - Condition: The string value of the condition.

Figure 3.2 shows how the queries are distributed through the dataset, and Figure 3.3 shows how the topic, query, and table are distributed throughout the dataset.

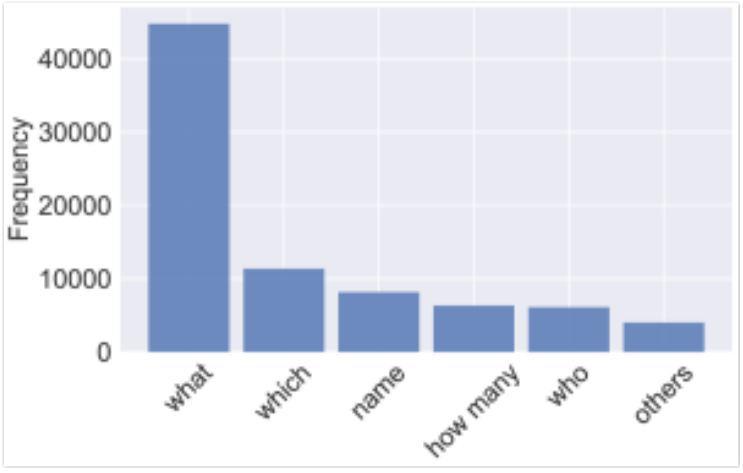

Figure 3.2: The questions distribution of the WikiSQL dataset (Zhong et al. 2017).

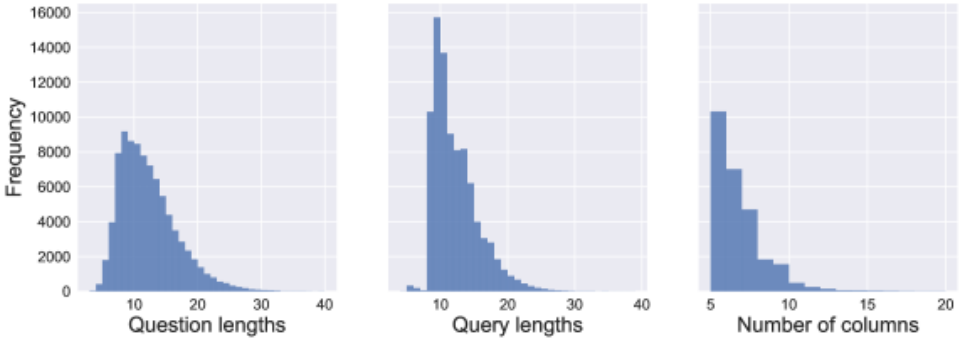

Figure 3.3: The distribution of question, query and table from the WikiSQL dataset (Zhong et al. 2017).

Five input knowledge groups that describe the questions are described by (Yavuz et al. 2018).

- **Exact match**: Each SQL query condition's column name appears in the vicinity issue of its condition value with the same surface form.
- **Paraphrase**: Column name is paraphrased for at least one of the SQL query conditions. Thus, certain paraphrasing capabilities are needed for inference.
- **Partial clue**: For at least one of the SQL query condition, there is no clear mention of its column name in the question, not even in paraphrased form. Limited semantic hints for inference, however, remain.
- **External knowledge**: For at least one of the SQL query condition, there's no clue to infer its column name in the question. To infer this column name from the question needs external information about the form of its condition value which can be identified from the query.
- **Ambiguous**: Although with external expertise, is almost impossible (even for humans) to reliably conclude the right state column from the query in this group..

Table 3.1 shows the examples of WikiSQL question and column header distribution. The model synthesises the SQL query under the condition that the table queried from

Table 3.1: Examples of WikiSQL question and column name over the distribution of knowledge group (Yavuz et al. 2018)

Category	Question	SQL Query	Column names
Exact match	In what state was the electorate fowler?	SELECT state WHERE electorate EQL fowler	member, party, state, electorate, erm in office
Paraphrase	What was the date of the game after week 5 against the Houston Oilers?	SELECT date WHERE week GT 5 AND opponent EQL houston oilers	week, date, opponent, result, attendance
Partial clue	Who had the most points in the game on March 7?	SELECT high points WHERE date EQL march 7	high points, game, date, team, score, ...
External knowledge	Name the callback date for amway arena	SELECT callback date WHERE audition venue EQL amway arena	audition venue, audition city, callback date, ...
Ambiguous	List the branding for krca-tv	SELECT branding WHERE callsign EQL krca-tv	branding, power (kw), callsign, channel, ...

is known. The SQL has a fixed structure with rules that govern information retrieval,

a typical SQL structure is written as: [SELECT COLUMN] [AGG] FROM TABLE [WHERE COLUMN2 EQUALS VALUE] [AND COLUMN3 EQUALS VALUE].

- AGGREGATION-retrieves the aggregation keywords of the information, which tells you the quantity of the information needed (e.g.,‘ ’, MAX, MIN, COUNT, SUM and AVG).
- SELECT-retrieves the table column names.
- WHERE-Clause-retrieves the condition under which the SQL query must pass to retrieve the information. Which consists of the column name, operation, and the condition under which information is synthesised.

Table 3.2 presents three different sets of question-query pairs. The first question-query is straight forward, does not contain an aggregation or multiple conditions. The second question-query consists of an aggregation and the third question-query consists of multiple conditions. Table 3.3 shows the three components of the SQL query, an optional

Table 3.2: Question and Query

Question	SQL
Who won the game where the Challenge Leader is ACC (2-1)?	SELECT winner WHERE challenge leader EQL acc (2-1)
How many losses did the Michigan State Spartans have	SELECT COUNT(loss) WHERE institution EQL michigan state spartans.
Which year's crew is varsity 8+ when the record is 7-3	SELECT year WHERE crew EQL varsity 8+ AND record EQL 7-3.

AGGREGATION, SELECT, and WHERE-Clause of conditions.

Table 3.3: The SQL query structure

SQL	AGGREGATION	SELECT	WHERE-Clause
SELECT winner WHERE challenge leader EQL acc(2-1)	None	Winner	challenge leader EQL acc(2-1)
SELECT max(loss) WHERE institution EQL michigan state spartans	max	loss	institution EQL michigan state spartans
SELECT year WHERE crew EQL varsity 8+ AND record EQL 7-3	none	Year	crew EQL varsity 8+ AND record EQL 7-3

Given a question: **What is the points of the South African player?**

- SELECT = [points]
- AGGREGATION [‘ ’]
- WHERE-clause [Country = South Africa]

Table 3.4 presents an example of a table database WikiSQL for given questions and table headers.

Table 3.4: WikiSQL question and table header

Player	Country	Points	Winnings
Stever Striker	United States	9000	120000
K.J Choi	South Korea	5400	76000
Rory Sabbani	South Africa	34000	450000
Mark Calcalvecchai	United States	2067	289330

3.2.3 Pre-processing

The WikiSQL dataset consists of pairs of natural language to corresponding SQL queries, to achieve the desired results during training, a good feature representation is required. The annotation of the data is done to capture the information about the dataset; this is achieved using the Standford Core-NLP (Manning et al. 2014); a technology tool that provide an understanding to human language; provide the base forms of words, their parts of speech and their relationships to other words. The dataset is first annotated using annotation.py from the original work that presented the WikSQL dataset (Zhong et al. 2017); this annotates the training, validation, and testing dataset. This dataset includes natural language questions, table ids, annotated SQL logical form, and data for each table as follows,

```
{
   "phase":1,
   "question":"What position does the player who played for butler cc (ks)?",
   "sql":{
      "conds":[
         [
            5,
            0,
            "1998"
         ]
      ],
      "sel":3,
      "agg":0
  \},
   "table_id":"1-10015132-11"
}
```

Tokenisation is the task of breaking down the raw text into words, sentence, called tokens. The training dataset consists of the questions and query and the training table consists of the column names and the condition. All the training, validation, and testing datasets are tokenized and represented as follows:


```
{"phase":1,"question":"What position does the player who played for butler cc (ks)
play?",
"sql":{"sel":3,"conds":[[5,0,"Butler CC (KS)"]],"agg":0},
"table_id":"1-10015132-11",
"tokenized_query":["SELECT","position","FROM","table_","WHERE","school\/club","team",
"EQL","butler","cc","(","ks",")"],
"query":"SELECT position FROM table_ WHERE school\/club team EQL butler cc ( ks )",
"tokenized_question":["what","position","does","the","player","who","played","for",
"butler","cc","(","ks",")","play","?"]}
```


After achieving a good annotation of the dataset and tokenisation, a collection of vocabulary is done, which are the collection of tokens achieved from the tokenisation for all the dataset step into each of the lists consisting of unique token for each data entry. Table 3.5 presents the training, validation, and testing datasets, and Table 3.6 are the tokenised vocabulary acquired after tokenisation.

Table 3.5: Data distribution for Training/Validation/Testing

Label	Values
Training_data	56355
Training table_data	18585
Validation_data	8421
Validation table_data	2716
Testing_data	15878
Testing table_data	5230

Table 3.6: The created vocabulary using pre-trained GloVe embeddings and FastText pre-trained embeddings, and the unknown vocabulary.

Pre-training embeddings	Created vocabulary	Used Vocab	Unknown Vocab
GloVe	7927	4515	1328349
FastText	8945	4885	1327979

3.2.3.1 Words Representation

To achieve a representation the training algorithms can use to process the high dimension of the vocabulary, embeddings are used as described in Chapter 2.2.1 on word representation is given in, the embedding model captures the characteristics of the neighbors (Young et al. 2017). This word embedding aims to make the algorithms better at identifying the semantics of a word in the question. The vocabulary created is trained on a

pre-trained word embedding to map the generated tokens to their corresponding vector representation using the pre-trained embedding. Two pre-trained word embeddings are explored on their effects of representing words, in a word level distribution. The **GloVe** pre-trained word embeddings is a large file of 5GB large word embedding trained on a **840 billion** token common web crawl corpus that maps words and phrases to vectors of real numbers where vectors of similar words have similar co-occurrences (Pennington et al. 2014). The mapping between the words and their indexes and the mapping between the indexes results in a dense vector with a fixed, arbitrary **300-dimensional** representations. Tokens that are not available in the word embedding are replaced with an $< UNK >$ unknown token, token to create its own embedding on a one-hot encoding. The vocabulary size created tokens are **4515** presented in Table 3.6 and with unknown tokens 1328349 also on Table 3.6. Another word embedding that is investigated is the **FastText**, as discussed in Chapter 2.2.1, that is an extension of the **word2vec** model that represents each word as an n-gram of characters (Joulin, Grave, Bojanowski, Douze, Jégou & Mikolov 2016). FastText consists of **2 million** word vectors trained on Common Crawl **600 billion** tokens. The vocabulary generated as presented in Table 3.6 is **8945**, used vocabulary **4885** and with unknown tokens **1327979**. The two word embedding are used because of their different relation to words, while the GloVe looks at vectors with similar co-occurance, the FastText capture the meaning of shorter words and allows the embeddings to understand suffixes and prefixes. Once the word has been represented using character n-grams, a skip-gram model is trained to learn the embeddings.

3.3 Research methods

The input is trained in batches of the word embeddings as explained in a representation of a sequence given as follows, given a sequence x as input; x is mapped to an output $\widehat{y}$, the SQL query, such that:

$$x = [x_1, ..., x_m] = [x_1^q, ..., x_m^q, x_{m+1}^c, ..., x_{m+n}^c, x_{m+n+1}^s, ..., x_{m+n}^s] \tag{3.1}$$

where m describes the vocabulary of tokens, x^q represents the question vocabulary, x^c represents the column name of the tables, and x^s the SQL query representation. The $< BEG >$ is the beginning token, and $< END >$ is the end token. The training process is divided into the three SQL components; each component is trained separately, each components with its embedding without sharing weights. These are then concatenated into a single result.

SELECT [AGG] [COLUMN] FROM TABLE [WHERE COLUMN2 EQUALS VALUE] [AND COLUMN3 EQUALS VALUE]

The [AGGREGATION] consists of 6 SQL keywords, ‘ ’, MAX, MIN, COUNT, SUM, and AVG, where the (‘ ’) represents not having an aggregation.
The [SELECT] column is based on two constraints, the question and the column names from the table.
The [WHERE-Clause] consisting of the three subcomponent, column name, operation, and condition.

Sequence-to-sequence models: The next step is to train the sequence-to-sequence

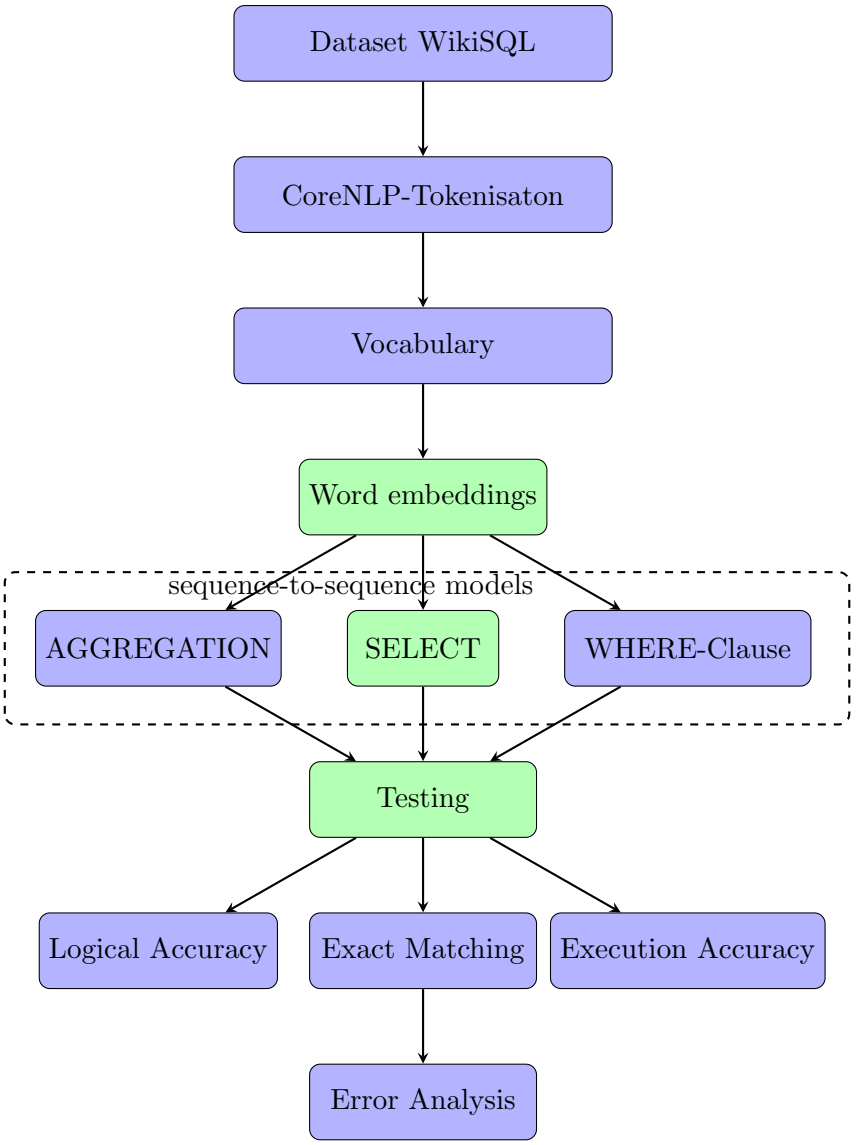

Figure 3.4: Methodology pipeline from data pre-processing to evaluation

models using the generated word embedding, five models are investigated, these models

are described in the literature review in Chapter 2. The sequence-to-sequence models used are discussed in Chapter 2.4.2, **Long-Short Term Memory**: Chapter 2.4.2.3 (Hochreiter & Schmidhuber 1997). **Bidirectional Long-Short Term Memory**: Chapter 2.4.2.2 outlines the bidirectional RNN; the BiLSTM comes from the RNN idea, which processes sequence data in both forward and backward directions with two separate hidden layers (Schuster & Paliwal 1997). **Encoder-Decoder** is outlined in Chapter 2.4.2.5. **Column Attention**: Chapter 2.4.2.6 discusses the different attention mechanism (Cho et al. 2014). The Column Attention computes an attention mechanism between tokens in the questions and the column names embeddings (Xu et al. 2017). **Pointer Network**: Chapter 2.4.2.7 (Vinyals et al. 2015) copies the input sequence to the output sequence. Pointer network is applied on the WHERE-Clause to generate the condition statement. Figure 3.4 shows the process of achieving the methodology.

3.4 Hardware and Software

3.4.1 Hardware

Table 3.7 presents the machine's hardware specification to run all the experiments for this research.

Table 3.7: Experimental hardware specifications

Computer Hardware	Specifications
Operating System	Linux b4.15.0-112
Version	113-Ubuntu
Architecture	Generic(x86_64)
RAM	128 GB
CPU Product	AMD Ryzen Threadripper 2950X 16-core processor
CPU Processor	1
CPU Cores	8
CPU threads	32
CPU Capacity	3800MHz
CPU Width	64 bits
Thread cores	32

3.4.2 Software

The implementations of the experiments are scripted using Python 3.7, computed using Pytorch; a Python-based scientific computing package that uses the power of graphics processing units. Scientific Learning Toolkit (SciKit-Learn) is used for the visualisation of the results. SQLite3 is used for the database connection.

3.5 Experimental Setup

In these experiments, the aim is to compare the performance of the five models LSTM, BiLSTM, BiLSTM Encoder-Decoder, BiLSTM Encoder-Decoder + Column Attention, and BiLSTM Encoder-Decoder + pointer network. Together with the WikiSQL dataset Zhong et al. (2017) made available the Table.py and DBengine.py that provides the connection to the query table and database. All architectures are trained separately

Table 3.8: Hyper-parameters of experimental models

Hyper Parameters	
Architecture	LSTM
Number of layers	2
Optimizer	Adam
Dropout	0.3
Hidden nodes	100, 200, 300
Learning rate	0.0001
Batch_size	64
Loss	Cross Entropy
Word Embedding	300
Epochs	50

for each component with the hyper-parameters as specified in Table 4.3. These hyper-parameters are the batch size, learning rates, number of epochs (discussed in Chapter 2.4.2.8). Gradient-based optimisation methods; **Adam**; optimisation explained in (discussed in Chapter 2.4.2.8). The model is trained by minimising the negative log likelihood of the training data using stochastic gradient (discussed in Chapter 2.4.2.8). The losses computed using cross-entropy loss (discussed in Chapter 2.4.2.8).

Testing and inference: Performance logs, prediction, and the models are saved during the training phase. The saved performance logs are then used separately on the new testing data for testing and inference. Similar to the training process, all architectures are tested separately for each dataset. Further performance logs and prediction from testing are saved for use in the analysis of results, error analysis, and discussion.

Performance evaluation: The performance measures follow Xu et al. (2017) to measure the query accuracy. To evaluate the generated predictions, the following matrices are employed:

1. **Logical-form accuracy**: Evaluates the predicted SQL query against the ground truth to check whether they match each other (Zhong et al. 2017).

2. **Execution accuracy**: The execution through the database both the predicted SQL query and the ground truth query, compare whether the results match to each other (Zhong et al. 2017).

3. **BLEU score**: Metric for assessing a predicted phrase to a reference phrase. A perfect match results in a score of 1.0, while a perfect mismatch results in a score of 0.0 (Papineni et al. 2002).

4. **Precision**: Calculates a percentage of positive predictions that were correct, to find all proportion of the model that the model says is relevant and are actually relevant. Given by the equation below:

$$Precision = \frac{TruePositive}{TruePositive + FalsePositive} \tag{3.2}$$

5. **Recall**: Calculates the true positive rate of the model performance to find all data points interest in the dataset. Given by the equation below:

$$Recall = \frac{TruePositive}{TruePositive + FalseNegative} \tag{3.3}$$

6. **F1-Score**: Calculates the weighted average of Precision and Recall. This score takes both false positives and false negatives into account. Given by the equation below:

$$F1 = 2 \times \frac{precision * recall}{precision + recall} \tag{3.4}$$

Error Analysis and Discussion: After collecting all the results from the training, validation, and testing phases a detailed error analysis is conducted. The final outputs of the execution accuracy are used to compute the generation of the error analysis. This is to quantify the quality of the output string with its generated ground truth to further identify where the performance of the models' has failed.

3.6 Summary

This chapter presents the research methodology, a comprehensive detail on the WikiSQL dataset, pre-processing of the dataset, and architectural design. It further outlines the experimental setup and the evaluation processed to conduct this research. Figure 3.1 presents the research pipeline, and Figure 3.4 outlines the training process of the architectures. The next chapter presents the results for each of the architecture described in this chapter; a discussion on the results, and extensive error analysis are also presented.

Chapter 4

Results and Analysis

4.1 Introduction

The previous chapter presented the research methodology that aims to address the research objective presented in Chapter 1, to comparatively analyse the sequence-to-sequence architectures to translate natural language to SQL queries. A detailed description of the pipeline that outlines the process of achieving the objective was presented. It also provides a comprehensive explanation of the WikiSQL dataset that consists of the question-query pairs used to evaluate the sequence-to-sequence architectures in section 3.2. Further detailing the material and methods that are used to meet the objective of the research, this includes a detailed outline on the pre-processing phase followed by the presentation of the five sequence-to-sequence models; **LSTM, BiLSTM, Encoder-Decoder, Column Attention, and Pointer Network** including the experimental-setup used to conduct the training process presented in section 3.3 to section 5. This chapter presents the experimental results and analysis that answers the research questions presented in Chapter 1; Which is the most effective sequence-to-sequence architecture for translating natural language to SQL?; What are the key factors contributing to the incorrect prediction of SQL queries?; and How can the prediction failures when generating semantically equivalent SQL queries be mitigated?. Section 4.2 presents the training level results where the hyper-paramer learning rates, hidden nodes, batch size and word embeddings are evaluated on the training, validation, and testing dataset. Section 4.3 presents the architectural level results and presents the results on the five evaluation metrics; BLEU, precision, recall F1-score and exact matching. Section 4.4 presents a comprehensive error analysis. Section 4.5 presents an overall discussion of all the results presented. Lastly, section 4.6 presents the summary of this chapter.

4.2 Training Level Results

Various factors contribute to the performance of sequence-to-sequence models, this includes the architectural design, number of hidden nodes, batch size, and the learning rate; these are presented in Chapter 2.4.2.9. This section details the training level results on the hyper-parameters including learning rate, number of hidden nodes, batch sizes, and word embeddings. The training process's main goal for sequence-to-sequence models is to find the parameters that optimise the best performance. The gradient of the error for the parameters accomplishes this, as discussed in Chapter 2.4.2.8. The **Convergence** of the trained network occurs when the error reduces with respect to the network's parameters. This decrease depends on the initialisation of the weights of the network. In the initial training stage, the models poorly define the relationship between training inputs to their corresponding ground truth; this is observed with the high error rate in the learning process. The error starts to decline as the network learns to generalise better parameters, explaining the relationship between the inputs and the corresponding ground truths. This observation is found in Figure 4.1; this shows the performance loss of training the **LSTM** using 50 epochs with a loss starting from loss of **3.58** to the loss of **0.63** at a steady decline where each component reached convergence at different epochs; AGGREGATION, SELECT, and WHERE-Clause achieving; **5, 46, 48** respectively.

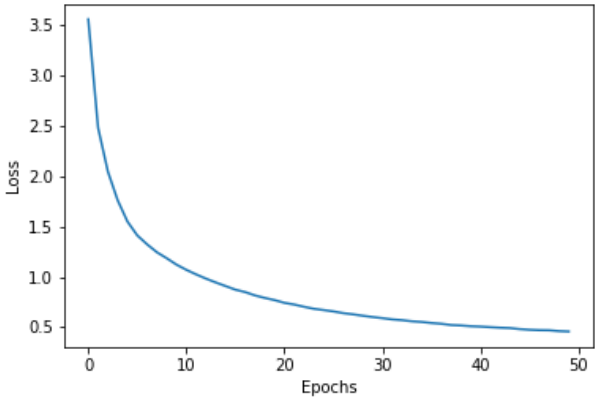

Figure 4.1: Training Loss vs epochs on the LSTM model for 50 epochs

This loss is achieved at an average of **125 minutes** per epoch. The loss is computed with the cross-entropy as discussed in Chapter 2.4.2.8. The convergence of the network depends on various factors such as optimisation, model setup, learning rate, and dataset size, in this case **Adam** discussed in Chapter 2.4.2.8, is used for the optimisation, which has shown to be an efficient technique to ensure convergence in previous literature (Zhong et al. 2017). As described in Chapter 3, a learning rate of **0.0001** is

used to controls the amount of change allocated to the error where the weights of the model are updated at each time time-step of each batch of training examples. This has shown effectiveness in previous research (Zhong et al. 2017). The sizes of the datasets used in this research are sufficiently large to ensure that the models learn the most optimal parameters for describing the relationship between the inputs and target, since all architectures displayed comparable convergence occurrence.

4.2.1 Effect on batch size

The training process's capacity depends on the batch size the sequence-to-sequence architectures train on, which is needed before starting of the training process. As described in Chapter 2, the batch sizes are the number of training examples used in the gradient estimation process per iteration. This affects the training and time performance of the models. Two batch sizes which are often used in literature are investigated the **batch size 32 and 64** (Goodfellow et al. 2016). The LSTM is noted to train only with the batch size of 32. However, the LSTM model did not train at all when using a batch size of 64. The **LSTM** took longer than all the top 4 models during train using a batch size of 32 computation time of **140 min** per epoch for **50** epochs. Figure 4.2 presents the **BiLSTM** training performance accuracy when using batch size of 32 presenting a higher performance accuracy than using the batch size of 64 on. The batch size of 32 trains at **120 min** per epoch and the batch size of 64 batch trains at **63 min** per epoch taking a much shorter time than the LSTM for both batch sizes 32 and 64.

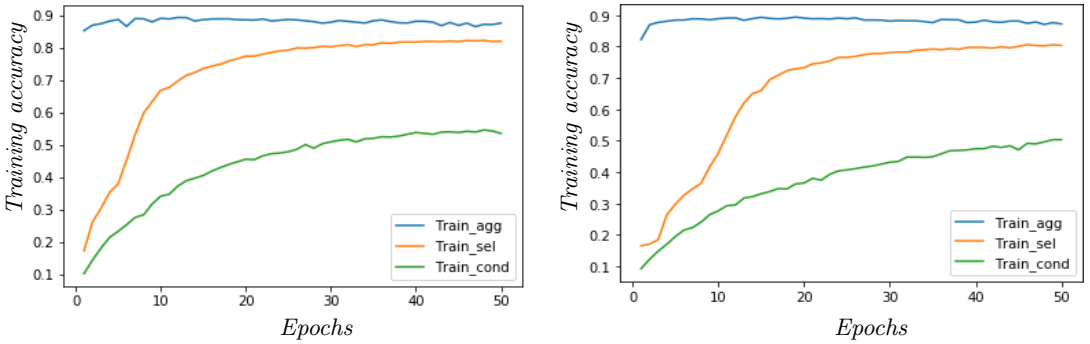

Figure 4.2: BiLSTM training accuracy on batch sizes 32 and 64

Figure 4.3 shows performance for the **Encoder-Decoder** model when training with the batch size of 32 and batch size of 64 respectively. Training with batch size of 32 took a much longer time with an average of **103 min** per epoch, while training with a batch size of 64 took **100 min** per epoch.

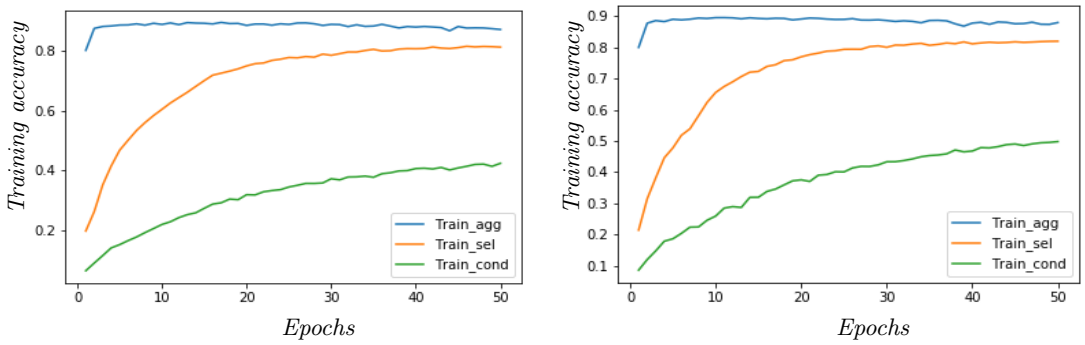

Figure 4.3: The Encoder-Decoder model training performance on batch sizes of 32 and 64 respectively

Figure 4.4 shows performance for the **Column Attention** when training with the batch size of 32 and batch size of 64 respectively. The batch size of 32 achieved **108 min** per epoch and the batch size of 64 achieves **65 min** per epoch, the column attention shows to train longer than both the Encoder-Decoder and Pointer Network.

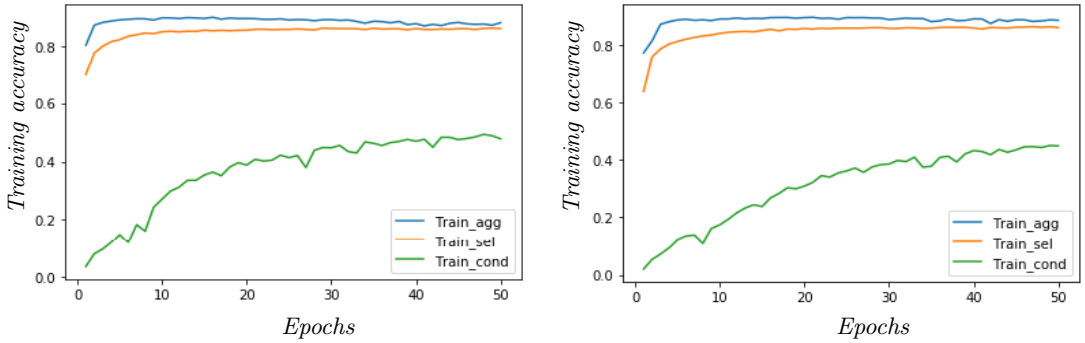

Figure 4.4: The Column Attention Training accuracy vs Epochs on batch sizes of 32 and 64 respectively

Figure 4.5 show performance for the **Pointer Network** when training with the batch size of 32 and batch size of 64 respectively. The Pointer Network performs closer to the Encoder-Decoder performing at **103 min** per epoch on a batch size of 32, while the batch size of 64 takes **60 min** per epoch. Overall for all the sequence-to-sequence

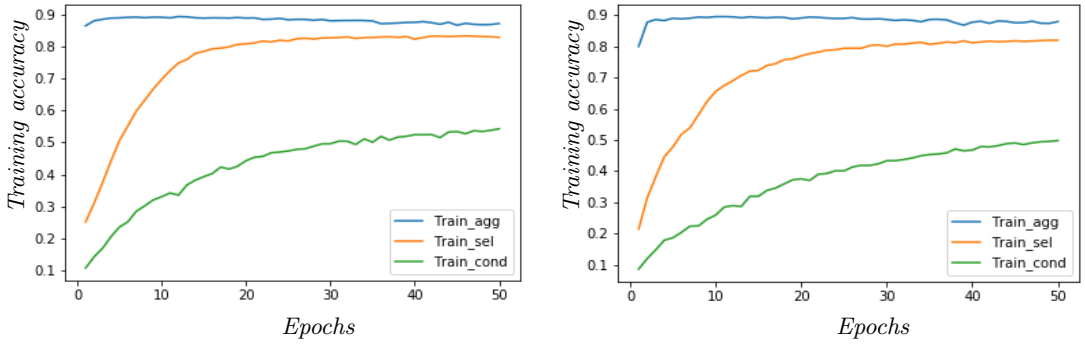

Figure 4.5: The Pointer Network training performance on batch sizes of 32 and 64 respectively

models, the batch size of 32 was an optimal training parameter, achieving a higher

performance accuracy than training with a batch size of 64. Although training with a batch size of 32 is noted to take double the training time to that the batch size of 64 takes, it achieves much higher performance accuracy. This highlights the importance of carefully choosing batch sizes for sequence-to-sequence models.

4.2.2 Effect on hidden nodes

An important parameter when training sequence-to-sequence architectures is the number of hidden nodes of the model because of their impact on the performance accuracy. The more hidden nodes, the better the reduction of the loss. Table 4.1 presents the performance results of the five sequence-to-sequence models on test dataset; the AGGREGATION, SELECT, WHERE-Clause, Logical, and Execution accuracies. As presented

Table 4.1: Performance results on the test dataset, AGGREGATION, SELECT, WHERE-Clause, Logical, Execution accuracies, and time taken for each epoch for different number of hidden nodes

Model	No. Nodes	TestAcc Agg (%)	TestAcc Sel (%)	TestAcc Where (%)	Logic Acc (%)	Exec Acc (%)	Training Time/Epoch(min)
LSTM	100	89.709	82.044	53.621	42.834	51.297	125
LSTM	200	89.677	83.769	56.877	45.673	54.011	125
LSTM	300	89.299	84.506	59.157	47.745	56.008	140
BiLSTM	100	89.759	83.278	56.002	45.25	53.797	63
BiLSTM	200	89.583	84.444	59.680	48.369	56.562	98
BiLSTM	300	89.703	84.727	59.535	48.593	56.272	120
Enc-Dec	100	89.664	80.526	41.516	32.982	42.436	65
Enc-Dec	200	89.778	83.877	49.861	40.660	48.929	95
Enc-Dec	300	89.912	85.975	59.767	48.373	56.654	103
Col Attention	100	89.600	84.756	53.645	41.754	50.490	90
Col Attention	200	89.761	85.789	54.184	43.258	52.781	103
Col Attention	300	89.917	86.893	55.315	45.452	53.671	108
Pointer Net	100	89.659	83.279	55.819	44.829	53.47	63
Pointer Net	200	89.747	84.658	59.623	48.526	56.984	93
Pointer Net	300	89.797	85.187	60.367	49.276	57.752	103

in Table 4.1 the models are evaluated on three different sets of hidden nodes **100, 200, 300**, to determine the optimal hidden nodes for each architectures. The performance accuracy is noted to continuously increase as the hidden nodes are increased from 100, 200, and 300 for the **Encoder-Decoder, Column Attention, and Pointer Network** on all the three SQL components. However, the **LSTM** and **BiLSTM** are noted to decrease as the hidden nodes are increased on the 300 hidden nodes. The LSTM is noted to generate an inconsistency across the components, the AGGREGATION is noted to decrease as the hidden nodes are increased while SELECT and WHERE-Clause components are noted to increase with the addition of hidden layers. Increasing the number of

hidden nodes more than the sufficient to train the model creates a performance decrease as evidence from the LSTM and the BiLSTM performance decrease on the AGGREGATION and the WHERE-Clause respectively. Interestingly different SQL components performance accuracy decrease with the increase of hidden nodes across the models. As the sequence-to-sequence models become complex, the more hidden nodes the models requires, this is evidence of the **Encoder-Decoder, Column Attention, and Pointer Network**'s performance accuracy continuing to increasing, while the increase in hidden nodes on basic sequence-to-sequence models LSTM and BiLSTM show a performance decrease. The generalisation error occurs because of the higher than needed hidden nodes for the LSTM and BiLSTM. This shows different SQL components achieve the optimal points with different hidden nodes, while the LSTM's AGGREGATION reaches its optimal training point at 100 hidden nodes, its SELECT and WHERE-Clause reaches their optimal training point at 300 hidden nodes, and BiLSTM reach training optimal point at 200 hidden nodes. The Encoder-Decoder, Column Attention, and Pointer Network reaches their optimal point at 300 hidden nodes. While adding hidden nodes show to improve performance accuracy, it is noted the more complex the model is the more hidden nodes it requires. The increase in the hidden nodes parameters is noted to lead to a longer training time, as presented in Table 4.1 where the training time increases for all the hidden nodes from 100-300 hidden nodes. Complex neural network architectures are prone to overfitting as discussed in Chapter 2. To overcome overfitting a regulation method **Dropout** of **0.3** is used for all the models (Srivastava et al. 2014). There is a direct trade-off between overfitting and model complexity.

4.2.3 Effects of Word embeddings

Word embeddings are used in the training of sequence-to-sequence models to capture the relationships of words. To evaluate effects of word embeddings in developing translation of natural language to SQL, two types of word embeddings **FastText**, that uses character n-grams and **GloVe** that uses discrete tokens, are used on the Pointer Network model. Table 4.2 presents the performance results of both the FastText word embedding on the Pointer Network and the GloVe word embedding on the Pointer Network. The

Table 4.2: Performance results on the test dataset, AGGREGATION, SELECT, WHERE-Clause, Logical, Execution accuracies, and time taken for each epoch for the FastText and GloVe word embedding on the Pointer Network model

Model	TestAcc Agg (%)	TestAcc Sel (%)	TestAcc Where (%)	Logic Acc (%)	Exec Acc (%)	Training Time/Epoch(min)
PointerFT	89.237	36.182	33.997	11.626	20.330	60.03
PointerGloVe	89.797	85.187	60.367	49.276	57.752	103

Pointer Network on both word embeddings are trained with the same hyper-parameters of batch size 32, learning rate 0.0001. Figure 4.6 presents the performance of FastText, and the GloVe word embeddings performance. The performance accuracy of using both

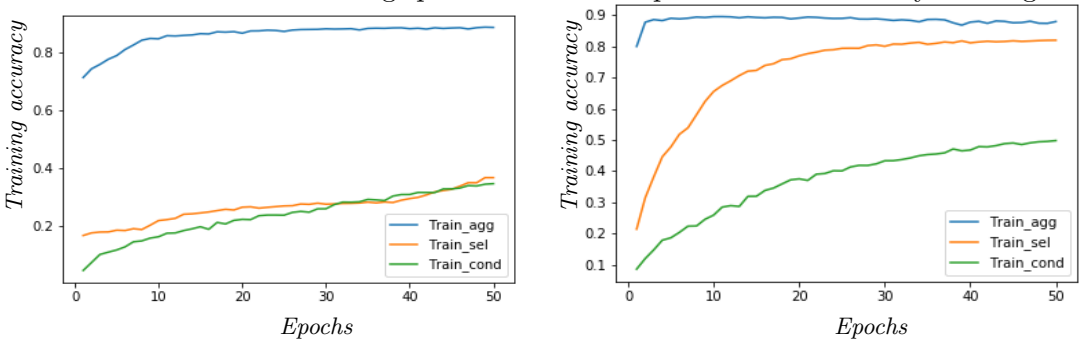

Figure 4.6: FastText vs GloVe word embeddings on the Pointer Network model

the FastText and GloVe on Pointer Network is noted to be comparative on the AGGREGATION as presented in Figure 4.6. Although the GloVe word embedding achieves a higher accuracy, the FastText is noted to achieve a performance accuracy closer to GloVe word embedding. The performance accuracy on both the SELECT and WHERE-Clause declines, however, using the FastText word show a decline when using FastText compared to the GloVe word embedding. This is seen on Table 4.2. The GloVe captures the information of the two components better than FastText on this task. Evaluating the two embeddings has shown the importance of word representation in NLP tasks of translation, specifically how each SQL component can be affected by the word representation used. These are somewhat unexpected results as FastText improves on Word2Vec by taking word parts into account. This thought to improve the Pointer Network performance on the WHERE-Clause for treating rare words or generalisation to unknown words. However, the GloVe embedding can have the advantage of words co-occurring in the same environment being able to predict the correct output token by token.

