An Integrated Search System for Korea Weather Data

Jinkyung Jo[1] **Dayeon Ki**[2] **Soyoung Yoon**[1] **Minjoon Seo**[1]
KAIST[1] Korea University[2]
{jinkyungjo, lovelife, minjoon}@kaist.ac.kr[1] dayeonki@gmail.com[2]

Abstract

We introduce WEATHERSEARCH, an integrated search system deployed at the Korea Meteorological Administration (KMA) [1]. WEATHERSEARCH enables users to retrieve all the relevant data for weather forecasting from a massive weather database with simple natural language queries. We carefully design and conduct multiple expert surveys and interviews for template creation and apply data augmentation techniques including template filling to collect 4 million data points with minimal human labors. We then finetune mT5 (Xue et al., 2021) on the collected dataset and achieve an average MRR of 0.66 and an average Recall of 0.82. We also discuss weather-data-specific characteristics that should be taken into account for creating such a system. We hope our paper serves as a simple and effective guideline for those designing similar systems in other regions of the world.

1 Introduction

Weather forecasting is an important task that involves predicting future weather conditions based on current and past meteorological observations. Accurate weather forecasting not only impacts our daily lives but also plays a crucial role in potentially saving lives and resources during natural disasters. In the case of South Korea, the diversity of weather phenomena (due to its three-sided coastline and approximately 70% of the land consisting of mountainous areas) increases the significance and challenges of weather forecasting.

Meteorological experts rely on two main types of data sources for weather forecasting. The first is the Comprehensive Meteorological Information System (COMIS), which provides access to radar images, cloud images, satellite imagery, and other relevant data. COMIS has a structure similar to a typical website, featuring a hierarchical tree structure with select boxes, drop-down menus, and other interactive elements. Navigating through this structure often requires multiple clicks to arrive at the final target information, and it can be time-consuming, especially when the exact location of the data is unknown. As a result, most experts tend to stick to the pages they are familiar with and rarely explore other pages and this limitation hinders the effective utilization of diverse weather data. The second data source is the Korea Meteorological Administration (KMA) database, which stores past weather observation data such as temperature, humidity, wind speed, and precipitation. The KMA database contains an incredibly extensive volume of data. Its one-minute interval data for 600 unique stations makes 900,000 data points per day, covering roughly 40 years since the 1980s. To retrieve the data from such a massive database, proficiency in using SQL queries is necessary. However, many meteorological experts, particularly newcomers unfamiliar with weather database and senior professionals who struggle with programming languages, face difficulties in using SQL queries. Consequently, they result in spending a significant amount of unnecessary time constructing queries or resort to using only basic SQL queries.

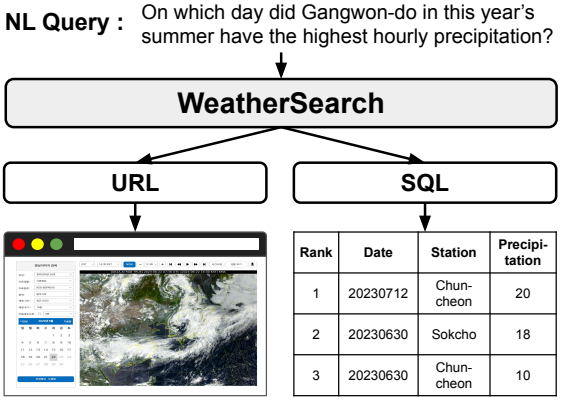

Figure 1: An Overview of WEATHERSEARCH.

[1] https://www.kma.go.kr/neng/index.do

Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing: Industry Track, pages 765–774

In this paper, we introduce WEATHERSEARCH, an integrated search system that allows the users to access to all the necessary weather data from the COMIS and the database through natural language queries. To the best of our knowledge, there are currently no existing search models specifically tailored to the meteorological field and the corresponding training data. Our focus has been on constructing datasets that actively incorporate the opinions of industry experts. Subsequently, using the constructed datasets, we fine-tune a pretrained mT5 model (Xue et al., 2021) to map each natural language query to a structured form.

