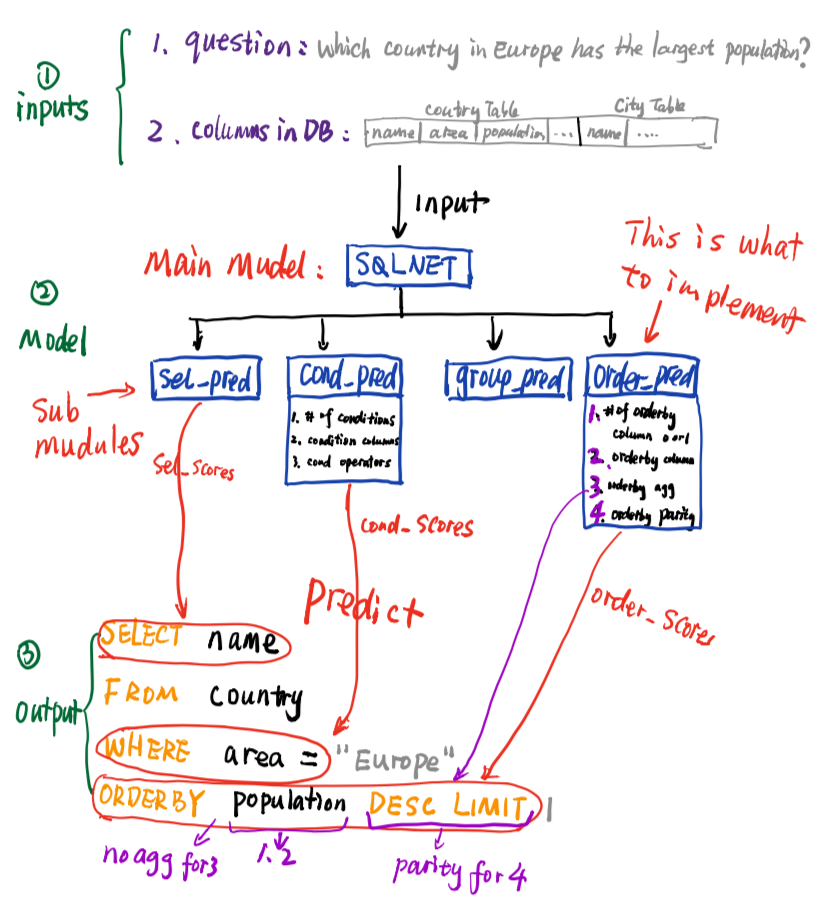
**More Instructions for Assignment 4**

**1. The Big Picture of SQLNet**

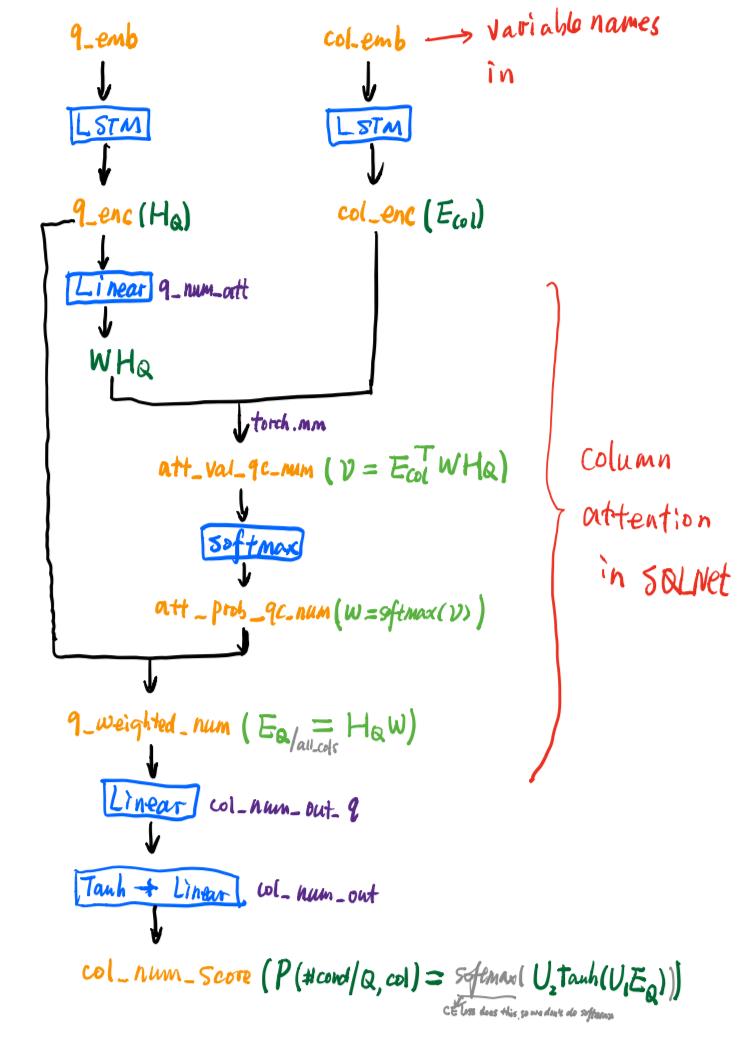
In this assignment, we extend SQLNet to the Spider task including more complex SELECT, GROUP BY and ORDER BY sub-modules. What you should do is to understand what is going on for WHERE condition module (the ‘CondPredictor’ class in the code). And then you should follow WHERE condition module and code comments to implement ORDER BY sub-module (the ‘OrderPredictor’ class in the code).

The picture below shows the big picture of SQLNet. The inputs for the SQLNet are questions and columns in the corresponding database. SQLNet is the main module containing 4 seperate sub-modules that predict 4 different components in SQL queries. Each sub-module has several steps to predict different parts in the SQL