4.3 Testing level results

The test level result uses the optimal hyper-parameters found during the training level results as specified in Table 4.3. The results presented are from the five sequence-to-sequence models investigated where the final results are obtained during the testing process where the checkpoints are inferred using an unseen test dataset with parameters learned from the training stage for each model.

The **LSTM** architecture as discussed in Chapter 2 struggled to train on the CPU; the optimal results were achieved on batch size of 32. Table 4.4 presents the architectural results from the unseen test dataset.

Table 4.3: Final Hyper-parameters of experimental models

Hyper Parameters	
Architecture	LSTM
Number of layers	2
Optimizer	Adam
Dropout	0.3
Hidden nodes	300
Learning rate	0.0001
Batch_size	32
Loss	Cross Entropy
Weight Decay	0.3
GloVe Word Embedding	300 dimension
Epochs	50

Table 4.4: Performance results on the test dataset, AGGREGATION, SELECT, WHERE-Clause, Logical, Execution accuracies, and time taken for each epoch for each model

Model	TestAcc Agg (%)	TestAcc Sel (%)	TestAcc Where (%)	Logic Acc (%)	Exec Acc (%)	Training Time/Epoch(min)
LSTM	89.299	84.506	59.157	47.745	56.008	140
BiLSTM	89.703	84.727	59.535	48.593	56.272	120
Enc-Dec	89.912	85.975	59.767	48.373	56.654	103
Col Attention	89.917	86.893	55.315	45.452	53.671	108
Pointer-Net	89.797	85.187	60.367	49.276	57.752	103

LSTM achieving **89.299%, 84.506%, and 59.157%** performance accuracy for the AGGREGATION, SELECT, and WHERE-Clause respectively. The models's performance are evaluated on the **Logical Accuracy** and **Execution Accuracy**, the logical accuracy as explained in Chapter 2 evaluates the predicted SQL query to the ground truth ground truth on a component level. The execution accuracy evaluates the predicted SQL query to the execution ground truth on the database executable on SQLite3. The LSTM achieves **47.745%** and **56.008%** on the logical and execution accuracies respectively. It is important to note that the Execution Accuracy can create false positive evaluation as predicted SQL could return the same result (for example, NULL) as the ground truth SQL when they are semantically different. Compared to the top four models, however, the results are comparative given the training difficulty the LSTM model suffered.

The **BiLSTM** as discussed in Chapter 2 architecturally compared to LSTM, takes two LSTMs instate of single LSTM, computing the sequence in a forward and backward direction. Unlike LSTM, the BiLSTM had no training problems. The performance results reported in Table 4.4 show the BiLSTM performance improving from the LSTM with

a small margin of **89.703%, 84.727%, and 59.535%**. The logical and execution accuracies receiving **48.593%** and **56.272%** respectively. This improvement is attributed to the BiLSTM being able to capture information from the forward direction and the backward direction, capturing information better than the LSTM model. A contributing factor of the BiLSTM the ability to process and train much easier with the large corpus.

As described in Chapter 2, the **Encoder-Decoder** consists of two BiLSTMs on that encodes and the other that decodes. Performs an encoding to the input and decoding to the output, it is noted to outperform both LSTM and BiLSTM. As shown in Table 4.4 the Encoder-Decoder takes a much longer training time than the BiLSTM with an extra **5 min** per epoch. The Encoder-Decoder is noted to achieve a higher performance accuracy than both the LSTM and BiLSTM on all components. Table 4.4 presents the Encoder-Decoder model **89.912%, 85.975%, and 59.767%** and the logical and execution accuracies **48.373%** and **56.654%** respectively. It is noted that although the each of the Encoder-Decoder model's achieve higher performance accuracy to the BiLSTM, it's logical accuracy decline compared to the BiLSTM. This can be due to having accurate component prediction, however, not in the exact logical order. This is because although the logical accuracy receives a decline compared to BiLSTM, however, the execution accuracy is noted to achieve a much higher accuracy to the BiLSTM.

The **Column Attention**, as discussed in Chapter 2 pays attention to column names for both the SELECT and WHERE-Clause components. The Column Attention in Table 4.4 achieves **89.917%, 86.893%, and 55.315%** on the AGGREGATION, SELECT, and WHERE-Clause respectively. The impact of adding column attention does not significantly improve the performance of the AGGREGATION component, compared to both the Encoder-Decoder and the BiLSTM. A decline in performance is observed on the WHERE-Clause than the Encoder-Decoder model with a drop of **3%**. The logical and execution accuracies were noted also to decline compared to the Encoder-Decoder model achieving **45.452%** and **53.671%** respectively. This results shows that adding Column Attention can significantly improve performance capturing the column names much better than using the Encoder-Decoder model as evidence in the results. However, it does not significantly capture WHERE-Clause condition as evidence with the decline of the WHERE-Clause. As the Encoder-Decoder is noted to perform much better than the Column Attention on the WHERE-Clause.

Chapter 2 presents the **Pointer Network**, which learns the conditional probability of the output sequence with elements that are discrete tokens corresponding to positions in an input sequence. Table 4.4 presents the results on Pointer Network, achieving

89.797 %, 85.187%, and 60.367% on the AGGREGATION, SELECT, and WHERE-Clause respectively. The AGGREGATION shows to have a small decline in performance compared to the Encoder-Decoder and the Column Attention. This result has shown the impact of using Pointer Network with the WHERE-Clause achieving more than 1% performance boost on the WHERE-Clause. The Pointer Network on the SELECT component does not improve as the Column Attention performs much better than the Pointer Network on the SELECT. The performance results of the logical and execution accuracies **49.276%** and **57.752%** respectively.

4.3.1 Evaluation

To further establish the predicted SQL query's correctness to its corresponding ground truth the evaluation metrics are performed on the final-output SQL query. Table 4.5 presents the results from the evaluation metrics, which are the **F1-score, Precision accuracy, Recall accuracy, Exact match, and BLEU-score**.

Table 4.5: The evaluation metrics for all five models, F1-Score, Precision, Recall, and BLEU Score

Model	F1-Score(%)	Precision(%)	Recall(%)	Exact Match(%)	BLEU Score(%)
LSTM	55.813	63.158	50.0	33.883	72.076
BiLSTM	64.516	52.632	83.300	36.485	74.79
Enc-Dec	78.651	76.253	84.211	39.642	77.269
Column Attention	77.700	73.680	82.350	37.300	75.670
Pointer Network	88.888	84.210	94.117	40.611	78.056

The **BLEU-score** (Papineni et al. 2002), as discussed in Chapter 3, evaluates the quality of translation from one natural language to another. Its strength is to correlate well with human judgment by averaging out individual sentence judgment errors over the data, rather than attempting to devise the exact human judgment for every sentence (Papineni et al. 2002). Taking the BLEU-score of each sentence in the corpus and then averaging across them may artificially inflate the score; this is because it does not consider sentence structure or capture the syntax of the prediction, which is relevant when evaluating the predicted SQL query. This can be a flawed score. For example, when the predicted components are not at their right location, this can bring incorrect inflation to the BLEU-score. Unsurprisingly, the Pointer Network achieves better BLEU-score than all the four models, achieving **78.056%**. A possible reason for this is that they tend to generate words frequently associated with ground truth SQL queries. The BiLSTM and LSTM are noted to achieve higher BLEU-score, which can be attributed to the false negative of having correct information but not at the right place and BLEU-score averaging

the results higher than they should be. Although getting a good BLEU-score signal that the model has a good prediction, it is important to get an exact match prediction to its ground truth. The BLEU-score limitation is noted and the prediction outputs are further evaluated using four metrics the exact match, precision, recall, and F1-score. The **Exact match** metrics check the predicted SQL exactly matches the ground truth SQL query. It is a performance of lower bound as a semantically correct SQL query may differ from the ground truth SQL query in surface form. The Pointer Network shows to have a higher exact match than the top four models, achieving **40.611%**. This is attributed to generating more correct SQL to the ground truth SQL query. The Encoder-Decoder model is noted to achieve a higher exact match than the Column Attention, achieving **39.642%**. It is also interesting to note that the exact matching accuracy can be much lower than the BLEU-score; this may indicate that some of what BLEU-score has considered as correct are infact incorrect predictions. as discussed in Chapter 3, the **Recall** is the ratio of correctly predicted positive observations on all observations. The top four of the models a higher recall, excluding the LSTM, indicating that the top four models performed well in predicting overall correct predictions. The LSTM's recall compared to the top four indeed puts a question on the results received which indicated the BLEU-score might have been inflated by inaccurate results. The **Precision** as discussed in Chapter 3 is the ratio of the correctly predicted positive observations to the total predicted positive observations. The LSTM achieves higher precision than the recall, this can be attributed to generating correct predictions than errors, however, compared to other models, does not capture all the correct prediction. The **F1-score** combines the precision and recall into a single metric by taking their harmonic mean as described in Chapter 3. In the results presented in Table 4.5, the Pointer Network achieves the highest F1-score **88.88%**, while top four models achieves as follows, **77.700%, 78.651%, and 64.516%, 55.813%** for the Column Attention, Encoder-Decoder, BiLSTM, and LSTM respectively. Based on the given F1-score results, the quality of translation by Pointer Network is better than the top four models. This is further verified with the achievements of the highest exact score compared to all the top four models investigated.

The results presented so far partial answers research question one stated in Chapter 1.4; Which is the most effective sequence-to-sequence architecture for translating natural language to SQL?; The results provided thus far in Table 4.5 provides evidence that the Pointer Network trained at a learning rate of 0.0001, batch size 32, hidden nodes of 300, achieved highest score on Exact Match and BLEU-score, furthermore, achieving higher performance on the Pointer Network achieves a higher accuracy on both the logical and execution accuracies on Table 4.4. The Pointer Network shows to be the most effective sequence-to-sequence architecture. The next subsection presents some examples of the output prediction for all the models evaluated.

4.3.2 SQL Prediction output

The prediction outputs are analysed according to their question knowledge groups categories as described in Chapter 3; where knowledge groups are Exact, Paraphrase, Partial, External knowledge, Ambiguous respectively. Tables 4.6, 4.7, 4.8, 4.9, and 4.10 presents the final prediction outputs for the five knowledge group described as Exact, Paraphrase, Partial, External knowledge, and Ambiguous respectively.

4.3.3 Exact Match

The **'Exact match'** as described in Chapter 3 consists of relevant information including the column names for both SELECT, WHERE-Clause, and condition explicitly defined. Below examples in Tables 4.6 show the output prediction of the five models where the questions can be classified as 'Exact match'. Upon observation of the output examples for all the sequence-to-sequence models, it is noted that the five models were proficient in predicting the SQL queries where for these questions, as evident in the examples.

4.3.4 Paraphrase

The questions referred as **'paraphrase'** as described in Chapter 3 where the column names are paraphrased. The Column Attention and Pointer Network consistently predicted the correct SELECT column name. However, the Column Attention further predicts correct WHERE-Clause column name whereas the Pointer Network predicted more incorrect column names. Table 4.7 examples show the output prediction of the five models where the questions can be classified as 'Paraphrase'. Questions consisting of paraphrasing may consist of more than the two column names for both the SELECT and WHERE-Clause, which shows prediction errors if column names' information is inside other column names available on the question.

Table 4.6: The predicted SQL statement with its corresponding ground truth SQL statement for exact match questions

Type	Sentence
Question 1	what is the total number of total w–l where doubles w-l is 11–11?
Table	['Player', 'Total W–L', 'Singles W–L', 'Doubles W–L', 'Ties played', 'Debut', 'Years played']
Ground Truth	SELECT COUNT(Total W-L) FROM table_1_10294071_1 WHERE Doubles W–L = 11–11
LSTM	SELECT COUNT(Total W–L) FROM table_1_10294071_1 WHERE Doubles W–L = 11–11
BiLSTM	SELECT COUNT(Total W–L) FROM table_1_10294071_1 WHERE Doubles W–L = 11–11
Encoder-Decoder	SELECT COUNT(Total W–L) FROM table_1_10294071_1 WHERE Doubles W–L = 11–11
Column Attention	SELECT COUNT(Total W–L) FROM table_1_10294071_1 WHERE Doubles W–L = 11–11
Pointer Network	SELECT COUNT(Total W–L) FROM table_1_10294071_1 WHERE Doubles W–L = 11–11
Question 2	What is the Frequency at the Market/Rank of Burlington - Plattsburgh , Vermont - New York /143?
Table	['Calls', 'Frequency', 'Branding','Format', 'Market/Rank', 'Timeslot', 'Group owner']
Ground Truth	SELECT COUNT(Frequency) FROM table_1_10333757_1 WHERE Market/Rank = Burlington - Plattsburgh , Vermont - New York /143
LSTM	SELECT Frequency FROM table_1_10333757_1 WHERE Branding = burlington - plattsburgh- vermont- new york/143, vermont
BiLSTM	SELECT Frequency FROM table_1_10333757_1 WHERE Branding = burlington - plattsburgh , vermont - new york /143
Encoder-Decoder	SELECT Frequency FROM table_1_10333757_1 WHERE Frequency = burlington - plattsburgh , vermont - new york /143
Column Attention	SELECT Frequency FROM table_1_10333757_1 WHERE Group owner = burlington - plattsburgh , vermont - new york /143
Pointer Network	SELECT Frequency FROM table_1_10333757_1 WHERE Format = burlington - plattsburgh , vermont - new york /143
Question 3	how many pick# does the chivas usa mls team have
Table	['Pick #', 'MLS team', 'Player', 'Position', 'Affiliation']
Ground Truth	SELECT MAX(Pick #) FROM table_1_29626583_1 WHERE MLS team = Chivas USA
LSTM	SELECT COUNT(Pick #) FROM table_1_29626583_1 WHERE MLS team = chivas usa mls
BiLSTM	SELECT COUNT(Pick #) FROM table_1_29626583_1 WHERE MLS team = chivas usa
Encoder-Decoder	SELECT COUNT(Pick #) FROM table_1_29626583_1 WHERE MLS team = chivas usa mls
Col Attention	SELECT COUNT(Pick #) FROM table_1_29626583_1 WHERE MLS team = chivas usa
Pointer Network	SELECT COUNT(Pick #) FROM table_1_29626583_1 WHERE MLS team = chivas usa
Question 4	How many picks did Mike Zaher have?
Table	['Round', 'Pick', 'Player', 'Position', 'School/Club Team']
Ground Truth	SELECT SUM(Pick #) FROM table_2_15214004_3 WHERE Player = mike zaher
LSTM	SELECT COUNT(Pick #) FROM table_2_15214004_3 WHERE Player = mike zaher
BiLSTM	SELECT COUNT(Pick #) FROM table_2_15214004_3 WHERE Player = mike zaher
Encoder-Decoder	SELECT COUNT(Pick #) FROM table_2_15214004_3 WHERE Player = mike zaher
Col Attention	SELECT COUNT(Pick #) FROM table_2_15214004_3 WHERE Player = mike zaher
Pointer Network	SELECT COUNT(Pick #)FROM table_2_15214004_3 WHERE Player = mike zaher
Question 5	How many silver medals when the total was 2 and less than 1 bronze?
Table	['Rank', 'Nation', 'Gold', 'Silver', 'Bronze', 'Total']
Ground Truth	SELECT SUM(Silver) FROM table_2_18661293_4 WHERE Bronze AND Total < AND = 1 AND 2
LSTM	SELECT SUM(Silver) FROM table_2_18661293_4 WHERE Total = 2 AND Bronze < 1
BiLSTM	SELECT COUNT(Silver) FROM table_2_18661293_4 WHERE Total = 2 AND Bronze < 1
Encoder-Decoder	SELECT SUM(Silver) FROM table_2_18661293_4 WHERE Total = 2 AND Silver < 1
Col Attention	SELECT COUNT(Silver) FROM table_2_18661293_4 WHERE Total = 1 AND Bronze < 1
Pointer Network	SELECT COUNT(Silver) WHERE table_2_18661293_4 Total AND Total = AND < 2 AND 1
Question 5	What is the notes distance for 2nd position earlier than 1986?
Table	['Year', 'Title', 'Role', 'Director', 'Notes']
Ground Truth	SELECT Notes FROM table_2_14617261_1 WHERE Position AND Year = AND < 2nd AND 1986
LSTM	SELECT Notes FROM table_2_14617261_1 WHERE Position = 2nd AND Year < 1986
BiLSTM	SELECT Notes FROM table_2_14617261_1 WHERE Position = 2nd AND Year< 1986
Encoder-Decoder	SELECT Notes FROM table_2_14617261_1 WHERE Position = 2nd AND Year < 1986
Column Attention	SELECT Notes FROM table_2_14617261_1 WHERE Position = 1986 AND Year < 1986
Pointer Network	SELECT Notes FROM table_2_14617261_1 WHERE Position AND Year = AND > 2nd AND 1986

4.3.5 Partial Clue

The questions referred to as **'Partial clue'** about its definition described in Chapter 3 where the question only consists of the partial clue for both the SELECT and WHERE-Clause column names. Column Attention is noted to predict correct column names for both the SELECT and WHERE-Clause. The top four models predicts the correct SELECT column names but struggles with the prediction of the WHERE-Clause column names. Table 4.8 show the output prediction examples of the five models where the questions can be classified as 'partial clue'.

Table 4.7: The predicted SQL statement with its corresponding ground-truth SQL statement for paraphrase questions

Type	Sentence
Question 1	If a radius is 10, what is the lowest possible mass?
Table	['Star (Pismis24-#)', 'Spectral type', 'Magnitude (M bol)', 'Temperature (K)', 'Radius (R ⊙)', 'Mass (M ⊙)']
Ground Truth	SELECT MIN(Mass (M ⊙)) table_1_10432351_1 WHERE Radius (R ⊙) = 10
LSTM	SELECT MIN(Temperature (K)) FROM table_1_10432351_1 WHERE Spectral type = 10
BiLSTM	SELECT MIN(Temperature (K)) FROM table_1_10432351_1 WHERE Star (Pismis24-#) = 10
Encoder-Decoder	SELECT MIN(Mass (M ⊙)) FROM table_1_10432351_1 WHERE Star (Pismis24-#) = 10
Column Attention	SELECT MIN(Mass (M ⊙)) FROM table_1_10432351_1 WHERE Radius (R ⊙) = ,
Pointer Network	SELECT MIN(Mass (M ⊙)) FROM table_1_10432351_1 WHERE Radius (R ⊙) = 10
Question 2	Which charts had debut sales of more than 339333.011497678?
Table	['Release', 'Oricon Albums Chart', 'Peak Position', 'Debut Sales (copies)', 'Sales Total (copies)', 'Chart Run']
Ground Truth	SELECT Oricon Albums Chart FROM table_1_23180638_1 WHERE Debut Sales (copies) > 339333.011497678
LSTM	SELECT Release FROM table_1_23180638_1 WHERE Release> 339333.011497678
BiLSTM	SELECT Oricon Albums Chart FROM table_1_23180638_1 WHERE Release > 339333.011497678
Encoder-Decoder	SELECT COUNT(Peak Position) FROM table_1_23180638_1 WHERE
Column Attention	SELECT Oricon Albums Chart FROM table_1_23180638_1 WHERE Debut Sales (copies) > 339333.011497678
Pointer Network	SELECT Oricon Albums Chart FROM table_1_23180638_1 WHERE Release = 339333.011497678
Question 3	How many hours were flown in each of the years where more than 64379058.0 kilometers were flown?
Table	['Year', 'Aircraft kilometers', 'Departures', 'Flying hours', 'Passengers', 'Seat factor', 'Employees', 'Profit/loss']
Ground Truth	SELECT Flying hours FROM table_1_105344_2 WHERE Aircraft kilometers > 64379058.0
LSTM	SELECT COUNT(Flying hours) FROM table_1_105344_2 WHERE Seat factor > 64379058.0
BiLSTM	SELECT COUNT(Flying hours) FROM table_1_105344_2 WHERE Flying hours > 64379058.0
Encoder-Decoder	SELECT Flying hours FROM table_1_105344_2 WHERE Aircraft kilometers > 64379058.0
Column Attention	SELECT Flying hours FROM table_1_105344_2 WHERE Departures > 64379058.0
Pointer Network	SELECT COUNT(Flying hours) FROM table_1_105344_2 WHERE Passengers > 64379058.0
Question 4	Give me a list of all spanish officer titles that could receive recognition as group captain in english
Table	['Equivalent NATO Rank Code', 'Rank in Spanish', 'Rank in English', 'Commonwealth equivalent', 'US Air Force equivalent']
Ground Truth	SELECT Rank in English FROM table_1_1015521_2 WHERE Commonwealth equivalent = Group Captain
LSTM	SELECT Rank in English FROM table_1_1015521_2 WHERE Equivalent NATO Rank Code = english
BiLSTM	SELECT Commonwealth equivalent FROM table_1_1015521_2 WHERE Equivalent NATO Rank Code = english
Encoder-Decoder	SELECT Rank in Spanish FROM table_1_1015521_2 WHERE Equivalent NATO Rank Code = in english group captain
Col Attention	SELECT Equivalent NATO Rank Code FROM table_1_1015521_2 WHERE Rank in English = english
Pointer Network	SELECT Equivalent NATO Rank Code FROM table_1_1015521_2 WHERE Rank in English = english
Question 5	Which wrestlers have had 2 reigns?
Table	['Rank Each wrestlers total number of days as champion are ranked highest to lowest; wrestlers with the same number mean that they are tied for that certain rank.', 'Wrestler', '# of reigns', 'Combined defenses', 'Combined days']
Ground Truth	SELECT Wrestler FROM table_1_10182508_5 WHERE # of reigns = 2
LSTM	SELECT COUNT(Wrestler) FROM table_1_10182508_5 WHERE Rank Each wrestlers total number of days as champion are ranked highest to lowest; wrestlers with the same number mean that they are tied for that certain rank. = wrestler AND Rank Each wrestlers total number of days as champion are ranked highest to lowest; wrestlers with the same number mean that they are tied for that certain rank. = combined AND Rank Each wrestlers total number of days as champion are ranked highest to lowest; wrestlers with the same number mean that they are tied for that certain rank. = combined
BiLSTM	SELECT Combined defenses FROM table_1_10182508_5 WHERE Rank Each wrestlers total number of days as champion are ranked highest to lowest; wrestlers with the same number mean that they are tied for that certain rank. = 2
Encoder-Decoder	SELECT COUNT(Wrestler) FROM table_1_10182508_5 WHERE Rank Each wrestlers total number of days as champion are ranked highest to lowest; wrestlers with the same number mean that they are tied for that certain rank. = 2
Column Attention	SELECT COUNT(Wrestler) FROM table_1_10182508_5 WHERE Wrestler =
Pointer Network	SELECT Wrestler FROM table_1_10182508_5 WHERE # of reigns = 2
Question 6	when english is tea how many nihon-shiki?
Table	['English', 'Japanese', 'Kana spelling', 'Revised Hepburn', 'Kunrei-shiki', 'Nihon-shiki']
Ground Truth	SELECT Nihon-shiki FROM table_1_26263954_1 WHERE English = tea
LSTM	SELECT COUNT(Revised Hepburn) FROM table_1_26263954_1 WHERE English = roman
BiLSTM	SELECT COUNT(Revised Hepburn) FROM table_1_26263954_1 WHERE English = english
Encoder-Decoder	SELECT COUNT(Japanese) FROM table_1_26263954_1 WHERE English = tea
Column Attention	SELECT COUNT(Japanese) FROM table_1_26263954_1 WHERE English = english is
Pointer Network	SELECT COUNT(Kana spelling) FROM table_1_26263954_1 WHERE English = english how many

4.3.6 External Knowledge

The questions referred to as **'External Knowledge'** in reference to its definition in Chapter 3 where the column names are not mentioned and require the knowledge from the table. Table 4.9 shows the output prediction where the questions can be classified as 'external knowledge'. The external knowledge question shows a higher prediction on the

Table 4.8: The predicted SQL statement with its corresponding ground-truth SQL statement for partial clue questions

Type	Sentence
Question 1	Which winning team beat the New York Yankees?
Table	['Year', 'Game or event', 'Date contested', 'League or governing body', 'Sport', 'Winning team', 'Losing team', 'Final score']
Ground Truth	SELECT Winning team FROM table_1_10548224_1 WHERE Losing team = New York Yankees
LSTM	SELECT Winning team FROM table_1_10548224_1 WHERE Game or event = new york yankees
BiLSTM	SELECT Winning team FROM table_1_10548224_1 WHERE Winning team = new york yankees
Encoder-Decoder	SELECT Winning team FROM table_1_10548224_1 WHERE Sport = new york yankees
Column Attention	SELECT Winning team FROM table_1_10548224_1 WHERE Losing team = new york yankees
Pointer Network	SELECT Losing team FROM table_1_10548224_1 WHERE Year = new york yankees
Question 2	What was the final score for the game that was contested on February 1, 2009?
Table	['Year', 'Game or event', 'Date contested', 'League or governing body', 'Sport', 'Winning team', 'Losing team', 'Final score']
Ground Truth	SELECT Final score FROM table_1_10548224_1 WHERE Date contested = February 1, 2009
LSTM	SELECT Final score FROM table_1_10548224_1 WHERE Year = february 1, 2009
BiLSTM	SELECT Final score FROM table_1_10548224_1 WHERE Date contested = february 1, 2009
Encoder-Decoder	SELECT Final score FROM table_1_10548224_1 WHERE Year = february 1, 2009
Column Attention	SELECT Final score FROM table_1_10548224_1 WHERE Date contested = february 1, 2009
Pointer Network	SELECT Final score FROM table_1_10548224_1 WHERE Year = february 1, 2009
Question 3	what's the total number of south australia with victoria value of 2173?
Table	['Crop (kilotonnes)', 'New South Wales', 'Victoria', 'Queensland', 'Western Australia', 'South Australia', 'Tasmania', 'Total']
Ground Truth	SELECT COUNT(South Australia) FROM table_1_1057262_2 WHERE Victoria = 2173
LSTM	SELECT COUNT(New South Wales) FROM table_1_1057262_2 WHERE Tasmania = 2173
BiLSTM	SELECT COUNT(South Australia) FROM table_1_1057262_2 WHERE Queensland = 2173
Encoder-Decoder	SELECT COUNT(South Australia) FROM table_1_1057262_2 WHERE Victoria = 2173
Column Attention	SELECT COUNT(New South Wales) FROM table_1_1057262_2 WHERE Victoria = 2173
Pointer Network	SELECT COUNT New South Wales FROM table_1_1057262_2 WHERE Queensland = 2173
Question 4	What was the first season of the club who last won a title in 2012?
Table	['Club', 'Position in 2012', 'First season', 'Number of seasons', 'First season of current spell', 'Titles','Last title']
Ground Truth	SELECT First season FROM table_1_1096793_1 WHERE Last title = 2012
LSTM	SELECT MIN(First season) FROM table_1_1096793_1 WHERE Club = 2012
BiLSTM	SELECT First season FROM table_1_1096793_1 WHERE First season = 2012
Encoder-Decoder	SELECT MIN(First season) FROM table_1_1096793_1 WHERE Titles = 2012
Column Attention	SELECT First season FROM table_1_1096793_1 WHERE Last title = 2012
Pointer Network	SELECT First season FROM table_1_1096793_1 WHERE Titles = 2012
Question 5	What was the total in Brooklyn when Manhattan was 3,139?
Table	['2013 Republican primary', 'Manhattan', 'The Bronx', 'Brooklyn', 'Queens', 'Staten Island', 'Total', '%']
Ground Truth	SELECT Brooklyn FROM table_1_1108394_6 WHERE Manhattan = 3,139
LSTM	SELECT Total FROM table_1_1108394_6 WHERE Queens = 3,139
BiLSTM	SELECT Manhattan FROM table_1_1108394_6 WHERE Manhattan = 3,139
Encoder-Decoder	SELECT Manhattan FROM table_1_1108394_6 WHERE Brooklyn = 3,139
Column Attention	SELECT Total FROM table_1_1108394_6 WHERE Manhattan = 3,139
Pointer Network	SELECT Total FROM table_1_1108394_6 WHERE 2013 Republican primary = 3,139
Question 6	Name the most ceiling temperature for very extra high?
Table	['Maximum Ceiling Temperature', 'Temperature Rating', 'Temperature Classification', 'Color Code (with Fusible Link)', 'Glass Bulb Color']
Ground Truth	SELECT Maximum Ceiling Temperature FROM table_1_1538516_1 WHERE Temperature Classification = Very Extra High
LSTM	SELECT MAX(Temperature Classification) FROM table_1_1538516_1 WHERE Glass Bulb Color = very (high)
BiLSTM	SELECT MAX(Maximum Ceiling Temperature) FROM table_1_1538516_1 WHERE Maximum Ceiling Temperature = very extra high
Encoder-Decoder	SELECT MAX(Maximum Ceiling Temperature) FROM table_1_1538516_1 WHERE Maximum Ceiling Temperature = very extra high
Column Attention	SELECT MAX(Maximum Ceiling Temperature) FROM table_1_1538516_1 WHERE Glass Bulb Color = extra high
Pointer Network	SELECT MAX(Maximum Ceiling Temperature) FROM table_1_1538516_1 WHERE Glass Bulb Color = extra high

Column Attention compared to the top 4 models where the prediction outputs show to be equivalent to the ground truth on 3 of the examples out of 5 examples; this shows the quality of the SQL query even though the column name is not mentioned in the question. The Column Attention focuses on capturing the attention on column; hence, this gives it an advantage in predicting questions that need external knowledge from the tables. The Column Attention consistently predicted correct SELECT and WHERE-Clause column names for questions needing external knowledge, however, the top four models failed to predict the correct WHERE-Clause column name. The Column Attention is successful in this questions, however, the Pointer Network is noted to struggle with predict the WHERE-Clause column name, especially where the column names are not mentioned at all as presented in Table 4.9.

Table 4.9: The predicted SQL statement with its corresponding ground-truth SQL statement for external knowledge questions

Type	Sentence
Question 1	What is the away team of the UEFA champions league?
Table	['Season', 'Competition', 'Round', 'Opponent', 'Home', 'Away']
Ground Truth	SELECT Away FROM table_2_1607312_1 WHERE Competition = uefa champions league
LSTM	SELECT Away FROM table_2_1607312_1 WHERE Competition = uefa champions
BiLSTM	SELECT Opponent FROM table_2_1607312_1 WHERE Competition = uefa champions
Encoder-Decoder	SELECT Away FROM table_2_1607312_1 WHERE Competition = uefa champions
Column Attention	SELECT Away FROM table_2_1607312_1 WHERE Competition = uefa champions
Pointer Network	SELECT Opponent FROM table_2_1607312_1 WHERE Opponent = uefa champions
Question 2	What was the elimination number of the fighter who fought within 26:15
Table	['Elimination number', 'Wrestler', 'Entered', 'Eliminated by', 'Method of elimination', 'Time']
Ground Truth	SELECT Elimination number FROM table_1_18598175_2 WHERE Method of elimination = 26:15
LSTM	SELECT Eliminated by FROM table_1_18598175_2 WHERE Time = 26:15
BiLSTM	SELECT Elimination number FROM table_1_18598175_2 WHERE Time = 26:15
Encoder-Decoder	SELECT MIN(Elimination number) FROM table_1_18598175_2 WHERE Time = 26:15
Column Attention	SELECT Elimination number FROM table_1_18598175_2 WHERE Method of elimination = 26:15
Pointer Network	SELECT MAX(Elimination number) FROM table_1_18598175_2 WHERE Time = 26:15
Question 3	Give me a list of all spanish officer titles that could receive recognition as group captain in english
Table	['Equivalent NATO Rank Code', 'Rank in Spanish', 'Rank in English', 'Commonwealth equivalent', 'US Air Force equivalent']
Ground Truth	SELECT Rank in English FROM table_1_1015521_2 WHERE Commonwealth equivalent = Group Captain
LSTM	SELECT Rank in English FROM table_1_1015521_2 WHERE Equivalent NATO Rank Code = english
BiLSTM	SELECT Commonwealth equivalent FROM table_1_1015521_2 WHERE Equivalent NATO Rank Code = english
Encoder-Decoder	SELECT Rank in Spanish FROM table_1_1015521_2 WHERE Equivalent NATO Rank Code = in english group captain
Column Attention	SELECT Equivalent NATO Rank Code FROM table_1_1015521_2 WHERE Rank in English = english
Pointer Network	SELECT Equivalent NATO Rank Code FROM table_1_1015521_2 WHERE Rank in English = english
Question 4	What number of voters did Queens have when Staten Island had 295 voters?
Table	['1953', 'party', 'Manhattan', 'The Bronx', 'Brooklyn', 'Queens', 'Richmond [Staten Is.]', 'Total', '%']
Ground Truth	SELECT Queens FROM table_1_1108394_34 WHERE Richmond [Staten Is.] = 295
LSTM	SELECT COUNT(Total) FROM table_1_1108394_6 WHERE 2013 Republican primary = staten island
BiLSTM	SELECT The Bronx FROM table_1_1108394_34 WHERE Brooklyn = 295
Encoder-Decoder	SELECT Total FROM table_1_1108394_34 WHERE % = 295
Column Attention	SELECT 1953 FROM table_1_1108394_34 WHERE The Bronx = staten island
Pointer Network	SELECT Total FROM table_1_1108394_34 WHERE party = staten
Question 5	What nation had 135 Bronze medals?
Table	['Sport', 'Gold', 'Silver','Bronze', 'Total']
Ground Truth	SELECT COUNT(Bronze) FROM table_2_12253523_2 WHERE Total = 135
LSTM	SELECT Nation FROM table_2_12253523_2 WHERE Bronze = 135
BiLSTM	SELECT Nation FROM table_2_12253523_2 WHERE Bronze = 135 AND Nation = 135
Encoder-Decoder	SELECT Nation FROM table_2_12253523_2 WHERE Bronze = 135
Column Attention	SELECT Nation FROM table_2_12253523_2 WHERE Bronze = 135
Pointer Network	SELECT Nation FROM table_2_12253523_2 WHERE Bronze = 135

4.3.7 Ambiguous

The questions which are referred to as **'Ambiguous'** in reference to its definition described in Chapter 3. All the sequence-to-sequence models show to suffer due to the question ambiguity. The prediction of column names are incorrect in most questions, as seen on Table 4.10, showing the output prediction where the questions can be classified as 'ambiguous'. The WHERE-Clause column names are noted to be correctly predicted by the Encoder-Decoder, Column Attention, and Pointer Network. The lack of information on the question showed to affect the prediction of the column names as the column attention also struggled to predict the SELECT column names when it was not explicitly mentioned. The prediction of the condition is noted to be successfully on most sequence-to-sequence models although some of the models struggled to capture the entire condition as evident in the examples. Although the condition is explicitly defined in the questions, the SQL query prediction have shown to differ per model indicating the different capacity each model has in learning the translation task. The Pointer Network showed to predict the WHERE-Clause condition most correctly. This

Table 4.10: The predicted SQL statement with its corresponding ground truth SQL statement for ambiguous questions

Type	Sentence
Question 1	What is the tone for gen.sing. plague?
Table	['Nominative', 'Old orthography', 'New orthography', '/e/ or /æ/ (IPA)', 'Tone (Latvian notation: / ∼ /- level, / ∧ / - broken)', 'Translation']
Ground Truth	SELECT Tone (Latvian notation: / ∼ / - level, /∧/ - broken) FROM table_1_10409754_5 WHERE Translation = Gen.Sing. plague
LSTM	SELECT /e/ or /æ/ (IPA) FROM table_1_10409754_5 WHERE Nominative = gen.sing
BiLSTM	SELECT Nominative FROM table_1_10409754_5 WHERE Old orthography = plague
Encoder-Decoder	SELECT Translation FROM table_1_10409754_5 WHERE Old orthography = gen.sing
Column Attention	SELECT /e/ or /æ/ (IPA) FROM table_1_10409754_5 WHERE /e/ or /æ/ (IPA) = .
Pointer Network	SELECT New orthography FROM table_1_10409754_5 WHERE New orthography = plague
Question 2	here was Assen held?
Table	['No', 'Date', 'Round', 'Circuit', 'Pole Position', 'Fastest Lap', 'Race winner', 'Report']
Ground Truth	SELECT Round FROM table_1_10083598_1 WHERE Circuit = Assen
LSTM	SELECT Circuit FROM table_1_10083598_1 WHERE Circuit = assen
BiLSTM	SELECT Pole Position FROM table_1_10083598_1 WHERE Circuit = assen
Encoder-Decoder	SELECT Circuit FROM table_1_10083598_1 WHERE Circuit = assen
Column Attention	SELECT Circuit FROM table_1_10083598_1 WHERE Circuit = assen
Pointer Network	SELECT Circuit FROM table_1_10083598_1 WHERE Circuit = assen
Question 3	Which show runs on Friday at 05:00 AM?
Table	['Time', '00:00 AM', '01:00 AM', '02:00 AM', '05:00 AM']
Ground Truth	SELECT 05:00 AM FROM table_2_15535243_1 WHERE Time = friday
LSTM	SELECT 00:00 AM FROM table_2_15535243_1 WHERE 01:00 AM = friday am AND Time = friday
BiLSTM	SELECT Time FROM table_2_15535243_1 WHERE Time = 05:00 am
Encoder-Decoder	SELECT Time FROM table_2_15535243_1 WHERE Time = 05:00 am
Column Attention	SELECT Time FROM table_2_15535243_1 WHERE Time = at 05:00 am
Pointer Network	SELECT Time FROM table_2_15535243_1 00:00 WHERE AM = 05:00 am
Question 4	What is the total number of James Bond 007: Everything or Nothing for each system?
Table	['Title', 'System', 'Publisher', 'Product No.', 'Compatible with 20/60GB NTSC PS3 (CECHA/CECHB)', 'Compatible with 60GB PAL/80GB NTSC PS3 (CECHC/CECHE)']
Ground Truth	SELECT COUNT(System) FROM table_1_10875694_11 WHERE Title = James Bond 007: Everything or Nothing
LSTM	SELECT COUNT(Compatible with 20/60GB NTSC PS3 (CECHA/CECHB)) FROM table_1_10875694_11 WHERE Title = james bond 007:. or)
BiLSTM	SELECT COUNT(Compatible with 60GB PAL/80GB NTSC PS3 (CECHC/CECHE)) FROM table_1_10875694_11 WHERE System = system
Encoder-Decoder	SELECT COUNT(Product No.) FROM table_1_10875694_11 WHERE Publisher = system
Column Attention	SELECT COUNT(Product No.) FROM table_1_10875694_11 WHERE Title = james bond 007: everything or nothing for each system AND Compatible with 60GB PAL/80GB NTSC PS3 (CECHC/CECHE) = 007: everything or nothing for
Pointer Network	SELECT COUNT(Product No.) FROM table_1_10875694_11 WHERE Title = james bond 007: everything or nothing for each system AND Compatible with 60GB PAL/80GB NTSC PS3 (CECHC/CECHE) = 007: everything or nothing for
Question 5	What number of voters did Queens have when Staten Island had 295 voters?
Table	['1953', 'party', 'Manhattan', 'The Bronx', 'Brooklyn', 'Queens', 'Richmond [Staten Is.]', 'Total', '%']
Ground Truth	SELECT Queens FROM table_1_1108394_34 WHERE Richmond [Staten Is.] = 295
LSTM	SELECT COUNT(Total) FROM table_1_1108394_6 WHERE 2013 Republican primary = staten island
BiLSTM	SELECT The Bronx FROM table_1_1108394_34 WHERE Brooklyn = 295
Encoder-Decoder	SELECT Total FROM table_1_1108394_34 WHERE % = 295
Col Attention	SELECT 1953 FROM table_1_1108394_34 WHERE The Bronx = staten island
Pointer Network	SELECT Total FROM table_1_1108394_34 WHERE party = staten

further conclusively answers the first research question presented in Chapter 1; Which is the most effective sequence-to-sequence architecture for translating natural language to SQL?. The type of evaluation that has been conducted shows that different architectures are capable of achieving best results on different SQL components. This informs that no one architecture can generalise and support all SQL components. The SQL output categorised in their knowledge bases shows that the Column Attention works well across the 4 knowledge bases and provides much more correct column names. This shows that while the Pointer Network gives an overall, the Column Attention is much better at predicting column names. This section further partially answers the second research question; What are the key factors contributing to the incorrect prediction of SQL queries?. The results of the output as per their knowledge base shows that not all models handle the same question the same. While the Column Attention is able to perform well on the 4 knowledge base the top 4 models struggled to generate correct SQL. The same can be said about the Pointer Network, the model is able to predict

the WHERE-Clause on ambiguous questions while the rest of the models suffer. The Encoder-Decoder suffers when the string is long as observed. These error factors can be thought as model level errors. [1]

4.4 Error Analysis

The error analysis is conducted to examine where each of the five sequence-to-sequence models has failed to predict the correct SQL query output. Often a model may not perform to its optimal level because of the influence in the architectural development, the dataset, in this case; the dataset pre-processing, questions, column names, and hyper-parameters. Other factors may be the genuine challenge of using natural languages and the sequence-to-sequence models' learning capability. The error analysis is performed to understand where each of the five sequence-to-sequence models struggles to generate exact matching SQL query to their corresponding ground truth SQL query. Four phases, AGGREGATION, Column name, Operation, and Condition, are presented in Table 4.11, Table 4.12, Table 4.13, and Table 4.14 respectively each with five examples randomly selected. Various factors were identified as some of the main reasons for the arising errors as follows:

- Errors in the WikiSQL dataset.
- Question not containing either the column names or the condition not explicitly mentioned.
- Mismatch and lexical problem, occurring between a natural language question and its corresponding SQL query since SQL.
- The column name does not contain enough information, i.e, abbreviations not matching information from the question or column names with similar names.
- Multiple valid SQL queries for each question.
- Error in the ground truth.