We construct two domain-specific datasets: (1) a natural language query-SQL query dataset and (2) a keyword-URL dataset. To collect the SQL dataset, we conduct multiple expert surveys and interviews targeting 24 experts to gather responses. Based on these responses, we manually create question templates and corresponding SQL query templates. Subsequently, various techniques, including template filling (Lee et al., 2023), are applied to cover the entire scope of the database, resulting in the final dataset. The URL dataset is collected by crawling all possible URLs from the COMIS and tagging them with corresponding keywords. We preprocess the keywords to make them similar to the actual search keywords by applying useful techniques.

Through the deployment of WEATHERSEARCH to real-world meteorological experts, we anticipate the following contributions:

1. We propose an effective development pipeline for the search system that works with a vast amount of real structured data and incorporates expert opinions in weather domain.

2. Our system enables weather experts to leverage wide range of data during weather forcasting, allowing them to work more efficiently.

Through the disclosure of our methodology, we hope to offer support to those seeking to create similar systems for different regions or languages.

2 Related Work

Machine Learning for Weather Machine learning techniques have been increasingly applied in the meteorological domain. Several notable applications of machine learning in meteorology include weather prediction (Pangu-Weather (Bi et al., 2022), GraphCast (Lam et al., 2022)), extreme weather event detection (ExtremeWeather (Racah et al., 2016), ClimateNet (Kashinath et al., 2021)), climate modeling (MetNet (Sønderby et al., 2020)), and data analysis. However, a single machine learning model capable of efficiently querying vast amounts of databases and websites to quickly access weather data is currently lacking.

Semantic Parsing Semantic parsing is a fundamental task that involves mapping natural language expressions to structured representations. It encompasses various applications, including SQL query generation and code generation. Notable models in SQL query generation include Seq2SQL (Zhong et al., 2018), Spider (Yu et al., 2018), CoSQL (Yu et al., 2019), and UNITE (Lan et al., 2023), which have demonstrated advancements in accurately generating SQL queries from natural language inputs. On the other hand, in the code generation, AlphaCode (Li et al., 2022), Synchromesh (Poesia et al., 2022), and CodeRL (Le et al., 2022) have emerged as prominent approaches, showcasing their ability to transform natural language instructions into executable code. We apply semantic parsing to the generation of SQL queries and URLs (structured representation) in the weather domain.

3 Data Collection

We construct domain-specific datasets necessary to train the search system. As mentioned in §Section 1, weather data comes from two different sources, and each source has its own structured query for accessing data. For this reason, we build separate datasets for each source. One is the SQL dataset, which comprises pairs of natural language queries and corresponding SQL queries (§Section 3.1). The other is the URL dataset, which consists of pairs of natural language keywords and corresponding URLs (§Section 3.2).

3.1 SQL Dataset

SQL data collection method is based on expert surveys and interviews to ensure that the constructed dataset can be applied and closely utilized in the real world. Indeed, gathering all possible natural language queries that experts may use through surveys is inefficient and impractical. Instead, we collect responses through surveys and transform them into templates, which are then filled in accordingly (Lee et al., 2023). Figure 2 illustrates SQL data collection process. The iterative template

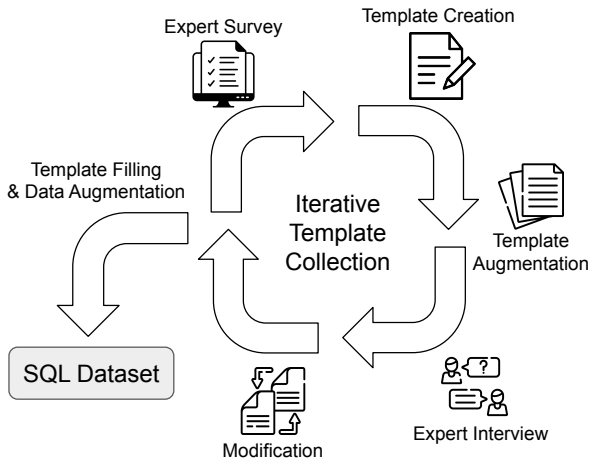

Figure 2: Process of collecting SQL dataset.

collection process is conducted in a total of five stages: (1) expert survey, (2) template creation, (3) template augmentation, (4) expert interview, and (5) modification. It takes approximately two weeks to complete one iteration. We repeat this four times to ensure high quality.

