

目次

1. 論文紹介
2. 実験1 (Geoquery)
3. 考察
4. 実験2 (Spider)
5. 追加実験
6. 考察
7. 感想・まとめ

Semantic Parsing とは

自然言語をなんらかの機械可読表現に変換するタスク

Geoquery:

- how many state border s0
- (count \$0 (and (state:t \$0) (next_to:t \$0 s0)))

Spider:

- Show the status shared by cities with population bigger than 1500 and smaller than 500.
- SELECT Status FROM city WHERE Population > 1500 INTERSECT
SELECT Status FROM city WHERE Population < 500

論文

Dong, Li, and Mirella Lapata. "Language to logical form with neural attention." arXiv preprint arXiv:1601.01280 (2016).

ベース:

- Encoder、Decoderモデル(LSTM)

工夫1:

- DecoderをTree Decoderにする

工夫2:

- Attention構造を入れる

工夫3:

- Argument Identification
 - (jobs with a salary of 40000 -> jobs with a salary of num0)

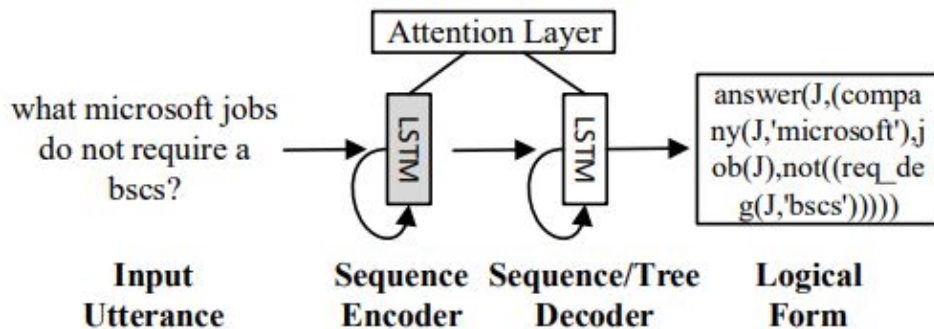


Figure 1: Input utterances and their logical forms are encoded and decoded with neural networks. An attention layer is used to learn soft alignments.

再現実装・実験

実装内容:

- Seq2Seqモデル(LSTM)
 - Attention機構あり
- 推論時はビームサーチ
- ハイパーパラメータはほぼ論文通り
 - バッチサイズ20、Epoch数90、dropout 0.5

データセット:

- Geoquery
 - train: 600データ
 - test: 280データ
- 環境: csc

結果 (Geoquery)

再現実装:

- Attentionなし: 57.5 %
- Attentionあり: **77.5 %**

論文:

- Attentionなし: 72.9 %
- Attentionあり: 84.6 %

考察 (Geoquery)

分析:

- “(“の対応関係(構造)がおかしい: 2/63のみ
- 惜しいもの: 20/63
 - 意味が近いtokenを混同:
 - capital/state, density/population
 - シンボルを混同
 - r0 / s0
 - 全体の意味はほぼ同じ
 - exp: (count \$0 (and (capital:t \$0) (loc:t \$0 s0)))
 - output: (count \$0 (and (major:t \$0) (city:t \$0) (loc:t \$0 s0)))

結論:

- Tree Decoderがなくても、構造を学習できる。
- vocab数が少なく簡単すぎる。

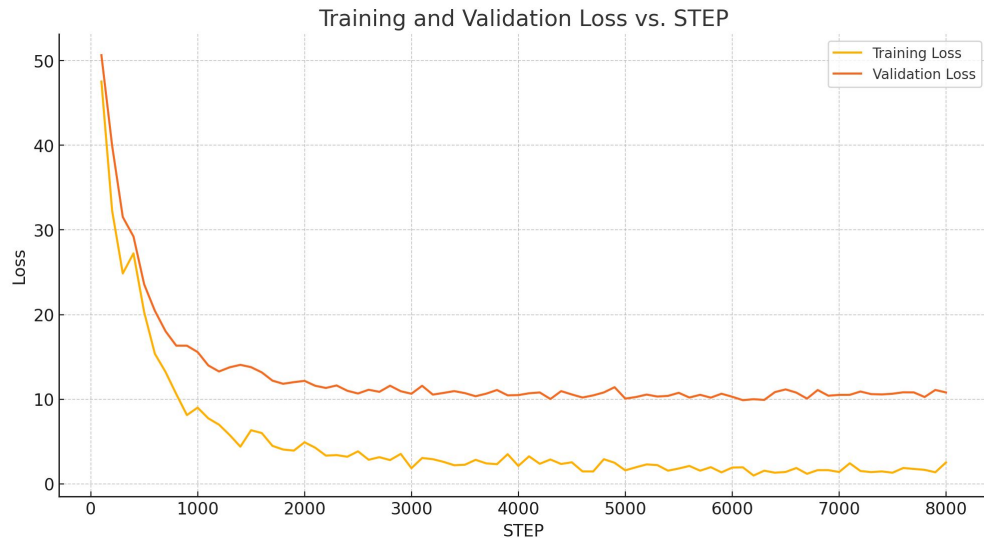
Spider Datasetで実験

Spiderとは:

- Text to SQLのデータセット
- db: 166件、scheme: 計876件
- 7000件のデータを分割
 - train: 4800 (バッチサイズ48)
 - validation: 100
 - test: 2100