These factors are elaborated further with the error analysis and discussion categorised in 4 phases; the AGGREGATION, Column name, Operation, and Condition. Some of the deep learning outputs are not interpretable, so a general hypothesis is discussed on some of the findings of the errors.

[1] Table 1 in the appendix provides 15 randomly selected prediction output results, predicted SQL query, ground truth SQL query, questions, and column name for each of the five models. Table 2 appendix provides 15 randomly selected prediction output results, the SQL query predictions are incorrect or do not match the ground truth SQL query, predicted SQL query, ground truth SQL query, questions, and column names, for each of the five model

4.4.1 Errors in the Aggregation

Table 4.11 presents five examples related to the errors on the AGGREGATION.

Table 4.11: The predicted SQL statement with its corresponding ground truth SQL statement where the AGGREGATION has incorrect predictions

Type	Sentence
Question 1	What was Go Shiozaki's rank?
Table	['Rank Each wrestlers total number of days as champion are ranked highest to lowest; wrestlers with the same number mean that they are tied for that certain rank.', 'Wrestler', '# of reigns', 'Combined defenses', 'Combined days']
Ground Truth	SELECT MIN(Rank Each wrestlers total number of days as champion are ranked highest to lowest; wrestlers with the same number mean that they are tied for that certain rank.) FROM table_1_10182508_5 WHERE Wrestler = Go Shiozaki
LSTM	SELECT COUNT(Combined days) FROM table_1_10182508_5 WHERE Rank Each wrestlers total number of days as champion are ranked highest to lowest; wrestlers with the same number mean that they are tied for that certain rank. = go shiozaki
BiLSTM	SELECT COUNT(# of reigns) FROM table_1_10182508_5 WHERE Rank Each wrestlers total number of days as champion are ranked highest to lowest; wrestlers with the same number mean that they are tied for that certain rank. = go shiozaki
Encoder-Decoder	SELECT COUNT(# of reigns) FROM table_1_10182508_5 WHERE Rank Each wrestlers total number of days as champion are ranked highest to lowest; wrestlers with the same number mean that they are tied for that certain rank. = shiozaki shiozaki
Column Attention	SELECT COUNT(Rank Each wrestlers total number of days as champion are ranked highest to lowest; wrestlers with the same number mean that they are tied for that certain rank.) FROM table_1_10182508_5 WHERE Wrestler =
Pointer Network	SELECT Rank Each wrestlers total number of days as champion are ranked highest to lowest; wrestlers with the same number mean that they are tied for that certain rank FROM table_1_10182508_5 WHERE Rank Each wrestlers total number of days as champion are ranked highest to lowest; wrestlers with the same number mean that they are tied for that certain rank. = go shiozaki
Error	MIN
Question 2	What was the number of race that Kevin Curtain won?
Table	['No', 'Date', 'Round', 'Circuit', 'Pole Position', 'Fastest Lap', 'Race winner', 'Report']
Ground Truth	SELECT COUNT(No) FROM table_1_10083598_1 WHERE Pole Position = Kevin Curtain
LSTM	SELECT COUNT(Round) FROM table_1_10083598_1 WHERE Circuit = kevin curtain
BiLSTM	SELECT COUNT(Fastest Lap) FROM table_1_10083598_1 WHERE Race winner = kevin curtain
Encoder-Decoder	SELECT COUNT(No) FROM table_1_10083598_1 WHERE Race winner = kevin curtain
Column Attention	SELECT Race winner FROM table_1_10083598_1 WHERE Race winner = kevin curtain
Pointer Network	SELECT Race winner FROM table_1_10083598_1 WHERE Race winner = kevin curtain
Error	COUNT
Question 3	What was the elimination number of the fighter who fought within 27:27
Table	['Elimination number', 'Wrestler', 'Entered', 'Eliminated by', 'Method of elimination', 'Time']
Ground Truth	SELECT Wrestler FROM table_1_18598175_2 Elimination number = 27:27
LSTM	SELECT Eliminated by FROM table_1_18598175_2 WHERE Time = 27:27
BiLSTM	SELECT Elimination number FROM table_1_18598175_2 WHERE Time = 27:27
Encoder-Decoder	SELECT MIN(Elimination number) FROM table_1_18598175_2 WHERE Time = 27:27
Column Attention	SELECT MIN(Elimination number) FROM table_1_18598175_2 WHERE Time = 27:27
Pointer Network	SELECT MIN(Elimination number) FROM table_1_18598175_2 WHERE Time = 27:27
Error	MIN
Question 4	How many picks did Mike Zaher have?
Table	['Round', 'Pick', 'Player', 'Position', 'School/Club Team']
Ground Truth	SELECT SUM(Pick #) FROM table_2_15214004_3 WHERE Player = mike zaher
LSTM	SELECT COUNT(Pick #) FROM table_2_15214004_3 WHERE Player = mike zaher
BiLSTM	SELECT COUNT(Pick #) FROM table_2_15214004_3 WHERE Player = mike zaher
Encoder-Decoder	SELECT COUNT(Pick #) FROM table_2_15214004_3 WHERE Player = mike zaher
Column Attention	SELECT COUNT(Pick #) FROM table_2_15214004_3 WHERE Player = mike zaher
Pointer Network	SELECT COUNT(Pick #)FROM table_2_15214004_3 WHERE Player = mike zaher
Error	SUM
Question 5	When did the stadium where Bursaspor is the tenant open?
Table	['Stadium', 'Capacity', 'City', 'Country', 'Tenant', 'Opening']
Ground Truth	SELECT MIN(Opening) FROM table_1_10601843_2 WHERE Tenant = Bursaspor
LSTM	SELECT Stadium FROM table_1_10601843_2 WHERE Tenant = bursaspor open
BiLST	SELECT Tenant FROM table_1_10601843_2 WHERE Country = bursaspor
Encoder-Decoder	SELECT Opening FROM table_1_10601843_2 WHERE Stadium = tenant open
Column Attention	SELECT Opening FROM table_1_10601843_2 WHERE Stadium = tenant open
Pointer Network	SELECT Opening FROM table_1_10601843_2 WHERE Stadium = tenant open
Error	MIN

Example 1 presented in Table 4.11, is noted to consists of the ground truth generating an unnecessary AGGREGATION; the question does not require an AGGREGATION or suggest 'MIN' as the AGGREGATION. This shows a typical example of errors in the ground truth; however, the predicted SQL query is noted to be correct for all the five sequence-to-sequence models. Example 2, the question requires a 'COUNT' as it

asks the number of race; however, as it has been observed, the model learns the keyword 'COUNT' from questions that consists of "how many", "how much", Hence this might be the reason why the Column Attention and Pointer Network did not predict the 'COUNT'. It is noted that the BiLSTM and Encoder-Decoder are able to predicts the correct AGGREGATION (Hwang et al. 2019). Example 3, it is noted that the top four models predict an error 'MIN', however, nothing suggests a minimum; interestingly the LSTM and BiLSTM do not predict any AGGREGATION, which is corresponding to the ground truth (Hwang et al. 2019). Example 4, the prediction of the 'COUNT' shows the association of the questions with "How many" and "How much" to the keyword 'COUNT' and 'SUM' by all the models, this may be confusing to predict depending on the interpretation of as words "How many" and "How much" that associates with the 'COUNT' relate to quantity. This shows that the models still need to understand the context better. It is noted that the way a question is asked influences the predictions. In this case the question does not suggest a 'COUNT'. The models might have learnt to generate the specific AGGREGATION on the association of ("How many"), hence, generating the 'COUNT' instate of 'SUM' AGGREGATION. It is worth noting that even human struggle with the 'SUM' and 'COUNT' keywords. Hwang et al. (2019) found humans made similar errors as the models. Example 5, the ground truth suggests 'MIN'. However, given the question, there is nothing that suggests the prediction of the AGGREGATION prediction. All the models did not predict an AGGREGATION which suggest that the ground truth is actually incorrect. More examples of the incorrect predictions on the AGGREGATION are presented in Table 3 in the appendix at the end of the thesis.

4.4.2 Error in the SELECT and WHERE-Clause column names

The errors for the column names for both SELECT and WHERE-Clause are mostly influenced by the unavailability of information on the questions (Zhong et al. 2017, Hwang et al. 2019). Five examples are presented for the SELECT and WHERE-Clause column names on Table 4.12.

Table 4.12: Incorrect matching between the Column names of both the SELECT and the WHERE-Clause

Type	Sentence
Question 1	Where was Assen held?
Table	['No', 'Date', 'Round', 'Circuit', 'Pole Position', 'Fastest Lap', 'Race winner', 'Report']
Ground Truth	SELECT Round FROM table_1_10083598_1 WHERE Circuit = Assen
LSTM	SELECT No FROM table_1_10083598_1 WHERE Circuit = assen
BiLSTM	SELECT Pole Position FROM table_1_10083598_1 WHERE Circuit = assen
Encoder-Decoder	SELECT Circuit FROM table_1_10083598_1 WHERE Circuit = assen
Col Attention	SELECT Circuit FROM table_1_10083598_1 WHERE Circuit = assen
Pointer Network	SELECT Circuit FROM table_1_10083598_1 WHERE Circuit = assen
Error	SELECT: Circuit
Question 2	When were ships laid down that were commissioned on october 29, 1965?
Table	['#', 'Shipyard', 'Laid down', 'Launched', 'Commissioned', 'Fleet', 'Status']
Ground Truth	SELECT Laid down FROM table_110142062_2 WHERE Commissioned = October 29, 1965
LSTM	SELECT Laid down FROM table_1_1014206_2 WHERE # = october 29, 1965 AND # > commissioned
BiLSTM	SELECT Shipyard FROM table_1_1014206_2 WHERE Commissioned = october 29, 1965
Encoder-Decoder	SELECT Laid down FROM table_1_1014206_2 WHERE Status = october 29, 1965
Col Attention	SELECT Laid down FROM table_1_1014206_2 WHERE Laid down = october 29, 1965
Pointer Network	SELECT Laid down FROM table_110142062_2 WHERE Laid down = October 29, 1965
Error	WHERE-Clause: Laid down
Question 3	Who remixed an album of dance remixes?
Table	['Version', 'Length', 'Album', 'Remixed by', 'Year']
Ground Truth	SELECT Remixed by FROM table_2_15173207_2 WHERE Album = dance remixes
LSTM	SELECT Remixed by FROM table_2_15173207_2 WHERE Album = dance remixes
BiLSTM	SELECT Remixed by FROM table_2_15173207_2 WHERE Album = dance remixes
Encoder-Decoder	SELECT Remixed by FROM table_2_15173207_2 WHERE Album = dance remixes
Column Attention	SELECT Remixed by FROM table_2_15173207_2 WHERE Album = dance remixes
Pointer Network	SELECT Album FROM table_2_15173207_2 WHERE Version = dance remixes
Error	SELECT Album
Question 4	Who remixed a music video in 1985?
Table	['Version', 'Length', 'Album', 'Remixed by', 'Year']
Ground Truth	SELECT Remixed by FROM table_2_15173207_2 WHERE Year AND Version = AND = 1985 AND music video
LSTM	SELECT Remixed by FROM table_2_15173207_2 WHERE Year = remixed AND Version = 1985
BiLSTM	SELECT Remixed by FROM table_2_15173207_2 WHERE Year = 1985 AND Version = music remixed
Encoder-Decoder	SELECT Remixed by FROM table_2_15173207_2 WHERE Album = 1985 AND Version = 1985
Column Attention	SELECT Remixed by FROM table_2_15173207_2 WHERE Year = 1985
Pointer Network	SELECT Remixed by FROM table_2_15173207_2 WHERE Album AND Length = AND = music video AND 1985
Error	WHERE-Clause: Album AND Length
Question 5	Which position ranked 10th overall?
Table	['Rnd', 'Pick', 'Player name', 'Position', 'College', 'Height', 'Weight']
Ground Truth:	SELECT Position FROM table_2_13864_1 WHERE Pick = 10th overall
LSTM	SELECT COUNT(Position) FROM table_2_13864_1 WHERE Pick = 10th
BiLSTM	SELECT Position FROM table_2_13864_1 WHERE Pick = 10th
Encoder-Decoder	SELECT Position FROM table_2_13864_1 WHERE Position = 10th
Column Attention	SELECT Position FROM table_2_13864_1 WHERE Position = 10th
Pointer Network	SELECT Position FROM table_2_13864_1 WHERE Position = 10th
Error	WHERE-Clause: Position

Example 1 presented in Table 4.12, all the models could not predict the SELECT column name, interestingly even the Column Attention was unable to predict the SELECT, is attributed to the ambiguity of the question, neither the SELECT nor WHERE-Clause are mentioned in the question. A common error across the models was the tendency of predicting the wrong WHERE-Clause column by taking the SELECT column name as its predicted output, when the column names are not explicitly mentioned in the question (Xu et al. 2017). Example 2, presents the same error of predicting the same

column names for both the SELECT and WHERE-Clause (Xu et al. 2017). Interestingly all information are available on the question, SELECT and WHERE-Clause. Example 3, only the Pointer Network predicts a wrong WHERE-Clause column name incorrectly, which seem to be random and unexplainable. Example 4, is a case of multiple correct SQL queries as both the ground truth and the predicted SQL queries can hold as the correct SQL query (Hwang et al. 2019). The models might have been confused because of the availability of many column names clues. This is seen in most examples that consist of more than two column names in the question. Example 5, although the LSTM and BiLSTM predict the correct WHERE-Clause column name, the Encoder-Decoder, Column attention, and Pointer Network incorrectly predicts the WHERE-Clause column name, this might be attributed to the question not having the WHERE-Clause column name.

4.4.3 Error in the Operation

The operation of the WHERE-Clause gives comparison and restriction to the information requested; this contributes to the overall errors of the WHERE-Clause. Five examples are presented in Table 4.14.

Example 1 presented in Table 4.14, the keyword prior signifies the column name 'Previous champion(s)', however, the Pointer Network and BiLSTM model predicts the wrong operation, this might be an association of 'prior' to an operation >. However, the Column Attention and Encoder-Decoder model predicts the correct operation. Example 2, the LSTM and Pointer Network models are noted to predict > which could be because of the words "most high points". However, the column name already takes the "High points" (Xu et al. 2017). Example 3, not predicting the correct column name subsequently shows to effect on the prediction of the operation. Interestingly only the Pointer Network model is the only one that predicted the wrong operation. A random prediction on the Pointer Network shows to predict the operation >; however, the other

Table 4.13: Incorrect WHERE-Clause Operation Prediction

Type	Sentence
Question 1	Who was the champion prior to Xix Xavant?
Ground Truth	SELECT Previous champion(s) FROM table_1_1272033_1 WHERE Champion(s) = Xix Xavant
LSTM	SELECT Championship FROM table_1_1272033_1 WHERE Date Won < xix
BiLSTM	SELECT Champion(s) FROM table_1_1272033_1 WHERE Championship > xix
Encoder-Decoder	SELECT Championship FROM table_1_1272033_1 WHERE Championship = xix xavant
Column Attention	SELECT Champion(s) FROM table_1_1272033_1 WHERE Location = xix xavant
Pointer Network	SELECT Date Won FROM table_1_1272033_1 WHERE Championship > xavant
Question 2	Who had the most high points and rebounds than charles oakley (9)?
Ground Truth	SELECT High points FROM table_1_13812785_8 High rebounds = Charles Oakley (9)
LSTM	SELECT High points FROM table_1_13812785_8 WHERE High rebounds > charles AND Team = oakley oakley (9)
BiLSTM	SELECT High points FROM table_1_13812785_8 WHERE High rebounds = charles oakley (9)
Encoder-Decoder	SELECT High points FROM table_1_13812785_8 WHERE High points = charles oakley (9)
Column Attention	SELECT High points FROM table_1_13812785_8 WHERE High rebounds = charles oakley (9)
Pointer Network	SELECT High points FROM table_1_13812785_8 Record AND Team > AND = charles AND 9 oakley (9)
Question 3	What's the change over same quarter the previous year in the period when the 89.6% of the trains arrive within 5 minutes of scheduled time (over three months)?
Ground Truth	SELECT Change over same quarter the previous year FROM table_1_171748_3 WHERE % trains arriving within 5 mins of scheduled time (over three months) = 89.6%
LSTM	SELECT Change over same quarter the previous year FROM table_1_171748_3 WHERE Period < 5 AND Period = 89.6%)
BiLSTM	SELECT Change over same quarter the previous year FROM table_1_171748_3 WHERE Period = 89.6%
Encoder-Decoder	SELECT % trains arriving within 5 mins of scheduled time (over three months) FROM table_1_171748_3 WHERE Period = 89.6%
Column Attention	SELECT Change over previous year as a whole FROM table_1_171748_3 WHERE % trains arriving within 5 mins of scheduled time (over three months) = the 89.6% of trains arrive arrive arrive

models are able to predict the correct operation, which is unexplainable as to why only the Pointer Network does not get the correct prediction (Zhong et al. 2017). Example 4, the Pointer Network and Encoder-Decoder models did not get 'larger than' clue subsequently getting the predicting an incorrect operation (Zhong et al. 2017). Interestingly both the BiLSTM and Column Attention are capable of predicting the correct operation. Example 5, according to the question, the operation that needs to be predicted is $>$ which all the models properly predicted. However, the ground truth did not generate the correct operation. This indicates the ground truth consists of errors. Table 5 in the appendix presents the rest of the WHERE-Clause operation errors.

4.4.4 Error in the WHERE-Clause condition

The WHERE-Clause achieved the lowest accuracy compared to the AGGREGATION and SELECT. This is attributed to the incorrect predictions of column names, operation, and the condition string. Several factors attributed to the incorrect predictions of the SQL queries; eight examples are presented in Table 4.14:
Example 1 presented in Table 4.14, semantically the generation of "overall" seems to be an extra generated information from the ground truth as the condition of 10th is correctly predicted by all the models. Example 2, the error seemed to have been generated by the ground truth as the "season" is the one equal to 6. The SQL query shows to need another condition with 'AND' where "episode = 15", the Pointer Network shows to have predicted the correct condition. Example 3, the last substring is taken as the prediction output instate of "Go Shiozaki", assume that both the Column Attention and Pointer Network models predict as the condition (Wang, Brockschmidt & Singh 2018). Example 4, the common error with neural network models is the repetition of prediction, text degeneration of condition prediction where unnecessary extra information is predicted (Holtzman et al. 2019). The Pointer Network shows to suffer from this especially when the condition is not the last string in the question; this is also seen on examples 5, 6, and 7. Example 8, the column name "tenant" is predicted as the condition; this could be attributed to information overload as all three "(stadium, tenant, opening)" are column name clues, the Column Attention and Pointer Network might to the "stadium" as the is the correct column name (Wang, Brockschmidt & Singh 2018).

Table 6 in the appendix gives further examples of the incorrect prediction of the WHERE-Clause condition. The results presented in this section further answers the second research questions; What are the key factors contributing to the incorrect prediction of SQL queries?; based on the error analysis categories on section 4.4.1, section 4.4.2, section 4.4.3, and section 4.4.4. This chapter shows main challenges that includes errors

Table 4.14: Incorrect WHERE-Clause Condition prediction

Text	Sentence
Question 1	Which position ranked 10th overall?
Ground Truth	SELECT Position FROM table_2_13864_1 WHERE Pick = 10th overall
LSTM	SELECT COUNT(Position) FROM table_2_13864_1 WHERE Pick = 10th
BiLSTM	SELECT Position FROM table_2_13864_1 WHERE Position = 10th
Encoder-Decoder	SELECT Position FROM table_2_13864_1 WHERE Position = 10th
Column Attention	SELECT Position FROM table_2_13864_1 WHERE Position = 10th
Pointer Network	SELECT Position FROM table_2_13864_1 WHERE Position = 10th
Question 2	What is the original air date for episode 15 of season 6?
Ground Truth	SELECT Original air date FROM table_1_10718868_2 WHERE No. in season = 15
LSTM	SELECT Original air date FROM table_1_10718868_2 WHERE No. in season = 15
BiLSTM	SELECT Original air date FROM table_1_10718868_2 WHERE No. in season = 15
Encoder-Decoder	SELECT Original air date FROM table_1_10718868_2 WHERE No. in season = 15
Column Attention	SELECT Original air date FROM table_1_10718868_2 WHERE No. in season = of
Pointer Network	SELECT Original air date FROM table_1_10718868_2 WHERE No. in season = 6
Question 3	How many combined days did Go Shiozaki have?
Ground Truth	SELECT Combined days FROM table_1_10182508_5 WHERE Wrestler = Go Shiozaki
LSTM	SELECT COUNT(Combined days) FROM table_1_10182508_5 WHERE Rank Each wrestlers total number of days as champion are ranked highest to lowest; wrestlers with the same number mean that they are tied for that certain rank. = go shiozaki
BiLSTM	SELECT COUNT(# of reigns) FROM table_1_10182508_5 WHERE Rank Each wrestlers total number of days as champion are ranked highest to lowest; wrestlers with the same number mean that they are tied for that certain rank. = go shiozaki
Encoder-Decoder	SELECT COUNT(# of reigns) FROM table_1_10182508_5 WHERE Rank Each wrestlers total number of days as champion are ranked highest to lowest; wrestlers with the same number mean that they are tied for that certain rank. = shiozaki shiozaki
Column Attention	SELECT COUNT(Combined days) FROM table_1_10182508_5 WHERE Wrestler = have
Pointer Network	SELECT COUNT(Combined days) FROM table_1_10182508_5 WHERE Wrestler = have
Question 4	Which song has picturisation by only vijay?
Ground Truth	SELECT Song FROM table_1_10848177_1 WHERE Picturization = Vijay
LSTM	SELECT Song FROM table_1_10848177_1 WHERE Lyricist = only vijay
BiLSTM	SELECT Song FROM table_1_10848177_1 WHERE Singers = only vijay
Encoder-Decoder	SELECT Song FROM table_1_10848177_1 WHERE Lyricist = only vijay
Column Attention	SELECT Song FROM table_1_10848177_1 WHERE Singers = vijay
Pointer Network	SELECT Song FROM table_1_10848177_1 WHERE Picturization = vijay vijay
Question 5	What is the maximum capacity of the Otkrytie Arena stadium?
Ground Truth	SELECT MAX(Capacity) FROM table_1_10601843_2 WHERE Stadium = Otkrytie Arena
LSTM	SELECT MAX(Capacity) FROM table_1_10601843_2 WHERE Stadium = otkrytie arena stadium
BiLSTM	SELECT MAX(Capacity) FROM table_1_10601843_2 WHERE Stadium = otkrytie arena stadium
Encoder-Decoder	SELECT MAX(Capacity) FROM table_1_10601843_2 WHERE Stadium = otkrytie arena stadium
Column Attention	SELECT MAX(Capacity) FROM table_1_10601843_2 WHERE Stadium = otkrytie arena stadium
Pointer Network	SELECT MAX(Capacity) FROM table_1_10601843_2 WHERE Stadium = otkrytie arena stadium
Question 6	What was the first season of the club that in 2012 was 2nd in Superettan?
Ground Truth	SELECT First season FROM table_1_1096793_1 WHERE Position in 2012 = 2nd
LSTM	SELECT First season FROM table_1_1096793_1 WHERE Position in 2012 = 2nd AND Position in 2012 = superettan
BiLSTM	SELECT First season FROM table_1_1096793_1 WHERE Club = 2nd
Encoder-Decoder	SELECT First season FROM table_1_1096793_1 WHERE Titles = 2nd AND Position in 2012 = superettan
Column Attention	SELECT First season FROM table_1_1096793_1 WHERE Position in 2012 = was AND First season of current spell = was
Pointer Network	SELECT First season FROM table_1_1096793_1 WHERE Position in 2012 = 2nd in Superettan
Question 7	What is the regulated retail price for the tariff code ff0 prs?
Ground Truth	SELECT BTs retail price (regulated) FROM table_1_10408617_5 WHERE Tariff code = ff0 PRS
LSTM	SELECT BTs retail price (regulated) FROM table_1_10408617_5 WHERE Tariff code = ff0 prs
BiLSTM	SELECT BTs retail price (regulated) FROM table_1_10408617_5 WHERE Tariff code = ff0 prs
Encoder-Decoder	SELECT BTs retail price (regulated) FROM table_1_10408617_5 WHERE Tariff code = ff0 prs
Column Attention	SELECT BTs retail price (regulated) FROM table_1_10408617_5 WHERE Tariff code = ff0 prs
Pointer Network	SELECT BTs retail price (regulated) FROM table_1_10408617_5 WHERE Tariff code = ff0
Question 8	When did the stadium where Bursaspor is the tenant open?
Ground Truth	SELECT MIN(Opening) FROM table_1_10601843_2 WHERE Tenant = Bursaspor
LSTM	SELECT Stadium FROM table_1_10601843_2 WHERE Tenant = bursaspor open
BiLSTM	SELECT Tenant FROM table_1_10601843_2 WHERE Tenant = tenant
Encoder-Decoder	SELECT Opening FROM table_1_10601843_2 WHERE Stadium = tenant open
Column Attention	SELECT Opening FROM table_1_10601843_2 WHERE Stadium = tenant open
Pointer Network	SELECT Opening FROM table_1_10601843_2 WHERE Stadium = tenant

in ground truth, this includes the generation of multiple correct SQL queries, however, with one ground truth. Errors in the mixing of AGGREGATION because of the ambiguity of the question which is often also confuses humans. Unexplainable errors where models randomly predicts incorrect information.

This section further partially answers the last research question; How can the prediction failures when generating semantically equivalent SQL queries be mitigated?. The errors in ground truth shows that the mitigation comes from fixing the ground truth. The need for multiple ground truth for multiple correct SQL queries. The errors in the mixing of AGGREGATION because of the ambiguity of the question which is often also confuses

humans. This can be specifically trained. Error in the WHERE-Clause, this includes the text degeneration, out-of-vocabulary models can assist with capturing vocabulary when WHERE-Clause.

4.5 Discussion

This research aims to comparatively evaluate the performance of the sequence-to-sequence architectures for the translation of natural language to SQL. To further analyse the incorrect predictions output generated from the sequence-to-sequence models. The research questions quest to be answer are as follows:
Which sequence-to-sequence architecture is more efficient in translating natural language to SQL?.
What are the key factors of failures in the generation of semantically equivalent SQL statements?.
How are failures mitigated in the generation semantically equivalent SQL queries?.
These were answered by performing sets of experiments using the WikiSQL dataset on the five sequence-to-sequence models. The following models performance LSTM, BiLSTM, Encoder-Decoder, Column Attention, and Pointer Network, were evaluated on a different set of training hyper-parameters. This is to evaluate their training process, the architectural performance, and their capacity to generate SQL queries

4.5.1 Training level Results

The attributes influencing the performance of the sequence-to-sequence models investigated include the architectural and dataset-setup, hidden nodes, batch sizes, and word embeddings for the successfully translation of natural language to SQL queries. The learning rate used was across all the architectures which showed effective in the convergence of all the sequence-to-sequence models. The investigated hidden nodes of 100, 200, and 300 highlights the importance of choosing correct hidden nodes for each model; not all equivalent hidden nodes showed to perform the same for each model across each component. The **LSTM** and **BiLSTM** showed to have a performance decrease with 300 hidden nodes. While it did not give any training performance issues when increasing the hidden nodes, increasing the hidden nodes is note to decrease performance on the AGGREGATION component. This shows that increasing the hidden nodes much more than needed overloads the models causing training accuracy to decrease, high generalisation error due to overfitting and high variance decreases the performance accuracy (Goodfellow et al. 2016, Goldberg 2017). The **Encoder-Decoder, Column Attention, and Pointer Network** showed to have a performance accuracy increase which

is noted and shows not all models can reach optimal with the same hidden nodes. This shows that the more complex the architecture gets the more the neural network hidden nodes are needed. None of the components decreased with the increase of hidden nodes. Neural network model capacity is controlled both by the number of nodes and the number of layers in the model (Goodfellow et al. 2016). A high-capacity such as a low capacity affects the performance of the models in such way that a decrease or increase can occurs (Goodfellow et al. 2016). An increase in the training time is noted when increasing the hidden nodes. Another important hyper-parameter is the batch size that was also investigated on two batch sizes of 32 and 64. Both the batch size of 32 and 64 is noted to affect the performance accuracy and the training times for all the models. The the LSTM only trained with on the batch size 32. When the batch sizes are increased to 64, the LSTM capacity to train on the large dataset is presented to not be able to handle a large data processing at a time, however, compared to the BiLSTM which did not have capacity issues it was able to handle training for both batch sizes of 32 and batch sizes of 64. Furthermore, the Encoder-Decoder, Column Attention, and Pointer Network did not have capacity issues when training with batch sizes of 64. The batch size of 32 provided an optimal performance accuracy better than batch sizes of 64. Although taking a much longer training time. It is worth noting the importance of the batch size choices when training sequence-to-sequence models as they affect the performance accuracy. Another important training information is how the words are represented for training of sequence-to-sequence models, the **FastText** and **GloVe** word embedding were evaluated on the performance of the Pointer Network model. The choice of word embedding shows the effects representing words using embedding in translation tasks. The GloVe word embedding showed its success over the FastText, specifically achieving a much better performance accuracy on the SELECT and WHERE-Clause. The performance of both FastText and GloVe word embedding for this task highlights the benefits of using word embedding to generate relation to words that are trained in sequence-to-sequence models. The general performance of FastText against the GloVe show the FastText capacity to generate similar performance to GloVe word embedding on the AGGREGATION. FastText is noted to have a poor performance on the SELECT and WHERE-Clause components, compared to the GloVe word embedding. This is an important finding on the word embedding capacity to sequence-to-sequence models accurately learn. On one hand, word embedding representation on n-gram level is capable of generating shorter tokens, as evident with the capturing of the AGGREGATION, on another hand, the longer tokens of the SELECT and WHERE-Clause are not accurately captured.

4.5.2 Testing level results

The testing level results present the final output of the sequence-to-sequence models. The results are from the final output using the optimal hyper-parameters from the training level results. The word embedding of choice for the final output trained and tested on is the GloVe word embedding, as it showed to perform well in the task of translating natural language to SQL for the SELECT and WHERE-Clause component. The **LSTM** model showed to only have training a capacity for training on a small batch size of 32. This is noted to only be the LSTM problem as all the top four models are noted to generally do not have capacity issues when training with different hyper-parameters. Although experiencing capacity problem, the LSTM on the batch of 32 showed to generate comparative results which is a surprise given the none performance when increasing batch size. However, performing lower than all the top four models. The AGGREGATION of the LSTM shows to perform as comparative as all the 4 models.
The performance of **BiLSTM** show to overcome training struggles of the LSTM and to handle much larger batch sizes, this is noted to show the capacity of the BiLSTM to train on larger training batch sizes, this also says the BiLSTM can be used in areas where resources are not enough to overcome the LSTM training problem. The BiLSTM is noted to have a performance increase compared to the LSTM. This is due to information capturing as the BiLSTM captures the information from both forward and backward, which is noted to capture and remember better. This shows not only to affect the training process and also the performance accuracy.
The **Encoder-Decoder** show the effectiveness of encoding the input, outputting the encoder of the questions and the schema of the database fed into the decoder to predict the appropriate SQL query (Ahkouk et al. 2020). The Encoder-Decoder shows to particularly boost the WHERE-Clause compared to the Column Attention, LSTM, and BiLSTM. This approach of translating natural language to SQL can yield satisfying result given that the WHERE-Clause is noted to improve. Increase from this approach can be helpful in the WHERE-Clause component if it is used appropriately, since each element on the WHERE-Clause depends on the previous one.
Column Attention is noted to be adequate to improve the SELECT component, and this can be attributed to the fact that the Column Attention captures the context of the column names which improves the performance accuracy of the column names (Xu et al. 2017). This shows consistency in the literature; even for other task attention mechanism has shown to better capture context better than general Encoder-Decoder model (Xu et al. 2017). The WHERE-Clause of the Column Attention is noted to decline. However, this does not correlate with the results (Xu et al. 2017); however, the WHERE-Clause in this literature is based on a slot fitting structure rather than as a complete sequence, this might be the reason the column attention does not reach as Encoder-Decoder model,

the Column Attention seems to not capture more on the WHERE-Clause (Ahkouk et al. 2020). When one token is faulty, and since each token is feed to the next generation operation, the rest of the process is affected, and hence, all the subsequent generated tokens are more likely to be faulty this maybe to not this could be the reason why the WHERE-Clause declines capturing all information from the WHERE-Clause, which is the column name, operation and condition. This is noted to decrease both the Logical and execution accuracies.
The **Pointer Network** is noted to capture the WHERE-Clause better than all the models even the Encoder-Decoder model, achieving a higher accuracy in the WHERE-Clause achieving much higher logical and execution accuracies, instate of token to token learning the Pointer Network copies from input to output since some queries includes with more than one token for the same condition. This shows to perform much better than the Column Attention particularly on the WHERE-Clause. This has shown to be comparative to the literature on the effects of Pointer Network to the possibility of copying from input to output (Wang, Brockschmidt & Singh 2018, Zhong et al. 2017).

The evaluation of the output measures the predicted SQL query versus the ground truth using the five metrics **BLEU-Score, Precision, Recall, F1-Score, and Exact Matching**. The **BLEU-Score** evaluates the quality of translation from one natural language to another is noted to have a high score on **LSTM**, and **BiLSTM** this is noted to be a higher BLEU-score than expected over the performance accuracy received compared to their Exact match and overall Precision and Recall accuracies. Taking the BLEU-score of each sentence in the corpus and then averaging across them may artificially inflate the score, which looking at the other performance metrics it looks like this is the case for the LSTM. This is because the BLEU-score does not consider sentence structure or capture the syntax of the prediction which is relevant when evaluating the predicted SQL query. Hence, this flow on the score, is countered by using four more metrics. The Exact Match only considers the exact string match against the ground truth, not bearing the fact that some SQL queries can be correct although not particularly equivalent to the ground truth. The exact match is noted to be particularly low, even lower than the logical and execution accuracies. This is somewhat expected results; Exact Match of the Pointer Network is higher than the rest of the models which makes is expected as it performed on both logical and execution accuracies better than all the models, a possible reason for this is that they tend to generate words frequently associated with ground truth SQL queries. The LSTM achieves a lower precision than all top four models, although the BLEU-score may seem high, it is important to keep in mind that LSTM's training and learning difficulty that have resulted in achieving significantly lower results. As the performance improve for top four models their precision is shown to improve which is also evident with their Exact Match and BLEU-score results. The

F1-score takes the harmonic mean of the precision and recalls this is noted to have an overall highest score for the Pointer Network. The investigation of the evaluation metrics shows the Pointer Network is overall best in predicting SQL queries equivalent to the ground truth compared to the top four models. It is important to have an evaluation metrics prediction that evaluates exact matching as the column names of one component can be mistakenly predicted by the other component. The testing evaluation results answers question one; by showing which models performs better on different SQL components, showing that the Column Attention works well on the column names, while the Pointer Network works well on the WHERE-Clause. These partially answers the first research question as presented in section 4.3.

4.5.3 SQL prediction output

The analysis of the output prediction presented in section 4.3.2 shows the dependency of learning from information in the question, each model showed to have different predictions given its capacity to learn. The questions are described in 5 knowledge categories as Exact Match, Paraphrased, Partial clue, External knowledge, Ambiguity, which show each model predicting the SQL queries. When the column names and conditions are explicitly mentioned on the questions, as defined as Exact Match, it is noted that all the models generate more accurate predictions. This indicates that with enough context in the question all sequence-to-sequence models are able to learn and infer to generate equivalent SQL queries. The questions with paraphrased, partial clue, and external knowledge, the Column Attention model showed to be more effective in capturing the column names for both SELECT and WHERE-Clause column names, performing better than all the four models on this task. This is because the column attention focuses on attending to the column names. Although the LSTM takes a long computationally time, its predictions are noted to be as comparable as the BiLSTM. However, while the prediction output show the capability of using single LSTM the computation overload may prevent the choice of using LSTM in this task and rather using the BiLSTM. The task of splitting the SQL queries in the three components might mean the LSTM can work on the more simpler task of predicting the AGGREGATION rather than being used in more complex components of the SELECT and WHERE-Clause. The AGGREGATION is noted to not require computationally expensive models, because the BiLSTM and LSTM can achieve comparative results to the Encoder-Decoder, Column Attention, and Pointer Network. The SELECT is noted to be successful on the Column Attention a hybrid of the Attention Mechanism has shown to output the most correct SELECT and WHERE-Clause column names as observed in the output examples. Compared to

the 4 models, the Column Attention was able to predict much better given the question being paraphrased or containing more information (Xu et al. 2017). Incorporating effective column name contextualisation may greatly improve the performance of predicting the column names (Devlin et al. 2018, Peters et al. 2018). The Encoder-Decoder and Pointer Network models predicted the most correct WHERE-Clause condition to their corresponding ground truths. The Encoder-Decoder model performs well in this tasks as some of the examples where the Pointer Network failed to predict the correct condition the Encoder-Decoder is noted to predict the correct condition. This is a particularly interesting point for the translation of natural language to SQL. However, the Pointer Network achieves more robust output strings compared to the overall predictions of all the models. This makes the Pointer Network important in that it predicts the most equivalent condition to the ground truth compared to the Encoder-Decoder model. The probability of the same vocabulary value occurring at different input indices or information being copied does not exist on the situated place it is supposed to be affects the copying mechanism (Wang, Brockschmidt & Singh 2018). This output analysis highlights an important aspects of translating natural language, and answers the first research question in Chapter 1.4; which sequence-to-sequence architecture is more efficient in translating natural language to SQL?; no one model achieves the best output for all the SQL components; the Column Attention which is a variant of Attention Mechanism can particularly focus on the column names, while the copy mechanism of the Pointer Network applied on WHERE-Clause makes a significant difference in the prediction of the condition. The AGGREGATION shows to also perform well on simple models such as LSTM and BiLSTM, however, taking into consideration the type of capacity one has, the BiLSTM shows to be capable of training in less resourced places.

