A Extension to Complex SQL

Table 8 shows the extended lexicon entries and grammar rules in NLG for applying our MISP-SQL agent to generate more complex SQL queries, such as those on Spider (Yu et al., 2018c). In this dataset, a SQL query can associate with multiple tables. Therefore, we name a column by combining the column name with its table name (i.e., "*col*" in table "*tab*" $\rightarrow$ COL[*col* (table *tab*)]). For simplicity, we omit "(table *tab*)" when referring to a column *col* in the grammar.

B Simulation Evaluation Results

The complete simulation experiment results of MISP-SQL agents (based on SQLNet and SQLova) are shown in Table 6 & 7.

C Error Detector Comparison

As a supplementary experiment to Figure 4, in this section, we show the performance of different error detectors under the same average number of questions ("*target budget*"). Specifically, for each base semantic parser and each kind of error detector, we tune its decision threshold (i.e., p^* and s^*) such that the resulting average number of questions ("*actual budget*") is as close to the target as possible. In practice, we relax the actual budget to be within ± 0.015 of the target budget, which empirically leads to merely negligible variance. The results are shown in Table 9-10 for SQLNet and Table 11-12 for SQLova.

	SQLNet		
System	**Acc_{qm}**	**Acc_{ex}**	**Avg. #q**
no interaction	0.615	0.681	N/A
MISP-SQL$^{\text{Unlimit10}}$	0.932	0.948	7.445
MISP-SQL$^{\text{Unlimit3}}$	0.870	0.900	7.052
MISP-SQL$^{p^*=0.95}$	0.782	0.824	1.713
MISP-SQL$^{p^*=0.8}$	0.729	0.779	1.104
MISP-SQL$^{p^*=0.5}$	0.661	0.722	0.421
MISP-SQL$^{s^*=0.01}$	0.796	0.845	2.106
MISP-SQL$^{s^*=0.05}$	0.725	0.786	1.348
MISP-SQL$^{s^*=0.1}$	0.695	0.758	1.009
MISP-SQL$^{s^*=0.2}$	0.650	0.714	0.413

Table 6: Simulation evaluation of MISP-SQL (based on SQLNet) on WikiSQL Test set.

	SQLova		
System	**Acc_{qm}**	**Acc_{ex}**	**Avg. #q**
no interaction	0.797	0.853	N/A
MISP-SQL$^{\text{Unlimit10}}$	0.985	0.991	6.591
MISP-SQL$^{\text{Unlimit3}}$	0.955	0.974	6.515
MISP-SQL$^{p^*=0.95}$	0.912	0.939	0.773
MISP-SQL$^{p^*=0.8}$	0.880	0.914	0.488
MISP-SQL$^{p^*=0.5}$	0.835	0.879	0.209
MISP-SQL$^{s^*=0.01}$	0.913	0.942	0.893
MISP-SQL$^{s^*=0.03}$	0.866	0.912	0.515
MISP-SQL$^{s^*=0.05}$	0.840	0.892	0.333
MISP-SQL$^{s^*=0.07}$	0.825	0.880	0.216

Table 7: Simulation evaluation of MISP-SQL (based on SQLova) on WikiSQL Test set.