Expert Survey Expert surveys are conducted targeting 24 experts from the Forecasting Department of the Daejeon Regional Meteorological Administration. On average, we obtain approximately 60 responses per survey, and the collected responses comprise weather-related natural language queries (*e.g.* How many days did it snow in the capital area last winter?) commonly used by experts in their practical situation, including variables (*e.g.* snow, capital area, last winter) within the queries. The collected queries encompass a wide range of difficulties, from simple questions that find a single climatic factor, to complex questions (*i.e.* requiring SQL table `JOIN`) that involve multiple regions or multiple climatic factors, and even complex inquiries that necessitate specific conditions to be satisfied. The examples of survey responses can be found in Appendix A.

Template Creation Following the survey, three-type templates are created based on the collected responses: (1) question template, (2) SQL template, and (3) time template. Figure 3 depicts the procedure of making question templates and SQL templates. First, we represent the query templates by using placeholders (*e.g.* `{date}`, `{region}`) for the words that can be replaced with variables from survey responses. The details of range for each variable can be found in §Section 3.1. Once the question templates are completed, corresponding SQL templates are created, which also include variables (*e.g.* `{date_sql}`, `{region_sql}`) corresponding to the variables in the question templates. In the end, we construct a total of 117 question-SQL template pairs.

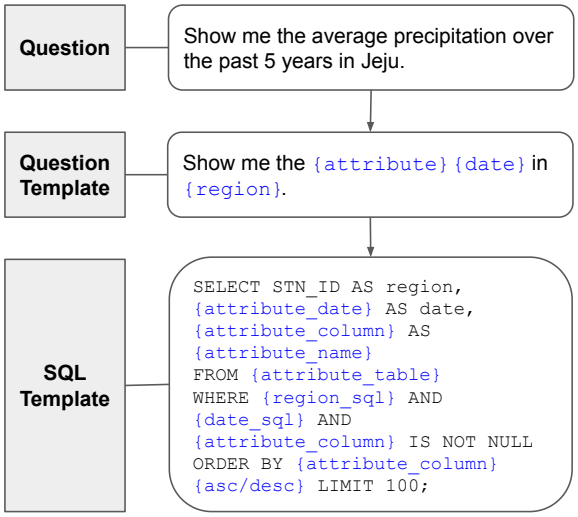

Figure 3: The procedure of making question template and SQL template.

Time expression is crucial in querying weather data as it exhibits significant diversity. For this reason, a separate time template is created to handle time expressions. Time expressions are categorized into daily, monthly, yearly, ordinal, and seasonal representations, considering their combinations. Additionally, colloquial date expressions (*e.g.* last, previous year, yesterday) are also included. As a result, a total of 755 time templates are obtained. All three templates are constructed and annotated manually by the authors of this paper. Appendix B provides the samples of the time templates.

Template Augmentation To accommodate the diversity of language, we employ instruction-tuned language models for template augmentation. As the survey is conducted exclusively with a small number of experts, the data might have biases (*i.e.* being influenced by their specific linguistic tendencies) and potentially limiting the usage of diverse vocabulary. Hence, we leverage language models such as ChatGPT[2] to augment the templates, aiming to encompass vocabulary that would be used by a broader range of individuals.

[2] https://openai.com/blog/chatgpt

Expert Interview & Modification Upon completion of template creation and augmentation, it is essential to undergo a review process involving experts. Six experts from the Forecasting Department of the Daejeon Regional Meteorological Administration are interviewed to review the data and provide feedback. The focus of the review is to examine whether the template content aligns with the actual queries used and if the variable ranges are correctly set. Following the interviews, the templates are modified based on the feedback received. This entire process constitutes one iteration. Subsequently, a new round of surveys is conducted to collect data, create templates, and receive feedback, repeating the iterative process.