This further answers the research question one by showing that not one model is fit for all the three SQL components to achieve best results. This analysis also partially answers the second research question; What are the key factors of failures in the generation of semantically equivalent SQL statements?; presenting the models capacity to learn on different question knowledge bases, excluding the Column Attention, all the models investigated struggled a lot questions which did not have enough information, partial clues, and external knowledge this highlights that model needs context to learn efficiently. Ambiguous questions also showed to suffer across all models except the Pointer Network largely because the condition was explicitly mentioned. These errors further emphasis the contextualisation need for this type of questions. This has answered the first two research questions.

4.5.4 Error Analysis

The results from the error analysis answers research question two specified in Chapter 1.4; What are the key factors of failures in the generation of semantically equivalent SQL statements? and research question three specified in Chapter 1.4; How are failures mitigated in the generation semantically equivalent SQL queries. A well known factor in using sequence-to-sequence architecture is the explainability problem as some of the errors are unexplainable. The importance of explainability comes with being able to solve the errors and improve upon the errors, this explainability problem is seen across deep learning architectures (Goodfellow et al. 2016). The errors generated by confusing the AGGREGATION keywords 'COUNT' and 'SUM' indicates the sequence-to-sequence models have not truly understood the usage of 'COUNT' and 'SUM'. This is a classic SQL query error that even human easily mix and can fall under explainability problem. As categorised by the knowledge base the more information was not in the question more errors were generated for both the SELECT and WHERE, this indicates the need to generate and get information from the table too. The IRNET (Guo et al. 2019) by using a linking and intermediate representation, this algorithms tackle the problem of SELECT and WHERE column name prediction. The idea is to define the columns and tables listed in a query, and to assign different forms to the columns depending on how they are mentioned in the question. The operations is influenced by the WHERE-Clause column as it follows the WHERE-Clause, this might mean the WHERE-Clause be generated as a slot filling problem where each component will be treated separately.This is to avoid the ripple effect of predicting wrong column name, then wrong operation and wrong condition (Xu et al. 2017). The Pointer Network has shown to improve the performance of the WHERE-Clause, the WHERE-Clause condition has various different output errors that needs various learning methods that might help improve. One of the greatest challenges occurring in natural languages is the ambiguity matter which is highlighted on the condition error analysis section, incorporating execution-guided decoding on the Pointer Network can potentially resolve the issue for generating accurate WHERE-Clause, the generator can deals with out-of-vocabulary and execution guidance with empty prediction (See et al. 2017, Wang, Tatwawadi, Brockschmidt, Huang, Mao, Polozov & Singh 2018). A major concern in this analysis is the ground truth's errors, as shown in the examples presented in the error analysis study. Some ground truth were incorrect as some of the questions in WikiSQL are created by paraphrasing queries generated automatically from the templates without considering the table contents, the meanings of the questions could change, possibly leading to ground truth errors (Wang, Tatwawadi, Brockschmidt, Huang, Mao, Polozov & Singh 2018). Although some ground truth were incorrect, it is noted that correct predictions occurred, indicating that a higher performance accuracy could have been achieved than what has been presented.

This is a major concern as retrieving information from the database should be the most correct, especially when querying databases are impact to the business. The ground truth errors fall in two categories, one where the generated ground truth is incorrect, another where there are multiple correct SQL; however, only one ground truth. Although the predicted SQL is correct it will be incorrectly classified as falls just because it does not match the ground truth. This brings the issue of having one standard ground truth given the context the question. Generate a comprehensive ground truth that can cater for multiple valid predictions is important. If the ground truth can not validate the prediction then errors generated would likely be blamed on the unexplainabilty of using sequence-to-sequence architecture although the problems are ground truth. While applying state of the art deep learning methods is effective as this study suggests, the basis comes from fixing the error analysis at the ground truth level. This helps understand the capacity of sequence-to-sequence architectures better, applying more powerful methods with the same ground truth might yield results but the underlying factor will persist which leaves us not knowing indeed if the natural language to SQL systems would ever be deployed in institutions where they are needed. The error analysis has highlighted one of the most important subjects when training natural languages. This answers both the research questions, the contributing factors for incorrect prediction of SQL queries using error analysis, and what factors can contribute in mitigating errors when generating natural language to SQL. This is to achieve better performance results and robust translation that can be trusted to be deployed in sensitive environments. Although outperforming conventional machine learning, deep learning approaches have shown to be the future of translating natural language to SQL. More research needs to be conducted on using deep learning models for the translation of natural language to SQL.

4.6 Summary

This chapter addressed the objective of this research stated in Chapter 1.3.1, to comparatively investigate the performance of sequence-to-sequence model for the translation of natural language to SQL using the WikiSQL dataset fully presented in Chapter 3. The WikiSQL consists of question-query pairs with three SQL components the AGGREGATION, SELECT, and WHERE-Clause that are trained separately. Three research questions stated in Chapter 1.4 are answered; Which sequence-to-sequence architecture is more efficient in translating natural language to SQL?, What are the key factors of failures in the generation of semantically equivalent SQL statements?, and How can prediction failures be mitigated in the generation of semantically equivalent SQL queries;

The first research question, which sequence-to-sequence architecture is more efficient in translating natural language to SQL?, is answered by a comparative investigating the five sequence-to-sequence architectures; LSTM, BiLSTM, Encoder-Decoder, Column Attention, Pointer Network on the WikiSQL dataset to translate natural language to SQL. The comparative investigation included the training on embeddings, hidden nodes, learning rate, and batch sizes. The components are trained individually to determine the performance accuracy of each model on each SQL component. The results on section 4.3 provides a partial answer to the research questions, the results presented show an overall the Pointer Network performs best. This is given by the high performance on the logical and execution accuracies, this is further confirmed with the Exact match and BLEU-score on section 4.3.1. However, a further analysis on output prediction in section 4.3.2 categorised in question knowledge groups where knowledge groups are Exact, Paraphrase, Partial, External knowledge, Ambiguous respectively; conclusively show that not one model is capable of effectively generating all SQL component. The BiLSTM/LSTM in the section show to effectively presents better AGGREGATION. According to the analysis on the knowledge group the Column attention shows to be effective at outputting correct column names even on questions that do not consist of enough table information. Finally the Pointer Network shows to have a prediction output of correct WHERE-Clause than all the investigated models. This is shown on section 4.3.6 that presents the quality of the where clause string. The copying method shows to be critical in the development of translation of natural language to SQL. This shows the impact of each model on each SQL component.

The second question, what are the key factors of failures in the generation of semantically equivalent SQL statements?, is partially answered in section 4.3.2 and finally concluded in the error analysis in section 4.4. The questions categorised in knowledge bases show the capabilities of the models to learn from information given. The Pointer Network, Encoder-Decoder, LSTM, and BiLSTM struggle to generate constant correct SQL queries when information is not Exact Match. this are model capacity errors where models are unable to fully learn due to various factors which include the leak of information or needing external information to infer from. Section 4.4 is an error analysis that investigates various factors contributing to the incorrect prediction of the SQL queries are identified. Four phases, AGGREGATION, Column names, Operation, and Condition, were investigated where error analysis looked at randomly selected errors that are prevalent in the query examples. The error analysis presents the challenges of using natural languages and the difficulty in understanding some of the contexts in the questions where the architectures are unable to compute and return the correct SQL queries. Some errors generated were confusing and random as nothing from question explained why they were generated. This is a big problem if the explainability of architecture output

is vague. Factors including the confusion of 'SUM' and 'COUNT' keywords were found to occur across all the architectures; however, this type of error can also be confused by humans. Although WHERE-Clause's performance was low across all models compared to AGGREGATION and SELECT, the error analysis showed that the performance decrease was from the various subcomponents including the column names, operation, and condition. The condition although consisted of various errors, the errors generated by text degeneration showed to be prevalent in many examples and across all the models. The error analysis indicated the need to capture context for the translation of natural language to SQL. The errors generated from the SELECT and WHERE-Clause were noted to occur when the column names were either unavailable, paraphrased, and not explicitly mentioned. Another factor included column names that have similar names; it has been noted that all models struggled with predicting the correct column names. The operation showed to have random errors or error generation on both the ground truth and the model prediction. The main challenges with translating natural language to SQL are the complexity of natural languages. The third and final research question, how are failures mitigated in the generation semantically equivalent SQL queries?, is answered by the error analysis, the ambiguous questions highlights the need to understand context in questions. Another important mitigating factor is fixing the ground truth, and generating multiple ground truth for the multiple valid SQL queries. The ground truth is the constant factor throughout all the models, even if improvements can be achieved, evaluating on wrong ground truth will not help achieve optimal results. This research questions achieves the all objectives presented in Chapter 1. The next chapter concludes this research.

Chapter 5

Conclusion

5.1 Overview

The use of artificial intelligence has significantly transformed the field of natural language processing research. In semantic parsing, a growth in using deep learning architectures has been implemented in many literatures, where systems are built to understand the context. This has been shown to outperform traditional rule-based methods, as evident from literature. In natural language processing, this research investigated the semantic parsing of natural language to SQL using a deep learning approach. The main objective of this research was to investigate specifically sequence-to-sequence models for the translation of natural language to SQL. The study answers three research questions as stated; Which is the most effective sequence-to-sequence architecture for translating natural language to SQL?; What are the main factors that cause SQL queries to be incorrectly predicted?; and How can the prediction failures when generating semantically equivalent SQL queries be mitigated?. Section 5.2 summarises the methods and results that were taken to answer these research questions, and these are the research achievements. Section 5.3 presents areas for future work.

5.2 Research Achievements

The main objective of this research is to comparatively evaluate sequence-to-sequence models. To achieve this, five sequence-to-sequence models were investigated; LSTM, BiLSTM, Encoder-Decoder, Column Attention, and Pointer Network with the use of the WikiSQL dataset that is a standard dataset for this task. To achieve the objective of this research, these research questions were investigated.

5.2.1 Which sequence-to-sequence architecture is more efficient in translating natural language to SQL?

Section 4.5.2 and Section 4.5.3 presented the architectural and SQL query output results respectively. These results provided the basis of the answer to research question one. The results on section 4.5.2 provided a partial answer that showed that the Pointer Network was the overall best performing model with the results in the evaluation metrics. However, the SQL output presented in section 4.5.3 showed that no one model is best at predicting all the SQL components. The results presented, highlighted an important information on the learning models and how the models learn from the information provided. According to the results presented in section 4.5.3, the ability of an architecture to learn high dependent on the information provided. The WikiSQL datasets consist of questions that are categorised into five knowledge groups; Exact match, Paraphrasing, Partial clue, External knowledge, ambiguity. Each model output was analysed to check how each model handles questions and learns from the question. The Column Attention showed to be robust in the three knowledge groups; paraphrasing, partial clue, external knowledge, where other models failed because of their lack of column information or external knowledge needed. This shows the strong learning capacity of the Column Attention towards the column names. While the Pointer Network was an overall best model amongst the investigated sequence-to-sequence model, it failed at this task that Column Attention successfully does well at. The Pointer Network also presented the ability to generate WHERE-Clause conditions even for questions which were ambiguous, this showed more strength compared to all models. The Encoder-Decoder generated good SQL components although failing in the three knowledge base when information was not contained in the question. Although performing well in the task of predicting showed to suffer from text degeneration in the prediction of the WHERE-Clause which made the Pointer Network as a preferred method to generate the WHERE-Clause. Section 4.5.2 also shows the quality of the predicted AGGREGATION by the LSTM and BiLSTM. Although the LSTM like the BiLSTM produces generally good AGGREGATION, it does not have good training capacity and fails to train in environments that have limited resources as seen when training the LSTM.

The results answers the research question by showing that no one model can fit all the SQL components, the BiLSTM showed to perform well on the AGGREGATION, the Column Attention showed to boost the performance of SELECT and WHERE-Clause column names, while the Pointer Network showed to boost the performance of the WHERE-Clause. It can be recommended that the models are used in parallel to each other to achieve best results for SQL component, that is the BiLSTM is used on

the AGGREGATION, the Column Attention is used on the Column names of both the SELECT and WHERE-Clause and the Pointer Network is used on the WHERE-Clause.

5.2.2 What are the main factors that cause SQL queries to be incorrectly predicted?

The second research question was answered with an analysis of the SQL predicted output on section 4.5.3 and further answered with an error analysis on section 4.5.4. Section 4.5.3 presented the capacity at which each model is able to learn with a category of five question knowledge bases. This shows that by the information available in the question some models are capable or not capable to learn. The LSTM, BiLSTM, Encoder-Decoder, and Pointer Network struggled with dealing with questions that fell into these categories; paraphrasing, partial clue, external knowledge, this is a source of errors from the model by not having the capacity to learn. The LSTM, BiLSTM, Encoder-Decoder, and Column Attention struggled with the prediction of the ambiguous questions. Some errors were not explainable as to why some models got them wrong; while others got them right, this is a common problem with deep learning models. Identifying the source of errors in models, especially when deep learning models are not explainable, may aid in determining which models are prone to which errors on which knowledge groups. The error analysis in section 4.5.3 further investigates the roots of errors with 4 categories; AGGREGATION, Column name, Operation, and Conditions. This revealed a number of errors, including those caused by multiple SQL predictions where there is only one ground truth, the ground truth where the expected SQL is right but the ground truth is incorrect, and other errors that were previously listed as unexplainable. Another aspect that affects output with incorrect SQL queries, but not with correct SQL queries, is getting several valid SQL queries per search.

The first step in improving the performance of sequence-to-sequence models is to identify the cause of failures. The error analysis looks at the parts of the entire generation process are responsible for prediction errors. This includes everything from data creation to sequence-to-sequence models.

5.2.3 How can prediction failures when generating semantically equivalent SQL queries be mitigated?

Section 4.5.3 identified which sequence-to-sequence model performed best on the different SQL tasks and section 4.5.4 identified the sources of errors. The third research

question investigated how the errors can be mitigated for the sequence-to-sequence models. The results on this were presented in section 4.4 obtained through error analysis. The error analysis was presented in 4 categories; AGGREGATION, Column names, operation, and condition, where each category investigated predictions of each model. The error analysis highlighted the areas where training with sequence-to-sequence models failed to predict correct SQL queries. Using contextual representation to further improve the learning of context errors given by ambiguity in questions and also with paraphrased, external knowledge, and questions consisting of many column name clues. Incorporating the column names from the table might improve the performance of the column names especially for questions that need external information. The errors are generated by only having a single SQL ground truth for questions that consist of multiple valid SQL queries, the need to generate multiple ground truth. Fixing errors in the ground truth can further help the evaluation process where the evaluation can be trusted to output true performance accuracies.

This research has provided results that are important for this field of parsing natural language to SQL. The answer to research question one shows that there is no one sequence-to-sequence architecture that can handle all the SQL components. In order to get achieve the best results a recommendation to consider using different architectures for different SQL components. The second research question investigated the sources of prediction errors which were found to be due to various factors, this includes dataset, lack of information, ground truth, and multiple valid SQL for each question. The third research question infers from the error analysis on ways to mitigate the generated errors by using various tools; using contextualise models for ambiguous questions, training models with their suitable question dataset that they can handle, developing multiple SQL ground truth for questions with multiple valid SQL queries. This section concludes this research thesis. The next section details the future work

5.3 Future Work

The research results and analysis of this work have brought several research avenues that can be explored to improve the translation of natural language to SQL querying the database. Further research can be done on the following areas as future work;

1. An investigation in the use of contextual embedding (A recent advancement in natural language processing); that captures context of question where the same word may have different context this helps to better disambiguate between the correct sense of a given the word.

2. An effort to purify the ground truth so as to provide more accurate results for deep learning models that are used for natural language to SQL.

3. An effort to querete the ground truth dataset that associates multiple valid SQL queries, in order to investigate a case where there is more than one SQL query for one natural language.

4. An investigation in the text degeneration algorithms for the sequence-to-sequence models because of their tendency to predict the same information specifically in the WHERE-Clause.

These are some of the high level topics that can be investigated for future work in the area of translating natural language to SQL.

Bibliography

Aditya, B., Bhalotia, G., Chakrabarti, S., Hulgeri, A., Nakhe, C. & Sudarshanxe, S. (2002), Banks: Browsing and keyword searching in relational databases, *in* 'VLDB'02: Proceedings of the 28th International Conference on Very Large Databases', Elsevier, pp. 1083–1086.

Affolter, K., Stockinger, K. & Bernstein, A. (2019), 'A comparative survey of recent natural language interfaces for databases', *The VLDB Journal* **28**(5), 793–819.

Ahkouk, K., Machkour, M., Majhadi, K. & Mama, R. (2020), 'Seq2seq vs sketch filling structure for natural language to sql translation', *The International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences* **44**, 7–11.

Androutsopoulos, I., Ritchie, G. D. & Thanisch, P. (1995), 'Natural language interfaces to databases–an introduction', *Natural language engineering* **1**(1), 29–81.

Artzi, Y., Das, D. & Petrov, S. (2014), 'Learning compact lexicons for ccg semantic parsing'.

Athiwaratkun, B., Wilson, A. G. & Anandkumar, A. (2018), 'Probabilistic fasttext for multi-sense word embeddings', *arXiv preprint arXiv:1806.02901* .

Audet, M. (2008), 'Document vectors'. US Patent App. 11/944,163.

Bahdanau, D., Cho, K. & Bengio, Y. (2014), 'Neural machine translation by jointly learning to align and translate', *arXiv preprint arXiv:1409.0473* .

Bahdanau, D., Chorowski, J., Serdyuk, D., Brakel, P. & Bengio, Y. (2016), End-to-end attention-based large vocabulary speech recognition, *in* 'Acoustics, Speech and Signal Processing (ICASSP), 2016 IEEE International Conference on', IEEE, pp. 4945–4949.

Bakshi, R. K., Kaur, N., Kaur, R. & Kaur, G. (2016), Opinion mining and sentiment analysis, *in* '2016 3rd International Conference on Computing for Sustainable Global Development (INDIACom)', IEEE, pp. 452–455.

Basik, F., Hättasch, B., Ilkhechi, A., Usta, A., Ramaswamy, S., Utama, P., Weir, N., Binnig, C. & Cetintemel, U. (2018), Dbpal: A learned nl-interface for databases, *in* 'Proceedings of the 2018 International Conference on Management of Data', ACM, pp. 1765–1768.

Bengio, Y. (2012), Practical recommendations for gradient-based training of deep architectures, *in* 'Neural networks: Tricks of the trade', Springer, pp. 437–478.

Bengio, Y., Ducharme, R., Vincent, P. & Jauvin, C. (2003), 'A neural probabilistic language model', *Journal of machine learning research* **3**(2), 1137–1155.

Berant, J., Chou, A., Frostig, R. & Liang, P. (2013), Semantic parsing on freebase from question-answer pairs, *in* 'Proceedings of the 2013 conference on empirical methods in natural language processing', pp. 1533–1544.

Bhirud, N. S., Bhavsar, R. & Pawar, B. (2017), 'Grammar checkers for natural languages: a review', *International Journal on Natural Language Computing (IJNLC)* **6**(4), 51–62.

Bishop, C. M. et al. (1995), *Neural networks for pattern recognition*, Oxford university press.

Bojanowski, P., Grave, E., Joulin, A. & Mikolov, T. (2017), 'Enriching word vectors with subword information', *Transactions of the Association for Computational Linguistics* **5**, 135–146.

Bordawekar, R. & Shmueli, O. (2017), Using word embedding to enable semantic queries in relational databases, *in* 'Proceedings of the 1st Workshop on Data Management for End-to-End Machine Learning', pp. 1–4.

Bottou, L., Curtis, F. E. & Nocedal, J. (2018), 'Optimization methods for large-scale machine learning', *Siam Review* **60**(2), 223–311.

Cai, H., Zheng, V. W. & Chang, K. C.-C. (2018), 'A comprehensive survey of graph embedding: Problems, techniques, and applications', *IEEE Transactions on Knowledge and Data Engineering* **30**(9), 1616–1637.

Cai, Q. & Yates, A. (2013), Large-scale semantic parsing via schema matching and lexicon extension, *in* 'Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)', pp. 423–433.

Cai, R., Xu, B., Yang, X., Zhang, Z., Li, Z. & Liang, Z. (2017), 'An encoder-decoder framework translating natural language to database queries', *arXiv preprint arXiv:1711.06061* .

Cambria, E. & White, B. (2014), 'Jumping nlp curves: A review of natural language processing research', *IEEE Computational intelligence magazine* **9**(2), 48–57.

Cheng, J., Reddy, S., Saraswat, V. & Lapata, M. (2017), 'Learning structured natural language representations for semantic parsing', *arXiv preprint arXiv:1704.08387* .

Cho, K., Van Merriënboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H. & Bengio, Y. (2014), 'Learning phrase representations using rnn encoder-decoder for statistical machine translation', *arXiv preprint arXiv:1406.1078* .

Chowdhary, K. (2020), Natural language processing, *in* 'Fundamentals of Artificial Intelligence', Springer, pp. 603–649.

Chung, J., Gulcehre, C., Cho, K. & Bengio, Y. (2014), 'Empirical evaluation of gated recurrent neural networks on sequence modeling', *arXiv preprint arXiv:1412.3555* .

Cichocki, A., Unbehauen, R. & Swiniarski, R. W. (1993), *Neural networks for optimization and signal processing*, Vol. 253, wiley New York.

Ciresan, D. C., Meier, U., Gambardella, L. M. & Schmidhuber, J. (2011), Convolutional neural network committees for handwritten character classification, *in* '2011 International Conference on Document Analysis and Recognition', IEEE, pp. 1135–1139.

Collobert, R. & Weston, J. (2008), A unified architecture for natural language processing: Deep neural networks with multitask learning, *in* 'Proceedings of the 25th international conference on Machine learning', pp. 160–167.

Collobert, R., Weston, J., Bottou, L., Karlen, M., Kavukcuoglu, K. & Kuksa, P. (2011), 'Natural language processing (almost) from scratch', *Journal of Machine Learning Research* **12**(8), 2493–2537.

Cumby, C. M. & Roth, D. (2003), On kernel methods for relational learning, *in* 'Proceedings of the 20th International Conference on Machine Learning (ICML-03)', pp. 107–114.

Dai, A. M., Olah, C. & Le, Q. V. (2015), 'Document embedding with paragraph vectors', *arXiv preprint arXiv:1507.07998* .

Dale, R. (2016), 'The return of the chatbots', *Natural Language Engineering* **22**(5), 811–817.

Dale, R., Moisl, H. & Somers, H. (2000), *Handbook of natural language processing*, CRC Press.

Damljanovic, D., Tablan, V. & Bontcheva, K. (2008), A text-based query interface to owl ontologies., *in* 'LREC'.

Devlin, J., Chang, M.-W., Lee, K. & Toutanova, K. (2018), 'Bert: Pre-training of deep bidirectional transformers for language understanding', *arXiv preprint arXiv:1810.04805* .

Dong, L. & Lapata, M. (2016), 'Language to logical form with neural attention', *arXiv preprint arXiv:1601.01280* .

Drakos, G. (2019), 'What is a recurrent neural networks(rnns) and gated recurrent unit(grus)', https://towardsdatascience.com/what-is-a-recurrent-nns-and-gated-recurrent-unit-grus-ea71d2a05a69. [Online; accessed February, 2019].

Duchi, J., Hazan, E. & Singer, Y. (2011), 'Adaptive subgradient methods for online learning and stochastic optimization.', *Journal of machine learning research* **12**(7).

Feldman, R. (2013), 'Techniques and applications for sentiment analysis', *Communications of the ACM* **56**(4), 82–89.

Filling, S. D. I. S. (2017), 'Conference on empirical methods in natural language processing (and forerunners)'.

Furbach, U., Glöckner, I. & Pelzer, B. (2010), 'An application of automated reasoning in natural language question answering', *Ai Communications* **23**(2-3), 241–265.

Ge, R. & Mooney, R. (2005), A statistical semantic parser that integrates syntax and semantics, *in* 'Proceedings of the Ninth Conference on Computational Natural Language Learning (CoNLL-2005)', pp. 9–16.

Giordani, A. & Moschitti, A. (2009), Semantic mapping between natural language questions and sql queries via syntactic pairing, *in* 'International Conference on Application of Natural Language to Information Systems', Springer, pp. 207–221.

Giordani, A. & Moschitti, A. (2012), Translating questions to sql queries with generative parsers discriminatively reranked, *in* 'Proceedings of COLING 2012: Posters', pp. 401–410.

Glorot, X. & Bengio, Y. (2010), Understanding the difficulty of training deep feedforward neural networks, *in* 'Proceedings of the thirteenth international conference on artificial intelligence and statistics', pp. 249–256.

Glorot, X., Bordes, A. & Bengio, Y. (2011), Deep sparse rectifier neural networks, *in* 'Proceedings of the fourteenth international conference on artificial intelligence and statistics', pp. 315–323.

Goddard, C. (2011), *Semantic analysis: A practical introduction*, Oxford University Press.

Goldberg, Y. (2017), 'Neural network methods for natural language processing', *Synthesis Lectures on Human Language Technologies* **10**(1), 1–309.

Goodfellow, I., Bengio, Y. & Courville, A. (2016), *Deep learning*, MIT press.

Graves, A., Jaitly, N. & Mohamed, A.-r. (2013), Hybrid speech recognition with deep bidirectional lstm, *in* '2013 IEEE workshop on automatic speech recognition and understanding', IEEE, pp. 273–278.

Grefenstette, E., Blunsom, P., De Freitas, N. & Hermann, K. M. (2014), 'A deep architecture for semantic parsing', *arXiv preprint arXiv:1404.7296* .

Grosz, B. (1983), Team: A transportable natural language interface system, *in* 'Proceedings of the Conference on Applied Natural Language Processing (1983)', Association for Computational Linguistics.

Gu, J., Lu, Z., Li, H. & Li, V. O. (2016), 'Incorporating copying mechanism in sequence-to-sequence learning', *arXiv preprint arXiv:1603.06393* .

Guo, J., Zhan, Z., Gao, Y., Xiao, Y., Lou, J.-G., Liu, T. & Zhang, D. (2019), 'Towards complex text-to-sql in cross-domain database with intermediate representation', *arXiv preprint arXiv:1905.08205* .

He, Y. & Young, S. (2006), 'Spoken language understanding using the hidden vector state model', *Speech Communication* **48**(3-4), 262–275.

Heinecke, J. (2010), 'Text language identification'. US Patent 7,689,409.

Hinton, G. E., Srivastava, N., Krizhevsky, A., Sutskever, I. & Salakhutdinov, R. R. (2012), 'Improving neural networks by preventing co-adaptation of feature detectors', *arXiv preprint arXiv:1207.0580* .

Hirschberg, J. & Manning, C. D. (2015), 'Advances in natural language processing', *Science* **349**(6245), 261–266.

Ho, Y. & Wookey, S. (2019), 'The real-world-weight cross-entropy loss function: Modeling the costs of mislabeling', *IEEE Access* **8**, 4806–4813.

Hochreiter, S. & Schmidhuber, J. (1997), 'Long short-term memory', *Neural computation* **9**(8), 1735–1780.

Holtzman, A., Buys, J., Du, L., Forbes, M. & Choi, Y. (2019), 'The curious case of neural text degeneration', *arXiv preprint arXiv:1904.09751* .

Huang, E. H., Socher, R., Manning, C. D. & Ng, A. Y. (2012), Improving word representations via global context and multiple word prototypes, *in* 'Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)', pp. 873–882.

Hwang, W., Yim, J., Park, S. & Seo, M. (2019), 'A comprehensive exploration on wikisql with table-aware word contextualization', *arXiv preprint arXiv:1902.01069* .

Ioffe, S. & Szegedy, C. (2015), 'Batch normalization: Accelerating deep network training by reducing internal covariate shift', *arXiv preprint arXiv:1502.03167* .

Iyer, S., Konstas, I., Cheung, A., Krishnamurthy, J. & Zettlemoyer, L. (2017), 'Learning a neural semantic parser from user feedback', *arXiv preprint arXiv:1704.08760* .

Jamil, H. M. (2017), Knowledge rich natural language queries over structured biological databases, *in* 'Proceedings of the 8th ACM International Conference on Bioinformatics, Computational Biology, and Health Informatics', pp. 352–361.

Jia, R. & Liang, P. (2016), 'Data recombination for neural semantic parsing', *arXiv preprint arXiv:1606.03622* .

Joulin, A., Grave, E., Bojanowski, P., Douze, M., Jégou, H. & Mikolov, T. (2016), 'Fasttext. zip: Compressing text classification models', *arXiv preprint arXiv:1612.03651* .

Joulin, A., Grave, E., Bojanowski, P. & Mikolov, T. (2016), 'Bag of tricks for efficient text classification', *arXiv preprint arXiv:1607.01759* .

Jozefowicz, R., Zaremba, W. & Sutskever, I. (2015), An empirical exploration of recurrent network architectures, *in* 'International Conference on Machine Learning', pp. 2342–2350.

Jun-Li, W., Xiao-Min, W. & Ya-Xing, Y. (2017), A word embedding generation method based on flexible length context, *in* 'Computer and Communications (ICCC), 2017 3rd IEEE International Conference on', IEEE, pp. 2564–2568.

Kamath, A. & Das, R. (2018), 'A survey on semantic parsing', *arXiv preprint arXiv:1812.00978* .

Kate, R. J., Wong, Y. W. & Mooney, R. J. (2005), Learning to transform natural to formal languages, *in* 'AAAI', pp. 1062–1068.

Kate, R. & Mooney, R. (2006), Using string-kernels for learning semantic parsers, *in* 'Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics', pp. 913–920.

Keskar, N. S., Mudigere, D., Nocedal, J., Smelyanskiy, M. & Tang, P. T. P. (2016), 'On large-batch training for deep learning: Generalization gap and sharp minima', *arXiv preprint arXiv:1609.04836* .

Kingma, D. P. & Ba, J. (2014), 'Adam: A method for stochastic optimization', *arXiv preprint arXiv:1412.6980* .

Krenker, A., Bester, J. & Kos, A. (2011), Introduction to the artificial neural networks, *in* 'Artificial neural networks-methodological advances and biomedical applications', InTech.

Krishnamurthy, J. (2016), Probabilistic models for learning a semantic parser lexicon, *in* 'Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies', pp. 606–616.

Krishnamurthy, J. & Mitchell, T. (2012), Weakly supervised training of semantic parsers, *in* 'Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning', pp. 754–765.

Krishnamurthy, J. & Mitchell, T. M. (2015), 'Learning a compositional semantics for freebase with an open predicate vocabulary', *Transactions of the Association for Computational Linguistics* **3**, 257–270.

Krizhevsky, A., Sutskever, I. & Hinton, G. E. (2012), Imagenet classification with deep convolutional neural networks, *in* 'Advances in neural information processing systems', pp. 1097–1105.

Kumar, A., Irsoy, O., Ondruska, P., Iyyer, M., Bradbury, J., Gulrajani, I., Zhong, V., Paulus, R. & Socher, R. (2016), Ask me anything: Dynamic memory networks for natural language processing, *in* 'International conference on machine learning', pp. 1378–1387.

Kwiatkowksi, T., Zettlemoyer, L., Goldwater, S. & Steedman, M. (2010), Inducing probabilistic ccg grammars from logical form with higher-order unification, *in* 'Proceedings of the 2010 conference on empirical methods in natural language processing', pp. 1223–1233.

Kwiatkowski, T., Zettlemoyer, L., Goldwater, S. & Steedman, M. (2011), Lexical generalization in ccg grammar induction for semantic parsing, *in* 'Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing', pp. 1512–1523.

Lample, G., Ballesteros, M., Subramanian, S., Kawakami, K. & Dyer, C. (2016), 'Neural architectures for named entity recognition', *arXiv preprint arXiv:1603.01360* .

Le, Q. & Mikolov, T. (2014), Distributed representations of sentences and documents, *in* 'International conference on machine learning', pp. 1188–1196.

Levy, O. & Goldberg, Y. (2014), Dependency-based word embeddings, *in* 'Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)', pp. 302–308.

Li, F. & Jagadish, H. (2014*a*), 'Constructing an interactive natural language interface for relational databases', *Proceedings of the VLDB Endowment* **8**(1), 73–84.

Li, F. & Jagadish, H. V. (2014*b*), Nalir: an interactive natural language interface for querying relational databases, *in* 'Proceedings of the 2014 ACM SIGMOD international conference on Management of data', pp. 709–712.

Li, J. & Jurafsky, D. (2015), 'Do multi-sense embeddings improve natural language understanding?', *arXiv preprint arXiv:1506.01070* .

Liang, P., Jordan, M. I. & Klein, D. (2013), 'Learning dependency-based compositional semantics', *Computational Linguistics* **39**(2), 389–446.

Ling, W., Grefenstette, E., Hermann, K. M., Kočiskỳ, T., Senior, A., Wang, F. & Blunsom, P. (2016), 'Latent predictor networks for code generation', *arXiv preprint arXiv:1603.06744* .

Liu, Z., Lin, Y. & Sun, M. (2020), Word representation, *in* 'Representation Learning for Natural Language Processing', Springer, pp. 13–41.

Lodhi, H., Saunders, C., Shawe-Taylor, J., Cristianini, N. & Watkins, C. (2002), 'Text classification using string kernels', *Journal of Machine Learning Research* **2**(2), 419–444.

Lu, Z. & Li, H. (2013), A deep architecture for matching short texts, *in* 'Advances in neural information processing systems', pp. 1367–1375.

Lukovnikov, D., Chakraborty, N., Lehmann, J. & Fischer, A. (2018), 'Translating natural language to sql using pointer-generator networks and how decoding order matters', *arXiv preprint arXiv:1811.05303* .

Luong, M.-T., Le, Q. V., Sutskever, I., Vinyals, O. & Kaiser, L. (2015), 'Multi-task sequence to sequence learning', *arXiv preprint arXiv:1511.06114* .

Luong, M.-T., Pham, H. & Manning, C. D. (2015), 'Effective approaches to attention-based neural machine translation', *arXiv preprint arXiv:1508.04025* .

Ma, Y. & Cambria, E. (2018), 'Concept-based embeddings for natural language processing', *arXiv preprint arXiv:1807.05519* .

Manning, C. D., Surdeanu, M., Bauer, J., Finkel, J. R., Bethard, S. & McClosky, D. (2014), The stanford corenlp natural language processing toolkit, *in* 'Proceedings of 52nd annual meeting of the association for computational linguistics: system demonstrations', pp. 55–60.

Marquez, L. & Salgado, J. G. (2000), 'Machine learning and natural language processing'.

Masters, D. & Luschi, C. (2018), 'Revisiting small batch training for deep neural networks', *arXiv preprint arXiv:1804.07612* .

McCulloch, W. S. & Pitts, W. (1943), 'A logical calculus of the ideas immanent in nervous activity', *The bulletin of mathematical biophysics* **5**(4), 115–133.

McDonald, D. D. (1992), 'Krisp'.

Melucci, M. (2005), Context modeling and discovery using vector space bases, *in* 'Proceedings of the 14th ACM international conference on Information and Knowledge Management', pp. 808–815.

Mikolov, T., Chen, K., Corrado, G. & Dean, J. (2013), 'Efficient estimation of word representations in vector space', *arXiv preprint arXiv:1301.3781* .

Mikolov, T., Karafiát, M. & Burget, L. (2010), Jan cernocky, and sanjeev khudanpur. 2010. recurrent neural network based language model, *in* 'Eleventh annual conference of the international speech communication association', pp. 1045–1048.

Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S. & Dean, J. (2013), Distributed representations of words and phrases and their compositionality, *in* 'Advances in neural information processing systems', pp. 3111–3119.

Mikolov, T., Yih, W.-t. & Zweig, G. (2013), Linguistic regularities in continuous space word representations, *in* 'Proceedings of the 2013 conference of the north american chapter of the association for computational linguistics: Human language technologies', pp. 746–751.

Minaee, S., Kalchbrenner, N., Cambria, E., Nikzad, N., Chenaghlu, M. & Gao, J. (2020), 'Deep learning based text classification: A comprehensive review', *arXiv preprint arXiv:2004.03705* .

Mitchell, T. (1997), 'Machine learning, mcgraw-hill higher education', *New York* .

Mooney, R. J. (2007), Learning for semantic parsing, *in* 'International Conference on Intelligent Text Processing and Computational Linguistics', Springer, pp. 311–324.

Mou, L., Lu, Z., Li, H. & Jin, Z. (2017), Coupling distributed and symbolic execution for natural language queries, *in* 'International Conference on Machine Learning', pp. 2518–2526.

Nair, V. & Hinton, G. E. (2010), Rectified linear units improve restricted boltzmann machines, *in* 'ICML'.

Nallapati, R., Xiang, B. & Zhou, B. (2016), 'Sequence-to-sequence rnns for text summarization'.

Nasr, G. E., Badr, E. & Joun, C. (2002), Cross entropy error function in neural networks: Forecasting gasoline demand., *in* 'FLAIRS conference', pp. 381–384.

Neubig, G. (2017), 'Neural machine translation and sequence-to-sequence models: A tutorial', *arXiv preprint arXiv:1703.01619* .

Nielsen, M. A. (2015), *Neural networks and deep learning*, Vol. 2018, Determination press San Francisco, CA.

Noda, K., Yamaguchi, Y., Nakadai, K., Okuno, H. G. & Ogata, T. (2015), 'Audio-visual speech recognition using deep learning', *Applied Intelligence* **42**(4), 722–737.

Olah, C. (2015), 'Understanding LSTMs', https://colah.github.io/posts/2015-08-Understanding-LSTMs/. [Online; accessed August 27, 2015].

Owen, B. & Steiner, J. (2009), 'Email filtering system and method'. US Patent 7,580,982.

Palmer, M., Gildea, D. & Xue, N. (2010), 'Semantic role labeling', *Synthesis Lectures on Human Language Technologies* **3**(1), 1–103.

Panchal, F. S. & Panchal, M. (2014), 'Review on methods of selecting number of hidden nodes in artificial neural network', *International Journal of Computer Science and Mobile Computing* **3**(11), 455–464.

Pandit, S. (2008), 'On a robust document classification approach using tf-idf scheme with learned, context-sensitive semantics.'.

Papineni, K., Roukos, S., Ward, T. & Zhu, W.-J. (2002), Bleu: A method for automatic evaluation of machine translation, *in* 'Proceedings of the 40th annual meeting on association for computational linguistics', Association for Computational Linguistics, pp. 311–318.

Pasupat, P. & Liang, P. (2015), 'Compositional semantic parsing on semi-structured tables', *arXiv preprint arXiv:1508.00305* .

Patil, M. G., Galande, M. V., Kekan, M. V. & Dange, M. K. (2014), 'International journal of innovative research in computer and communication engineering', *Sentiment analysis using support vector machine* **2**(1), 2607–2612.

Pedregosa, F. (2016), 'Hyperparameter optimization with approximate gradient', *arXiv preprint arXiv:1602.02355* .

Pennington, J., Socher, R. & Manning, C. (2014), Glove: Global vectors for word representation, *in* 'Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)', pp. 1532–1543.

Peters, M. E., Neumann, M., Iyyer, M., Gardner, M., Clark, C., Lee, K. & Zettlemoyer, L. (2018), 'Deep contextualized word representations', *arXiv preprint arXiv:1802.05365* .

Poon, H. (2013), Grounded unsupervised semantic parsing, *in* 'Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics', Vol. 1, pp. 933–943.

