ディープラーニングを用いた SQL の自動生成に関する研究

松　山　　摩　偉　児
津山高専　　電子・情報システム工学専攻

1．は　じ　め　に

関係データベース(RDB)は，表形式のデータ集合を互いに関連付け，その関係を利用したデータベース(DB)のことであり，大量の情報が保管できる．しかし，RDB を利用するにはプログラム初心者には習得が困難な SQL を理解する必要がある．さらに近年，システムエンジニアが不足しており，何度も同じような SQL を記述しなければならないことで，エンジニアの負担も大きくなっている．

これを解決する研究として，Zhong らの Seq2SQL[1] と呼ばれる手法がある．彼らの研究では，大きく二つの貢献をしている．

一つは，自然言語の問合せ文，SQL クエリ，および Wikipedia の 24241 個の HTML テーブルから抽出した SQL テーブルの 80654 個の注釈付きコーパスをまとめた WikiSQL と呼ばれる DB を提供したこと．

もう一つは，自然言語を対応する SQL クエリに翻訳するための DNN である Seq2SQL を提案したことである．しかし，これは SQL の[“MAX”, “MIN”, “COUNT”, “SUM”, “AVG”]のような集約関数の生成に対する再現率が低く，正しく予測されない可能性がある．

本研究は，Seq2SQL の集約関数の生成に注目し，自然言語の問合せ文と集約関数の依存関係を表す補助情報を加えることで再現率の向上と SQL 生成全体の精度の向上を図る．また，Zhong らの手法と提案手法を用いた実験を実施し，結果の比較分析を行う．ただし，データセットは彼らが提供した WikiSQL を用いる．

2．Seq2SQL

Seq2SQL の概要図を図 1 に示す．Seq2SQL では自然言語の問合せ文とテーブルのカラム名を入力することで SQL 文が生成される．

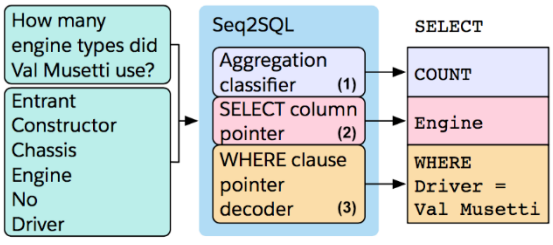

図 1　Seq2SQL モデルの概要図

Seq2SQL は三つの機能により構成される: (1) 集約関数を生成する部分, (2) SELECT のカラムを選択する部分, (3) WHERE の条件文を生成する部分である．このモデルはまず, SQL クエリの集約関数を生成する．ただし，集約関数を必要としない場合に対応する NULL の操作も追加する．次に, SELECT のカラムに対応する入力テーブルの列を指定する．最後にポインタを使用して, SQL クエリの条件文を生成する．つまり, Seq2SQL は 3 つの学習ネットワークを有している．

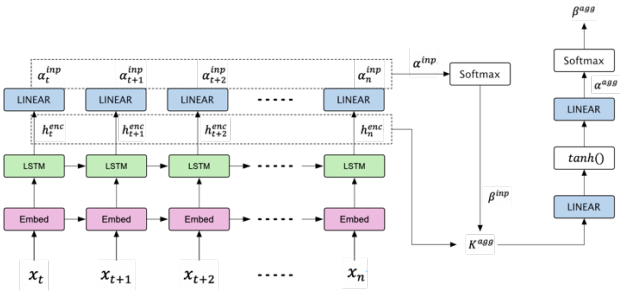

図 2　集約関数の生成ネットワーク

本研究は集約関数の生成に注目するため，図 2 に集約関数の生成ネットワークを示す．図2の入力xは全てのカラム名, SQL トークン，問合せ文の連結で定義される:

$$x = [< col >; x_1^c; x_2^c; \ldots; x_N^c; < sql >; x^s; < Qs >; x^q] \quad (1)$$

ここで，$[a; b]$はaとbの連結を示している．まず，ネットワークは入力xを埋込み層でベクトルに変換する．その後, LSTM [2]を用いてエンコードされる．エンコーダーの出力はh^{enc}，ステップ t のときの単語x_tに対応した出力はh_t^{enc}で表される．さらに，h_t^{enc}をそれぞれ線形層に通すことで，入力配列に対する注目度α_t^{inp}を計算することができる．例えば，問合せ文に "How many"という語が含まれていると正しい集約関数は"COUNT" の可能性が高い．注目度は，この LSTM の出力に注目し重みを掛け，計算する:

$$\alpha_t^{inp} = W^{inp} h_t^{enc} \quad (2)$$

計算されたα_t^{inp}を Softmax 関数によって正規化しβ^{inp}を計算する．正規化した注目度β^{inp}によって重み付けされたエンコーダーの出力h^{enc}の合計をK^{agg}とする:

$$K^{agg} = \sum_t \beta_t^{inp} h_t^{enc} \quad (3)$$

K^{agg}は注目度が高い特徴量が保持され，注目度が低い特徴量は破棄される．つまり，集約関数に関する特

徴を持つ値のみが保持される．もし，集約関数を持たない場合はこの値も小さくなる．この値を入力とした多層パーセプトロンを用いることで，どの集約関数であるかを表すスコアα^{agg}を計算する:

$$\alpha^{agg} = W^{agg}\tanh(V^{agg}K^{agg} + b^{agg}) + c^{agg} \quad (4)$$

α^{agg}を Softmax 関数によって正規化すると，集約関数の確率分布β^{agg}が得られ，集約関数のうち確率β^{agg}が高いものが予測結果となる．この予測結果の値と正解データの値との差分をクロスエントロピー損失L^{agg}として計算し，この値が小さくなるように学習する．