Template Filling Once template collection is complete, the next step involves populating the variables within the templates with values to generate actual training data. The key aspect of this process is to fill in values that cover the entire range of the variables in the standardized templates, creating training data that closely resembles the real world and covers all possible questions.

Before proceeding with template filling, we have to explore the variables that need to be filled and their respective ranges. There are five main variables that require population: {date}, {region}, {attribute}, {extreme_expression}, and {value}. Here is a detailed description of each variable:

- {date}: This variable represents the specific date or time period for which the query is being made. It is populated with one of the time templates mentioned earlier in the paper.

- {region}: This variable indicates the geographical area or location of interest for the query. In Korea, there are originally 728 observation stations nationwide. However, we categorize them into a total of 183 regions by grouping them at the provincial level (*e.g.* Gyeongju-si, Cheorwon-gun) and also at the metropolitan area level (*e.g.* Gyeongsangnam-do, Jeollabuk-do).

- {attribute}: This variable pertains to the specific climatic factor or meteorological parameter that is being queried, such as temperature, humidity, precipitation, wind speed, and snowfall amount. There are a total of 30 climatic factors.

- {extreme_expression}: This variable accounts for expressions related to extreme conditions, such as "highest" and "lowest". It covaries with the {attribute} variable, showing a significant influence.

- {value}: This variable represents the actual value or range of values associated with the {attribute} being queried. It is usually a numerical value.

Taking the characteristics and ranges of these variables into consideration, we will now proceed to the template filling stage to actually populate the values.

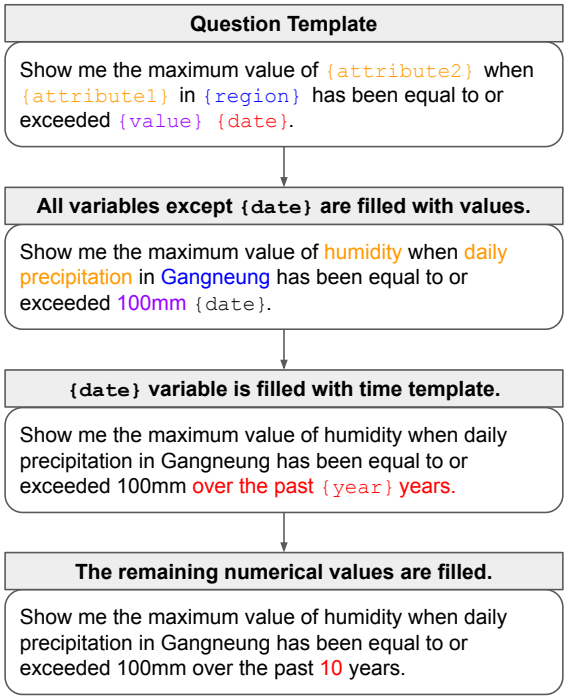

Figure 4: The procedure of template filling.

Template filling begins with the question template. Firstly, random values are assigned to variables other than {date} in the question template. Next, a random time template is selected from the time templates and inserted into the {date} variable. Finally, if there are variables in the selected time template, numerical values that meet the variable conditions are inserted (see Figure 4). In the case of the {value} variable and variables within the time template, it is necessary to ensure that the assigned values fall within the specified ranges. For example, if there is a {month} variable within the time template, it should be restricted to numbers between 1 and 12.

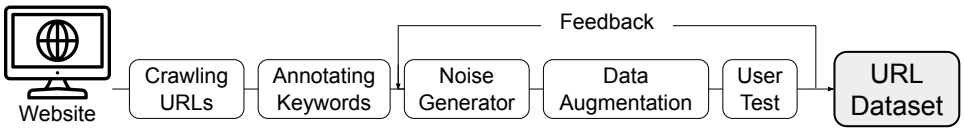

Figure 5: Process of collecting URL dataset.

In order to account for the variation in the amount of data that can be generated through template filling, we apply different weights to each template. These weights are determined based on factors such as the number of variables and the range of values within each template. The amount of data generated from a single template is then proportional to its assigned weight.