Popescu, A.-M., Armanasu, A., Etzioni, O., Ko, D. & Yates, A. (2004), Modern natural language interfaces to databases: Composing statistical parsing with semantic tractability, *in* 'COLING 2004: Proceedings of the 20th International Conference on Computational Linguistics', pp. 141–147.

Popescu, A.-M., Etzioni, O. & Kautz, H. (2003), Towards a theory of natural language interfaces to databases, *in* 'Proceedings of the 8th international conference on Intelligent user interfaces', ACM, pp. 149–157.

Quirk, C., Mooney, R. & Galley, M. (2015), 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', Vol. 1, pp. 878–888.

Recognition-Fifth, S. I. C. S. (2013), 'Generation computer corporation', *Fifthgen. com. Archived from the original on* **11**.

Robertson, A. M. & Willett, P. (1998), 'Applications of n-grams in textual information systems', *Journal of Documentation* **54**(1), 48–67.

Rong, X. (2014), 'Word2vec parameter learning explained', *arXiv preprint arXiv:1411.2738* .

Rosenblatt, F. (1961), Principles of neurodynamics. perceptrons and the theory of brain mechanisms, Technical report, Cornell Aeronautical Lab Inc Buffalo NY.

Ryan, J., Lin, M.-J. & Miikkulainen, R. (1998), Intrusion detection with neural networks, *in* 'Advances in neural information processing systems', pp. 943–949.

Sadowski, P. (2016), 'Notes on backpropagation', *homepage: https://www. ics. uci. edu/pjsadows/notes. pdf (online)* .

Saha, D., Floratou, A., Sankaranarayanan, K., Minhas, U. F., Mittal, A. R. & Özcan, F. (2016), 'Athena: an ontology-driven system for natural language querying over relational data stores', *Proceedings of the VLDB Endowment* **9**(12), 1209–1220.

Sak, H., Senior, A. & Beaufays, F. (2014), Long short-term memory recurrent neural network architectures for large scale acoustic modeling, *in* 'Fifteenth annual conference of the international speech communication association'.

Salton, G., Wong, A. & Yang, C.-S. (1975), 'A vector space model for automatic indexing', *Communications of the ACM* **18**(11), 613–620.

Sanh, V., Wolf, T. & Ruder, S. (2019), A hierarchical multi-task approach for learning embeddings from semantic tasks, *in* 'Proceedings of the AAAI Conference on Artificial Intelligence', Vol. 33, pp. 6949–6956.

Sathick, K. J. & Jaya, A. (2015), 'Natural language to sql generation for semantic knowledge extraction in social web sources', *Indian Journal of Science and Technology* **8**(1), 1.

Saxen, S. (2017), 'Artificial Neuron Networks(Basics)— Introduction to Neural Networks', https://becominghuman.ai/artificial-neuron-networks-basics-introduction-to-neural-networks-3082f1dcca8c. [Online; accessed 26-October-2017].

Schaefer, A. M., Udluft, S. & Zimmermann, H.-G. (2008), 'Learning long-term dependencies with recurrent neural networks', *Neurocomputing* **71**(13-15), 2481–2488.

Schalkoff, R. J. (1989), *Digital image processing and computer vision*, Vol. 286, Wiley New York.

Schuster, M. & Paliwal, K. K. (1997), 'Bidirectional recurrent neural networks', *IEEE transactions on Signal Processing* **45**(11), 2673–2681.

See, A., Liu, P. J. & Manning, C. D. (2017), 'Get to the point: Summarization with pointer-generator networks', *arXiv preprint arXiv:1704.04368* .

Shawe-Taylor, J. & Cristianini, N. (2000), *Support vector machines*, Vol. 2, Cambridge University Press Cambridge.

Shi, Z., Shi, M. & Li, C. (2017), The prediction of character based on recurrent neural network language model, *in* 'Computer and Information Science (ICIS), 2017 IEEE/ACIS 16th International Conference on', IEEE, pp. 613–616.

Singh, S. P., Kumar, A., Singh, L., Bhargava, M., Goyal, K. & Sharma, B. (2016), Frequency based spell checking and rule based grammar checking, *in* '2016 International Conference on Electrical, Electronics, and Optimization Techniques (ICEEOT)', IEEE, pp. 4435–4439.

Socher, R. (2014), Recursive deep learning for natural language processing and computer vision, PhD thesis, Citeseer.

Socher, R., Mohammadi, M. & Mundra, R. (2015), 'Cs 224d: Deep learning for nlp'.

Song, D., Schilder, F., Smiley, C., Brew, C., Zielund, T., Bretz, H., Martin, R., Dale, C., Duprey, J. & Miller, T. (2015), Tr discover: A natural language interface for querying and analyzing interlinked datasets, *in* 'International Semantic Web Conference', Springer, pp. 21–37.

Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I. & Salakhutdinov, R. (2014), 'Dropout: A simple way to prevent neural networks from overfitting', *The journal of machine learning research* **15**(1), 1929–1958.

Steedman, M. (2000), *The syntactic process*, Vol. 24, MIT press Cambridge, MA.

Steinbiss, V., Tran, B.-H. & Ney, H. (1994), Improvements in beam search, *in* 'Third International Conference on Spoken Language Processing'.

Stevenson, A. (2010), *Oxford dictionary of English*, Oxford University Press, USA.

Su, Y., Liu, Q., Liu, Q., Huang, Z., Yin, Y., Chen, E., Ding, C., Wei, S. & Hu, G. (2018), Exercise-enhanced sequential modeling for student performance prediction, *in* 'Thirty-Second AAAI Conference on Artificial Intelligence'.

Suliman, A. & Zhang, Y. (2015), 'A review on back-propagation neural networks in the application of remote sensing image classification', *Journal of Earth Science and Engineering* **5**, 52–65.

Sun, Y., Tang, D., Duan, N., Ji, J., Cao, G., Feng, X., Qin, B., Liu, T. & Zhou, M. (2018), 'Semantic parsing with syntax-and table-aware sql generation', *arXiv preprint arXiv:1804.08338* .

Sutskever, I., Vinyals, O. & Le, Q. V. (2014*a*), Sequence-to-sequence learning with neural networks, *in* 'Advances in neural information processing systems', pp. 3104–3112.

Sutskever, I., Vinyals, O. & Le, Q. V. (2014*b*), Sequence to sequence learning with neural networks, *in* 'Advances in neural information processing systems', pp. 3104–3112.

Tang, L. R. & Mooney, R. J. (2001), Using multiple clause constructors in inductive logic programming for semantic parsing, *in* 'European Conference on Machine Learning', Springer, pp. 466–477.

Taylor, P. (2009), *Text-to-speech synthesis*, Cambridge university press.

Tieleman, T. & Hinton, G. (2012), 'Lecture 6.5-rmsprop: Divide the gradient by a running average of its recent magnitude', *COURSERA: Neural networks for machine learning* **4**(2), 26–31.

Turian, J., Ratinov, L. & Bengio, Y. (2010), Word representations: a simple and general method for semi-supervised learning, *in* 'Proceedings of the 48th annual meeting of the association for computational linguistics', pp. 384–394.

Turney, P. D. & Pantel, P. (2010), 'From frequency to meaning: Vector space models of semantics', *Journal of artificial intelligence research* **37**, 141–188.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł. & Polosukhin, I. (2017), Attention is all you need, *in* 'Advances in neural information processing systems', pp. 5998–6008.

Vinyals, O., Fortunato, M. & Jaitly, N. (2015), Pointer networks, *in* 'Advances in Neural Information Processing Systems', pp. 2692–2700.

Walter, S., Unger, C., Cimiano, P. & Bär, D. (2012), Evaluation of a layered approach to question answering over linked data, *in* 'International Semantic Web Conference', Springer, pp. 362–374.

Waltinger, U., Tecuci, D., Olteanu, M., Mocanu, V. & Sullivan, S. (2013), Usi answers: Natural language question answering over (semi-) structured industry data, *in* 'Proceedings of the Twenty-Seventh AAAI Conference on Artificial Intelligence', pp. 1471–1478.

Wang, C., Brockschmidt, M. & Singh, R. (2018), 'Pointing out sql queries from text'.

Wang, C., Tatwawadi, K., Brockschmidt, M., Huang, P.-S., Mao, Y., Polozov, O. & Singh, R. (2018), 'Robust text-to-sql generation with execution-guided decoding', *arXiv preprint arXiv:1807.03100* .

Wang, M., Lu, S., Zhu, D., Lin, J. & Wang, Z. (2018), A high-speed and low-complexity architecture for softmax function in deep learning, *in* '2018 IEEE Asia Pacific Conference on Circuits and Systems (APCCAS)', IEEE, pp. 223–226.

Wang, W. Y. & Yang, D. (2015), That's so annoying!!!: A lexical and frame-semantic embedding based data augmentation approach to automatic categorization of annoying behaviors using# petpeeve tweets, *in* 'Proceedings of the 2015 conference on empirical methods in natural language processing', pp. 2557–2563.

Warren, D. H. & Pereira, F. C. (1982), 'An efficient easily adaptable system for interpreting natural language queries', *American journal of computational linguistics* **8**(3-4), 110–122.

Wong, Y. W. (2005), *Learning for semantic parsing using statistical machine translation techniques*, Computer Science Department, University of Texas at Austin.

Wong, Y. W. & Mooney, R. (2006), Learning for semantic parsing with statistical machine translation, *in* 'Proceedings of the Human Language Technology Conference of the NAACL, Main Conference', pp. 439–446.

Woods, W. (1972), 'The lunar sciences natural language information system', *BBN report* .

Woods, W. A. (1979), *Semantics for a question-answering system*, Vol. 27, Dissertations-G.

Xu, J. & Du, Q. (2019), A deep investigation into fasttext, *in* '2019 IEEE 21st International Conference on High Performance Computing and Communications; IEEE 17th International Conference on Smart City; IEEE 5th International Conference on Data Science and Systems (HPCC/SmartCity/DSS)', IEEE, pp. 1714–1719.

Xu, X., Liu, C. & Song, D. (2017), 'Sqlnet: Generating structured queries from natural language without reinforcement learning', *arXiv preprint arXiv:1711.04436* .

Yang, Z., Wang, C., Zhang, Z. & Li, J. (2019), 'Mini-batch algorithms with online step size', *Knowledge-Based Systems* **165**, 228–240.

Yavuz, S., Gur, I., Su, Y. & Yan, X. (2018), What it takes to achieve 100% condition accuracy on wikisql, *in* 'Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing', pp. 1702–1711.

Yih, W.-t., He, X. & Meek, C. (2014), Semantic parsing for single-relation question answering, *in* 'Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)', pp. 643–648.

Yin, Y., Huang, Z., Chen, E., Liu, Q., Zhang, F., Xie, X. & Hu, G. (2018), Transcribing content from structural images with spotlight mechanism, *in* 'Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining', pp. 2643–2652.

Young, T., Hazarika, D., Poria, S. & Cambria, E. (2017), 'Recent trends in deep learning based natural language processing', *arXiv preprint arXiv:1708.02709* .

Yu, D. & Li, J. (2017), 'Recent progresses in deep learning based acoustic models', *IEEE/CAA Journal of Automatica Sinica* **4**(3), 396–409.

Yu, L., Hermann, K. M., Blunsom, P. & Pulman, S. (2014), 'Deep learning for answer sentence selection', *arXiv preprint arXiv:1412.1632* .

Yu, T., Li, Z., Zhang, Z., Zhang, R. & Radev, D. (2018), 'Typesql: Knowledge-based type-aware neural text-to-sql generation', *arXiv preprint arXiv:1804.09769* .

Zelle, J. M. & Mooney, R. J. (1996), Learning to parse database queries using inductive logic programming, *in* 'Proceedings of the national conference on artificial intelligence', pp. 1050–1055.

Zettlemoyer, L. & Collins, M. (2007), Online learning of relaxed ccg grammars for parsing to logical form, *in* 'Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)', pp. 678–687.

Zettlemoyer, L. S. & Collins, M. (2009), 'Learning context-dependent mappings from sentences to logical form'.

Zhang, D. & Lee, W. S. (2003), Question classification using support vector machines, *in* 'Proceedings of the 26th annual international ACM SIGIR conference on Research and development in informaion retrieval', pp. 26–32.

Zhong, H., Chen, Z., Qin, C., Huang, Z., Zheng, V. W., Xu, T. & Chen, E. (2020), 'Adam revisited: a weighted past gradients perspective', *Frontiers of Computer Science* **14**(5), 1–16.

Zhong, V., Xiong, C. & Socher, R. (2017), 'Seq2sql: Generating structured queries from natural language using reinforcement learning', *arXiv preprint arXiv:1709.00103* .

Appendix

Table 1: Generated output results of the SQL query, ground-truth SQL query, questions, and column name

Type	Sentence
Question 1	What is terrence ross' nationality?
Table	['Player', 'No.', 'Nationality', 'Position', 'Years in Toronto', 'School/Club Team']
Ground Truth	SELECT Nationality FROM table_1_10015132_16 WHERE Player = terrence ross
LSTM	SELECT Years in Toronto FROM table_1_10015132_16 WHERE Player = terrence ross
BiLSTM	SELECT Years in Toronto FROM table_1_10015132_16 WHERE Player = terrence ross
Encoder-Decoder	SELECT Nationality FROM table_1_10015132_16 WHERE Player = Terrence Ross
Col Attention	SELECT Nationality FROM table_1_10015132_16 WHERE Player = Terrence Ross
Pointer Network	SELECT Nationality FROM table_1_10015132_16 WHERE Player = Terrence Ross
Question 2	how many schools or teams had jalen rose?
Table	['Player', 'No.', 'Nationality', 'Position', 'Years in Toronto', 'School/Club Team']
Ground Truth	SELECT COUNT(School/Club Team) FROM table_1_10015132_16 WHERE Player = Jalen Rose
LSTM	SELECT COUNT(School/Club Team) FROM table_1_10015132_16 WHERE Player = Jalen Rose
BiLSTM	SELECT COUNT(School/Club Team) FROM table_1_10015132_16 WHERE Player = jalen rose
Encoder-Decoder	SELECT COUNT(No.) FROM table_1_10015132_16 WHERE Player = EQL rose
Col Attention	SELECT COUNT(School/Club Team) FROM table_1_10015132_16 WHERE School/Club Team = jalen rose
Pointer Network	SELECT COUNT(School/Club Team) FROM Team table_1_10015132_16 School/Club Team = jalen rose
Question 3	What clu was in toronto 1995-96?

Continue on the next page

Table 1: Generated output results of the SQL query, ground-truth SQL query, questions, and column name

Type	Sentence
Table	['Player', 'No.', 'Nationality', 'Position', 'Years in Toronto', 'School/Club Team']
Ground Truth	SELECT School/Club Team FROM table_1_10015132_16 WHERE Years in Toronto = 1995-96
LSTM	SELECT School/Club Team FROM table_1_10015132_16 WHERE Years in Toronto = 1995-96
BiLSTM	SELECT Nationality FROM table_1_10015132_16 WHERE Player = toronto 1995-96
Encoder-Decoder	SELECT Nationality FROM table_1_10015132_16 WHERE Player = toronto 1995-96
Col Attention	SELECT Nationality FROM table_1_10015132_16 WHERE School/Club Team = toronto 1995-96
Pointer Network	SELECT Nationality FROM table_1_10015132_16 WHERE Player = toronto 1995-96
Question 4	List the # for ships commissioned on december 18, 1965
Table	['#', 'Shipyard', 'Laid down', 'Launched', 'Commissioned', 'Fleet', 'Status']
Ground Truth	SELECT # FROM table_1_1014206_2 WHERE Commissioned = December 18, 1965
LSTM	SELECT # FROM table_1_1014206_2 WHERE Fleet = december 18, 1965
BiLSTM	SELECT # FROM table_1_1014206_2 WHERE Fleet = commissioned commissioned AND Launched = december 18, 1965
Encoder-Decoder	SELECT # FROM table_1_1014206_2 WHERE Status = commissioned 18, 1965, 18
Col Attention	SELECT # FROM table_1_1014206_2 WHERE Launched = december 18, 1965
Pointer Network	SELECT # FROM table_1_1014206_2 WHERE Launched = december 18, 1965
Question 5	What could a spanish coronel be addressed as in the commonwealth military?
Table	['Equivalent NATO Rank Code', 'Rank in Spanish', 'Rank in English', 'Commonwealth equivalent', 'US Air Force equivalent']
Ground Truth	SELECT Commonwealth equivalent FROM table_1_1015521_2 WHERE Rank in Spanish = Coronel

Continue on the next page

Table 1: Generated output results of the SQL query, ground-truth SQL query, questions, and column name

Type	Sentence
LSTM	SELECT Rank in Spanish FROM table_1_1015521_2 WHERE Equivalent NATO Rank Code = spanish
BiLSTM	SELECT Rank in Spanish FROM table_1_1015521_2 WHERE Equivalent NATO Rank Code = commonwealth
Encoder-Decoder	SELECT Commonwealth equivalent FROM table_1_1015521_2 WHERE Equivalent NATO Rank Code = spanish coronel
Col Attention	SELECT Commonwealth equivalent FROM table_1_1015521_2 WHERE Rank in English = commonwealth military
Pointer Network	SELECT Commonwealth equivalent FROM table_1_1015521_2 WHERE Commonwealth equivalent = commonwealth military
Question 6	Which wrestlers have had 2 reigns?
Table	[‘Rank Each wrestlers total number of days as champion are ranked highest to lowest; wrestlers with the same number mean that they are tied for that certain rank.’, ‘Wrestler’, ‘# of reigns’, ‘Combined defenses’, ‘Combined days’]
Ground Truth	SELECT Wrestler FROM table_1_10182508_5 WHERE # of reigns = 2
LSTM	SELECT COUNT(Wrestler) FROM table_1_10182508_5 WHERE Rank Each wrestlers total number of days as champion are ranked highest to lowest; wrestlers with the same number mean that they are tied for that certain rank. = wrestler AND Rank Each wrestlers total number of days as champion are ranked highest to lowest; wrestlers with the same number mean that they are tied for that certain rank. = combined AND Rank Each wrestlers total number of days as champion are ranked highest to lowest; wrestlers with the same number mean that they are tied for that certain rank. = combined
BiLSTM	SELECT Combined defenses FROM table_1_10182508_5 WHERE Rank Each wrestlers total number of days as champion are ranked highest to lowest; wrestlers with the same number mean that they are tied for that certain rank. = 2
Encoder-Decoder	SELECT COUNT(Wrestler) FROM table_1_10182508_5 WHERE Rank Each wrestlers total number of days as champion are ranked highest to lowest; wrestlers with the same number mean that they are tied for that certain rank. = 2

Continue on the next page

Table 1: Generated output results of the SQL query, ground-truth SQL query, questions, and column name

Type	Sentence
Col Attention	SELECT COUNT(Wrestler) FROM table_1_10182508_5 WHERE Wrestler =
Pointer Network	Wrestler FROM table_1_10182508_5 WHERE # of reigns = 2
Question 7	what is the total number of total w–l where doubles w-l is 11–11
Table	[‘Player’, ‘Total W–L’, ‘Singles W–L’, ‘Doubles W–L’, ‘Ties played’, ‘Debut’, ‘Years played’]
Ground Truth	SELECT COUNT(Total W-L) FROM table_1_10294071_1 WHERE Doubles W–L = 11–11
LSTM	SELECT COUNT(Total W–L) FROM table_1_10294071_1 WHERE Doubles W–L = 11–11
BiLSTM	SELECT COUNT(Total W–L) FROM table_1_10294071_1 WHERE Doubles W–L = 11–11
Encoder-Decoder	SELECT COUNT(Total W–L) FROM table_1_10294071_1 WHERE Doubles W–L = 11–11
Col Attention	SELECT COUNT(Total W–L) FROM table_1_10294071_1 WHERE Doubles W–L = 11–11
Pointer Network	SELECT COUNT(Total W–L) FROM table_1_10294071_1 WHERE Doubles W–L = 11–11
Question 8	What is the Frequency at the Market/Rank of Burlington - Plattsburgh , Vermont - New York /143?
Table	[‘Calls’, ‘Frequency’, ‘Branding’,‘Format’, ‘Market/Rank’, ‘Timeslot’, ‘Group owner’]
Ground Truth	SELECT COUNT(Frequency) FROM table_1_10333757_1 WHERE Market/Rank = Burlington - Plattsburgh , Vermont - New York /143
LSTM	SELECT Frequency FROM table_1_10333757_1 WHERE Branding = burlington - plattsburgh- vermont- new york/143, vermont
BiLSTM	SELECT Frequency FROM table_1_10333757_1 WHERE Branding = burlington - plattsburgh , vermont - new york /143
Encoder-Decoder	SELECT Frequency FROM table_1_10333757_1 WHERE Frequency = burlington - plattsburgh , vermont - new york /143
Col Attention	SELECT Frequency FROM table_1_10333757_1 WHERE Group owner = burlington - plattsburgh , vermont - new york /143

Continue on the next page

Table 1: Generated output results of the SQL query, ground-truth SQL query, questions, and column name

Type	Sentence
Pointer Network	SELECT Frequency FROM table_1_10333757_1 WHERE Format = burlington - plattsburgh , vermont - new york /143
Question 9	What is the Format for Branding of 1290 wkbk w281au 104.1?
Table	['Calls', 'Frequency', 'Branding', 'Format', 'Market/Rank', 'Timeslot', 'Group owner']
Ground Truth	SELECT Format FROM table_1_10333757_1 WHERE Branding = 1290 WKBK W281AU 104.1
LSTM	SELECT Format FROM table_1_10333757_1 WHERE Branding = 1290 wkbk w281au 104.1
BiLSTM	SELECT Format FROM table_1_10333757_1 WHERE Branding = 1290 wkbk w281au 104.1
Encoder-Decoder	SELECT Format FROM table_1_10333757_1 WHERE Branding = 1290 WKBK W281AU 104.1
Col Attention	SELECT Format FROM table_1_10333757_1 WHERE Branding = 1290 wkbk w281au 104.1
Pointer network	SELECT Format FROM table_1_10333757_1 WHERE Timeslot = 1290 wkbk w281au 104.1
Question 10	Which Market/Rank is associated with WCRN calls?
Table	['Calls', 'Frequency', 'Branding', 'Format', 'Market/Rank', 'Timeslot', 'Group owner']
Ground Truth	SELECT Market/Rank FROM table_1_10333757_1 WHERE Calls = WCRN
LSTM	SELECT Branding FROM table_1_10333757_1 WHERE Calls = wcrn
BiLSTM	SELECT Branding FROM table_1_10333757_1 WHERE Calls = wcrn
Encoder-Decoder	SELECT Market/Rank FROM table_1_10333757_1 WHERE Calls = WCRN
Col Attention	SELECT Branding FROM table_1_10333757_1 WHERE Calls = wcrn
Pointer Network	SELECT Branding FROM table_1_10333757_1 WHERE Calls = wcrn calls
Question 11	What is the regulated retail price for the tariff code ff0 prs?
Table	['Scheme', 'Tariff code', 'BTs retail price (regulated)', 'Approx premium', 'Prefixes']
Ground Truth	SELECT BTs retail price (regulated) FROM table_1_10408617_5 WHERE Tariff code = ff0 PRS
LSTM	SELECT BTs retail price (regulated) FROM table_1_10408617_5 WHERE Tariff code = ff0 prs

Continue on the next page

Table 1: Generated output results of the SQL query, ground-truth SQL query, questions, and column name

Type	Sentence
BiLSTM	SELECT BTs retail price (regulated) FROM table_1_10408617_5 WHERE Tariff code = ff0 prs
Encoder-Decoder	SELECT BTs retail price (regulated) FROM table_1_10408617_5 WHERE Tariff code = ff0 prs
Col Attention	SELECT BTs retail price (regulated) FROM table_1_10408617_5 WHERE Tariff code = ff0 prs
Pointer Network	SELECT BTs retail price (regulated) FROM table_1_10408617_5 WHERE Tariff code = ff0
Question 12	How many tariff codes have a bts retail price of 2p/min or inclusive?
Table	['Scheme', 'Tariff code', 'BTs retail price (regulated)', 'Approx premium', 'Prefixes']
Ground Truth	SELECT COUNT(Tariff code) FROM table_1_10408617_5 WHERE BTs retail price (regulated) = 2p/min or inclusive
LSTM	SELECT COUNT(Tariff code) FROM table_1_10408617_5 WHERE BTs retail price (regulated) = inclusive AND Tariff code = 2p/min
BiLSTM	SELECT COUNT(Tariff code) FROM table_1_10408617_5 WHERE BTs retail price (regulated) = 2p/min
Encoder-Decoder	SELECT COUNT(Tariff code) FROM table_1_10408617_5 WHERE BTs retail price (regulated) = inclusive inclusive
Col Attention	SELECT COUNT(Tariff code) FROM table_1_10408617_5 WHERE Approx premium = or inclusive
Pointer Network	SELECT COUNT(Tariff code) FROM table_1_10408617_5 WHERE BTs retail price (regulated) = 2p/min
Question 13	How many hours were flown in each of the years where more than 64379058.0 kilometers were flown?
Table	['Year', 'Aircraft kilometers', 'Departures', 'Flying hours', 'Passengers', 'Seat factor', 'Employees', 'Profit/loss']
Ground Truth	SELECT Flying hours FROM table_1_105344_2 WHERE Aircraft kilometers > 64379058.0
LSTM	SELECT COUNT(Flying hours) FROM table_1_105344_2 WHERE Seat factor > 64379058.0
BiLSTM	SELECT COUNT(Flying hours) FROM table_1_105344_2 WHERE Flying hours > 64379058.0

Continue on the next page

Table 1: Generated output results of the SQL query, ground-truth SQL query, questions, and column name

Type	Sentence
Encoder-Decoder	SELECT COUNT(Flying hours) FROM table_1_105344_2 WHERE Seat factor = 64379058.0
Col Attention	SELECT Flying hours FROM table_1_105344_2 WHERE Departures > 64379058.0
Pointer Network	SELECT COUNT(Flying hours) FROM table_1_105344_2 WHERE Passengers > 64379058.0
Question 14	What is the lowest group to earn 886.6 points?
Table	['Rank', 'Member Association', 'Points', 'Group stage', 'Play-off', 'AFC Cup']
Ground Truth	SELECT MIN(Group stage) FROM table_1_1059743_2 WHERE Points = 886.6
LSTM	SELECT MIN(Group stage) FROM table_1_1059743_2 WHERE
BiLSTM	SELECT MIN(Group stage) FROM table_1_1059743_2 WHERE Points = 886.6
Encoder-Decoder	SELECT MIN(Rank) FROM table_1_1059743_2 WHERE Rank = 886.6
Col Attention	SELECT MIN(Play-off) FROM table_1_1059743_2 WHERE
Pointer Network	SELECT MIN(Member Association) FROM table_1_1059743_2 WHERE Points = 886.6
Question 15	If % lunsford is 51.82% what is the % mcconnell in Letcher?
Table	['County', 'Precincts', 'Lunsford', '% Lunsford', 'McConnell', '% McConnell', 'Total']
Ground Truth	SELECT % McConnell FROM table_1_10586064_2 WHERE % Lunsford = 51.82%
LSTM	SELECT % Lunsford FROM table_1_10586064_2 WHERE % Lunsford = 51.82%
BiLSTM	SELECT % Lunsford FROM table_1_10586064_2 WHERE % Lunsford = 51.82%
Encoder-Decoder	SELECT % Lunsford FROM table_1_10586064_2 WHERE % Lunsford = 51.82%
Col Attention	SELECT % Lunsford FROM table_1_10586064_2 WHERE % Lunsford = % lunsford is 51.82% what
Pointer Network	SELECT % Lunsford FROM table_1_10586064_2 WHERE % Lunsford = 51.82%

Table 2: Randomly selected generated output where incorrect prediction are found of the prediction for the sequence-to-sequence models

Model	Model
Question 1	What club was in toronto 1995-96
Table	['Player', 'No.', 'Nationality', 'Position', 'Years in Toronto', 'School/Club Team']
Ground Truth	SELECT School/Club Team FROM table_1_10015132_16 WHERE Years in Toronto = 1995-96
LSTM	SELECT Nationality FROM table_1_10015132_16 WHERE School/Club Team = toronto 1995-96
BiLSTM	SELECT Nationality FROM table_1_10015132_16 WHERE Player = toronto 1995-96
Encoder-Decoder	SELECT Nationality FROM table_1_10015132_16 WHERE Player = toronto 1995-96
Col Attention	SELECT Nationality FROM table_1_10015132_16 WHERE School/Club Team = toronto 1995-96
Pointer Network	SELECT Nationality FROM table_1_10015132_16 WHERE Player = toronto 1995-96
Question 2	What is the catalog number named callanetics: 10 years younger in 10 hours
Table	['Year', 'Title', 'Format', 'Studio', 'Release Date', 'Copyright Information', 'Catalog Number']
Ground Truth	SELECT Catalog Number FROM table_1_11222744_3 WHERE Title = Callanetics: 10 Years Younger In 10 Hours
LSTM	SELECT Catalog Number FROM table_1_11222744_3 WHERE Studio = callanetics: 10 years younger in 10 AND Year = 10 hours
BiLSTM	SELECT Catalog Number FROM table_1_11222744_3 WHERE Year = 10 AND Format = callanetics: 10 years younger in 10
Encoder-Decoder	SELECT Copyright Information FROM table_1_11222744_3 WHERE Year = callanetics: 10 younger AND Studio = 10 hours
Col Attention	SELECT Catalog Number FROM table_1_11222744_3 WHERE Year = hours AND Catalog Number = hours
Pointer Network	SELECT Copyright Information FROM table_1_11222744_3 WHERE Release Date = callanetics: 10 years younger in 10
Question 3	What is the tone for gen.sing. plague?

Continue on the next page

Table 2: Randomly selected generated output where incorrect prediction are found of the prediction for the sequence-to-sequence models

Model	Model
Table	[‘Nominative’, ‘Old orthography’, ‘New orthography’, ‘/e/ or /æ/ (IPA)’, ‘Tone (Latvian notation: / ∼ /- level, / ∧ / - broken)’, ‘Translation’]
Ground Truth	SELECT Tone (Latvian notation: / ∼ / - level, / ∧ / - broken) FROM table_1_10409754_5 WHERE Translation = Gen.Sing. plague
LSTM	SELECT /e/ or /æ/ (IPA) FROM table_1_10409754_5 WHERE Nominative = gen.sing
BiLSTM	SELECT Nominative FROM table_1_10409754_5 WHERE Old orthography = plague
Encoder-Decoder	SELECT Translation FROM table_1_10409754_5 WHERE Old orthography = gen.sing
Col Attention	SELECT /e/ or /æ/ (IPA) FROM table_1_10409754_5 WHERE /e/ or /æ/ (IPA) = .
Pointer Network	SELECT New orthography FROM table_1_10409754_5 WHERE New orthography = plague
Question 4	What is the smallest possible radius?
Table	[‘Star (Pismis24-#)’, ‘Spectral type’, ‘Magnitude (M bol)’, ‘Temperature (K)’, ‘Radius (R ⊙)’, ‘Mass (M ⊙)’]
Ground Truth	SELECT MIN(Radius (R ⊙)) FROM table_1_10432351_1
LSTM	SELECT Star (Pismis24-#) FROM table_1_10432351_1 WHERE
BiLSTM	SELECT Magnitude (M bol) FROM table_1_10432351_1 WHERE
Encoder-Decoder	SELECT Magnitude (M bol) FROM table_1_10432351_1 WHERE
Col Attention	SELECT Radius (R ⊙) FROM table_1_10432351_1 WHERE Radius (R ⊙) = radius
Pointer Network	SELECT Spectral type FROM table_1_10432351_1
Question 5	If a radius is 10, what is the lowest possible mass?
Table	[‘Star (Pismis24-#)’, ‘Spectral type’, ‘Magnitude (M bol)’, ‘Temperature (K)’, ‘Radius (R ⊙)’, ‘Mass (M ⊙)’]
Ground Truth	SELECT MIN(Mass (M ⊙)) table_1_10432351_1 WHERE Radius (R ⊙) = 10
LSTM	SELECT MIN(Temperature (K)) FROM table_1_10432351_1 WHERE Spectral type = 10
BiLSTM	SELECT MIN(Temperature (K)) FROM table_1_10432351_1 WHERE Star (Pismis24-#) = 10

Continue on the next page

Table 2: Randomly selected generated output where incorrect prediction are found of the prediction for the sequence-to-sequence models

Model	Model
Encoder-Decoder	SELECT MIN(Mass (M ⊙)) FROM table_1_10432351_1 WHERE Star (Pismis24-#) = 10
Col Attention	SELECT MIN(Mass (M ⊙)) FROM table_1_10432351_1 WHERE Radius (R ⊙) = ,
Pointer Network	SELECT MIN(Mass (M ⊙)) FROM table_1_10432351_1 WHERE Radius (R ⊙) = 10
Question 6	What percentage of seats were filled in 2006?
Table	['Year', 'Aircraft kilometers', 'Departures', 'Flying hours', 'Passengers', 'Seat factor', 'Employees', 'Profit/loss']
Ground Truth	SELECT Seat factor FROM table_1_105344_2 WHERE Year = 2006
LSTM	SELECT Flying hours FROM table_1_105344_2 WHERE Year = 2006
BiLSTM	SELECT Aircraft kilometers FROM table_1_105344_2 WHERE Year = 2006
Encoder-Decoder	SELECT Passengers FROM table_1_105344_2 WHERE Year = 2006
Col Attention	SELECT Departures FROM table_1_105344_2 WHERE Year = 2006
Pointer Network	SELECT Seat factor FROM table_1_105344_2 WHERE Year = 2006
Question 7	How many hours were flown in each of the years where more than 64379058.0 kilometers were flown?
Table	['Year', 'Aircraft kilometers', 'Departures', 'Flying hours', 'Passengers', 'Seat factor', 'Employees', 'Profit/loss']
Ground Truth	SELECT Flying hours FROM table_1_105344_2 WHERE Aircraft kilometers > 64379058.0
LSTM	SELECT COUNT(Flying hours) FROM table_1_105344_2 WHERE Seat factor > 64379058.0
BiLSTM	SELECT COUNT(Flying hours) FROM table_1_105344_2 WHERE Flying hours > 64379058.0
Encoder-Decoder	SELECT Flying hours FROM table_1_105344_2 WHERE Aircraft kilometers > 64379058.0
Col Attention	SELECT Flying hours FROM table_1_105344_2 WHERE Departures > 64379058.0
Pointer Network	SELECT COUNT(Flying hours) FROM table_1_105344_2 WHERE Passengers > 64379058.0
Question 8	Of the years that had exactly 17096 departures, what is the greatest number of aircraft kilometers flown?

