自然言語の問い合わせをSQLに変換する系の論文を読む

はじめに

SQLってどんなもの？

SQLはデータベースの定義や操作を行う言語。
下記の3つに大別できる。
・データを定義するためのDDL(データ定義言語)
・データを制御するためのDCL(データ制御言語)
・データを操作するための**DML(データ操作言語)**

DDL・DCLについては割愛。
DMLは下記の4つに大別できる。

SELECT：データベースを検索する
INSERT：データを挿入する
DELETE：データを削除する
UPDATE：データを更新する

Natural language to SQLの分野ではDMLのSELECTについて扱われている論文が多かった。

調べたいこと

「炎タイプ の ポケモン って 何匹いるの？」

↓ 予測

```
SELECT COUNT(id)
FROM pokemon_table
WHERE type='Fire';
```

を実現している手法。

目標

関連する分野の論文を5本サクッと読んで、
簡潔にまとめる。(落合先生のフォーマット？を用いる)
足りない知識はなるべく論文ベースで補完する。

SyntaxSQLNet: Syntax Tree Networks for Complex and Cross-Domain Text-to-SQL Task

https://arxiv.org/abs/1810.05237

(*Submitted on 11 Oct 2018 (v1), last revised 25 Oct 2018*)

Tao Yu, Michihiro Yasunaga, Kai Yang, Rui Zhang, Dongxu Wang, Zifan Li, Dragomir Radev

どんなもの？

複雑かつ分野を超えたテキストから、複数の句・サブクエリを含むような複雑なSQLを生成する手法。
（選定理由 -> 比較的新しかったから

どうやって有効だと検証した？

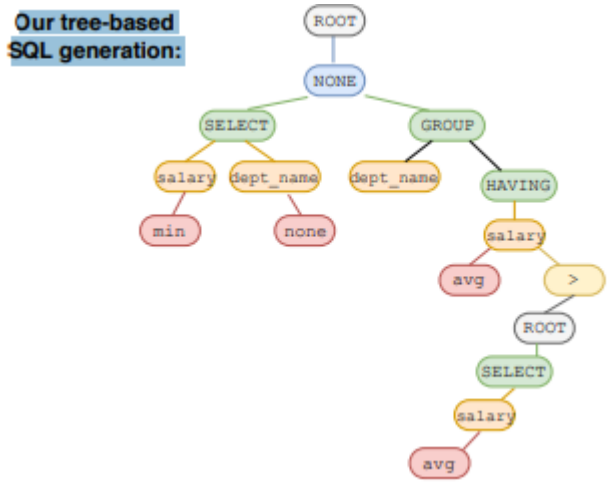
predictとground truthを「SELECT, WHERE, GROUP BY , ORDER BY, KEYWORDS」に分解したのち、スコアリング(F1)を行い評価している。

先行研究と比べて何がすごい？

複雑なSQL文に対応しつつ、SQLNetやTypeSQLなどの過去のモデルよりも15%程度F1値が上昇している点。

技術の手法や肝は？

Syntaxをツリー構造化してSQLを生成する。



各ノードで9つに細分化されたモジュールによりデコードし、その履歴を保持し次ノードのインプットとするのも特徴。

次に読むべき論文は？

そもそもモジュールの詳細設計を読み解くのに、先行研究（seq2SQL）や使用してるモデル（LSTM）などの知識を前提としてそうなのでその辺を読む

Seq2SQL: Generating Structured Queries from Natural Language using Reinforcement Learning

https://arxiv.org/abs/1709.00103

(*Submitted on 31 Aug 2017 (v1), last revised 9 Nov 2017*)

Victor Zhong, Caiming Xiong, Richard Socher

どんなもの？

自然言語の質問をSQLクエリに変換する手法。

先行研究と比べて何がすごい？ どうやって有効だと検証した？

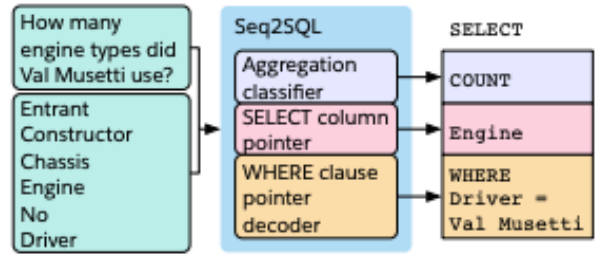
ニューラルネットワークを使わない過去のモデルとのスコアリング比較を行った。結果は下記。