Data Augmentation Although language diversity is ensured through template augmentation, data augmentation is also necessary for the values that go into the placeholders. There exist synonyms for meteorological terminology. If a general model does not encounter these words during training, it may not recognize them as synonyms. To address this, we create a synonym dictionary for meteorological terminology and apply it to the training data. We construct synonyms for a total of 197 words, mainly focusing on weather elements and regions (*e.g.* Precipitation = Rainfall = Water accumulation, Gwangju Metropolitan City = Gwangju Jeollado).

3.2 URL Dataset

Similarly to the SQL dataset collection process, the URL dataset also incorporates an extensive amount of expert opinions. However, in the case of the URL dataset, the template collection process is omitted due to the availability of pre-collected keyword-URL pairs. Instead, the data preprocessing for the URL dataset is iteratively refined through expert feedback and modifications, aiming to enhance its quality and relevance (see Figure 5).

We crawl and collect all possible URLs within the COMIS website and annotate the collected URLs with matching keywords. These collected URL-keyword pairs undergo several preprocessing steps based on expert feedback. Step (1), we add noise to the keyword data in the training set to ensure robust performance even with changes in keyword order or partial keyword omissions. Step (2), we apply a synonym dictionary to augment the data, ensuring proper functioning with synonyms of meteorological terminology. We collect synonyms for a total of 196 words (see Appendix C). Step (3), we incorporate new feedback from weather experts (users) for further improvements. Step (4), we go back to Step (1) and repeat the process again. We repeat the process twice, but the number of iterations can be increased for better alignment with the users.

3.3 Data Statistics

In our study, we collect two distinct datasets: URL dataset (keywords-URL pairs) and SQL dataset (NL-SQL pairs). These datasets contain a comprehensive range of weather-related information, showcasing their ability to capture diverse and extensive elements (see Table 1).

URL		**SQL**	
Elements	*# of data*	*Elements*	*# of data*
Radar lightning	854,607	Temperature	746,495
Marine	504,989	Humidity	467,182
AWS	280,810	Rainfall	406,121
Surface	224,543	Wind	274,473
High altitude	69,281	Snow	213,669
Weather bulletin	45,619	Pressure	138,022
Weather forecast	41,393	Cloud	54,496
Yellow dust	24,874	Evaporation	54,105
Weather map	18,001	Radiation	39,879
Satellite	10,075	Fog	16,501
Storm	7,797		
Aviation	1,345		

Table 1: Composition of weather elements in URL and SQL dataset.

We align the sizes of the URL dataset and the SQL dataset, aiming for a balanced distribution of data between the two. The URL dataset contains a total of 2,083,334 records and includes weather imagery and images related to radar lightning, marine conditions, and more. Users can access a variety of climate-related visuals and images through this dataset. The SQL dataset consists of a total of 1,983,800 records and allows for querying observation measurements such as temperature, humidity, and other variables from the database. Due to the ability to perform table `JOIN` in SQL queries (data involving `JOIN` accounts for approximately one-

	SQL	URL							
	EM	*MRR*				*Recall*			
		@5	*@10*	*@20*	*Avg.*	*@5*	*@10*	*@20*	*Avg.*
mT5	0.99	0.63	0.64	0.65	0.64	0.75	0.82	0.86	0.81
mT5 *w/ C.D.*	0.99	0.65	0.66	0.67	0.66	0.76	0.83	0.88	0.82

Table 2: Results of WEATHERSEARCH. *Avg.* represents the average of *@5*, *@10*, and *@20* scores and *w/ C.D.* indicates "with constrained decoding"

third of the entire SQL dataset), it is possible to access multiple elements simultaneously. As a result, the sum of records for individual elements may differ from the total dataset count.

4 Experiment

Model We use mT5 (Xue et al., 2021) as the base model for our WEATHERSEARCH system. mT5 is pre-trained on a massive corpus of multilingual text data, and possesses the capability to encode and decode both Korean and English within its outputs. In addition, to further improve accuracy, we incorporate constrained decoding following De Cao et al. (2021). Constrained decoding, utilizing a prefix tree, involves guiding the generation process during natural language generation tasks by constraining the output based on predefined rules represented in the form of a tree-like data structure. This technique ensures that the generated outputs adhere to specific patterns or formats, improving the quality and coherence of the generated text.