Continue on the next page

Table 2: Randomly selected generated output where incorrect prediction are found of the prediction for the sequence-to-sequence models

Model	Model
Table	['Year', 'Aircraft kilometers', 'Departures', 'Flying hours', 'Passengers', 'Seat factor', 'Employees', 'Profit/loss']
Ground Truth	SELECT MAX(Aircraft kilometers) FROM table_1_105344_2 WHERE Departures = 17096
LSTM	SELECT MAX(Aircraft kilometers) FROM table_1_105344_2 WHERE Flying hours = 17096 flown
BiLSTM	SELECT MAX(Aircraft kilometers) FROM table_1_105344_2 WHERE Departures = 17096 departures
Encoder-Decoder	SELECT MAX(Passengers) FROM table_1_105344_2 WHERE Year = 17096
Col Attention	SELECT MAX(Year) FROM table_1_105344_2 WHERE Departures = departures
Pointer Network	SELECT MAX(Aircraft kilometers) FROM table_1_105344_2 WHERE Profit/loss = 17096 departures
Question 9	Which winning team beat the New York Yankees?
Table	['Year', 'Game or event', 'Date contested', 'League or governing body', 'Sport', 'Winning team', 'Losing team', 'Final score']
Ground Truth	SELECT Winning team FROM table_1_10548224_1 WHERE Losing team = New York Yankees
LSTM	SELECT Winning team FROM table_1_10548224_1 WHERE Game or event = new york yankees
BiLSTM	SELECT Winning team FROM table_1_10548224_1 WHERE Winning team = new york yankees
Encoder-Decoder	SELECT Winning team FROM table_1_10548224_1 WHERE Sport = new york yankees
Col Attention	SELECT Winning team FROM table_1_10548224_1 WHERE Losing team = new york yankees
Pointer Network	SELECT Losing team FROM table_1_10548224_1 WHERE Year = new york yankees
Question 10	What was the final score for the game that was contested on February 1, 2009?
Table	['Year', 'Game or event', 'Date contested', 'League or governing body', 'Sport', 'Winning team', 'Losing team', 'Final score']

Continue on the next page

Table 2: Randomly selected generated output where incorrect prediction are found of the prediction for the sequence-to-sequence models

Model	Model
Ground Truth	SELECT Final score FROM table_1_10548224_1 WHERE Date contested = February 1, 2009
LSTM	SELECT Final score FROM table_1_10548224_1 WHERE Year = february 1, 2009
BiLSTM	SELECT Final score FROM table_1_10548224_1 WHERE Date contested = february 1, 2009
Encoder-Decoder	SELECT Final score FROM table_1_10548224_1 WHERE Year = february 1, 2009
Col Attention	SELECT Final score FROM table_1_10548224_1 WHERE Date contested = february 1, 2009
Pointer Network	SELECT Final score FROM table_1_10548224_1 WHERE Year = february 1, 2009
Question 11	what's the total number of south australia with victoria value of 2173
Table	['Crop (kilotonnes)', 'New South Wales', 'Victoria', 'Queensland', 'Western Australia', 'South Australia', 'Tasmania', 'Total']
Ground Truth	SELECT COUNT(South Australia) FROM table_1_1057262_2 WHERE Victoria = 2173
LSTM	SELECT COUNT(New South Wales) FROM table_1_1057262_2 WHERE Tasmania = 2173
BiLSTM	SELECT COUNT(South Australia) FROM table_1_1057262_2 WHERE Queensland = 2173
Encoder-Decoder	SELECT COUNT(South Australia) FROM table_1_1057262_2 WHERE Victoria = 2173
Col Attention	SELECT COUNT(New South Wales) FROM table_1_1057262_2 WHERE Victoria = 2173
Pointer Network	SELECT COUNT New South Wales FROM table_1_1057262_2 WHERE Queensland = 2173
Question 12	What is the county of precints 515?
Table	['County', 'Precincts', 'Lunsford', '% Lunsford', 'McConnell', '% McConnell', 'Total']
Ground Truth	SELECT County FROM table_1_10586064_2 WHERE Precincts = 515
LSTM	SELECT County FROM table_1_10586064_2 WHERE Precincts = precints

Continue on the next page

Table 2: Randomly selected generated output where incorrect prediction are found of the prediction for the sequence-to-sequence models

Model	Model
BiLSTM	SELECT County FROM table_1_10586064_2 WHERE County = precints 515
Encoder-Decoder	SELECT County FROM table_1_10586064_2 WHERE County = precints 515
Col Attention	SELECT County FROM table_1_10586064_2 WHERE Precincts = 515
Pointer Network	SELECT County FROM table_1_10586064_2 WHERE County = precints 515
Question 13	What is the percentage of the trance- n himalaya zone that corresponds with the high hill zone is 25%?
Table	['Particulars and Characteristics', 'Shivalik Zone', 'Mid-Hill Zone', 'High hill zone', 'Trance- n Himalaya Zone']
Ground Truth	SELECT Trance- n Himalaya Zone FROM table_1_10638523_1 WHERE High hill zone = 25%
LSTM	SELECT Particulars and Characteristics FROM table_1_10638523_1 WHERE Particulars and Characteristics = 25% AND Particulars and Characteristics = corresponds
BiLSTM	SELECT Particulars and Characteristics FROM table_1_10638523_1 WHERE Particulars and Characteristics = corresponds% AND Particulars and Characteristics = 25%
Encoder-Decoder	SELECT Particulars and Characteristics FROM table_1_10638523_1 WHERE Particulars and Characteristics = trance- himalaya zone
Col Attention	SELECT Particulars and Characteristics FROM table_1_10638523_1 WHERE Shivalik Zone = trance- n himalaya
Pointer Network	SELECT Trance- n Himalaya Zone FROM table_1_10638523_1 WHERE Particulars and Characteristics = corresponds AND Particulars and Characteristics = 25%
Question 14	What is the state of Ted Stevens?
Table	['Total tenure rank', 'Uninterrupted rank', 'Name', 'State represented', 'Dates of service', 'Total tenure time', 'Uninterrupted time']
Ground Truth	SELECT State represented FROM table_1_10644188_3 WHERE Name = Ted Stevens
LSTM	SELECT State represented FROM table_1_10644188_3 WHERE Name = ted stevens

Continue on the next page

Table 2: Randomly selected generated output where incorrect prediction are found of the prediction for the sequence-to-sequence models

Model	Model
BiLSTM	SELECT State represented FROM table_1_10644188_3 WHERE Total tenure rank = ted stevens
Encoder-Decoder	SELECT Total tenure time FROM table_1_10644188_3 WHERE Name = ted ted stevens
Col Attention	SELECT State represented FROM table_1_10644188_3 WHERE Name = ted stevens
Pointer Network	SELECT State represented FROM table_1_10644188_3 WHERE Name = ted stevens
Question 15	What's the smallest number of players?
Table	['Country', 'Players', 'Standard', 'Minor', 'First title', 'Last title']
Ground Truth	SELECT MIN(Players) FROM table_1_10682862_68 WHERE
LSTM	SELECT MIN(Players) FROM table_1_10682862_68 WHERE
BiLSTM	SELECT MIN(Players) FROM table_1_10682862_68 WHERE Country = smallest
Encoder-Decoder	SELECT MIN(Standard) FROM table_1_10682862_68 WHERE
Col Attention	SELECT COUNT(Players) FROM table_1_10682862_68 WHERE Country = players
Pointer Network	SELECT MIN(Players) FROM table_1_10682862_68 WHERE

Table 3: The predicted SQL statement with its corresponding ground-truth SQL statement where no exact matching on the AGGREGATION

Type	Sentences
Question 1	What's Dorain Anneck's pick number?
Table	['Pick', 'Player', 'Position', 'Nationality', 'NHL team', 'College/junior/club team']
Ground Truth	SELECT Pick FROM table_110131293_3 WHERE Player = Dorain Anneck
LSTM	SELECT COUNT(Pick) FROM table_1_1013129_3 WHERE Player = dorain anneck
BiLSTM	SELECT Pick FROM table_1_1013129_3 WHERE Player = dorain anneck
Encoder-Decoder	SELECT Pick FROM table_1_1013129_3 WHERE Player = dorain anneck

Continue on the next page

Table 3: The predicted SQL statement with its corresponding ground-truth SQL statement where no exact matching on the AGGREGATION

Type	Sentences
Col Attention	SELECT COUNT(Pick) FROM table_1_1013129_3 WHERE Player = dorain anneck
Pointer Network	SELECT COUNT(Pick) FROM table_110131293_3 WHERE Player = dorain anne
Error	COUNT
Question 2	What is the lowest pick number?
Table	['Pick #', 'Brand (to)', 'Employee (Real name)', 'Role(s)', 'Brand (from)']
Ground Truth	SELECT Min(Pick #) FROM table_1_15353123_1 WHERE
LSTM	SELECT MAX(Pick #) FROM table_1_15353123_1 WHERE
BiLSTM	SELECT MAX(Pick #) FROM table_1_15353123_1 WHERE
Encoder-Decoder	SELECT MAX(Pick #) FROM table_1_15353123_1 WHERE
Col Attention	SELECT MAX(Pick #) FROM table_1_15353123_1 WHERE
Pointer Network	SELECT MAX(Pick #) FROM table_1_15353123_1 WHERE
Error	MAX
Question 3	What was the # for the episode "the birthday present"
Table	['#', 'Episode', 'Air Date', 'Timeslot (EST)', 'Rating', 'Share', '18–49 Rating', 'Viewers (millions)', 'Rank']
Ground Truth	SELECT MAX(#) FROM table_1_11238597_4 WHERE Episode = "The Birthday Present"
LSTM	SELECT # FROM table_1_11238597_4 WHERE Episode = "the birthday present"
BiLSTM	SELECT # FROM table_1_11238597_4 WHERE Episode = "the birthday present"
Encoder-Decoder	SELECT # FROM table_1_11238597_4 WHERE Episode = "the birthday present"
Col Attention	SELECT # FROM table_1_11238597_4 WHERE Episode = "the birthday present"
Pointer Network	SELECT # FROM table_1_11238597_4 WHERE Episode = "the birthday present"
Error	MAX
Question 4	When did the stadium where Bursaspor is the tenant open?
Table	['Stadium', 'Capacity', 'City', 'Country', 'Tenant', 'Opening']

Continue on the next page

Table 3: The predicted SQL statement with its corresponding ground-truth SQL statement where no exact matching on the AGGREGATION

Type	Sentences
Ground Truth	SELECT MIN(Opening) FROM table_1_10601843_2 WHERE Tenant = Bursaspor
LSTM	SELECT Stadium FROM table_1_10601843_2 WHERE Tenant = bursaspor open
BiLSTM	SELECT Tenant FROM table_1_10601843_2 WHERE Country = bursaspor
Encoder-Decoder	SELECT Opening FROM table_1_10601843_2 WHERE Stadium = tenant open
Col Attention	SELECT Opening FROM table_1_10601843_2 WHERE Stadium = tenant open
Pointer Network	SELECT Opening FROM table_1_10601843_2 WHERE Stadium = tenant open
Error	MIN
Question 5	What was the elimination number of the fighter who fought within 27:27?
Table	[‘Elimination number’, ‘Wrestler’, ‘Entered’, ‘Eliminated by’, ‘Method of elimination’, ‘Time’]
Ground Truth	SELECT Elimination number FROM table_1_18598175_2 WHERE Method of elimination = 27:27
LSTM	SELECT Eliminated by FROM table_1_18598175_2 WHERE Time = 27:27
BiLSTM	SELECT Elimination number FROM table_1_18598175_2 WHERE Time = 27:27
Encoder-Decoder	SELECT MIN(Elimination number) FROM table_1_18598175_2 WHERE Time = 27:27
Col Attention	SELECT MIN(Elimination number) FROM table_1_18598175_2 WHERE Time = 27:27
Pointer Network	SELECT MIN(Elimination number) FROM table_1_18598175_2 WHERE Time = 27:27
Error	MIN
Question 6	What was the elimination number of the fighter who fought within 26:15
Table	[‘Elimination number’, ‘Wrestler’, ‘Entered’, ‘Eliminated by’, ‘Method of elimination’, ‘Time’]
Ground Truth	SELECT Elimination number FROM table_1_18598175_2 WHERE Method of elimination = 26:15
LSTM	SELECT Eliminated by FROM table_1_18598175_2 WHERE Time = 26:15

Continue on the next page

Table 3: The predicted SQL statement with its corresponding ground-truth SQL statement where no exact matching on the AGGREGATION

Type	Sentences
BiLSTM	SELECT Elimination number FROM table_1_18598175_2 WHERE Time = 26:15
Encoder-Decoder	SELECT MIN(Elimination number) FROM table_1_18598175_2 WHERE Time = 26:15
Col Attention	SELECT Elimination number FROM table_1_18598175_2 WHERE Method of elimination = 26:15
Pointer Network	SELECT MAX(Elimination number) FROM table_1_18598175_2 WHERE Time = 26:15
Error	MAX
Question 7	Name the most ceiling temperature for very extra high
Table	['Maximum Ceiling Temperature', 'Temperature Rating', 'Temperature Classification', 'Color Code (with Fusible Link)', 'Glass Bulb Color']
Ground Truth	SELECT Maximum Ceiling Temperature FROM table_1_1538516_1 WHERE Temperature Classification = Very Extra High
LSTM	SELECT MAX(Temperature Classification) FROM table_1_1538516_1 WHERE Glass Bulb Color = very (high)
BiLSTM	SELECT MAX(Maximum Ceiling Temperature) FROM table_1_1538516_1 WHERE Maximum Ceiling Temperature = very extra high
Encoder-Decoder	SELECT MAX(Maximum Ceiling Temperature) FROM table_1_1538516_1 WHERE Maximum Ceiling Temperature = very extra high
Col Attention	SELECT MAX(Maximum Ceiling Temperature) FROM table_1_1538516_1 WHERE Glass Bulb Color = extra high
Pointer Network	SELECT MAX(Maximum Ceiling Temperature) FROM table_1_1538516_1 WHERE Glass Bulb Color = extra high
Error	MAX
Question 8	What was the number of race that Kevin Curtain won?
Table	['No', 'Date', 'Round', 'Circuit', 'Pole Position', 'Fastest Lap', 'Race winner', 'Report']
Ground Truth	SELECT COUNT(No) FROM table_1_10083598_1 WHERE Pole Position = Kevin Curtain
LSTM	SELECT COUNT(Round) FROM table_1_10083598_1 WHERE Circuit = kevin curtain

Continue on the next page

Table 3: The predicted SQL statement with its corresponding ground-truth SQL statement where no exact matching on the AGGREGATION

Type	Sentences
BiLSTM	SELECT COUNT(Fastest Lap) FROM table_1_10083598_1 WHERE Race winner = kevin curtain
Encoder-Decoder	SELECT COUNT(No) FROM table_1_10083598_1 WHERE Race winner = kevin curtain
Col Attention	SELECT Race winner FROM table_1_10083598_1 WHERE Race winner = kevin curtain
Predicted	SELECT Race winner FROM table_1_10083598_1 WHERE Race winner = kevin curtain
Error	COUNT
Question 9	What percent of the population has phone in Russia?
Table	['Country or region', 'Number of mobile phones', 'Population', '# of phones as % of population', 'Last updated date']
Ground Truth	SELECT AVG(# of phones as % of population) FROM table_2_15653040_1 WHERE Country or region = russia
LSTM	SELECT # of phones as % of population FROM table_2_15653040_1 WHERE Country or region = russia
BiLSTM	SELECT Population FROM table_2_15653040_1 WHERE Number of mobile phones = russia
Encoder-Decoder	SELECT # of phones as % of population FROM table_2_15653040_1 WHERE Country or region = russia
Col Attention	SELECT # of phones as % of population FROM table_2_15653040_1 WHERE Country or region = russia
Pointer Network	SELECT # of phones as % of population FROM table_2_15653040_1 WHERE Country or region = russia
Error	AVG
Question 10	How many silver medals when the total was 2 and less than 1 bronze?
Table	['Rank', 'Nation', 'Gold', 'Silver', 'Bronze', 'Total']
Ground Truth	SELECT SUM(Silver) FROM table_2_18661293_4 WHERE Bronze AND-
BiLSTM	Total < AND = 1 AND 2
LSTM	SELECT SUM(Silver) FROM table_2_18661293_4 WHERE Total = 2 AND Bronze < 1
BiLSTM	SELECT COUNT(Silver) FROM table_2_18661293_4 WHERE Total = 2 AND Bronze < 1

Continue on the next page

Table 3: The predicted SQL statement with its corresponding ground-truth SQL statement where no exact matching on the AGGREGATION

Type	Sentences
Encoder-Decoder	SELECT SUM(Silver) FROM table_2_18661293_4 WHERE Total = 2 AND Silver < 1
Col Attention	SELECT COUNT(Silver) FROM table_2_18661293_4 WHERE Total = 1 AND Bronze < 1
Pointer Network	SELECT COUNT(Silver) WHERE table_2_18661293_4 Total AND Total = AND < 2 AND 1
Error	COUNT
Question 11	When was the earliest with 4 seasons?
Table	[‘Season’, ‘Year(s)’, ‘Winner’, ‘Runner-Up’, ‘3rd Place’, ‘Host’]
Ground Truth	SELECT MIN(Year(s)) FROM table_2_17671150_8 WHERE Season = 4
LSTM	SELECT Year(s) FROM table_2_17671150_8 WHERE Season = 4
BiLSTM	SELECT Season FROM table_2_17671150_8 WHERE Season = 4
Encoder-Decoder	SELECT Host FROM table_2_17671150_8 WHERE Season = 4
Col Attention	SELECT 3rd Place FROM table_2_17671150_8 WHERE Season = 4
Pointer Network	SELECT 3rd Place FROM table_2_17671150_8 WHERE Season = 4
Error	MIN
Question 12	What is the Frequency at the Market/Rank of Burlington - Plattsburgh , Vermont - New York /143?
Table	[‘Calls’, ‘Frequency’, ‘Branding’, ‘Format’, ‘Market/Rank’, ‘Timeslot’, ‘Group owner’]
Ground Truth	SELECT COUNT(Frequency) FROM table_1_10333757_1 WHERE Market/Rank = Burlington - Plattsburgh , Vermont - New York /143
LSTM	SELECT Frequency FROM table_1_10333757_1 WHERE Branding = burlington - plattsburgh- vermont- new york/143, vermont
BiLSTM	SELECT Frequency FROM table_1_10333757_1 WHERE Timeslot = burlington - plattsburgh , vermont - new york /143
Encoder-Decoder	SELECT Frequency FROM table_1_10333757_1 WHERE Frequency = burlington - plattsburgh , vermont - new york /143
Col Attention	SELECT Frequency FROM table_1_10333757_1 WHERE Group owner = burlington - plattsburgh , vermont - new york /143
Pointer Network	SELECT Frequency FROM table_1_10333757_1 WHERE Group owner = burlington - plattsburgh , vermont - new york /143

Continue on the next page

Table 3: The predicted SQL statement with its corresponding ground-truth SQL statement where no exact matching on the AGGREGATION

Type	Sentences
Error	COUNT
Question 13	In what group stage were 177.2 points awarded?
Table	[‘Rank’, ‘Member Association’, ‘Points’, ‘Group stage’, ‘Play-off’, ‘AFC Cup’]
Ground Truth	SELECT COUNT(Group stage) FROM table_1_1059743_2 WHERE Points = 177.2
LSTM	SELECT Group stage FROM table_1_1059743_2 WHERE Points = 177.2
BiLSTM	SELECT COUNT(Member Association) FROM table_1_1059743_2 WHERE Points = 177.2
Encoder-Decoder	SELECT Group stage FROM table_1_1059743_2 WHERE Points = 177.2
Col Attention	SELECT Group stage FROM table_1_1059743_2 WHERE Points = 177.2
Pointer Network	SELECT Group stage FROM table_1_1059743_2 WHERE Points = 177.2
Error	COUNT
Question 14	What nation had 135 Bronze medals?
Table	[‘Sport’, ‘Gold’, ‘Silver’,‘Bronze’, ‘Total’]
Ground Truth	SELECT COUNT(Bronze) FROM table_2_12253523_2 WHERE Total = 135
LSTM	SELECT Nation FROM table_2_12253523_2 WHERE Bronze = 135
BiLSTM	SELECT Nation FROM table_2_12253523_2 WHERE Bronze = 135 AND Nation = 135
Encoder-Decoder	SELECT Nation FROM table_2_12253523_2 WHERE Bronze = 135
Col Attention	SELECT Nation FROM table_2_12253523_2 WHERE Bronze = 135
Pointer Network	SELECT Nation FROM table_2_12253523_2 WHERE Bronze = 135
Error	COUNT
Question 15	How many games were lost when the goals scored was 37?
Table	[‘Position’, ‘Club’, ‘Games played’, ‘Wins’, ‘Draws’, ‘Loses’, ‘Goals scored’, ‘Goals conceded’, ‘Points’]
Ground Truth	SELECT COUNT(Loses) FROM table_1_16034882_2 WHERE Goals scored = 37
LSTM	SELECT Loses FROM table_1_16034882_2 WHERE Goals scored = 37
BiLSTM	SELECT COUNT(Loses) FROM table_1_16034882_2 WHERE Goals scored = 37
Encoder-Decoder	SELECT COUNT(Loses) FROM table_1_16034882_2 WHERE Goals scored = 37

Continue on the next page

Table 3: The predicted SQL statement with its corresponding ground-truth SQL statement where no exact matching on the AGGREGATION

Type	Sentences
Col Attention	SELECT Loses FROM table_1_16034882_2 WHERE Goals scored = 37
Pointer Network	SELECT Loses FROM table_1_16034882_2 WHERE Goals scored = 37
Error	COUNT
Question 16	how many kana spelling when english is mount fuji
Table	['English', 'Japanese', 'Kana spelling', 'Revised Hepburn', 'Kunrei-shiki', 'Nihon-shiki']
Ground Truth	SELECT Kana spelling FROM table_1_26263954_1 WHERE English = Mount Fuji
LSTM	SELECT COUNT(Kana spelling) FROM table_1_26263954_1 WHERE English = mount fuji
BiLSTM	SELECT COUNT(Kana spelling) FROM table_1_26263954_1 WHERE English = mount fuji
Encoder-Decoder	SELECT COUNT(Revised Hepburn) FROM table_1_26263954_1 WHERE English = mount fuji
Col Attention	SELECT COUNT(Kana spelling) FROM table_1_26263954_1 WHERE English = mount fuji
Pointer Network	SELECT COUNT(Kana spelling) FROM table_1_26263954_1 WHERE English = mount fuji
Error	AVG
Question 17	How many people went to the game that had 17-13?
Table	['Week', 'Date', 'Kickoff', 'Opponent', 'Final score', 'Team record', 'Game site', 'Attendance']
Ground Truth	SELECT SUM(Attendance) FROM table_2_15117450_2 WHERE Score = 17-13
LSTM	SELECT Attendance FROM table_2_15117450_2 WHERE Record = 17-13
BiLSTM	SELECT Attendance FROM table_2_15117450_2 WHERE Record = 17-13
Encoder-Decoder	SELECT Attendance FROM table_2_15117450_2 WHERE Record = 17-13
Col Attention	SELECT Attendance FROM table_2_15117450_2 WHERE Record = 17-13
Pointer Network	SELECT Attendance FROM table_2_15117450_2 WHERE Record = 17-13
Error	SUM
Question 18	when english is tea how many nihon-shiki?
Table	['English', 'Japanese', 'Kana spelling', 'Revised Hepburn', 'Kunrei-shiki', 'Nihon-shiki']

Continue on the next page

Table 3: The predicted SQL statement with its corresponding ground-truth SQL statement where no exact matching on the AGGREGATION

Type	Sentences
Ground Truth	SELECT Nihon-shiki FROM table_1_26263954_1 WHERE English = tea
LSTM	SELECT COUNT(Revised Hepburn) FROM table_1_26263954_1 WHERE English = roman
BiLSTM	SELECT COUNT(Revised Hepburn) FROM table_1_26263954_1 WHERE English = english
Encoder-Decoder	SELECT COUNT(Japanese) FROM table_1_26263954_1 WHERE English = tea
Col Attention	SELECT COUNT(Japanese) FROM table_1_26263954_1 WHERE English = english is
Pointer Network	SELECT COUNT(Kana spelling) FROM table_1_26263954_1 WHERE English = english how many
Error	COUNT
Question 19	How many silver when the team is northwest territories and gold is less than 34?
Table	[‘Team’, ‘Gold’, ‘Silver’, ‘Bronze’, ‘Overall’]
Ground Truth	SELECT COUNT(Silver) FROM table_2_16360419_1 WHERE Team AND Gold = AND < northwest territories AND 34
LSTM	
BiLSTM	SELECT SUM(Silver) FROM table_2_16360419_1 WHERE Team = northwest territories AND Gold < 34
Encoder-Decoder	SELECT SUM(Silver) FROM table_2_16360419_1 WHERE Team = northwest territories AND Silver <34
Col Attention	SELECT SUM(Silver) FROM table_2_16360419_1 WHERE Team = northwest territories AND Gold < 34
Pointer Network	SELECT SUM(Silver) FROM table_2_16360419_1 WHERE Team AND Overall = AND < northwest territories AND 34
Error	SUM
Question 20	In what Round was the draft pick from Florida with an Overall less than 166?
Table	[‘Round’, ‘Pick’, ‘Overall’, ‘Name’, ‘Position’, ‘College’]
Ground Truth	SELECT AVG(Round) FROM table_2_15198842_33 WHERE College AND Overall = AND < florida AND 166

Continue on the next page

Table 3: The predicted SQL statement with its corresponding ground-truth SQL statement where no exact matching on the AGGREGATION

Type	Sentences
LSTM	SELECT SUM(Round) FROM table_2_15198842_33 WHERE Round < 166 AND Overall = florida
BiLSTM	SELECT MAX(Round) FROM table_2_15198842_33 WHERE College = florida AND Overall < 166
Encoder-Decoder	SELECT SUM(Round) FROM table_2_15198842_33 WHERE Position = florida AND Overall < 166
Col Attention	SELECT Round FROM table_2_15198842_33 WHERE Overall < 166 AND College = 166
Pointer Network	SELECT SUM(Round) FROM table_2_15198842_33 WHERE College AND Overall = AND < florida AND 166
Error	SUM
Question 21	What is the Pick # of the player with an Overall less than 203 from West Virginia?
Table	['Round', 'Pick #', 'Overall', 'Name', 'Position', 'College']
Ground Truth	SELECT COUNT(Pick #) FROM table_2_15198842_33 WHERE College AND Overall = AND < west virginia AND 203
LSTM	SELECT SUM(Pick #) FROM table_2_15198842_33 WHERE Overall < 203 AND Position = west virginia
BiLSTM	SELECT MAX(Pick #) FROM table_2_15198842_33 WHERE Overall < 203 AND College = west virginia
Encoder-Decoder	SELECT Pick # FROM table_2_15198842_33 WHERE Overall < 203 AND Round = AND Round = virginia
Col Attention	SELECT SUM(Pick #) FROM table_2_15198842_33 WHERE Overall < 203 AND College = 203
Pointer Network	SELECT SUM(Pick #) FROM table_2_15198842_33 WHERE Overall AND College < AND = 203 AND west virginia
Question 22	Which Silver has a Total smaller than 2 and a Rank larger than 8
Table	['Rank', 'Nation', 'Gold', 'Silver', 'Bronze', 'Total']
Ground Truth	SELECT MIN(Silver) FROM table_2_156359_1 WHERE Total AND Rank < AND > 2 AND 8
LSTM	SELECT AVG(Silver) FROM table_2_156359_1 WHERE Total < 2 AND Rank > 8

Continue on the next page

Table 3: The predicted SQL statement with its corresponding ground-truth SQL statement where no exact matching on the AGGREGATION

Type	Sentences
BiLSTM	SELECT AVG(Silver) FROM table_2_156359_1 WHERE Total < 2 AND Rank > 8
Encoder-Decoder	SELECT AVG(Silver) FROM table_2_156359_1 WHERE Total < 2 AND Rank > 8
Col Attention	SELECT AVG(Silver) FROM table_2_156359_1 WHERE Rank > 8 AND Total < 8
Pointer Network	SELECT AVG(Silver) FROM table_2_156359_1 WHERE Total AND Rank < AND > 2 AND 8
Error	AVG
Question 23	What was the Pick Number when the position was wide receiver, the college was Southern Miss with an overall less than 186?
Table	['Round', 'Pick #', 'Overall', 'Name', 'Position', 'College']
Ground Truth	SELECT AVG(Pick #) FROM table_2_15198842_39 WHERE Position AND College AND Overall = AND = AND < wide receiver AND southern miss AND 186
LSTM	
BiLSTM	SELECT SUM(Pick #) FROM table_2_15198842_39 WHERE Position = wide receiver AND Position = southern miss AND Overall < 186
Encoder-Decoder	SELECT Pick # FROM table_2_15198842_39 WHERE Position = wide receiver AND Round = southern miss AND Overall < 186
Col Attention	SELECT AVG(Pick #) FROM table_2_15198842_39 WHERE Position = 186 AND Overall < 186 AND College = 186
Pointer Network	SELECT COUNT(Pick #)FROM table_2_15198842_39 WHERE Position AND College AND Overall = AND = AND < wide receiver AND southern miss AND 186
Error	AVG
Question 24	How many picks did Mike Zaher have?
Table	['Round', 'Pick', 'Player', 'Position', 'School/Club Team']
Ground Truth	SELECT SUM(Pick #) FROM table_2_15214004_3 WHERE Player = mike zaher
LSTM	SELECT COUNT(Pick #) FROM table_2_15214004_3 WHERE Player = mike zaher

Continue on the next page

Table 3: The predicted SQL statement with its corresponding ground-truth SQL statement where no exact matching on the AGGREGATION

Type	Sentences
BiLSTM	SELECT COUNT(Pick #) FROM table_2_15214004_3 WHERE Player = mike zaher
Encoder-Decoder	SELECT COUNT(Pick #) FROM table_2_15214004_3 WHERE Player = mike zaher
Col Attention	SELECT COUNT(Pick #) FROM table_2_15214004_3 WHERE Player = mike zaher
Pointer Network	SELECT COUNT(Pick #)FROM table_2_15214004_3 WHERE Player = mike zaher
Error	SUM
Question 25	how many pick# does the chivas usa mls team have
Table	[‘Pick #’, ‘MLS team’, ‘Player’, ‘Position’, ‘Affiliation’]
Ground Truth	SELECT MAX(Pick #) FROM table_1_29626583_1 WHERE MLS team = Chivas USA
LSTM	SELECT COUNT(Pick #) FROM table_1_29626583_1 WHERE MLS team = chivas usa mls
BiLSTM	SELECT COUNT(Pick #) FROM table_1_29626583_1 WHERE MLS team = chivas usa
Encoder-Decoder	SELECT COUNT(Pick #) FROM table_1_29626583_1 WHERE MLS team = chivas usa mls
Col Attention	SELECT COUNT(Pick #) FROM table_1_29626583_1 WHERE MLS team = chivas usa
Pointer Network	SELECT COUNT(Pick #) FROM table_1_29626583_1 WHERE MLS team = chivas usa
Error	Max
Question 26	Which Yards has a Long smaller than 3?
Table	[‘Player’, ‘Games Played’, ‘Attempts’, ‘Yards’,‘Touchdowns’,‘Longest run’]
Ground Truth	SELECT AVG(Yards) FROM table_2_15598908_5 WHERE Long < 3
LSTM	SELECT MAX(Yards) FROM table_2_15598908_5 WHERE Long < 3
BiLSTM	SELECT MAX(Yards) FROM table_2_15598908_5 WHERE Long < 3
Encoder-Decoder	SELECT Yards FROM table_2_15598908_5 WHERE Long <3
Col Attention	SELECT Yards FROM table_2_15598908_5 WHERE Long < 3
Pointer Network	SELECT MAX(Yards) FROM table_2_15598908_5 WHERE Long < 3

Continue on the next page

Table 3: The predicted SQL statement with its corresponding ground-truth SQL statement where no exact matching on the AGGREGATION

Type	Sentences
Error	Both Ground Truth and Predicted
Question 27	What is the lowest Extras for Chamara Kapugedera, with an E.R. less than 4.59?
Table	['Name', 'Overs Bowled', 'Maidens', 'Runs Conceded', 'Wickets', 'Extras', 'E.R.']
Ground Truth	SELECT Extras FROM table_2_15700367_6 WHERE Name AND E.R. = AND < chamara kapugedera AND 4.59
LSTM	SELECT MIN(Overs Bowled) FROM table_2_15700367_6 WHERE Name = chamara kapugedera AND E.R. < 4.59
BiLSTM	SELECT MIN(Maidens) FROM table_2_15700367_6 WHERE Name = chamara kapugedera AND Runs Conceded <4.59
Encoder-Decoder	SELECT MIN(Maidens) FROM table_2_15700367_6 WHERE Name = chamara kapugedera AND Runs Conceded <4.59
Col Attention	SELECT MIN(Overs Bowled) FROM table_2_15700367_6 WHERE Name = chamara kapugedera AND E.R. < 4.59
Pointer Network	SELECT MIN(Overs Bowled) FROM table_2_15700367_6 WHERE Name AND Name = AND < chamara kapug WHERE -edera AND 4.59
Error	Ground Truth
Question 28	What year was the No. 7 charts?
Table	['Year', 'Álbum', 'Charts', 'Sales', 'Certification']
Ground Truth	SELECT AVG(Year) FROM table_2_12937449_4 WHERE Charts = no. 7
LSTM	SELECT Year FROM table_2_12937449_4 WHERE Charts = no
BiLSTM	SELECT Year FROM table_2_12937449_4 WHERE Charts = 7
Encoder-Decoder	SELECT Year FROM table_2_12937449_4 WHERE Year = 7
Col Attention	SELECT Year FROM table_2_12937449_4 WHERE Álbum = 7
Pointer Network	SELECT Year FROM table_2_12937449_4 WHERE Sales = 7
Error	Ground Truth
Question 29	What is the chart number of the Requebra album with sales smaller than 32,000?
Table	['Year', 'Álbum', 'Charts', 'Sales', 'Certification']
Ground Truth	SELECT Charts FROM table_2_12937449_4 WHERE Sales AND Álbum < AND = 32,000 AND requebra

Continue on the next page

Table 3: The predicted SQL statement with its corresponding ground-truth SQL statement where no exact matching on the AGGREGATION

Type	Sentences
LSTM	SELECT COUNT(Álbum) FROM table_2_12937449_4 WHERE Sales = requebra AND Charts < 32,000
BiLSTM	SELECT COUNT(Charts) FROM table_2_12937449_4 WHERE Certification = requebra AND Sales < 32,000
Encoder-Decoder	SELECT COUNT(Charts) FROM table_2_12937449_4 WHERE Certification < requebra AND Charts = requebra
Col Attention	SELECT COUNT(Charts) FROM table_2_12937449_4 WHERE Sales < 32,000 AND Álbum = 32,000
Pointer Network	SELECT COUNT(Charts) FROM table_2_12937449_4 WHERE Sales AND Charts = AND < requebra AND 32,000
Error	Predicted
Question 30	What was the earliest release-year of the first charted record of Abba?
Table	['Artist', 'Country of origin', 'Period active', 'Release-year of first charted record', 'Genre', 'Claimed sales']
Ground Truth	SELECT MIN(Release-year of first charted record) FROM table_2_1291598_2 WHERE Artist = abba
LSTM	SELECT Period active FROM table_2_1291598_2 WHERE Artist = abba
BiLSTM	SELECT Claimed sales FROM table_2_1291598_2 WHERE Artist = abba
Encoder-Decoder	SELECT Genre FROM table_2_1291598_2 WHERE Artist = abba
Col Attention	SELECT Genre FROM table_2_1291598_2 WHERE Release-year of first charted record = abba
Pointer Network	SELECT Period active FROM table_2_1291598_2 WHERE Claimed sales = abba

Table 4: Incorrect matching between the Column names of both the SELECT and the WHERE-Clause

Type	Sentences
Question 1	how many schools or teams had jalen rose
Table	['Player', 'No.', 'Nationality', 'Position', 'Years in Toronto', 'School/Club Team']
Ground Truth	SELECT COUNT(School/Club Team) FROM table_1_10015132_16 WHERE Player = Jalen Rose

Continue on the next page

Table 4: Incorrect matching between the Column names of both the SELECT and the WHERE-Clause

Type	Sentences
LSTM	SELECT COUNT(School/Club Team) FROM table_1_10015132_16 WHERE Player = jalen rose
BiLSTM	SELECT COUNT(School/Club Team) FROM table_1_10015132_16 WHERE Player = jalen rose
Encoder-Decoder	SELECT COUNT(No.) FROM table_1_10015132_16 WHERE Player = EQL rose
Col Attention	SELECT COUNT(School/Club Team) FROM table_1_10015132_16 WHERE School/Club Team = jalen rose
Pointer Network	SELECT COUNT(School/Club Team) FROM table_1_10015132_16 WHERE School/Club Team = jalen rose
Question 2	What could a spanish coronel be addressed as in the commonwealth military?
Table	['Equivalent NATO Rank Code', 'Rank in Spanish', 'Rank in English', 'Commonwealth equivalent', 'US Air Force equivalent']
Ground Truth	SELECT Commonwealth equivalent FROM table_1_1015521_2 WHERE Rank in Spanish = Coronel
LSTM	SELECT Rank in Spanish FROM table_1_1015521_2 WHERE Equivalent NATO Rank Code = commonwealth
BiLSTM	SELECT Rank in Spanish FROM table_1_1015521_2 WHERE Equivalent NATO Rank Code = commonwealth
Encoder-Decoder	SELECT US Air Force equivalent FROM table_1_1015521_2 WHERE Rank in English = commonwealth
Col Attention	SELECT Commonwealth equivalent FROM table_1_1015521_2 WHERE Rank in English = commonwealth military
Pointer Network	SELECT Commonwealth equivalent FROM table_1_1015521_2 WHERE Rank in English = commonwealth military
Question 3	Give me a list of all spanish officer titles that could receive recognition as group captain in english
Table	['Equivalent NATO Rank Code', 'Rank in Spanish', 'Rank in English', 'Commonwealth equivalent', 'US Air Force equivalent']
Ground Truth	SELECT Rank in English FROM table_1_1015521_2 WHERE Commonwealth equivalent = Group Captain

Continue on the next page

Table 4: Incorrect matching between the Column names of both the SELECT and the WHERE-Clause

Type	Sentences
LSTM	SELECT Rank in English FROM table_1_1015521_2 WHERE Equivalent NATO Rank Code = english
BiLSTM	SELECT Commonwealth equivalent FROM table_1_1015521_2 WHERE Equivalent NATO Rank Code = english
Encoder-Decoder	SELECT Rank in Spanish FROM table_1_1015521_2 WHERE Equivalent NATO Rank Code = in english group captain
Col Attention	SELECT Equivalent NATO Rank Code FROM table_1_1015521_2 WHERE Rank in English = english
Pointer Network	SELECT Equivalent NATO Rank Code FROM table_1_1015521_2 WHERE Rank in English = english
Question 4	What is the total number of churches named Høyanger Kyrkje?
Table	['Parish (Prestegjeld)', 'Sub-Parish (Sogn)', 'Church Name', 'Year Built', 'Location of the Church']
Ground Truth	SELECT COUNT(Sub-Parish (Sogn)) FROM table_1_178389_1 WHERE Church Name = Høyanger kyrkje
LSTM	SELECT Parish (Prestegjeld) FROM table_1_178389_1 WHERE Church Name = høyanger kyrkje
BiLSTM	SELECT COUNT(Parish (Prestegjeld)) FROM table_1_178389_1 WHERE Church Name = høyanger kyrkje
Encoder-Decoder	SELECT Parish (Prestegjeld) FROM table_1_178389_1 WHERE Church Name = høyanger kyrkje
Col Attention	SELECT COUNT(Parish (Prestegjeld)) FROM table_1_178389_1 WHERE Church Name = høyanger kyrkje
Pointer Network	SELECT COUNT(Parish (Prestegjeld)) FROM table_1_178389_1 WHERE Church Name = høyanger kyrkje
Question 5	How many episodes did Cliff Bole directed in season 3?
Table	['No. in series', 'No. in season', 'Title', 'Directed by', 'Written by', 'Original air date', 'U.S. viewers (millions)']
Ground Truth	SELECT COUNT(Title) FROM table_1_17861265_1 WHERE Directed by = Cliff Bole
LSTM	SELECT COUNT(No. in season) FROM table_1_17861265_1 WHERE No. in series = cliff bole

Continue on the next page

Table 4: Incorrect matching between the Column names of both the SELECT and the WHERE-Clause

Type	Sentences
BiLSTM	SELECT COUNT(Written by) FROM table_1_17861265_1 WHERE Title = cliff bole
Encoder-Decoder	SELECT COUNT(No. in series) FROM table_1_17861265_1 WHERE No. in series = cliff bole AND No. in series = 3
Col Attention	SELECT COUNT(No. in series) FROM table_1_17861265_1 WHERE Directed by = cliff bole AND No. in season = cliff bole directed
Pointer Network	SELECT COUNT(No. in series) FROM table_1_17861265_1 WHERE Directed by = cliff bole AND No. in season = cliff bole directed
Question 6	Where was Assen held?
Table	['No', 'Date', 'Round', 'Circuit', 'Pole Position', 'Fastest Lap', 'Race winner', 'Report']
Ground Truth	SELECT Round FROM table_1_10083598_1 WHERE Circuit = Assen
BiLSTM	SELECT Pole Position FROM table_1_10083598_1 WHERE Circuit = assen
Encoder-Decoder	SELECT Circuit FROM table_1_10083598_1 WHERE Circuit = assen
Col Attention	SELECT Circuit FROM table_1_10083598_1 WHERE Circuit = assen
Pointer Network	SELECT Circuit FROM table_1_10083598_1 WHERE Circuit = assen
Question 7	What was the number of race that Kevin Curtain won?
Table	['No', 'Date', 'Round', 'Circuit', 'Pole Position', 'Fastest Lap', 'Race winner', 'Report']
Ground Truth	SELECT COUNT(No) FROM table_1_10083598_1 WHERE Pole Position = Kevin Curtain
LSTM	SELECT COUNT(Round) FROM table_1_10083598_1 WHERE Circuit = kevin curtain
BiLSTM	SELECT COUNT(Fastest Lap) FROM table_1_10083598_1 WHERE Race winner = kevin curtain
Encoder-Decoder	SELECT COUNT(No) FROM table_1_10083598_1 WHERE Race winner = kevin curtain
Attention	SELECT Race winner FROM table_1_10083598_1 WHERE Race winner = kevin curtain
Pointer Network	SELECT Race winner FROM table_1_10083598_1 WHERE Race winner = kevin curtain
Question 8	How many different college/junior/club teams provided a player to the Washington Capitals NHL Team?