Model	Precision	Recall	F1
Aug Ptr Network	66.3%	64.4%	65.4%
Seq2SQL	72.6%	66.2%	69.2%

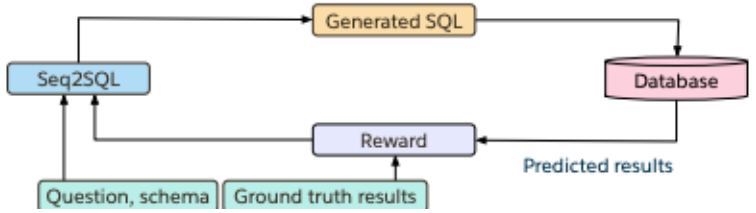
いずれの値も向上している。

技術の手法や肝は？

集計演算子、カラム、WHERE句の三つのコンポーネントで構成されている。集計演算子、カラムはクロスエントロピー誤差+softmaxを用いて分類。Where句は方策勾配法で強化学習（←よくわからない）



クエリの実行結果は強化学習アルゴリズムの報酬となる。



次に読むべき論文は？

seq2SQLより新しい手法の論文

SQLNet: Generating Structured Queries From Natural Language Without Reinforcement Learning

https://arxiv.org/abs/1711.04436

(*Submitted on 13 Nov 2017*)

Xiaojun Xu, Chang Liu, Dawn Song

どんなもの？

seq2SQLをベースに強化学習を使わないで、より高精度なSQLを自然言語から作成する手法。

先行研究と比べて何がすごい？どうやって有効だと検証した？

seq2SQLとのスコアリング比較を行った。結果は下記。

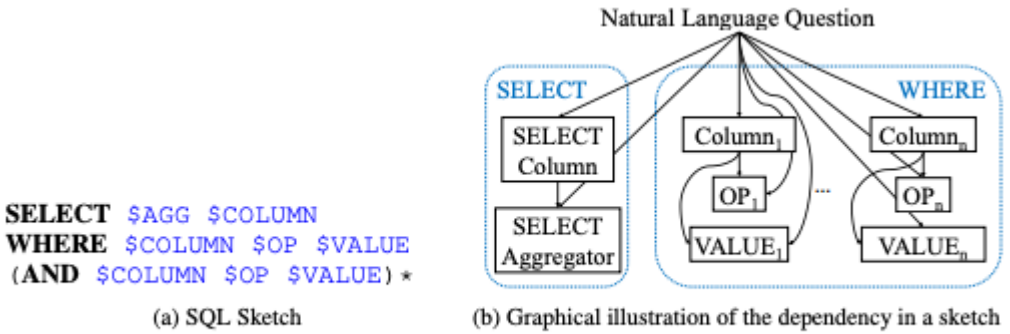
	dev			test		
	Acc_{lf}	Acc_{qm}	Acc_{ex}	Acc_{lf}	Acc_{qm}	Acc_{ex}
Seq2SQL (Zhong et al. (2017))	49.5%	-	60.8%	48.3%	-	59.4%
Seq2SQL (ours)	52.5%	53.5%	62.1%	50.8%	51.6%	60.4%
SQLNet	-	**63.2%**	**69.8%**	-	**61.3%**	**68.0%**

Table 1: Overall result on the WikiSQL task. Acc_{lf}, Acc_{qm}, and Acc_{ex} indicate the logical form, query-match and the execution accuracy respectively.

いずれの正確さも向上している。

技術の手法や肝は？

強化学習を使わずに下記スケッチをベースにSQL文を合成しているところ。((a) * indicates zero or more and clauses.)



(a) SQL Sketch (b) Graphical illustration of the dependency in a sketch

Figure 2: Sketch syntax and the dependency in a sketch

TypeSQL: Knowledge-based Type-Aware Neural Text-to-SQL Generation

https://arxiv.org/abs/1804.09769

(*Submitted on 25 Apr 2018*)