Evaluation Metrics For SQL query generation, we employ the Exact Match (EM) metric. On the other hand, for URL search, there may be multiple possible answers, similarly to the evaluation of canonical web search. For this reason, we use Mean Reciprocal Rank (MRR) and Recall metrics which are widely used in the search engine evaluation.

$$MRR@k = \frac{1}{|Q|}\sum_{i=1}^{|Q|}\frac{1}{rank_i}$$

$$Recall@k = \frac{\text{relevant recommended items}}{\text{all the possible relevant items}}$$

where $|Q|$ is the number of queries, $rank_i$ is the rank of correct answer, and k represents the number of outputs generated from a single query. Both MRR and recall provide valuable insights about the retrieval quality and user experience. MRR emphasizes the ranking quality of the retrieved results, giving more weight to the top-ranked items. Recall, on the other hand, focuses on the system's ability to retrieve all relevant items, ensuring comprehensive coverage.

Results Table 2 shows the experimental results of WEATHERSEARCH on the SQL evaluation dataset and URL evaluation dataset, respectively. the experimental results. In the experiments conducted on the SQL evaluation dataset, both with and without constrained decoding, the system exhibits saturated performance with an exact match score of 0.99. When evaluating on the URL dataset, the experimental results show an average MRR of 0.64 and an average Recall of 0.81. With the addition of constrained decoding, the performance improves to MRR 0.66 and Recall 0.82, respectively. The lower performance compared to the SQL evaluation dataset is expected due to the presence of noise (*i.e.*, some of the input keywords being missing or their order being changed) in the evaluation data. Yet, in search systems, it is more crucial to have a robust functionality even when some parts of the search query are missing. Therefore, it is important to also consider the setting with noisy.

5 Conclusion

We introduces WEATHERSEARCH, a model specifically designed for searching on Korean weather data. To develop the model, we construct two datasets tailored to the weather domain and finetune mT5 on the two datasets. These datasets effectively incorporate the expertise and feedback from the weather experts, making the model directly applicable to real-world scenarios. The experimental results demonstrate sufficiently accurate performance for deployment across various metrics. Future work includes conducting human evaluation for qualitative assessment of the system. WEATHERSEARCH is expected to provide valuable assistance in accessing and utilizing weather data to the weather experts, ultimately improving decision-making process and productivity for weather forecasting.

Limitations

Since we generate datasets based on templates, there may be grammatical errors or inconsistencies as mismatches in prepositions and postpositions. Although this issue can be resolved by manually editing the data later, for now, it is not changed since it does not significantly impact the model's performance during training.

The current evaluation datasets used in the experiments are created in a similar manner to the training datasets, which might have resulted in relatively favorable results. However, we anticipate that there could be a gap between these results and the actual user experience in real-world scenarios. To bridge this gap, we need to collect new evaluation data comprising actual search queries used by users and conduct human (weather expert) evaluations to further refine and validate the system's performance.

Due to security concerns regarding national data, we cannot publicly disclose the original training data. However, in the future, there is a possibility of releasing the data through data masking techniques. By applying data masking, we can enhance the security of the data while preserving its original characteristics, allowing for potential public release while safeguarding sensitive information.

Acknowledgements

We would like to express our sincere gratitude to all those who have contributed to the successful completion of this paper. We would also like to thank the support from National Institute of Meteorological Sciences and Daejeon Regional Meteorological Administration.

References

Kaifeng Bi, Lingxi Xie, Hengheng Zhang, Xin Chen, Xiaotao Gu, and Qi Tian. 2022. Pangu-weather: A 3d high-resolution model for fast and accurate global weather forecast. *ArXiv*, abs/2211.02556.

Nicola De Cao, Gautier Izacard, Sebastian Riedel, and Fabio Petroni. 2021. Autoregressive entity retrieval. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net.