	[Lexicon]
	is greater than (or equivalent to)|equals to|is less than (or equivalent to)|does not equal to → OP[> (=) |= |< (=) |! =]
	is IN|is NOT IN|follows a pattern like|is between → OP[in|not in|like|between]
	sum of values in|average value in|number of|minimum value in|maximum value in → AGG[sum|avg|count|min|max]
	in descending order (and limited to top N)|in ascending order (and limited to top N) → ORDER[desc (limit N) |asc (limit N)]
	[Grammar]
(R1)	"col" in table "tab" → COL[col (table tab)]
(R2)	Does the system need to return information about COL[col] ? → Q[col‖SELECT $agg?$ col]
(R3)	Does the system need to return AGG[agg] COL[col] ? → Q[agg‖SELECT agg col]
(R4)	Does the system need to return a value after any mathematical calculations on COL[col] ? → Q[agg=None‖SELECT agg col]
(R5)	Does the system need to consider any conditions about COL[col] ? → Q[col‖WHERE col op val]
(R6)	The system considers the following condition: COL[col] OP[op] a given literal value. Is this condition correct? → Q[$terminal$‖WHERE col op $terminal$]
(R7)	The system considers the following condition: COL[col] OP[op] a value to be calculated. Is this condition correct? → Q[$root$‖WHERE col op $root$]
(R8)	Do the conditions about COL[col_i] and COL[col_j] hold at the same time? → Q[AND‖WHERE col_i .. AND col_j ..]
(R9)	Do the conditions about COL[col_i] and COL[col_j] hold alternatively? → Q[OR‖WHERE col_i .. OR col_j ..]
(R10)	Does the system need to group items in table tab based on COL[col] before doing any mathematical calculations? → Q[col‖GROUP BY col]
(R11)	Given that the system groups items in table tab^g based on COL[col^g] before doing any mathematical calculations, does the system need to consider any conditions about COL[col] ? → Q[col‖GROUP BY col^g HAVING $agg?$ col]
(R12)	Given that the system groups items in table tab^g based on COL[col^g] before doing any mathematical calculations, does the system need to consider any conditions about AGG[agg] COL[col] ? → Q[agg‖GROUP BY col^g HAVING agg col]
(R13)	Given that the system groups items in table tab^g based on COL[col^g] before doing any mathematical calculations, does the system need to consider a value after any mathematical calculations on COL[col] ? → Q[agg=None‖GROUP BY col^g HAVING agg col]
(R14)	The system groups items in table tab^g based on COL[col^g] before doing any mathematical calculations, then considers the following condition: COL[col] OP[op] a value. Is this condition correct? → Q[op‖GROUP BY col^g HAVING $agg?$ col op val]
(R15)	Given that the system groups items in table tab^g based on COL[col^g] before doing any mathematical calculations, does it need to consider any conditions? → Q[NONE_HAVING ‖GROUP BY col^g NONE_HAVING]
(R16)	Does the system need to order results based on COL[col] ? → Q[col‖ORDER BY $agg?$ col]
(R17)	Does the system need to order results based on AGG[agg] COL[col] ? → Q[agg‖ORDER BY agg col]
(R18)	Does the system need to order results based on a value after any mathematical calculations on COL[col] ? → Q[agg=None‖ORDER BY agg col]
(R19)	Given that the system orders the results based on (AGG[agg]) COL[col], does it need to be ORDER[od] ? → Q[od‖ORDER BY $agg?$ col od]

Table 8: Extended lexicon and grammar for MISP-SQL NLG module to handle complex SQL on Spider.

Avg. #q	Probability-based		Dropout-based	
	Acc_{qm}	Acc_{ex}	Acc_{qm}	Acc_{ex}
0.5	0.672	0.732	0.663	0.726
1.0	0.725	0.775	0.706	0.765
1.5	0.778	0.820	0.749	0.809
2.0	0.812	0.848	0.796	0.845

Table 9: Comparison of error detectors for SQLNet with a target average number of questions on WikiSQL Dev set.

Avg. #q	Probability-based		Dropout-based	
	Acc_{qm}	Acc_{ex}	Acc_{qm}	Acc_{ex}
0.5	0.669	0.729	0.656	0.720
1.0	0.722	0.773	0.695	0.758
1.5	0.765	0.810	0.740	0.801
2.0	0.805	0.844	0.790	0.842

Table 10: Comparison of error detectors for SQLNet with a target average number of questions on WikiSQL Test set.

Avg. #q	Probability-based		Dropout-based	
	Acc_{qm}	Acc_{ex}	Acc_{qm}	Acc_{ex}
0.2	0.844	0.885	0.829	0.881
0.4	0.876	0.910	0.856	0.905
0.6	0.902	0.932	0.887	0.927
0.8	0.921	0.947	0.913	0.941

Table 11: Comparison of error detectors for SQLova with a target average number of questions on WikiSQL Dev set.

Avg. #q	Probability-based		Dropout-based	
	Acc_{qm}	Acc_{ex}	Acc_{qm}	Acc_{ex}
0.2	0.832	0.877	0.823	0.878
0.4	0.865	0.902	0.851	0.901
0.6	0.895	0.926	0.881	0.922
0.8	0.915	0.941	0.904	0.936

Table 12: Comparison of error detectors for SQLova with a target average number of questions on WikiSQL Test set.