component. For example, to predict WHERE condition in the SQL, condition module ‘cond\_pred’ (‘‘CondPredictor’’) first predicts the number of conditions, then the actual condition columns, and finally condition operators.

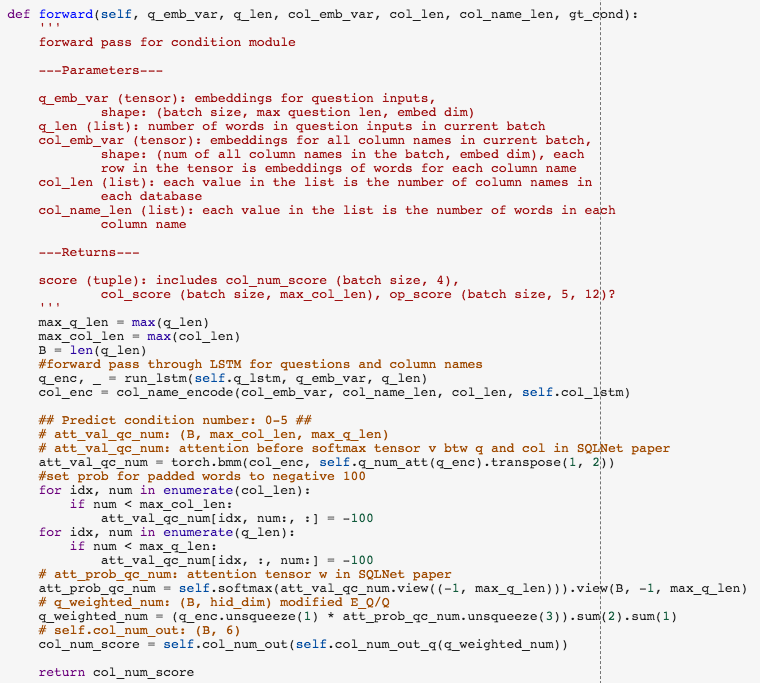


**2. Understand Code in CondPredictor and OrderPredictor**

The following picture illustrates how WHERE condition sub-module computes the probability of the number of conditions (**output: col\_num\_score**) from question and column embeddings as **inputs**. Names in yellow are the same as variable names in the code of the `CondPredictor` class. Blue blocks represent different layers, and green symbols refer to the math notations in the SQLNet paper. **This helps you understand the code you are going to write in `OrderPredictor`.** The corresponding code snippet is shown right after the picture.