Karthik Kashinath, Mayur Mudigonda, Sol Kim, Lukas Kapp-Schwoerer, Andre Graubner, Ege Karaismailoglu, Leo von Kleist, Thorsten Kurth, Annette Greiner, Ankur Mahesh, Kevin Yang, Colby Lewis, Jiayi Chen, Andrew Lou, Sathyavat Chandran, Benjamin A. Toms, William Chapman, Katherine Dagon, Christine A. Shields, Travis O'brien, Michael F. Wehner, and William D. Collins. 2021. Climatenet: an expert-labeled open dataset and deep learning architecture for enabling high-precision analyses of extreme weather. *Geoscientific Model Development*, 14:107–124.

Rémi R. Lam, Alvaro Sanchez-Gonzalez, Matthew Willson, Peter Wirnsberger, Meire Fortunato, Alexander Pritzel, Suman V. Ravuri, Timo Ewalds, Ferran Alet, Zach Eaton-Rosen, Weihua Hu, Alexander Merose, Stephan Hoyer, George Holland, Jacklynn Stott, Oriol Vinyals, Shakir Mohamed, and Peter W. Battaglia. 2022. Graphcast: Learning skillful medium-range global weather forecasting. *ArXiv*, abs/2212.12794.

Wuwei Lan, Zhiguo Wang, Anuj Chauhan, Henghui Zhu, Alexander Hanbo Li, Jiang Guo, Shenmin Zhang, Chung-Wei Hang, Joseph Lilien, Yiqun Hu, Lin Pan, Mingwen Dong, J. Wang, Jiarong Jiang, Stephen M. Ash, Vittorio Castelli, Patrick Ng, and Bing Xiang. 2023. Unite: A unified benchmark for text-to-sql evaluation. *ArXiv*, abs/2305.16265.

Hung Le, Yue Wang, Akhilesh Deepak Gotmare, Silvio Savarese, and Steven C. H. Hoi. 2022. Coderl: Mastering code generation through pretrained models and deep reinforcement learning. *ArXiv*, abs/2207.01780.

Gyubok Lee, Hyeonji Hwang, Seongsu Bae, Yeonsu Kwon, Won Young Shin, Seongjun Yang, Minjoon Seo, Jong-Yeup Kim, and E. Choi. 2023. Ehrsql: A practical text-to-sql benchmark for electronic health records. *ArXiv*, abs/2301.07695.

Yujia Li, David H. Choi, Junyoung Chung, Nate Kushman, Julian Schrittwieser, Rémi Leblond, Tom, Eccles, James Keeling, Felix Gimeno, Agustin Dal Lago, Thomas Hubert, Peter Choy, Cyprien de, Masson d'Autume, Igor Babuschkin, Xinyun Chen, Po-Sen Huang, Johannes Welbl, Sven Gowal, Alexey, Cherepanov, James Molloy, Daniel Jaymin Mankowitz, Esme Sutherland Robson, Pushmeet Kohli, Nando de, Freitas, Koray Kavukcuoglu, and Oriol Vinyals. 2022. Competition-level code generation with alphacode. *Science*, 378:1092 – 1097.

Gabriel Poesia, Oleksandr Polozov, Vu Le, Ashish Tiwari, Gustavo Soares, Christopher Meek, and Sumit Gulwani. 2022. Synchromesh: Reliable code generation from pre-trained language models. *ArXiv*, abs/2201.11227.

Evan Racah, Christopher Beckham, Tegan Maharaj, Samira Ebrahimi Kahou, Prabhat, and Christopher Joseph Pal. 2016. Extremeweather: A large-scale climate dataset for semi-supervised detection, localization, and understanding of extreme weather events. In *NIPS*.

Casper Kaae Sønderby, Lasse Espeholt, Jonathan Heek, Mostafa Dehghani, Avital Oliver, Tim Salimans,

Shreya Agrawal, Jason Hickey, and Nal Kalchbrenner. 2020. Metnet: A neural weather model for precipitation forecasting. *ArXiv*, abs/2003.12140.