Continue on the next page

Table 4: Incorrect matching between the Column names of both the SELECT and the WHERE-Clause

Type	Sentences
Table	['Pick', 'Player', 'Position', 'Nationality', 'NHL team', 'College/junior/club team']
Ground Truth	SELECT COUNT(College/junior/club team) FROM table_1_1013129_2 WHERE NHL team = Washington Capitals
LSTM	SELECT COUNT(NHL team) FROM table_1_1013129_2 WHERE NHL team = washington capitals nhl
BiLSTM	SELECT Nationality FROM table_1_1013129_2 WHERE College/junior/-club team = thunder bay flyers (ushl)
Encoder-Decoder	SELECT Nationality FROM table_1_1013129_2 WHERE College/junior/-club team = thunder bay flyers (ushl)
Col Attention	SELECT COUNT(NHL team) FROM table_1_1013129_2 WHERE NHL team = washington capitals nhl
Pointer Network	SELECT COUNT(NHL team) FROM table_1_1013129_2 WHERE NHL team = washington capitals nhl
Question 9	What are the nationalities of the player picked from Thunder Bay Flyers (ushl)
Table	['Pick', 'Player', 'Position', 'Nationality', 'NHL team', 'College/junior/club team']
Ground Truth	SELECT Nationality FROM table_1_1013129_2 WHERE College/junior/-club team = Thunder Bay Flyers (USHL)
LSTM	SELECT Nationality FROM table_1_1013129_2 WHERE College/junior/-club team = thunder bay flyers (ushl)
BiLSTM	SELECT Nationality FROM table_1_1013129_2 WHERE College/junior/-club team = thunder bay flyers (ushl)
Encoder-Decoder	SELECT Nationality FROM table_1_1013129_2 WHERE College/junior/-club team = thunder bay flyers (ushl)
Col Attention	SELECT Nationality FROM table_1_1013129_2 WHERE NHL team = thunder bay flyers (ushl)
Pointer Network	SELECT Nationality FROM table_1_1013129_2 WHERE NHL team = thunder bay flyers (ushl)
Question 10	When were ships laid down that were commissioned on october 29, 1965?
Table	['#', 'Shipyard', 'Laid down', 'Launched', 'Commissioned', 'Fleet', 'Status']

Continue on the next page

Table 4: Incorrect matching between the Column names of both the SELECT and the WHERE-Clause

Type	Sentences
Ground Truth	SELECT Laid down FROM table_1_1014206_2 WHERE Commissioned = October 29, 1965
LSTM	SELECT Laid down FROM table_1_1014206_2 WHERE # = october 29, 1965 AND # > commissioned
BiLSTM	SELECT Shipyard FROM table_1_1014206_2 WHERE Commissioned = october 29, 1965
Encoder-Decoder	SELECT Laid down FROM table_1_1014206_2 WHERE Status = october 29, 1965
Col Attention	SELECT Laid down FROM table_1_1014206_2 WHERE Laid down = october 29, 1965
Pointer Network	SELECT Laid down FROM table_1_1014206_2 WHERE Laid down = october 29, 1965
Question 11	Which show runs on Friday at 05:00 AM?
Table	[‘Time’, ‘00:00 AM’, ‘01:00 AM’, ‘02:00 AM’, ‘05:00 AM’]
Ground Truth	SELECT 05:00 AM FROM table_2_15535243_1 WHERE Time = friday
LSTM	SELECT 00:00 AM FROM table_2_15535243_1 WHERE 01:00 AM = friday am AND Time = friday
BiLSTM	SELECT Time FROM table_2_15535243_1 WHERE Time = 05:00 am
Encoder-Decoder	SELECT Time FROM table_2_15535243_1 WHERE Time = 05:00 am
Col Attention	SELECT Time FROM table_2_15535243_1 WHERE Time = at 05:00 am
Pointer Network	SELECT Time FROM table_2_15535243_1 00:00 WHERE AM = 05:00 am
Question 12	What is the away team of the UEFA champions league?
Table	[‘Season’, ‘Competition’, ‘Round’, ‘Opponent’, ‘Home’, ‘Away’]
Ground Truth	SELECT Away FROM table_2_1607312_1 WHERE Competition = uefa champions league
LSTM	SELECT Away FROM table_2_1607312_1 WHERE Competition = uefa champions
BiLSTM	SELECT Opponent FROM table_2_1607312_1 WHERE Competition = uefa champions
Encoder-Decoder	SELECT Away FROM table_2_1607312_1 WHERE Competition = uefa champions
Col Attention	SELECT Away FROM table_2_1607312_1 WHERE Competition = uefa champions

Continue on the next page

Table 4: Incorrect matching between the Column names of both the SELECT and the WHERE-Clause

Type	Sentences
Pointer Network	SELECT Opponent FROM table_2_1607312_1 WHERE Opponent = uefa champions
Question 13	Which claimant's rank is 200?
Table	['Rank', 'Margin', 'Opponent', 'Venue', 'Season']
Ground Truth	SELECT Claimant FROM table_2_1594415_2 WHERE Rank = 200
LSTM	SELECT Rank FROM table_2_1594415_2 WHERE Rank = 200
BiLSTM	SELECT Claimant FROM table_2_1594415_2 WHERE Rank = 200
Encoder-Decoder	SELECT Rank FROM table_2_1594415_2 WHERE Rank = 200
Col Attention	SELECT Claimant FROM table_2_1594415_2 WHERE Rank = 200
Pointer Network	SELECT Rank FROM table_2_1594415_2 WHERE Rank = 200
Question 14	What is the most points for South Ossetia?
Table	['Rank', 'Country', 'Claimant', 'Highest point', 'Height']
Ground Truth	SELECT Highest point FROM table_2_1594415_2 WHERE Country = south ossetia
LSTM	SELECT MAX(Highest point) FROM table_2_1594415_2 WHERE Country = south ossetia
BiLSTM	SELECT Highest point FROM table_2_1594415_2 WHERE Country = south ossetia
Encoder-Decoder	SELECT Highest point FROM table_2_1594415_2 WHERE Country = south ossetia
Col Attention	SELECT Highest point FROM table_2_1594415_2 WHERE Country = south ossetia
Pointer Network	SELECT Height FROM table_2_1594415_2 WHERE Country = south ossetia
Question 15	What is the highest amount of points for rank 95?
Table	['Rank', 'Country', 'Claimant', 'Highest point', 'Height']
Ground Truth	SELECT Highest point FROM table_2_1594415_2 WHERE Rank = 95
LSTM	SELECT MAX(Highest point) FROM table_2_1594415_2 WHERE Rank = 95
BiLSTM	SELECT Highest point FROM table_2_1594415_2 WHERE Rank = 95
Encoder-Decoder	SELECT MAX(Highest point) FROM table_2_1594415_2 WHERE Rank = 95

Continue on the next page

Table 4: Incorrect matching between the Column names of both the SELECT and the WHERE-Clause

Type	Sentences
Col Attention	SELECT MAX(Highest point) FROM table_2_1594415_2 WHERE Rank = 95
Pointer Network	SELECT Height FROM table_2_1594415_2 WHERE Rank = 95
Question 16	Robert S. Flanzer was the Free Libertarian ticket with who listed as the Liberal Ticket?
Table	['Office', 'Democratic ticket', 'Republican ticket', 'Conservative ticket', 'Liberal ticket', 'Free Libertarian ticket', 'Socialist Labor ticket']
Ground Truth	SELECT Liberal ticket FROM table_2_15558943_2 WHERE Free Libertarian ticket = robert s. flanzer
LSTM	SELECT Free Libertarian ticket FROM table_2_15558943_2 WHERE Liberal ticket = robert s. flanzer
BiLSTM	SELECT Liberal ticket FROM table_2_15558943_2 WHERE Office = robert s. flanzer
Encoder-Decoder	SELECT Socialist Labor ticket FROM table_2_15558943_2 WHERE Office = robert s. flanzer
Col Attention	SELECT Socialist Labor ticket FROM table_2_15558943_2 WHERE Conservative ticket = s. flanzer was the free????
Pointer Network	SELECT Democratic ticket FROM table_2_15558943_2 WHERE Office = robert s.
Question 17	What is the lowest Extras for Chamara Kapugedera, with an E.R. less than 4.59?
Table	['Name', 'Overs Bowled', 'Maidens', 'Runs Conceded', 'Wickets', 'Extras', 'E.R.']
Ground Truth	SELECT Extras FROM table_2_15700367_6 WHERE Name AND E.R. = AND < chamara kapugedera AND 4.59
LSTM	SELECT MIN(Overs Bowled) FROM table_2_15700367_6 WHERE Name = chamara kapugedera AND E.R. < 4.59
BiLSTM	SELECT MIN(Maidens) FROM table_2_15700367_6 WHERE Name = chamara kapugedera AND Runs Conceded < 4.59
Encoder-Decoder	
Col Attention	SELECT MIN(Overs Bowled) FROM table_2_15700367_6 WHERE Name = chamara kapugedera AND E.R. < 4.59

Continue on the next page

Table 4: Incorrect matching between the Column names of both the SELECT and the WHERE-Clause

Type	Sentences
Pointer Network	SELECT MIN(Overs Bowled) FROM table_2_15700367_6 WHERE Name AND Name = AND < chamara kapugedera AND 4.59
Question 18	What is the total number of Maidens when E.R. is less than 9.5, and a Overs Bowled larger than 57, and a Wickets of 9?
Table	['Name', 'Overs Bowled', 'Maidens', 'Runs Conceded', 'Wickets', 'Extras', 'E.R.']
Ground Truth	SELECT COUNT(Maidens) FROM table_2_15700367_6 WHERE E.R. AND Overs Bowled AND Wickets < AND > AND = 9.5 AND 57 AND 9
LSTM	SELECT COUNT(Maidens) FROM table_2_15700367_6 WHERE Name < 9.5 AND Overs Bowled > 57 AND Overs Bowled = 9
BiLSTM	SELECT COUNT(Maidens) FROM table_2_15700367_6 WHERE E.R. < 9.5 AND Runs Conceded > 57 AND Wickets = 9
Encoder-Decoder	SELECT MIN(Maidens) FROM table_2_15700367_6 WHERE Name = chamara kapugedera AND Runs Conceded < 4.59
Col Attention	SELECT COUNT(Extras) FROM table_2_15700367_6 WHERE E.R. > , AND Overs Bowled > , AND Maidens > 9
Pointer Network	SELECT COUNT(Overs Bowled) FROM table_2_15700367_6 WHERE Name AND Name AND Name < AND > AND = 9.5 AND 57 AND 9
Question 19	What is the second team that has first team of Africa sports?
Table	['Tie no', 'Team 1', 'Agg.', 'Team 2', '1st leg', '2nd leg']
Ground Truth	SELECT Team 2 FROM table_2_12330040_2 WHERE Team 1 = africa sports
LSTM	SELECT Team 2 FROM table_2_12330040_2 WHERE Team 1 = africa sports
BiLSTM	SELECT Team 2 FROM table_2_12330040_2 WHERE Team 1 = africa sports
Encoder-Decoder	SELECT Team 2 FROM table_2_12330040_2 WHERE Team 1 = africa sports
Col Attention	SELECT Team 1 FROM table_2_12330040_2 WHERE Team 1 = africa sports
Pointer Network	SELECT Team 1 FROM table_2_12330040_2 WHERE Team 1 = africa sports

Continue on the next page

Table 4: Incorrect matching between the Column names of both the SELECT and the WHERE-Clause

Type	Sentences
Question 20	Which margin of defeats had points of 30?
Table	['Season', 'Driver', 'Team', 'Engine', 'Poles', 'Wins', 'Podiums', 'Points', 'Margin of defeat']
Ground Truth	SELECT Margin of defeat FROM table_1_10753917_1 WHERE Points = 30
LSTM	SELECT COUNT(Driver) FROM table_1_10753917_1 WHERE Season = 2
BiLSTM	SELECT Margin of defeat FROM table_1_10753917_1 WHERE Points = 30
Encoder-Decoder	SELECT Margin of defeat FROM table_1_10753917_1 WHERE Points = 30
Col Attention	SELECT Margin of defeat FROM table_1_10753917_1 WHERE Points = 30 AND Driver = 30
Pointer Network	SELECT Poles FROM table_1_10753917_1 WHERE Margin of defeat = 30
Question 21	For which singers was Alangudi Somu the lyricist?
Table	['Track #', 'Song', 'Singers', 'Picturization', 'Length', 'Lyricist']
Ground Truth	SELECT Singers FROM table_1_10848177_1 WHERE Lyricist = Alangudi Somu
LSTM	SELECT Singers FROM table_1_10848177_1 WHERE Picturization = alangudi somu
BiLSTM	SELECT Singers FROM table_1_10848177_1 WHERE Singers = alangudi somu
Encoder-Decoder	SELECT Singers FROM table_1_10848177_1 WHERE Picturization = alangudi somu
Col Attention	SELECT Singers FROM table_1_10848177_1 WHERE Singers = alangudi somu
Pointer Network	SELECT Singers FROM table_1_10848177_1 WHERE Picturization = alangudi somu
Question 22	What publishers were involved with product number SCUS-97265?
Table	['Title', 'System', 'Publisher', 'Product No.', 'Compatible with 20/60GB NTSC PS3 (CECHA/CECHB)', 'Compatible with 60GB PAL/80GB NTSC PS3 (CECHC/CECHE)']
Ground Truth	SELECT Publisher FROM table_1_10875694_11 WHERE Product No. = SCUS-97265
LSTM	SELECT Product No. FROM table_1_10875694_11 WHERE Title = scus-97265

Continue on the next page

Table 4: Incorrect matching between the Column names of both the SELECT and the WHERE-Clause

Type	Sentences
BiLSTM	SELECT Publisher FROM table_1_10875694_11 WHERE Product No. = scus-97265 system AND System = scus-97265
Encoder-Decoder	SELECT Compatible with 60GB PAL/80GB NTSC PS3 (CECHC/CECHE) FROM table_1_10875694_11 WHERE Title = . ntsc publisherEQL 20/60gbEQLEQLEQL (((cechc/ceche cechc/ceche cechc/ceche cechc/ceche
Col Attention	SELECT Compatible with 20/60GB NTSC PS3 (CECHA/CECHB) FROM table_1_10875694_11 WHERE Compatible with 60GB PAL/80GB NTSC PS3 (CECHC/CECHE) = scus-97265 AND Compatible with 20/60GB NTSC PS3 (CECHA/CECHB) = scus-97265
Pointer Network	SELECT Publisher FROM table_1_10875694_11 WHERE Product No. = scus-97265 pal/80gb ntsc (cecha/cechb)) (((EQL scus-97265
Question 23	What is the total number of James Bond 007: Everything or Nothing for each system?
Table	['Title', 'System', 'Publisher', 'Product No.', 'Compatible with 20/60GB NTSC PS3 (CECHA/CECHB)', 'Compatible with 60GB PAL/80GB NTSC PS3 (CECHC/CECHE)']
Ground Truth	SELECT COUNT(System) FROM table_1_10875694_11 WHERE Title = James Bond 007: Everything or Nothing
LSTM	SELECT COUNT(Compatible with 20/60GB NTSC PS3 (CECHA/CECHB)) FROM table_1_10875694_11 WHERE Title = james bond 007:. or)
BiLSTM	SELECT COUNT(Compatible with 60GB PAL/80GB NTSC PS3 (CECHC/CECHE)) FROM table_1_10875694_11 WHERE System = system
Encoder-Decoder	SELECT COUNT(Product No.) FROM table_1_10875694_11 WHERE Publisher = system
Col Attention	SELECT COUNT(Product No.) FROM table_1_10875694_11 WHERE Title = james bond 007: everything or nothing for each system AND Compatible with 60GB PAL/80GB NTSC PS3 (CECHC/CECHE) = 007: everything or nothing for

Continue on the next page

Table 4: Incorrect matching between the Column names of both the SELECT and the WHERE-Clause

Type	Sentences
Pointer Network	SELECT COUNT(Product No.) FROM table_1_10875694_11 WHERE Title = james bond 007: everything or nothing for each system AND Compatible with 60GB PAL/80GB NTSC PS3 (CECHC/CECHE) = 007: everything or nothing for
Question 24	How many products have the title "Jak 3"?
Table	['Title', 'System', 'Publisher', 'Product No.', 'Compatible with 20/60GB NTSC PS3 (CECHA/CECHB)', 'Compatible with 60GB PAL/80GB NTSC PS3 (CECHC/CECHE)']
Ground Truth	SELECT COUNT(Product No.) FROM table_1_10875694_11 WHERE Title = Jak 3
LSTM	SELECT COUNT(Product No.) FROM table_1_10875694_11 WHERE Title = "jak 3"
BiLSTM	SELECT COUNT(Product No.) FROM table_1_10875694_11 WHERE Title = system jak 3
Encoder-Decoder	SELECT COUNT(Product No.) FROM table_1_10875694_11 WHERE Title = jak 3"(cecha/cechb)EQL 60gb
Col Attention	SELECT COUNT(Product No.) FROM table_1_10875694_11 WHERE Compatible with 60GB PAL/80GB NTSC PS3 (CECHC/CECHE) ="jak 3" AND Compatible with 20/60GB NTSC PS3 (CECHA/CECHB) = "jak 3"
Pointer Network	SELECT COUNT(System) FROM table_1_10875694_11 WHERE Publisher = jak 3 AND Title = 60gb pal/80gb ntsc (() 60gb ntsc ntsc
Question 25	What was the first season of the club who last won a title in 2012?
Table	['Club', 'Position in 2012', 'First season', 'Number of seasons', 'First season of current spell', 'Titles','Last title']
Ground Truth	SELECT First season FROM table_1_1096793_1 WHERE Last title = 2012
LSTM	SELECT MIN(First season) FROM table_1_1096793_1 WHERE Club = 2012
BiLSTM	SELECT First season FROM table_1_1096793_1 WHERE First season = 2012
Encoder-Decoder	SELECT MIN(First season) FROM table_1_1096793_1 WHERE Titles = 2012
Col Attention	SELECT First season FROM table_1_1096793_1 WHERE Last title = 2012

Continue on the next page

Table 4: Incorrect matching between the Column names of both the SELECT and the WHERE-Clause

Type	Sentences
Pointer Network	SELECT First season FROM table_1_1096793_1 WHERE Titles = 2012
Question 26	What percentage of The Bronx voters occured when Manhattan had 29.9% of voters?
Table	['1953', 'party', 'Manhattan', 'The Bronx', 'Brooklyn', 'Queens', 'Richmond [Staten Is.]', 'Total', '%']
Ground Truth	SELECT The Bronx FROM table_1_1108394_34 WHERE Manhattan = 29.9%
LSTM	SELECT % FROM table_1_1108394_34 WHERE % = 29.9%
BiLSTM	SELECT % FROM table_1_1108394_34 WHERE Brooklyn = 29.9%
Encoder-Decoder	SELECT % FROM table_1_1108394_34 WHERE 1953 = [staten is]] 29.9
Col Attention	SELECT % FROM table_1_1108394_34 WHERE % = had 29.9 AND Richmond [Staten Is.] = had 29.9
Pointer Network	SELECT Richmond [Staten Is.] FROM table_1_1108394_34 WHERE Richmond [Staten Is.] = 29.9
Question 27	What number of voters did the Bronx have when the total number was 2054?
Table	['1953', 'party', 'Manhattan', 'The Bronx', 'Brooklyn', 'Queens', 'Richmond [Staten Is.]', 'Total', '%']
Ground Truth	SELECT The Bronx FROM table_1_1108394_34 WHERE Total = 2054
LSTM	SELECT 1953 FROM table_1_1108394_34 WHERE 1953 = 2054
BiLSTM	SELECT 1953 FROM table_1_1108394_34 WHERE Total = 2054
Encoder-Decoder	SELECT 1953 FROM table_1_1108394_34 WHERE Total = 2054
Col Attention	SELECT 1953 FROM table_1_1108394_34 WHERE Total = 2054
Pointer Network	SELECT 1953 FROM table_1_1108394_34 WHERE party = 2054
Question 28	What number of voters did Queens have when Staten Island had 295 voters?
Table	['1953', 'party', 'Manhattan', 'The Bronx', 'Brooklyn', 'Queens', 'Richmond [Staten Is.]', 'Total', '%']
Ground Truth	SELECT Queens FROM table_1_1108394_34 WHERE Richmond [Staten Is.] = 295
LSTM	SELECT COUNT(Total) FROM table_1_1108394_6 WHERE 2013 Republican primary = staten island
BiLSTM	SELECT The Bronx FROM table_1_1108394_34 WHERE Brooklyn = 295
Encoder-Decoder	SELECT Total FROM table_1_1108394_34 WHERE % = 295

Continue on the next page

Table 4: Incorrect matching between the Column names of both the SELECT and the WHERE-Clause

Type	Sentences
Col Attention	SELECT 1953 FROM table_1_1108394_34 WHERE The Bronx = staten island
Pointer Network	SELECT Total FROM table_1_1108394_34 WHERE party = staten
Question 29	What was Staten Island when Brooklyn was 940?
Table	[‘2013 Republican primary’, ‘Manhattan’, ‘The Bronx’, ‘Brooklyn’, ‘Queens’, ‘Staten Island’, ‘Total’, ‘%’]
Ground Truth	SELECT Staten Island FROM table_1_1108394_6 WHERE Brooklyn = 940
LSTM	SELECT Staten Island FROM table_1_1108394_6 WHERE Manhattan = 940
BiLSTM	SELECT Staten Island FROM table_1_1108394_6 WHERE Manhattan = 940
Encoder-Decoder	SELECT Staten Island FROM table_1_1108394_6 WHERE Brooklyn = 940
Col Attention	SELECT Staten Island FROM table_1_1108394_6 WHERE Manhattan = 940
Pointer Network	SELECT Staten Island FROM table_1_1108394_6 WHERE Manhattan = 940
Question 30	What was the total in Brooklyn when Manhattan was 3,139?
Table	[‘2013 Republican primary’, ‘Manhattan’, ‘The Bronx’, ‘Brooklyn’, ‘Queens’, ‘Staten Island’, ‘Total’, ‘%’]
Ground Truth	SELECT Brooklyn FROM table_1_1108394_6 WHERE Manhattan = 3,139
LSTM	SELECT Total FROM table_1_1108394_6 WHERE Queens = 3,139
BiLSTM	SELECT Manhattan FROM table_1_1108394_6 WHERE Manhattan = 3,139
Encoder-Decoder	SELECT Manhattan FROM table_1_1108394_6 WHERE Brooklyn = 3,139
Col Attention	SELECT Total FROM table_1_1108394_6 WHERE Manhattan = 3,139
Pointer Network	SELECT Total FROM table_1_1108394_6 WHERE 2013 Republican primary = 3,139

Table 5: Incorrect WHERE-Clause Operation Prediction

Type	Sentences
Question 1	what’s the width with frame size being 4.5k

Continue on the next page

Table 5: Incorrect WHERE-Clause Operation Prediction

Type	Sentences
Table	[‘Frame size’, ‘Width’, ‘Height’, ‘Mpix’, ‘Aspect Ratio’, ‘max. fps’, ‘lowest possible REDCODE at 24 fps’]
Ground Truth	SELECT Width FROM table_1_1251878_1 WHERE Frame size = 4.5K
LSTM	SELECT Frame size FROM table_1_1251878_1 WHERE Frame size = 4.5k
BiLSTM	SELECT Width FROM table_1_1251878_1 WHERE Frame size = 4.5k
Encoder-Decoder	SELECT Frame size FROM table_1_1251878_1 WHERE Frame size = EQL 4.5k
Col Attention	SELECT Width FROM table_1_1251878_1 WHERE lowest possible REDCODE at 24 fps = 4.5k
Pointer Network	SELECT Width FROM table_1_1251878_1 WHERE Mpix AND Frame size = AND > 24 fps AND 4.5k
Question 2	How many home wins occured win the road games are larger than 3.0 but road win percentage is .000?
Table	[‘Num’, ‘Team’, ‘W’, ‘L’, ‘PCT’, ‘PF’, ‘PA’, ‘Last appearance’, ‘Last championship’, ‘HOME games’, ‘Home wins’, ‘Home losses’, ‘Home Win Pct.‘, ‘ROAD games’, ‘Road wins’, ‘Road losses’, ‘Road Win Pct.’]
Ground Truth	SELECT Home wins FROM table_1_1409106_2 WHERE Road Win Pct. AND ROAD games = AND > .000 AND 3.0
LSTM	SELECT Home wins FROM table_1_1409106_2 WHERE ROAD games > 3.0 AND Last championship = .000
BiLSTM	SELECT Home wins FROM table_1_1409106_2 WHERE Num > 3.0 AND Last championship = .000
Encoder-Decoder	SELECT COUNT(Home wins) FROM table_1_1409106_2 WHERE L >occured AND Num = .000
Col Attention	SELECT COUNT(Home wins) FROM table_1_1409106_2 WHERE ROAD games >.000 AND Road wins = .000
Pointer Network	SELECT Home wins FROM table_1_1409106_2 WHERE W AND Last appearance = AND = .000 AND .000
Question 3	What’s the change over same quarter the previous year in the period when the 89.6% of the trains arrive within 5 minutes of scheduled time (over three months)?

Continue on the next page

Table 5: Incorrect WHERE-Clause Operation Prediction

Type	Sentences
Table	['Period', '% trains arriving within 5 mins of scheduled time (over three months)', 'Change over same quarter the previous year', '% trains arriving within 5 mins of scheduled time Moving Annual Average (MAA)', 'Change over previous year as a whole']
Ground Truth	SELECT Change over same quarter the previous year FROM table_1_171748_3 WHERE % trains arriving within 5 mins of scheduled time (over three months) = 89.6%
LSTM	SELECT High points FROM table_1_13812785_8 WHERE High rebounds > charles AND Team = oakley oakley (9)
BiLSTM	SELECT Change over same quarter the previous year FROM table_1_171748_3 WHERE Period = 89.6%
Encoder-Decoder	SELECT % trains arriving within 5 mins of scheduled time (over three months) FROM table_1_171748_3 WHERE Period = 89.6%
Col Attention	SELECT Change over previous year as a whole FROM table_1_171748_3 WHERE % trains arriving within 5 mins of scheduled time (over three months) = the 89.6% of trains arrive arrive arrive

Table 5: Incorrect WHERE-Clause Operation Prediction

Type	Sentences
Col Attention	SELECT MAX(One year) FROM table_1_174266_6 WHERE 25 and more > 1.0
Pointer Network	SELECT MAX(Four years) FROM table_1_174266_6 WHERE Year = unknown
Question 5	Who weighed 135 lbs that was on the same team as Doug Del Porto (Galena) at 125 lbs?
Table	[‘Class’, ‘Team’, ‘103lbs’, ‘112lbs’, ‘119lbs’, ‘125lbs’, ‘130lbs’, ‘135lbs’, ‘140lbs’]
Ground Truth	SELECT 135lbs FROM table_1_18341614_11 WHERE 125lbs = Doug Del Porto (Galena)
LSTM	SELECT 103lbs FROM table_1_18341614_11 WHERE 130lbs = 135 lbs AND 119lbs > 125
BiLSTM	SELECT 103lbs FROM table_1_18341614_11 WHERE 125lbs = 125 del porto (galena) AND 135lbs = 135 lbs
Encoder-Decoder	SELECT 103lbs FROM table_1_18341614_11 WHERE 130lbs = tallon ricci (white pine) at teamEQL 103
Col Attention	SELECT 103lbs FROM table_1_18341614_11 WHERE 103lbs = 125 lbs
Pointer Network	SELECT 103lbs FROM table_1_18341614_11 WHERE Team AND 140lbs = AND > doug del porto (galena) AND 125
Question 6	What was game 3 when there were under 2.61m viewers?
Table	[‘Episode’, ‘Airdate’, ‘Game 1’, ‘Game 2’, ‘Game 3’,‘Viewers’]
Ground Truth	SELECT Game 3 FROM table_1_27039190_3 WHERE Viewers = Under 2.61m
LSTM	SELECT Game 3 FROM table_1_27039190_3 WHERE Viewers < 2.61m
BiLSTM	SELECT Game 3 FROM table_1_27039190_3 WHERE Viewers < 2.61m
Encoder-Decoder	SELECT Game 3 FROM table_1_27039190_3 WHERE Game 1 = 2.61m
Col Attention	SELECT Game 3 FROM table_1_27039190_3 WHERE Viewers = 2.61m AND Airdate = 2.61m
Pointer Network	SELECT Game 2 FROM table_1_27039190_3 WHERE Viewers > 2.61m
Question 7	What frame of time is listed under years during the 72nd congress?
Table	[‘Representative’, ‘Party’, ‘Years’, ‘Congress’, ‘District Home’, ‘Occupation’, ‘Electoral history’]
Ground Truth	SELECT Years FROM table_1_2841865_2 WHERE Congress = 72nd

Continue on the next page

Table 5: Incorrect WHERE-Clause Operation Prediction

Type	Sentences
LSTM	SELECT Years FROM table_1_2841865_2 WHERE Representative = 72nd congress
BiLSTM	SELECT Occupation FROM table_1_2841865_2 WHERE Years = 72nd
Encoder-Decoder	SELECT Years FROM table_1_2841865_2 WHERE Representative = 72nd congress
Col Attention	SELECT Years FROM table_1_2841865_2 WHERE Representative = 72nd congress
Pointer Network	SELECT Years FROM table_1_2841865_2 WHERE Years > 72nd
Question 8	Tell me the highest overall rank for heat rank more than 4 and lane less than 2
Table	[‘Heat Rank’, ‘Lane’, ‘Swimmer’, ‘Country’, ‘Time’, ‘Overall Rank’]
Ground Truth	SELECT MAX(Overall Rank) FROM table_2_10289926_6 WHERE Heat Rank AND Lane > AND < 4 AND 2
LSTM	SELECT MAX(Overall Rank) FROM table_2_10289926_6 WHERE Heat Rank > 4 AND Lane < 2
BiLSTM	SELECT MAX(Overall Rank) FROM table_2_10289926_6 WHERE Heat Rank > 4 AND Lane<2
Encoder-Decoder	SELECT MAX(Overall Rank) FROM table_2_10289926_6 WHERE Heat Rank > 4 AND Lane < 2
Col Attention	SELECT MAX(Overall Rank) FROM table_2_10289926_6 WHERE Lane < 4 AND Heat Rank >4
Pointer Network	SELECT MAX(Overall Rank) FROM table_2_10289926_6 WHERE Heat Rank AND Heat Rank> AND > 4 AND 2
Question 9	Before round 9, when Takashi Kogure held pole position, who was the team?
Table	[‘Round’, ‘Track’, ‘Date’, ‘Pole Position’, ‘Fastest Lap’, ‘Winner’, ‘Team’]
Ground Truth	SELECT Team FROM table_2_10154733_2 WHERE Round AND Pole Position <AND = 9 AND takashi kogure
LSTM	SELECT Track FROM table_2_10154733_2 WHERE Fastest Lap = takashi kogure
BiLSTM	SELECT Team FROM table_2_10154733_2 WHERE Pole Position = takashi kogure AND Round > 9
Encoder-Decoder	SELECT Team FROM table_2_10154733_2 WHERE Pole Position = takashi AND Date = takashi kogure

Continue on the next page

Table 5: Incorrect WHERE-Clause Operation Prediction

Type	Sentences
Attention	SELECT Track FROM table_2_10154733_2 WHERE Pole Position = takashi kogure
Pointer Network	SELECT Team FROM table_2_10154733_2 WHERE Round AND Fastest Lap > AND = 9 AND takashi kogure
Question 10	Which province and flag which has a water area (km 2) smaller than 176,928 has a total water area percentage of 0.0%?
Table	['Rank', 'Name and flag', 'Water area (km 2)', 'Water area as percentage of total area', 'Percentage of national freshwater area']
Ground Truth	SELECT Name and flag FROM table_2_106094_3 WHERE Water area (km 2) AND Water area as percentage of total area < AND = 176,928 AND 0.0%
LSTM	SELECT Name and flag FROM table_2_106094_3 WHERE Rank < 176,928 AND Water area (km 2) = 0.0
BiLSTM	SELECT Percentage of national freshwater area FROM table_2_106094_3 WHERE Water area (km 2) < 176,928 AND Name and flag = 0.0
Encoder-Decoder	SELECT Name and flag FROM table_2_106094_3 WHERE Rank > 176,928 AND Rank = area (km area 0.0 0.0
Col Attention	SELECT Name and flag FROM table_2_106094_3 WHERE Water area as percentage of total area = 0.0 AND Water area (km 2) < 0.0
Pointer Network	SELECT Percentage of national freshwater area FROM table_2_106094_3 WHERE Water area (km 2) AND Rank > AND > 176,928 AND 176,928
Question 11	Who has a rank below 125 and time of 00: 56.30?
Table	['Rank', 'Heat (Lane)', 'Name', 'Nationality', 'Time']
Ground Truth	SELECT Name FROM table_2_10647310_3 WHERE Rank AND Time < AND = 125 AND 00: 56.30
LSTM	SELECT Name FROM table_2_10647310_3 WHERE Rank < 125 AND Nationality = 00: 56.30
BiLSTM	SELECT Name FROM table_2_10647310_3 WHERE Rank < 125 AND Time = 00: 56.30
Encoder-Decoder	SELECT Name FROM table_2_10647310_3 WHERE Rank < 125 AND Rank = (00 00: 56.30
Col Attention	SELECT Name FROM table_2_10647310_3 WHERE Rank > 125 AND Time = 125

Continue on the next page

Table 5: Incorrect WHERE-Clause Operation Prediction

Type	Sentences
Pointer Network	SELECT Name FROM table_2_10647310_3 WHERE Rank AND Time > AND = 125 AND 00: 56.30
Question 12	Daily ridership greater that 414 is associated with which length?
Table	[‘Line’, ‘Color’, ‘Terminus’, ‘Length’, ‘Stations’, ‘Daily Ridership’]
Ground Truth	SELECT Length FROM table_2_1055707_1 WHERE Daily Ridership > 414
LSTM	SELECT Length FROM table_2_1055707_1 WHERE Length = daily ridership greater
BiLSTM	SELECT Terminus FROM table_2_1055707_1 WHERE Length = 414
Encoder-Decoder	SELECT Terminus FROM table_2_1055707_1 WHERE Stations = 414 AND Length = 414
Col Attention	SELECT Terminus FROM table_2_1055707_1 WHERE Stations = 414 length length length?
Pointer Network	SELECT Daily Ridership FROM table_2_1055707_1 WHERE Length = 414
Question 13	Which competition did T*â*rlea come in 4th place, in 2003 at Budapest, Hungary?
Table	[‘Year’, ‘Competition’, ‘Venue’, ‘Position’, ‘Notes’]
Ground Truth	SELECT Competition FROM table_2_1054502_1 WHERE Position AND Year AND Venue = AND > AND = 4th AND 2003 AND budapest, hungary
LSTM	SELECT Competition FROM table_2_1054502_1 WHERE Position = 4th AND Venue = budapest, hungary AND Venue = budapest
BiLSTM	SELECT Competition FROM table_2_1054502_1 WHERE Notes = târlea AND Year = 2003 AND Venue = budapest, hungary
Encoder-Decoder	SELECT Competition FROM table_2_1054502_1 WHERE Position = târlea AND Venue = EQL place AND Year = budapest, hungary
Col Attention	SELECT Competition FROM table_2_1054502_1 WHERE Year = come AND Position = târlea
Pointer Network	SELECT Competition FROM table_2_1054502_1 WHERE Position AND Year AND Year = AND = AND = 4th place AND budapest AND budapest, hungary
Question 14	Which building was built in 2015, has less than 40 floors and is larger than 509 feet?
Table	[‘Name’, ‘City’, ‘Metres’, ‘Feet’, ‘Floors’, ‘Finalized’]
Ground Truth	SELECT MAX(Metres) FROM table_2_10728418_3 WHERE Finalized AND Floors AND Feet = AND < AND > 2015 AND 40 AND 509

Continue on the next page

Table 5: Incorrect WHERE-Clause Operation Prediction

Type	Sentences
LSTM	SELECT Name FROM table_2_10728418_3 WHERE City = 2015 AND Feet < 40 AND Metres > 509
BiLSTM	SELECT City FROM table_2_10728418_3 WHERE Feet > 40 AND City = 2015 AND Floors > 509
Encoder-Decoder	SELECT Name FROM table_2_10728418_3 WHERE Finalized = 40 AND Metres > 40 AND Finalized> 509
Col Attention	SELECT Name FROM table_2_10728418_3 WHERE Floors > 509 AND Feet > 509 AND City = 2015
Pointer Network	SELECT City FROM table_2_10728418_3 WHERE Finalized AND Floors > AND < 40 AND 509
Question 15	What is the highest floor for the building measuring 220 meters?
Table	['Name', 'City', 'Metres', 'Feet', 'Floors', 'Finalized']
Ground Truth	SELECT MAX(Floors) FROM table_2_10728418_3 WHERE Metres > 220
LSTM	SELECT MAX(Metres) FROM table_2_10728418_3 WHERE Feet = 220
BiLSTM	SELECT MAX(Floors) FROM table_2_10728418_3 WHERE Feet = 220
Encoder-Decoder	SELECT MAX(Floors) FROM table_2_10728418_3 WHERE Feet = 220
Col Attention	SELECT MAX(Floors) FROM table_2_10728418_3 WHERE Metres = 220
Pointer Network	SELECT MAX(Metres) FROM table_2_10728418_3 WHERE Metres = 220
Question 16	What is the smallest draws for a player larger than 2 with a 100% wins?
Table	['Played', 'Lost', 'Drawn', 'Tries', 'Points', '% Won']
Ground Truth	SELECT MIN(Drawn) FROM table_2_1064022_2 WHERE Played AND % Won > AND > 2 AND 100
LSTM	SELECT MIN(Drawn) FROM table_2_1064022_2 WHERE Tries = 100% AND Points > 2
BiLSTM	SELECT MIN(Drawn) FROM table_2_1064022_2 WHERE Drawn > 2 AND % Won = 100
Encoder-Decoder	SELECT MIN(Drawn) FROM table_2_1064022_2 WHERE Tries > 2 AND % Won = 100
Attention	MIN(Drawn) FROM table_2_1064022_2 WHERE % Won = 100 AND Played > 100
Pointer Network	SELECT MIN(Drawn) FROM table_2_1064022_2 WHERE Drawn AND % Won > AND = 2 AND 100
Question 17	What is the average draws for a player larger than 16 with more than 1 tries and a win percentage smaller than 56.15%?