The corresponding code snippet of the picture above is shown below:



**3. Parts You Should Focus on**

**We added more detailed instructions in what you should do in the** [**Google Colab Notebook of the homework**](https://colab.research.google.com/drive/1RTzq36agSECCY770Gfa4JyOzXZpbk9qu)**. Please check out this version (updated on 04/10/2019) to help you focus on understanding the parts you need to fill in.**

1. You don't need to understand the code in the functions "process", "load\_dataset", "load\_word\_emb", "WordEmbedding", "run\_lstm".

2. You should look at "CondPredictor". To finish "OrderPredictor", you should follow exactly what condition number prediction does for orderby column number, orderby aggregation, and orderby parity prediction, also condition column prediction for orderby column prediction.

3. For the SQLNet class, you don't need to go over all of the code. What you need to do is to follow exactly the loss and accuracy for condition prediction to fill in orderby's loss and accuracy.

4. You can go very quickly through "to\_batch\_seq", "epoch\_train", and "epoch\_acc". There is no code for you to complete there.

5. For part 2, you only need to understand "get\_fields" and "reconstruct\_input", and "epoch\_train" functions in detail. The other functions are just for your reference.

**4. Reference Result Prints**

Below are result prints for part 2 and 3 for your reference. Your results are not supposed to be exactly the same as ours because of randomness in weight Initialization and dropout regularization, optimization etc,. However, your loss in part 2 should starts from ~6-7 and keeps going downing to ~2 after 20 epoches. Also, the best accuracy for ORDER BY in 30 epoches is ~ 86. For part 3, loss goes to ~0.08 from ~1.6, and the best accuracy for WHERE condition is ~54.

**For part 2:**

Epoch 1 @ 2019-04-10 07:34:09.240756

Loss = 7.305574634007045

Train acc\_qm: 0.01657142857142857

Breakdown results: sel: 0.062, cond: 0.4238571428571429, group: 0.7527142857142857, order: 0.7601428571428571

Dev acc\_qm: 0.031914893617021274

Breakdown results: sel: 0.10154738878143134, cond: 0.40522243713733075, group: 0.7398452611218569, order: 0.7485493230174082

Saving sel model...

Saving cond model...

Saving group model...

Saving order model...

Best val sel = 0.10154738878143134, cond = 0.40522243713733075, group = 0.7398452611218569, order = 0.7485493230174082, tot = 0.031914893617021274

Epoch 2 @ 2019-04-10 07:36:45.530488

Loss = 5.3860344870431085

Train acc\_qm: 0.010571428571428572

Breakdown results: sel: 0.09357142857142857, cond: 0.5107142857142857, group: 0.6344285714285715, order: 0.8058571428571428

Dev acc\_qm: 0.010638297872340425

Breakdown results: sel: 0.11411992263056092, cond: 0.5154738878143134, group: 0.6228239845261122, order: 0.8104448742746615

Saving sel model...

Saving cond model...

Saving order model...

Best val sel = 0.11411992263056092, cond = 0.5154738878143134, group = 0.7398452611218569, order = 0.8104448742746615, tot = 0.031914893617021274

Epoch 3 @ 2019-04-10 07:39:19.802871

Loss = 4.5505832579476495

Train acc\_qm: 0.04071428571428572

Breakdown results: sel: 0.1382857142857143, cond: 0.43542857142857144, group: 0.7127142857142857, order: 0.8175714285714286

Dev acc\_qm: 0.043520309477756286

Breakdown results: sel: 0.17311411992263057, cond: 0.4206963249516441, group: 0.6924564796905223, order: 0.8249516441005803

Saving sel model...