Linting Xue, Noah Constant, Adam Roberts, Mihir Kale, Rami Al-Rfou, Aditya Siddhant, Aditya Barua, and Colin Raffel. 2021. mT5: A massively multilingual pre-trained text-to-text transformer. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 483–498, Online. Association for Computational Linguistics.

Tao Yu, Rui Zhang, Heyang Er, Suyi Li, Eric Xue, Bo Pang, Xi Victoria Lin, Yi Chern Tan, Tianze Shi, Zihan Li, Youxuan Jiang, Michihiro Yasunaga, Sungrok Shim, Tao Chen, Alexander Fabbri, Zifan Li, Luyao Chen, Yuwen Zhang, Shreya Dixit, Vincent Zhang, Caiming Xiong, Richard Socher, Walter Lasecki, and Dragomir Radev. 2019. CoSQL: A conversational text-to-SQL challenge towards cross-domain natural language interfaces to databases. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 1962–1979, Hong Kong, China. Association for Computational Linguistics.

Tao Yu, Rui Zhang, Kai Yang, Michihiro Yasunaga, Dongxu Wang, Zifan Li, James Ma, Irene Li, Qingning Yao, Shanelle Roman, Zilin Zhang, and Dragomir Radev. 2018. Spider: A large-scale human-labeled dataset for complex and cross-domain semantic parsing and text-to-SQL task. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 3911–3921, Brussels, Belgium. Association for Computational Linguistics.

Victor Zhong, Caiming Xiong, and Richard Socher. 2018. Seq2sql: Generating structured queries from natural language using reinforcement learning. *ArXiv*, abs/1709.00103.

A Expert Survey Responses

Natural Language Query	Variable 1	Variable 2	Variable 3	Variable 4
Let me know the area where the rainfall is over 90mm for 3 hours.	rainfall	90mm	3 hours	
Please tell me the average and maximum precipitation over the past 10 years in Daejeon.	average precipitation	maximum precipitation	10 years	Daejeon
Show me the minimum temperature when the snowfall was more than 10mm.	minimum temperature	snowfall	10mm	
Show me the average summer temperature in Gangwondo and Gyeongsangdo.	average temperature	summer	Gangwondo	Gyeongsangdo
How many days did it snow in the capital area last winter?	snow	capital area	last winter	
Please show me the rainfall duration and number of storm days in Chungcheongdo in July.	rainfall duration	storm days	Chung cheong-do	July
Could you provide me with the average wind speed values from September 1st to 10th?	average wind speed	September 1st to 10th		
Please provide the relative humidity of Boryeong and Inje during the mid of three months ago.	relative humidity	Boryeong	Inje	the mid of three months

Table 3: Examples of responses collected from expert surveys. The columns named "Variable" represents words that can be used as variables in the natural language query.

B Time Template

Type	Time expression
None	since observation, all time
Day	from the {day1}th to the {day2}th, in the top ten days of a month
Month	in {month}, in the summer, from {month1} to {month2}
Year	over the last {year} years, last year, the {year}s, before {year}
Year + Month	last {month}, {year} springs ago, in {month} from {year1} to {year2}
Month + Day	on {month} {day}th, in the last {day} days of last month
Year + Month + Day	today, yesterday, the past {day} days, last month {day}th

Table 4: Samples of time templates.

C Synonyms

Terminology	Synonyms
short-term forecast	6-hour forecast, real-time weather forecast
MOS	correction model, BEST model, statistical prediction model
wind rose	wind distribution, wind direction distribution
GK-2A	COMS, weather satellite, geostationary satellites, korean satellite
IR	infrared thermography, infrared satellite, infrared channel
WISSDOM	radar synthetic aperture, radar wind synthesis
PM10	yellow dust observation data, asian dust observation data

Table 5: Samples of meteorological synonyms.

D Experimental Setup

We trained our model using eight A100 GPUs with a total memory capacity of 80GB. The maximum token length was set to 256, and we used a batch size of 16. The training process consisted of five epochs, and we employed a learning rate of 1e-5, which linearly decreased during training. The evaluation step followed the same conditions as the training process.