また，Seq2SQL 全体の損失関数は，3 つのネットワークのクロスエントロピー損失の合計値である．

3．提　案　手　法

提案手法では，Seq2SQL の集約関数の生成の再現率の低さに注目し，WikiSQL にある自然言語の問合せ文と集約関数の依存関係を表す補助情報を加えることで，モデルの集約関数の生成を改善し，最終的な SQL 生成全体の精度向上を目指す．まず，データセットからすべての問合せ文を単語に分割し，単語トークンX，集約関数カテゴリYを定義する:

$$X = \{"the","?","what",\dots\} = \{x_1, x_2, \dots, x_k\} \quad (5)$$

$$Y = \{"NULL","MAX","MIN","COUNT","SUM","AVG"\} = \{y_1, y_2, \dots, y_6\} \quad (6)$$

データセットの正解クエリを見れば，問合せ文がどの集約関数カテゴリに属しているのかがわかる．そのため，カテゴリ別単語出現頻度からXとYの周辺および同時確率を計算することができる．これを元に，二つの確率変数の相互依存の尺度を相互情報量$I(X,Y)$によって計算する．集約関数ごとに単語の相互情報量を求め，入力データの形式に合わせた次元tにおける合計I^{agg}を計算しこれを補助情報として図 2 の式(4)の学習パラメータに辞書として加えることで，分類精度を向上させる:

$$\alpha^{agg} = W^{agg}\tanh(V^{agg}K^{agg} + V^{agg}I^{agg} + b^{agg}) + c^{agg} \quad (7)$$

4．実　　験

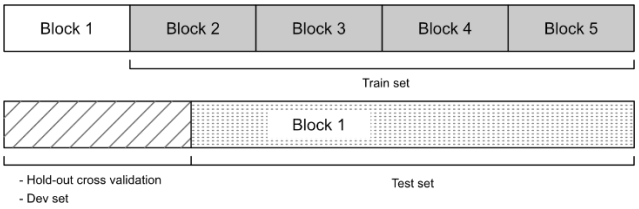

図 3 に乱数的に 5 つに分割した WikiSQL のデータセットを示す．WikiSQL はテーブル，問合せ文，SQL クエリで構成され，活用する際には学習用，開発用，テスト用のデータに分割する必要がある．よって，5 分割データのうち 1 ブロックを開発用，テスト用とし，残りを学習用とする．また，同様の操作を 5 回繰返し実施し，全ての組合せで実験を行う．ただし，開発用とテスト用のデータ比率はランダムで 33.3%, 66.7%である．

4.1 Zhong らの手法と提案手法による実験

Zhong らが提案した Seq2SQL と提案手法の実験結果の平均値を表 1 に示す．ただし，Acc_{lf}は正解クエリと予測クエリが完全一致している場合の論理精度を，Acc_{ex}はクエリ実行結果が正解クエリと一致した場合の実行精度を表している．

表 1　Seq2SQL と提案手法の実験結果の平均値

Model	Dev Acc_{lf}	Dev Acc_{ex}	Test Acc_{lf}	Test Acc_{ex}
Seq2SQL(Zhong et al.)	53.41%	61.70%	52.53%	60.85%
Seq2SQL(with mutual information)	**53.94%**	**62.03%**	**52.75%**	**61.05%**

各平均値を見ると，論理精度，実行精度のいずれも Zhong らの手法よりわずかに向上していることがわかる．すなわち，問合せ文と集約関数の依存関係を表す補助情報を加えたことによって，学習過程に影響が出ていると考えられる．

ここで，表 2 にそれぞれの手法が有する 3 つのネットワークに関する生成精度の平均値を示す．

表 2　3 つのネットワークに関する生成精度の平均値

Model	Dev agg	Dev sel	Dev where	Test agg	Test sel	Test where
Seq2SQL(Zhong et al.)	90.58%	90.24%	62.55%	**90.17%**	**89.82%**	62.03%
Seq2SQL(with mutual information)	**90.59%**	**90.45%**	**63.20%**	90.10%	89.72%	**62.49%**

表 7 を見ると，WHERE の条件文生成の精度が 0.5%ほど向上している．この結果，モデル全体の精度が向上したと考えられる．

提案手法では，集約関数ネットワークに対して補助情報を加え，その生成精度向上を図ったが，これによるクロスエントロピー損失L^{agg}の変化から，WHERE の条件文生成のネットワークにまで影響が伝搬し，このような結果になったと推察される．

5．ま　と　め

本研究は，Seq2SQL の集約関数の生成に対する再現率の低さに注目し，自然言語の問合せ文と集約関数の依存関係を表す補助情報を集約関数の生成ネットワークに加えることで，その再現率の向上と SQL 生成全体の精度を向上させるという新たな学習モデルを提案した．また，本研究のベースとなった Zhong らの手法と我々が提案した手法による比較実験を実施した．

実験の結果，Zhong らの手法より提案手法のほうが優れていることを示したが，その要因は WHERE の条件文生成の精度向上によるものであった．

文　　献

1) Zhong, Victor and Xiong, Caiming and Socher, Richard, Seq2sql: Generating structured queries from natural language using reinforcement learning, arXiv preprint arXiv:1709.00103 (2017).

2) Hochreiter, Sepp, and Jürgen Schmidhuber. "Long short-term memory." Neural computation 9.8 (1997): 1735-1780.