Saving order model...

Best val sel = 0.17311411992263057, cond = 0.5154738878143134, group = 0.7398452611218569, order = 0.8249516441005803, tot = 0.043520309477756286

Epoch 4 @ 2019-04-10 07:41:54.597956

Loss = 4.086432349886213

Train acc\_qm: 0.04585714285714286

Breakdown results: sel: 0.17185714285714285, cond: 0.5084285714285715, group: 0.7334285714285714, order: 0.802

Dev acc\_qm: 0.05705996131528046

Breakdown results: sel: 0.2369439071566731, cond: 0.5222437137330754, group: 0.7127659574468085, order: 0.7862669245647969

Saving sel model...

Saving cond model...

Best val sel = 0.2369439071566731, cond = 0.5222437137330754, group = 0.7398452611218569, order = 0.8249516441005803, tot = 0.05705996131528046

Epoch 5 @ 2019-04-10 07:44:44.104452

Loss = 3.846842564174107

Train acc\_qm: 0.047857142857142855

Breakdown results: sel: 0.18357142857142858, cond: 0.5102857142857142, group: 0.7232857142857143, order: 0.8204285714285714

Dev acc\_qm: 0.0609284332688588

Breakdown results: sel: 0.2504835589941973, cond: 0.5183752417794971, group: 0.6914893617021277, order: 0.8172147001934236

Saving sel model...

Best val sel = 0.2504835589941973, cond = 0.5222437137330754, group = 0.7398452611218569, order = 0.8249516441005803, tot = 0.0609284332688588

.

.

.

Epoch 22 @ 2019-04-10 08:41:46.102584

Loss = 2.3163977988106863

Train acc\_qm: 0.07357142857142857

Breakdown results: sel: 0.22071428571428572, cond: 0.5577142857142857, group: 0.7628571428571429, order: 0.8777142857142857

Dev acc\_qm: 0.07930367504835589

Breakdown results: sel: 0.2640232108317215, cond: 0.5193423597678917, group: 0.7195357833655706, order: 0.8588007736943907

Best val sel = 0.2736943907156673, cond = 0.539651837524178, group = 0.7398452611218569, order = 0.8597678916827853, tot = 0.086073500967118

Epoch 23 @ 2019-04-10 08:44:25.682845

Loss = 2.3217497435978482

Train acc\_qm: 0.07

Breakdown results: sel: 0.20985714285714285, cond: 0.552, group: 0.7658571428571429, order: 0.8804285714285714

Dev acc\_qm: 0.07930367504835589

Breakdown results: sel: 0.2620889748549323, cond: 0.5067698259187621, group: 0.7166344294003868, order: 0.8617021276595744

Saving order model...

Best val sel = 0.2736943907156673, cond = 0.539651837524178, group = 0.7398452611218569, order = 0.8617021276595744, tot = 0.086073500967118

Epoch 24 @ 2019-04-10 08:47:03.792844

Loss = 2.3159656451089043

Train acc\_qm: 0.07585714285714286

Breakdown results: sel: 0.2257142857142857, cond: 0.5548571428571428, group: 0.7687142857142857, order: 0.878

Dev acc\_qm: 0.07833655705996131

Breakdown results: sel: 0.2736943907156673, cond: 0.5154738878143134, group: 0.7330754352030948, order: 0.8597678916827853

Best val sel = 0.2736943907156673, cond = 0.539651837524178, group = 0.7398452611218569, order = 0.8617021276595744, tot = 0.086073500967118

**For part 3:**

Epoch 1 @ 2019-04-10 07:46:28.601473

Loss = 1.5659863141450014

Train acc\_qm: 0.44585714285714284

Breakdown results: sel: 1.0, cond: 0.44585714285714284, group: 1.0, order: 1.0

Dev acc\_qm: 0.45164410058027077

Breakdown results: sel: 1.0, cond: 0.45164410058027077, group: 1.0, order: 1.0

Saving cond model...

