

A Re-implementation

We perform a simple modification to Bogin et al. (2019). We add *cell values* to the graph, in a similar fashion to Krishnamurthy et al. (2017). Specifically, we extract the cells of the first 5000 rows of all tables in the schema, during the pre-processing phase. We then consider every cell q of a column c , which has a partial match with any of the question words $(x_1, \dots, x_{|x|})$. We then add nodes representing these cells to all of the model’s graphs, with extra edges (c, q) and (q, c) .

B Selected examples

Selected examples are given in Table 3.

Category	Question	Schema	Predicted Queries
Coverage	Show the name of the teacher for the math course.	course: course_id, starting_date, course teacher: teacher_id, name, age, hometown course_arrange: course_id, teacher_id, grade	Baseline: SELECT teacher.name FROM teacher WHERE teacher.name = 'math' Our Model: SELECT teacher.name FROM teacher JOIN course_arrange ON teacher.teacher_id = course_arrange.teacher_id JOIN course ON course_arrange.course_id = course.course_id WHERE course.course = 'math'
Coverage	Who is the first student to register? List the first name, middle name and last name.	students: student_id, current_address_id, first_name, middle_name, last_name, ... student_enrolment: student_enrolment_id, degree_program_id, ...	Baseline: SELECT students.first_name, students.middle_name, students.last_name FROM students Our Model: SELECT students.first_name, students.middle_name, students.last_name FROM students ORDER BY students.date_first_registered LIMIT 1
Coverage	List all singer names in concerts in year 2014.	singer: singer_id, name, country, song_name, song_release_year, ... concert: concert_id, concert_name, year, ... singer_in_concert: concert_id, singer_id	Baseline: SELECT singer.name FROM singer WHERE singer.song_release_year = 2014 Our Model: SELECT singer.name FROM singer JOIN singer_in_concert ON singer.singer_id = singer_in_concert.singer_id JOIN concert ON singer_in_concert.concert_id = concert.concert_id WHERE concert.year = 2014
Precision	What are the makers and models?	car_makers: id, maker, fullname, country model_list: modelid, maker, model	Baseline: SELECT car_makers.maker, model_list.model FROM car_makers JOIN model_list ON car_makers.id = model_list.maker Our Model: SELECT model_list.maker, model_list.model FROM model_list
Precision	Return the id of the document with the fewest paragraphs.	documents: document_id, document_name, document.description, ... paragraphs: paragraph_id, document_id, paragraph_text, ...	Baseline: SELECT documents.document_id FROM documents JOIN paragraphs ON documents.document_id = paragraphs.document_id GROUP BY documents.document_id ORDER BY COUNT(*) LIMIT 1 Our Model: SELECT paragraphs.document_id FROM paragraphs GROUP BY paragraphs.document_id ORDER BY COUNT(*) ASC LIMIT 1

Table 3: Selected correct examples where the baseline model is wrong, but our parser is correct.