Tao Yu, Zifan Li, Zilin Zhang, Rui Zhang, Dragomir Radev

どんなもの？

SQLNetをベースに下記のようなスケッチに入るsql（$AGG~など）をスロットに見立てて充填していくように処理していく手法。

SELECT $AGG $SELECT_COL
WHERE $COND_COL $OP $COND_VAL
(**AND** $COND_COL $OP $COND_VAL)*

先行研究と比べて何がすごい？どうやって有効だと検証した？

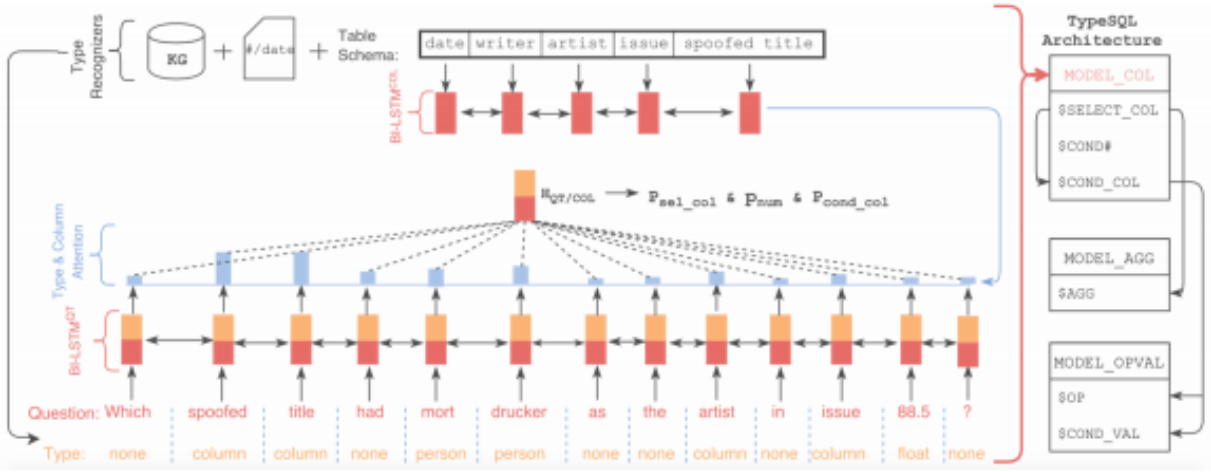
SQLNetとのスコアリング比較を行った結果は下記。

	Dev			Test		
	Acc_{agg}	Acc_{sel}	Acc_{where}	Acc_{agg}	Acc_{sel}	Acc_{where}
Seq2SQL (Zhong et al., 2017)	90.0%	89.6%	62.1%	90.1%	88.9%	60.2%
SQLNet (Xu et al., 2017)	90.1%	91.5%	74.1%	90.3%	90.9%	71.9%
TypeSQL (ours)	90.3%	**93.1%**	**78.5%**	90.5%	**92.2%**	**77.8%**
TypeSQL+TC (ours)	90.3%	**93.5%**	**92.8%**	90.5%	**92.1%**	**87.9%**

Select・where句の正確さが向上している。

技術の手法や肝は？

質問をタイプ別に分類したのちに、3つのコンポーネントごとに分けてスロットを充填していくように処理するところ。



Recent Trends in Deep Learning Based Natural Language Processing

https://arxiv.org/abs/1708.02709

(*Submitted on 9 Aug 2017 (v1), last revised 25 Nov 2018*)

Tom Young, Devamanyu Hazarika, Soujanya Poria, Erik Cambria

どんなもの？

自然言語に用いられるディープラーニングベースのモデル(RNN)とその変遷をまとめた論文。

どうやって有効だと検証した？

NLP(natural language processing)の領域において、モデル(LSTMなど)を適用することでF1値などが向上した。

先行研究と比べて何がすごい？

文脈に依存する自然言語処理において、順次情報処理を行うCNN等のモデルは時系列的にデータを扱えないという弱点があった。
RNN：時系列データを各層の入力に利用する作りにすることで、文脈を捉えることができるようになった。
LSTM：深いニューラルネットで発生する勾配消失(爆発)問題を解決している。

技術の手法や肝は？

前の時系列のデータの重みが次のデータに共有される。

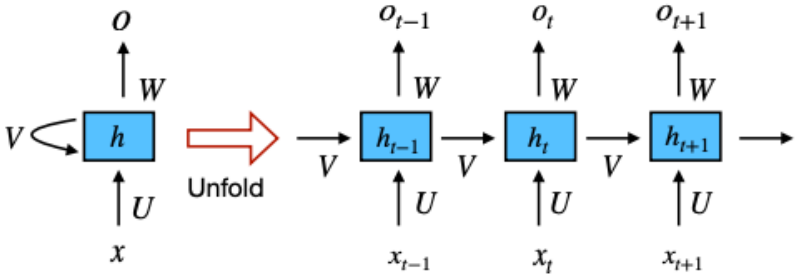


Fig. 9: Simple RNN network (Figure Source: LeCun et al. [90])

またLSTMでは、勾配が消失(爆発)しないように忘却ゲートというものを設けている。