Best val sel = 1.0, cond = 0.45164410058027077, group = 1.0, order = 1.0, tot = 0.45164410058027077

Epoch 2 @ 2019-04-10 07:47:29.293561

Loss = 1.180575482411818

Train acc\_qm: 0.4674285714285714

Breakdown results: sel: 1.0, cond: 0.4674285714285714, group: 1.0, order: 1.0

Dev acc\_qm: 0.45551257253384914

Breakdown results: sel: 1.0, cond: 0.45551257253384914, group: 1.0, order: 1.0

Saving cond model...

Best val sel = 1.0, cond = 0.45551257253384914, group = 1.0, order = 1.0, tot = 0.45551257253384914

Epoch 3 @ 2019-04-10 07:48:29.248706

Loss = 0.9335813262245872

Train acc\_qm: 0.4897142857142857

Breakdown results: sel: 1.0, cond: 0.4897142857142857, group: 1.0, order: 1.0

Dev acc\_qm: 0.4835589941972921

Breakdown results: sel: 1.0, cond: 0.4835589941972921, group: 1.0, order: 1.0

Saving cond model...

Best val sel = 1.0, cond = 0.4835589941972921, group = 1.0, order = 1.0, tot = 0.4835589941972921

Epoch 4 @ 2019-04-10 07:49:31.588205

Loss = 0.7802212715148926

Train acc\_qm: 0.5194285714285715

Breakdown results: sel: 1.0, cond: 0.5194285714285715, group: 1.0, order: 1.0

Dev acc\_qm: 0.5125725338491296

Breakdown results: sel: 1.0, cond: 0.5125725338491296, group: 1.0, order: 1.0

Saving cond model...

Best val sel = 1.0, cond = 0.5125725338491296, group = 1.0, order = 1.0, tot = 0.5125725338491296

Epoch 5 @ 2019-04-10 07:50:33.710432

Loss = 0.6939695119857788

Train acc\_qm: 0.5164285714285715

Breakdown results: sel: 1.0, cond: 0.5164285714285715, group: 1.0, order: 1.0

Dev acc\_qm: 0.488394584139265

Breakdown results: sel: 1.0, cond: 0.488394584139265, group: 1.0, order: 1.0

Best val sel = 1.0, cond = 0.5125725338491296, group = 1.0, order = 1.0, tot = 0.5125725338491296

.

.

.

Epoch 28 @ 2019-04-10 08:14:21.836785

Loss = 0.1147824809632518

Train acc\_qm: 0.6025714285714285

Breakdown results: sel: 1.0, cond: 0.6025714285714285, group: 1.0, order: 1.0

Dev acc\_qm: 0.5212765957446809

Breakdown results: sel: 1.0, cond: 0.5212765957446809, group: 1.0, order: 1.0

Best val sel = 1.0, cond = 0.5425531914893617, group = 1.0, order = 1.0, tot = 0.5425531914893617

Epoch 29 @ 2019-04-10 08:15:24.271671

Loss = 0.09201703071594239

Train acc\_qm: 0.6054285714285714

Breakdown results: sel: 1.0, cond: 0.6054285714285714, group: 1.0, order: 1.0

Dev acc\_qm: 0.52321083172147

Breakdown results: sel: 1.0, cond: 0.52321083172147, group: 1.0, order: 1.0

Best val sel = 1.0, cond = 0.5425531914893617, group = 1.0, order = 1.0, tot = 0.5425531914893617

Epoch 30 @ 2019-04-10 08:16:27.785008

Loss = 0.08492797696962953

Train acc\_qm: 0.6037142857142858

Breakdown results: sel: 1.0, cond: 0.6037142857142858, group: 1.0, order: 1.0

Dev acc\_qm: 0.5241779497098646

Breakdown results: sel: 1.0, cond: 0.5241779497098646, group: 1.0, order: 1.0

Best val sel = 1.0, cond = 0.5425531914893617, group = 1.0, order = 1.0, tot = 0.5425531914893617