Continue on the next page

Table 5: Incorrect WHERE-Clause Operation Prediction

Type	Sentences
Table	['Played', 'Lost', 'Drawn', 'Tries', 'Points', '% Won']
Ground Truth	SELECT AVG(Drawn) FROM table_2_1064022_2 WHERE Tries AND % Won AND Played > AND < AND > 1 AND 56.15 AND 16
LSTM	SELECT AVG(Drawn) FROM table_2_1064022_2 WHERE Tries > 16 AND Points < 56.15 AND % Won > 56.15
BiLSTM	SELECT AVG(Drawn) FROM table_2_1064022_2 WHERE Played>16 AND Tries > 1 AND % Won <56.15
Encoder-Decoder	SELECT AVG(Drawn) FROM table_2_1064022_2 WHERE Played > 16 AND Played = won 1 AND % Won < 56.15
Col Attention	SELECT AVG(Drawn) FROM table_2_1064022_2 WHERE Tries > 56.15 AND % Won < 16 AND Played > 56.15
Pointer Network	SELECT AVG(Drawn) FROM table_2_1064022_2 WHERE Drawn AND Tries AND % Won > AND = AND < 16 AND 56.15 AND 56.15
Question 18	What is the highest number of losses with 23 points and 22 plays?
Table	['Played', 'Lost', 'Drawn', 'Tries', 'Points', '% Won']
Ground Truth	SELECT MAX(Lost) FROM table_2_10516034_1 WHERE Points AND Played = AND > 23 AND 22
LSTM	SELECT SUM(Tries) FROM table_2_1064022_2 WHERE Points > 2 AND Played < 2
BiLSTM	
Encoder-Decoder	SELECT MAX(Lost) FROM table_2_10516034_1 WHERE Points = 23 AND Played = 22
Col Attention	SELECT MAX(Lost) FROM table_2_10516034_1 WHERE Points = 22 AND Played = 22
Pointer Network	SELECT MAX(Lost) FROM table_2_10516034_1 WHERE Points AND Points = AND = 23 AND 22
Question 19	What are the average points White made with 0 extra points?
Table	['Played', 'Lost', 'Drawn', 'Tries', 'Points', '% Won']
Ground Truth	SELECT AVG(Points) FROM table_2_14342210_2 WHERE Extra points AND Player = AND = 0 AND white
LSTM	SELECT AVG(Extra points) FROM table_2_14342210_2 WHERE Extra points > 0
BiLSTM	SELECT AVG(Extra points) FROM table_2_14342210_2 WHERE Points = white AND Extra points < 0

Continue on the next page

Table 5: Incorrect WHERE-Clause Operation Prediction

Type	Sentences
Encoder-Decoder	SELECT AVG(Field goals) FROM table_2_14342210_2 WHERE Extra points = 0
Col Attention	SELECT AVG(Points) FROM table_2_14342210_2 WHERE Extra points < 0
Pointer Network	SELECT AVG(Extra points) FROM table_2_14342210_2 WHERE Extra points > 0
Question 20	What is the notes distance for 2nd position earlier than 1986?
Table	[‘Year’, ‘Title’, ‘Role’, ‘Director’, ‘Notes’]
Ground Truth	SELECT Notes FROM table_2_14617261_1 WHERE Position AND Year = AND < 2nd AND 1986
LSTM	SELECT Notes FROM table_2_14617261_1 WHERE Position = 2nd AND Year < 1986
BiLSTM	SELECT Notes FROM table_2_14617261_1 WHERE Position = 2nd AND Year< 1986
Encoder-Decoder	SELECT Notes FROM table_2_14617261_1 WHERE Position = 2nd AND Year < 1986
Col Attention	SELECT Notes FROM table_2_14617261_1 WHERE Position = 1986 AND Year < 1986
Pointer Network	SELECT Notes FROM table_2_14617261_1 WHERE Position AND Year = AND > 2nd AND 1986
Question 21	On May 9 after Game 3, what was the Opponent?
Table	[‘Date’, ‘Opponent’, ‘Home / Away’, ‘Score’, ‘High points’, ‘High rebounds’, ‘High assists’, ‘Location/Attendance’, ‘Record’]
Ground Truth	SELECT Opponent FROM table_2_14323142_11 WHERE Game AND Date > AND = 3 AND may 9
LSTM	SELECT Opponent FROM table_2_14323142_11 WHERE Game > 3 AND Date = may 9, 3
BiLSTM	SELECT Opponent FROM table_2_14323142_11 WHERE Date > 3 AND Series = may 9
Encoder-Decoder	SELECT Opponent FROM table_2_14323142_11 WHERE Game = 3 AND Date = may 9
Col Attention	SELECT Opponent FROM table_2_14323142_11 WHERE Date = 9 after AND Game > 9 after

Continue on the next page

Table 5: Incorrect WHERE-Clause Operation Prediction

Type	Sentences
Pointer Network	SELECT Opponent FROM table_2_14323142_11 WHERE Game AND Date = AND = 3 AND may 9
Question 22	What Goals with a Level of 5 or greater is the lowest?
Table	['Season', 'Level', 'Position', 'Nationality', 'Apps', 'Goals']
Ground Truth	SELECT MIN(Goals) FROM table_2_13874304_2 WHERE Level > 5
LSTM	SELECT MIN(Goals) FROM table_2_13874304_2 WHERE Level = 5 or greater
BiLSTM	SELECT MIN(Goals) FROM table_2_13874304_2 WHERE Level = 5 or greater
Encoder-Decoder	SELECT MIN(Goals) FROM table_2_13874304_2 WHERE Goals = 5
Col Attention	SELECT MIN(Goals) FROM table_2_13874304_2 WHERE Level > 5
Pointer Network	SELECT MIN(Goals) FROM table_2_13874304_2 WHERE Position AND Level = AND > 5 AND 5
Question 23	Name the date for 2011 and position larger than 2.0
Table	['Season', 'Date', 'Location', 'Discipline', 'Position', 'FIS points']
Ground Truth	SELECT Date FROM table_1_30073089_2 WHERE Season = 2011 AND Position > 2.0
LSTM	
BiLSTM	SELECT Date FROM table_1_30073089_2 WHERE Season > 2.0 AND Position = 2011
Encoder-Decoder	SELECT Date FROM table_1_30073089_2 WHERE Position = 2011 AND Position = 2.0
Col Attention	SELECT Date FROM table_1_30073089_2 WHERE Position > 2.0 AND Date = 2.0
Pointer Network	SELECT Date FROM table_1_30073089_2 WHERE Season = 2011 AND Position > 2.0
Question 24	What team did Derek Black Play for prior to being drafted?
Table	['Pick #', 'Player', 'Position', 'Nationality', 'NHL team', 'College/junior/-club team']
Ground Truth	SELECT College/junior/club team FROM table_1_1473672_6 WHERE Player = Derek Black
LSTM	SELECT Nationality FROM table_1_1473672_6 WHERE NHL team = derek black

Continue on the next page

Table 5: Incorrect WHERE-Clause Operation Prediction

Type	Sentences
BiLSTM	SELECT Nationality FROM table_1_1473672_6 WHERE Player = derek black
Encoder-Decoder	SELECT Nationality FROM table_1_1473672_6 WHERE Player = derek black
Col Attention	SELECT NHL team FROM table_1_1473672_6 WHERE Player = derek black
Pointer Network	SELECT NHL team FROM table_1_1473672_6 WHERE Player = derek black AND Position < drafted
Question 25	If unknown is larger than 1.0, what is the maximum two year amount?
Table	['Year', 'Total', 'Less than a year', 'One year', 'Two years', 'Three years', 'Four years', '5–9 years', '10–14 years', '15–19 years', '20–24 years', '25 and more', 'Unknown']
Ground Truth	SELECT MAX(Two years) FROM table_1_174266_6 WHERE Unknown > 1.0
LSTM	SELECT MAX(Three years) FROM table_1_174266_6 WHERE Less than a year > 1.0
BiLSTM	SELECT MAX(Four years) FROM table_1_174266_6 WHERE Unknown > 1.0
Encoder-Decoder	SELECT MAX(Year) FROM table_1_174266_6 WHERE 5–9 years = 1.0
Col Attention	SELECT MAX(One year) FROM table_1_174266_6 WHERE 25 and more > 1.0
Pointer Network	SELECT MAX(Four years) FROM table_1_174266_6 WHERE Year = unknown
Question 26	Which charts had debut sales of of more than 339333.011497678?
Table	['Release', 'Oricon Albums Chart', 'Peak Position', 'Debut Sales (copies)', 'Sales Total (copies)', 'Chart Run']
Ground Truth	SELECT Oricon Albums Chart FROM table_1_23180638_1 WHERE Debut Sales (copies) > 339333.011497678
LSTM	SELECT Release FROM table_1_23180638_1 WHERE Release > 339333.011497678
BiLSTM	SELECT Peak Position FROM table_1_23180638_1 WHERE Oricon Albums Chart > 339333.011497678
Encoder-Decoder	SELECT COUNT(Peak Position) FROM table_1_23180638_1 WHERE

Continue on the next page

Table 5: Incorrect WHERE-Clause Operation Prediction

Type	Sentences
Col Attention	SELECT Oricon Albums Chart FROM table_1_23180638_1 WHERE Debut Sales (copies) > 339333.011497678
Pointer Network	SELECT Oricon Albums Chart FROM table_1_23180638_1 WHERE Release = 339333.011497678
Question 27	What was game 3 when there were under 2.61m viewers?
Table	['Episode', 'Airdate', 'Game 1', 'Game 2', 'Game 3', 'Viewers']
Ground Truth	SELECT Game 3 FROM table_1_27039190_3 WHERE Viewers = Under 2.61m
LSTM	SELECT Game 3 FROM table_1_27039190_3 WHERE Viewers < 2.61m
BiLSTM	SELECT Game 3 FROM table_1_27039190_3 WHERE Viewers = 2.61m
Encoder-Decoder	SELECT Game 3 FROM table_1_27039190_3 WHERE Game 1 = 2.61m
Col Attention	SELECT Game 3 FROM table_1_27039190_3 WHERE Viewers = 2.61m AND Airdate = 2.61m
Pointer Network	SELECT Game 2 FROM table_1_27039190_3 WHERE Viewers > 2.61m
Question 28	What was the series record at after game 3?
Table	['Game', 'Date', 'Team', 'Score', 'High points', 'High rebounds', 'High assists', 'Location Attendance', 'Series']
Ground Truth	SELECT Series FROM table_1_27700530_15 WHERE Game = 3
LSTM	SELECT Team FROM table_1_27700530_15 WHERE Game > 3
BiLSTM	SELECT Series FROM table_1_27700530_15 WHERE Game > 3
Encoder-Decoder	SELECT Series FROM table_1_27700530_15 WHERE Game > 3
Col Attention	SELECT Series FROM table_1_27700530_15 WHERE Game > 3
Pointer Network	SELECT Series FROM table_1_27700530_15 WHERE Game > 3
Question 29	Tell me the highest overall rank for heat rank more than 4 and lane less than 2
Table	['Heat Rank', 'Lane', 'Swimmer', 'Country', 'Time', 'Overall Rank']
Ground Truth	SELECT MAX(Overall Rank) FROM table_2_10289926_6 WHERE Heat Rank > 4 AND Lane < 2
LSTM	SELECT MAX(Overall Rank) FROM table_2_10289926_6 WHERE Heat Rank > 4 AND Lane < 2
BiLSTM	SELECT MAX(Overall Rank) FROM table_2_10289926_6 WHERE Heat Rank > 4 AND Time < 2
Encoder-Decoder	SELECT MAX(Overall Rank) FROM table_2_10289926_6 WHERE Heat Rank > 4 AND Lane < 2

Continue on the next page

Table 5: Incorrect WHERE-Clause Operation Prediction

Type	Sentences
Col Attention	SELECT MAX(Overall Rank) FROM table_2_10289926_6 WHERE Lane < 4 AND Heat Rank > 4
Pointer Network	SELECT MAX(Overall Rank) FROM table_2_10289926_6 WHERE Heat Rank > 4 AND Heat Rank > 2
Question 30	What are the election totals for David Cook who has 6,963 votes and 21.3 percentage?
Table	['Party', 'Constituency', 'Election', 'Candidate', 'Votes', 'Percentage', 'Position']
Ground Truth	SELECT COUNT(Election) FROM table_2_10107334_6 WHERE Candidate = david cook AND Votes = 6,963 AND Percentage > 21.3
LSTM	SELECT Election FROM table_2_10107334_6 WHERE Percentage = 21.3 AND Candidate = david cook AND Party = david cook
BiLSTM	SELECT Election FROM table_2_10107334_6 WHERE Percentage = 6,963 AND Constituency = david cook AND Percentage = 6,963
Encoder-Decoder	SELECT Election FROM table_2_10107334_6 WHERE Votes = 6,963 AND Candidate = david AND Candidate = david cook
Col Attention	SELECT Election FROM table_2_10107334_6 WHERE Percentage = 21.3 AND Votes < 21.3
Pointer Network	SELECT Constituency FROM table_2_10107334_6 WHERE Constituency = david cook AND Percentage = 21.3

Table 6: Incorrect WHERE-Clause Condition prediction

Type	Description
Question 1	How many different college/junior/club teams provided a player to the Washington Capitals NHL Team?
Table	['Pick', 'Player', 'Position', 'Nationality', 'NHL team', 'College/junior/club team']
Ground Truth	SELECT COUNT(College/junior/club team) FROM table_1_1013129_2 WHERE NHL team = Washington Capitals
LSTM	SELECT COUNT(NHL team) FROM table_1_1013129_2 WHERE NHL team = washington capitals nhl
BiLSTM	SELECT COUNT(NHL team) FROM table_1_1013129_2 WHERE NHL team = washington capitals

Continue on the next page

Table 6: Incorrect WHERE-Clause Condition prediction

Type	Description
Encoder-Decoder	SELECT COUNT(NHL team) FROM table_1_1013129_2 WHERE NHL team = washington capitals nhl
Col Attention	SELECT COUNT(NHL team) FROM table_1_1013129_2 WHERE NHL team = washington capitals nh
Pointer Network	SELECT COUNT(NHL team) FROM table_1_1013129_2 WHERE NHL team = washington capitals nhl
Question 2	What's Dorain Anneck's pick number?
Table	['Pick', 'Player', 'Position', 'Nationality', 'NHL team', 'College/junior/club team']
Ground Truth	SELECT Pick FROM table_1_1013129_3 WHERE Player = Dorain Anneck
LSTM	SELECT COUNT(Pick) FROM table_1_1013129_3 WHERE Player = dorain anneck
BiLSTM	SELECT Pick FROM table_1_1013129_3 WHERE Player = dorain anneck
Encoder-Decoder	SELECT Pick FROM table_1_1013129_3 WHERE Player = dorain anneck
Col Attention	SELECT COUNT(Pick) FROM table_1_1013129_3 WHERE Player = dorain anneck
Pointer Network	SELECT COUNT(Pick) FROM table_1_1013129_3 WHERE Player = dorain anneck
Question 3	What could a spanish coronel be addressed as in the commonwealth military?
Table	['Equivalent NATO Rank Code', 'Rank in Spanish', 'Rank in English', 'Commonwealth equivalent', 'US Air Force equivalent']
Ground Truth	SELECT Commonwealth equivalent FROM table_1_1015521_2 WHERE Rank in Spanish = Coronel
LSTM	SELECT Rank in Spanish FROM table_1_1015521_2 WHERE Equivalent NATO Rank Code = commonwealth
BiLSTM	SELECT Rank in Spanish FROM table_1_1015521_2 WHERE Equivalent NATO Rank Code = commonwealth
Encoder-Decoder	SELECT US Air Force equivalent FROM table_1_1015521_2 WHERE Rank in English = commonwealth
Col Attention	SELECT Commonwealth equivalent FROM table_1_1015521_2 WHERE Rank in English = commonwealth military
Pointer Network	SELECT Commonwealth equivalent FROM table_1_1015521_2 WHERE Rank in English = commonwealth military

Continue on the next page

Table 6: Incorrect WHERE-Clause Condition prediction

Type	Description
Question 4	who is the opponent where tv is abc and game site is tampa stadium?
Table	[‘Week’, ‘Date’, ‘Opponent’,‘Result’, ‘Kickoff [a]’, ‘Venue’, ‘TV’, ‘Attendance’, ‘Record’]
Ground Truth	SELECT Opponent FROM table_1_11406866_2 WHERE TV = ABC AND Game site = Tampa Stadium
LSTM	SELECT Opponent FROM table_1_11406866_2 WHERE Week = abc AND Opponent = tampa stadium
BiLSTM	SELECT Opponent FROM table_1_11406866_2 WHERE TV = abc AND Game site = tampa stadium
Encoder-Decoder	SELECT Opponent FROM table_1_11406866_2 WHERE Week =] AND Week = tampa stadium stadium
Col Attention	SELECT Opponent FROM table_1_11406866_2 WHERE Game site = tampa stadium AND TV = abc
Pointer Network	SELECT Opponent FROM table_1_11406866_2 WHERE Game site = tampa stadium AND TV = abc
Question 5	what’s the total w–l where player is boro jovanović category:articles with hcards?
Table	[‘Player’, ‘Total W–L’, ‘Singles W–L’, ‘Doubles W–L’, ‘Ties played’, ‘Debut’, ‘Years played’]
Ground Truth	SELECT Total W–L FROM table_1_10294071_1 WHERE Player = Boro Jovanović Category:Articles with hCards
LSTM	SELECT COUNT(Total W–L) FROM table_1_10294071_1 WHERE Player = boro jovanović category:articles with hcards
BiLSTM	SELECT Total W–L FROM table_1_10294071_1 WHERE Player = boro jovanović category:articles with hcards
Encoder-Decoder	SELECT COUNT(Total W–L) FROM table_1_10294071_1 WHERE Player = boro jovanović category:articles with hcards
Col Attention	SELECT Total W–L FROM table_1_10294071_1 WHERE Player = jovanović category:articles with
Pointer Network	SELECT Total W–L FROM table_1_10294071_1 WHERE Player = jovanović category:articles with
Question 6	what’s the total number of tasmania with new south wales crop of 190 kilotonnes?

Continue on the next page

Table 6: Incorrect WHERE-Clause Condition prediction

Type	Description
Table	[‘Crop (kilotonnes)’, ‘New South Wales’, ‘Victoria’, ‘Queensland’, ‘Western Australia’, ‘South Australia’, ‘Tasmania’, ‘Total’]
Ground Truth	SELECT COUNT(Tasmania) FROM table_1_1057262_2 WHERE New South Wales = 190
LSTM	SELECT COUNT(Total) FROM table_1_1057262_2 WHERE New South Wales = 190 kilotonnes (
BiLSTM	SELECT COUNT(Victoria) FROM table_1_1057262_2 WHERE Western Australia = 190 kilotonnes
Encoder-Decoder	SELECT COUNT(Total) FROM table_1_1057262_2 WHERE Crop (kilotonnes) = 190 kilotonnes
Col Attention	SELECT COUNT(Total) FROM table_1_1057262_2 WHERE New South Wales = 190 kilotonne
Pointer Network	SELECT COUNT(Total) FROM table_1_1057262_2 WHERE South Australia = 190 kilotonnes
Question 7	Which wrestlers have had 2 reigns?
Table	[‘Rank Each wrestlers total number of days as champion are ranked highest to lowest; wrestlers with the same number mean that they are tied for that certain rank.’,‘Wrestler’, ‘# of reigns’, ‘Combined defenses’, ‘Combined days’]
Ground Truth	SELECT Wrestler FROM table_1_10182508_5 WHERE # of reigns = 2
LSTM	SELECT COUNT(Wrestler) FROM table_1_10182508_5 WHERE Rank Each wrestlers total number of days as champion are ranked highest to lowest; wrestlers with the same number mean that they are tied for that certain rank. = wrestler AND Rank Each wrestlers total number of days as champion are ranked highest to lowest; wrestlers with the same number mean that they are tied for that certain rank. = combined AND Rank Each wrestlers total number of days as champion are ranked highest to lowest; wrestlers with the same number mean that they are tied for that certain rank. = combined
BiLSTM	SELECT Combined defenses FROM table_1_10182508_5 WHERE Rank Each wrestlers total number of days as champion are ranked highest to lowest; wrestlers with the same number mean that they are tied for that certain rank. = 2

Continue on the next page

Table 6: Incorrect WHERE-Clause Condition prediction

Type	Description
Encoder-Decoder	SELECT COUNT(Wrestler) FROM table_1_10182508_5 WHERE Rank Each wrestlers total number of days as champion are ranked highest to lowest; wrestlers with the same number mean that they are tied for that certain rank. = 2
Col Attention	SELECT COUNT(Wrestler) FROM table_1_10182508_5 WHERE Wrestler =
Pointer Network	SELECT COUNT(Wrestler) FROM table_1_10182508_5 WHERE Wrestler =
Question 8	Which province is bay of islands in
Table	['Member', 'Electorate', 'Province', 'MPs term', 'Election date']
Ground Truth	SELECT Province FROM table_1_1024710_2 WHERE Electorate = Bay of Islands
LSTM	SELECT Province FROM table_1_1024710_2 WHERE Electorate = bay of islands
BiLSTM	SELECT Province FROM table_1_1024710_2 WHERE Province = islands, islands
Encoder-Decoder	SELECT Province FROM table_1_1024710_2 WHERE Member = bay of islands in
Col Attention	SELECT Province FROM table_1_1024710_2 WHERE Province = in
Pointer Network	SELECT Province FROM table_1_1024710_2 WHERE Province = in
Question 9	Which province is grey and bell electorate in
Table	['Member', 'Electorate', 'Province', 'MPs term', 'Election date']
Ground Truth	SELECT Province FROM table_1_1024710_2 WHERE Electorate = Grey and Bell
LSTM	SELECT Province FROM table_1_1024710_2 WHERE Electorate = grey and bell electorate
BiLSTM	SELECT Province FROM table_1_1024710_2 WHERE Electorate = grey AND Member = bell electorate bell electorate
Encoder-Decoder	SELECT Province FROM table_1_1024710_2 WHERE Member = grey and bell electorate
Col Attention	SELECT Province FROM table_1_1024710_2 WHERE Electorate = grey and AND MPs term = grey and bell
Pointer Network	SELECT Province FROM table_1_1024710_2 WHERE Electorate = grey and AND MPs term = grey and bell

Continue on the next page

Table 6: Incorrect WHERE-Clause Condition prediction

Type	Description
Question 10	Give me a list of all spanish officer titles that could receive recognition as group captain in english
Table	['Equivalent NATO Rank Code', 'Rank in Spanish', 'Rank in English', 'Commonwealth equivalent', 'US Air Force equivalent']
Ground Truth	SELECT Rank in English FROM table_1_1015521_2 WHERE Commonwealth equivalent = Group Captain
LSTM	SELECT Rank in Spanish FROM table_1_1015521_2 WHERE Equivalent NATO Rank Code = spanish
BiLSTM	SELECT Commonwealth equivalent FROM table_1_1015521_2 WHERE Equivalent NATO Rank Code = english
Encoder-Decoder	SELECT Rank in Spanish FROM table_1_1015521_2 WHERE Equivalent NATO Rank Code = in english group captain
Col Attention	SELECT Equivalent NATO Rank Code FROM table_1_1015521_2 WHERE Rank in English = english
Pointer Network	SELECT Equivalent NATO Rank Code FROM table_1_1015521_2 WHERE Rank in English = english
Question 11	Where was d: ∼50nm, l: ∼600nm geometry researched?
Table	['Material', 'Type', 'Geometry', 'Output voltage', 'Output power', 'Synthesis', 'Researched at']
Ground Truth	SELECT Researched at FROM table_1_30057479_1 WHERE Geometry = D: ∼50nm, L: ∼600nm
LSTM	SELECT Geometry FROM table_1_30057479_1 WHERE Material = d: ∼50nm, l: ∼600nm geometry researched
BiLSTM	SSELECT Researched at FROM table_1_30057479_1 WHERE Geometry = d: ∼50nm, l: ∼600nm geometry researched
Encoder-Decoder	SELECT Geometry FROM table_1_30057479_1 WHERE Type = d: ∼50nm, l: ∼600nm
Col Attention	SELECT Researched at FROM table_1_30057479_1 WHERE Type = d: ∼50nm, l: ∼600nm geometry researched
Pointer Network	SELECT Geometry FROM table_1_30057479_1 WHERE Type = d: ∼50nm l:∼600nm geometry researched
Question 12	How many combined days did Go Shiozaki have?

Continue on the next page

Table 6: Incorrect WHERE-Clause Condition prediction

Type	Description
Table	['Rank Each wrestlers total number of days as champion are ranked highest to lowest;wrestlers with the same number mean that they are tied for that certain rank.', 'Wrestler', '# of reigns', 'Combined defenses', 'Combined days']
Ground Truth	SELECT Combined days FROM table_1_10182508_5 WHERE Wrestler = Go Shiozaki
LSTM	SELECT COUNT(Combined days) FROM table_1_10182508_5 WHERE Rank Each wrestlers total number of days as champion are ranked highest to lowest; wrestlers with the same number mean that they are tied for that certain rank. = go shiozaki
BiLSTM	SELECT COUNT(# of reigns) FROM table_1_10182508_5 WHERE Rank Each wrestlers total number of days as champion are ranked highest to lowest; wrestlers with the same number mean that they are tied for that certain rank. = go shiozaki
Encoder-Decoder	SELECT COUNT(# of reigns) FROM table_1_10182508_5 WHERE Rank Each wrestlers total number of days as champion are ranked highest to lowest; wrestlers with the same number mean that they are tied for that certain rank. = shiozaki shiozaki
Col Attention	SELECT COUNT(Combined days) FROM table_1_10182508_5 WHERE Wrestler = have
Pointer Network	SELECT COUNT(Combined days) FROM table_1_10182508_5 WHERE Wrestler = have
Question 13	How many different college/junior/club teams provided a player to the Washington Capitals NHL Team?
Table	['Pick', 'Player', 'Position', 'Nationality', 'NHL team', 'College/junior/club team']
Ground Truth	SELECT COUNT(College/junior/club team) FROM table_1_1013129_2 WHERE NHL team = Washington Capitals
LSTM	SELECT COUNT(NHL team) FROM table_1_1013129_2 WHERE NHL team = washington capitals nhl
BiLSTM	SELECT COUNT(NHL team) FROM table_1_1013129_2 WHERE NHL team = washington capitals
Encoder-Decoder	SELECT COUNT(NHL team) FROM table_1_1013129_2 WHERE NHL team = washington capitals nhl

Continue on the next page

Table 6: Incorrect WHERE-Clause Condition prediction

Type	Description
Attention	SELECT COUNT(NHL team) FROM table_1_1013129_2 WHERE NHL team = washington capitals nhl
Pointer Network	SELECT COUNT(NHL team) FROM table_1_1013129_2 WHERE NHL team = washington capitals nhl
Question 14	what's the minimum tasmania value
Table	['Crop (kilotonnes)', 'New South Wales', 'Victoria', 'Queensland', 'Western Australia', 'South Australia', 'Tasmania', 'Total']
Ground Truth	SELECT MIN(Tasmania) FROM table_1_1057262_2 WHERE
LSTM	SELECT MIN(Victoria) FROM table_1_1057262_2 WHERE
BiLSTM	SELECT MIN(Victoria) FROM table_1_1057262_2 WHERE
Encoder-Decoder	SELECT MIN(Victoria) FROM table_1_1057262_2 WHERE
Col Attention	SELECT MIN(Victoria) FROM table_1_1057262_2 WHERE
Pointer Network	SELECT MIN(Crop (kilotonnes)) FROM table_1_1057262_2 WHERE
Question 15	Which examples ask the existential question "Can I Love?"
Table	['Approximate Age', 'Virtues', 'Psycho Social Crisis', 'Significant Relationship', 'Existential Question [not in citation given]', 'Examples']
Ground Truth	SELECT Examples FROM table_1_1058787_1 WHERE Existential Question [not in citation given] = Can I Love?
LSTM	SELECT Examples FROM table_1_1058787_1 WHERE Approximate Age = can i love
BiLSTM	SELECT Examples FROM table_1_1058787_1 WHERE Significant Relationship = can i love
Encoder-Decoder	SELECT Psycho Social Crisis FROM table_1_1058787_1 WHERE Virtues = can i love
Col Attention	SELECT Examples FROM table_1_1058787_1 WHERE Virtues = "can i?"
Pointer Network	SELECT Virtues table_1_1058787_1 Virtues = can i love love
Question 16	who were the winners in west virginia
Table	['State (linked to summaries below)', 'Senator', 'Party', 'Electoral history', 'Result', 'Candidates Winning candidate in bold']
Ground Truth	SELECT Candidates Winning candidate in bold FROM table_1_1133844_4 WHERE State (linked to summaries below) = West Virginia
LSTM	SELECT Electoral history FROM table_1_1133844_4 WHERE State (linked to summaries below) = west virginia AND Senator = bold

Continue on the next page

Table 6: Incorrect WHERE-Clause Condition prediction

Type	Description
BiLSTM	SELECT Candidates Winning candidate in bold FROM table_1_1133844_4 WHERE Electoral history = west virginia
Encoder-Decoder	SELECT Senator FROM table_1_1133844_4 WHERE State (linked to summaries below) = west virginia
Col Attention	SELECT Party FROM table_1_1133844_4 WHERE Electoral history = west west west west west west west west west west west west
Pointer Network	SELECT Electoral history FROM table_1_1133844_4 WHERE State (linked to summaries below) = west virginia
Question 17	who won the senate seat in arkansas
Table	['State (linked to summaries below)', 'Senator', 'Party', 'Electoral history', 'Result', 'Candidates Winning candidate in bold']
Ground Truth	SELECT Senator FROM table_1_1133844_4 WHERE State (linked to summaries below) = Arkansas
LSTM	SELECT Electoral history FROM table_1_1133844_4 WHERE State (linked to summaries below) = arkansas
BiLSTM	SELECT Senator FROM table_1_1133844_4 WHERE State (linked to summaries below) = arkansas
Encoder-Decoder	SELECT State (linked to summaries below) FROM table_1_1133844_4 WHERE State (linked to summaries below) = (linked arkansas
Attention	SELECT Senator FROM table_1_1133844_4 WHERE Party = seat in arkansas
Pointer Network	SELECT Senator FROM table_1_1133844_4 WHERE Party = seat in arkansas
Question 18	what's the original air date with title "hell"
Table	['Series no.', 'Season no.', 'Title', 'Directed by', 'Written by', 'Original air date', 'Production Code']
Ground Truth	SELECT Original air date FROM table_1_10610087_5 WHERE Title = "Hell"
LSTM	SELECT Original air date FROM table_1_10610087_5 WHERE Title = "hell"
BiLSTM	SELECT Original air date FROM table_1_10610087_5 WHERE Title = "hell"
Encoder-Decoder	SELECT Original air date FROM table_1_10610087_5 WHERE Title = "hell"

Continue on the next page

Table 6: Incorrect WHERE-Clause Condition prediction

Type	Description
Col Attention	SELECT Original air date FROM table_1_10610087_5 WHERE Title = “hell
Pointer Network	SELECT Original air date FROM table_1_10610087_5 WHERE Title = “hell”
Question 19	Name the grand prix with a driver of Michael Schumacher and a round of 7?
Table	[‘Round’, ‘Grand Prix’, ‘Pole Position’, ‘Fastest Lap’, ‘Winning Driver’, ‘Winning Constructor’, ‘Report’]
Ground Truth	SELECT Grand Prix FROM table_1_1137696_3 WHERE Winning Driver = Michael Schumacher AND Fastest Lap = Michael Schumacher AND Round = 7
LSTM	SELECT Grand Prix FROM table_1_1137696_3 WHERE Grand Prix = 7 AND Grand Prix = michael schumacher
BiLSTM	SELECT Grand Prix FROM table_1_1137696_3 WHERE Winning Driver = michael schumacher AND Round = 7
Encoder-Decoder	SELECT Grand Prix FROM table_1_1137696_3 WHERE Round = michael schumacher AND Round = 7
Col Attention	SELECT Grand Prix FROM table_1_1137696_3 WHERE Round = 7 AND Winning Driver = michael schumacher
Pointer Network	SELECT Grand Prix FROM table_1_1137696_3 WHERE Round = 7 AND Winning Driver = michael schumacher
Question 20	Which junior high school has male and female genders?
Table	[‘Specification’, ‘Gender’, ‘Junior High School (12–15 yrs)’, ‘Senior High School (15–18 yrs)’, ‘University students and Adults (18yrs+)’]
Ground Truth	SELECT Junior High School (12–15 yrs) FROM table_2_13555999_1 WHERE Gender = male and female
LSTM	SELECT Gender FROM table_2_13555999_1 WHERE Specification = female
BiLSTM	SELECT Senior High School (15–18 yrs) FROM table_2_13555999_1 WHERE Gender = male
Encoder-Decoder	SELECT Senior High School (15–18 yrs) FROM table_2_13555999_1 WHERE Gender = male AND Specification = female
Col Attention	SELECT Specification FROM table_2_13555999_1 WHERE Gender = genders AND University students and Adults (18yrs+) = genders

Continue on the next page

Table 6: Incorrect WHERE-Clause Condition prediction

Type	Description
Pointer Network	SELECT Specification FROM table_2_13555999_1 WHERE Junior High School (12–15 yrs) = male
Question 21	Which gender is associated with University students and adults of 25mm?
Table	['Specification', 'Gender', 'Junior High School (12–15 yrs)', 'Senior High School (15–18 yrs)', 'University students and Adults (18yrs+)']
Ground Truth	SELECT Gender FROM table_2_13555999_1 WHERE University students and Adults (18yrs+) = 25mm
LSTM	SELECT Gender FROM table_2_13555999_1 WHERE Junior High School (12–15 yrs) = 25mm
BiLSTM	SELECT Senior High School (15–18 yrs) FROM table_2_13555999_1 WHERE Senior High School (15–18 yrs) = university
Encoder-Decoder	SELECT Senior High School (15–18 yrs) FROM table_2_13555999_1 WHERE Gender = university AND Specification = high school () 26mm 26mm
Col Attention	SELECT Gender FROM table_2_13555999_1 WHERE University students and Adults (18yrs+) = 25mm
Pointer Network	SELECT Gender FROM table_2_13555999_1 WHERE Specification = university
Question 22	On what Date was Palhinha Brazil Scorers?
Table	['Date', 'Result', 'Score', 'Brazil scorers', 'Competition']
Ground Truth	SELECT Date FROM table_2_15152729_6 WHERE Brazil scorers = palhinha
LSTM	SELECT Date FROM table_2_15152729_6 WHERE Brazil scorers = palhinha brazil
BiLSTM	SELECT Date FROM table_2_15152729_6 WHERE Brazil scorers = palhinha brazil
Encoder-Decoder	SELECT Date FROM table_2_15152729_6 WHERE Date = palhinha brazil
Col Attention	SELECT Date FROM table_2_15152729_6 WHERE Brazil scorers = palhinha brazil scorers
Pointer Network	SELECT Date FROM table_2_15152729_6 WHERE Brazil scorers = palhinha brazil
Question 23	what is the name of the netherlands head quarters
Table	['Rank', 'Name', 'Headquarters', 'Revenue Million', 'Profit Million', 'Employees (World)']

Continue on the next page

Table 6: Incorrect WHERE-Clause Condition prediction

Type	Description
Ground Truth	SELECT Name FROM table_2_14094649_10 WHERE Headquarters = the netherlands
LSTM	SELECT Name FROM table_2_14094649_10 WHERE Market value Headquarters = quarters
BiLSTM	SELECT Name FROM table_2_14094649_10 WHERE Primary industry = netherlands head
Encoder-Decoder	SELECT Name FROM table_2_14094649_10 WHERE Rank = netherlands
Col Attention	SELECT Name FROM table_2_14094649_10 WHERE Headquarters = head
Pointer Network	SELECT Name FROM table_2_14094649_10 WHERE Rank = netherlands
Question 24	Capital of brześć nad bugiem has what area (1930) in 1000skm?
Table	['s Car plate (starting 1937)', 'Voivodeship or city', 'Capital', 'Area (1930) in 1,000skm 2', 'Population (1931) in 1,000s']
Ground Truth	SELECT Area (1930) in 1,000skm 2 FROM table_2_14245_3 WHERE Capital = brześć nad bugiem
LSTM	SELECT Area (1930) in 1,000skm 2 FROM table_2_14245_3 WHERE Voivodeship or city = brześć nad bugiem)
BiLSTM	SELECT Area (1930) in 1,000skm 2 FROM table_2_14245_3 WHERE Capital = brześć, nad)
Encoder-Decoder	SELECT s Car plate (starting 1937) FROM table_2_14245_3 WHERE s Car plate (starting 1937) = brześć nad
Col Attention	SELECT Area (1930) in 1,000skm 2 FROM table_2_14245_3 WHERE Capital = of brześć nad bugiem has
Pointer Network	SELECT s Car plate (starting 1937) FROM table_2_14245_3 WHERE s Car plate (starting 1937) = 1000skm
Question 25	Capital of brześć nad bugiem has what population (1931) in 1,000s?
Table	['s Car plate (starting 1937)', 'Voivodeship or city', 'Capital', 'Area (1930) in 1,000skm 2', 'Population (1931) in 1,000s']
Ground Truth	SELECT Population (1931) in 1,000s FROM table_2_14245_3 WHERE Capital = brześć nad bugiem
LSTM	SELECT s Car plate (starting 1937) FROM table_2_14245_3 WHERE Voivodeship or city = brześć nad bugiem)
BiLSTM	SELECT Population (1931) in 1,000s FROM table_2_14245_3 WHERE Capital = brześć, nad)

Continue on the next page

Table 6: Incorrect WHERE-Clause Condition prediction

Type	Description
Encoder-Decoder	SELECT Population (1931) in 1,000s FROM table_2_14245_3 WHERE s Car plate (starting 1937) = or city brześć nad 1937 1,000skm nad
Col Attention	SELECT s Car plate (starting 1937) FROM table_2_14245_3 WHERE Capital = of brześć nad bugiem has
Pointer Network	SELECT s Car plate (starting 1937) FROM table_2_14245_3 WHERE Capital = brześć
Question 26	What is the number of rounds that took place for the fight that lasted 6:21?
Table	['Res.', 'Record', 'Opponent', 'Method', 'Event', 'Round', 'Time', 'Location']
Ground Truth	SELECT COUNT(Round) FROM table_2_1370208_2 WHERE Time = 6:21
LSTM	SELECT COUNT(Round) FROM table_2_1370208_2 WHERE Time = 6:21
BiLSTM	SELECT COUNT(Round) FROM table_2_1370208_2 WHERE Time = 6:21
Encoder-Decoder	SELECT COUNT(Round) FROM table_2_1370208_2 WHERE Time = 6:21
Col Attention	SELECT COUNT(Round) FROM table_2_1370208_2 WHERE Time = 6:21
Pointer Network	SELECT COUNT(Round) FROM table_2_1370208_2 WHERE Method = lasted 6:21
Question 27	What's the highest frequency of the KBDR callsign?
Table	['Callsign', 'Area served', 'Frequency', 'Band', 'On-air ID', 'Purpose']
Ground Truth	SELECT MAX(Frequency) FROM table_2_1436627_7 WHERE Callsign = kbdr
LSTM	SELECT MAX(Frequency) FROM table_2_1436627_7 WHERE Brand = kbdr callsign
BiLSTM	SELECT MAX(Frequency) FROM table_2_1436627_7 WHERE Brand = kbdr
Encoder-Decoder	SELECT MAX(Frequency) FROM table_2_1436627_7 WHERE Callsign = kbdr callsign
Col Attention	SELECT MAX(Frequency) FROM table_2_1436627_7 WHERE Callsign = kbdr callsign
Pointer Network	SELECT MAX(Frequency) FROM table_2_1436627_7 WHERE Brand = kbdr callsign
Question 28	What is the minimum capacity of Dundee United's stadium?
Table	['Team', 'Stadium', 'Capacity', 'Lowest', 'Highest', 'Average']
Ground Truth	SELECT MIN(Capacity) FROM table_1_11208143_9 WHERE Team = Dundee United

Continue on the next page

Table 6: Incorrect WHERE-Clause Condition prediction

Type	Description
LSTM	SELECT MIN(Capacity) FROM table_1_11208143_9 WHERE Stadium = dundee united
BiLSTM	SELECT Capacity FROM table_1_11208143_9 WHERE Stadium = dundee united stadium
Encoder-Decoder	SELECT MIN(Capacity) FROM table_1_11208143_9 WHERE Team = dundee united
Col Attention	SELECT MIN(Capacity) FROM table_1_11208143_9 WHERE Stadium = dundee united's stadium
Pointer Network	SELECT MIN(Capacity) FROM table_1_11208143_9 WHERE Team = dundee united stadium
Question 29	What is the symbol for Windpower in China?
Table	['Country', 'Renewable energy (RE)', 'Solar power (SP)', 'Wind power (WP)', 'Geothermal energy (GE)', 'Biofuel (BF)', 'Hydro power (Hydro)']
Ground Truth	SELECT Wind power (WP) FROM table_1_11347578_1 WHERE Country = China
LSTM	SELECT Country FROM table_1_11347578_1 WHERE Country = china
BiLSTM	SELECT Biofuel (BF) FROM table_1_11347578_1 WHERE Country = windpower
Encoder-Decoder	SELECT Biofuel (BF) FROM table_1_11347578_1 WHERE Country = windpower
Col Attention	SELECT Country FROM table_1_11347578_1 WHERE Country = china
Pointer Network	SELECT Country FROM table_1_11347578_1 WHERE Country = windpower
Question 30	Which driver won the i race of champions?
Table	['Race Name', 'Circuit', 'Date', 'Winning driver', 'Constructor', 'Report']
Ground Truth	SELECT Winning driver FROM table_1_1140099_6 WHERE Race Name = I Race of Champions
LSTM	SELECT Winning driver FROM table_1_1140099_6 WHERE Constructor = champions AND Race Name = i
BiLSTM	SELECT Winning driver FROM table_1_1140099_6 WHERE Race Name = champions
Encoder-Decoder	SELECT Winning driver FROM table_1_1140099_6 WHERE Race Name = champions

Continue on the next page

Table 6: Incorrect WHERE-Clause Condition prediction

Type	Description
Col Attention	SELECT Winning driver FROM table_1_1140099_6 WHERE Race Name = champions
Pointer Network	SELECT Winning driver FROM table_1_1140099_6 WHERE Race Name = i