モデルをそのまま適用した結果:

- 完全一致: **43%**
- SQLの構造一致: 65%



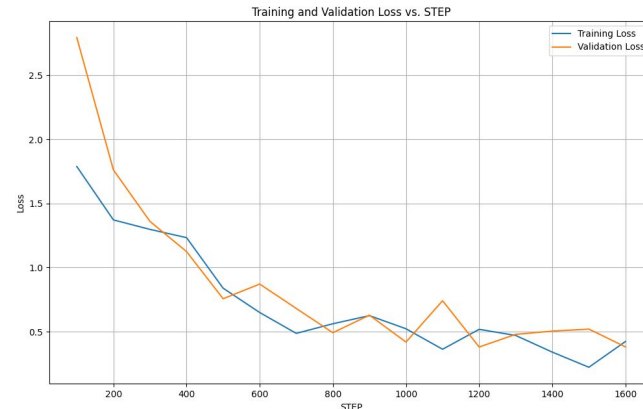
※構造一致:

- select _ from _ group by _ order by _ desc limit _
- select _ from _ where _ intersect select _ from _ where _

追加実験

① SQL構造だけdecodeする

- join as on あり: 45.4 %
- join as on なし: **72.2 %**
- joinが必要かどうかはScheme情報も必要なので難しい。
- Scheme情報が不要な構造に絞れば、高精度を実現できる。

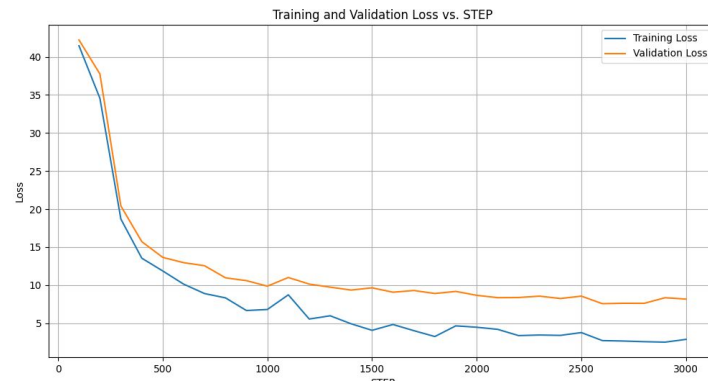


② Resdsql (SpiderでのSOTA)の工夫取り入れ

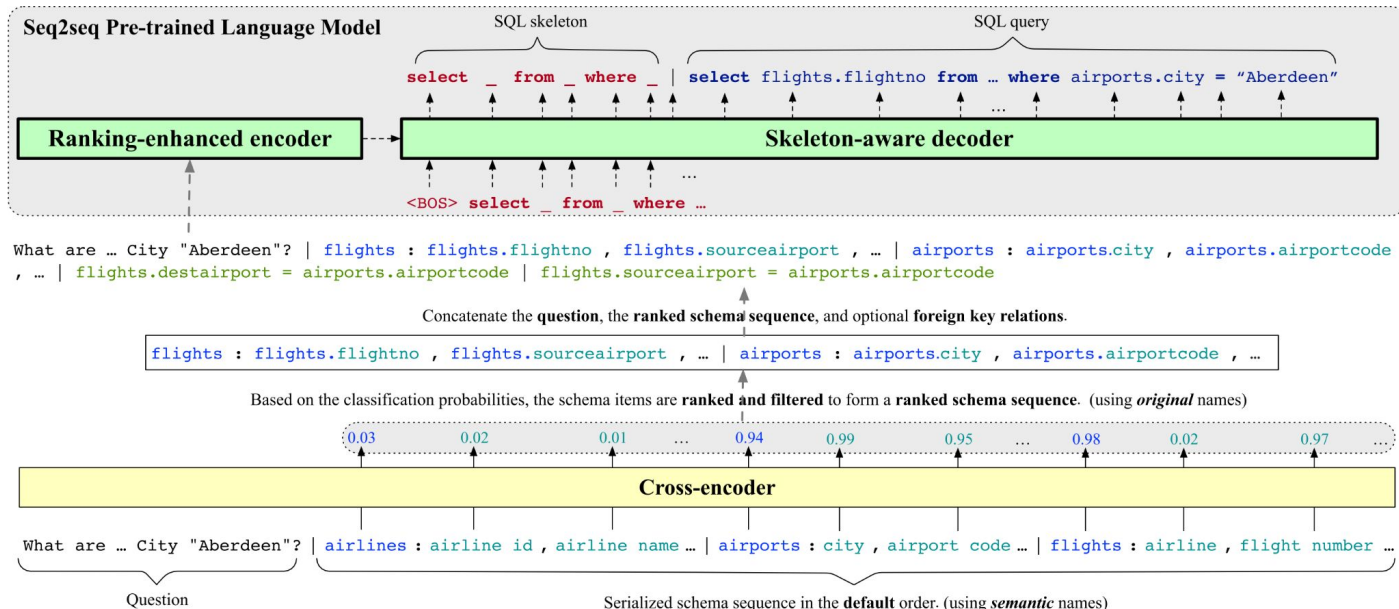
1. encoderにScheme情報も入れる
2. decoderでSQL構造の後にSQLを出力させる

結果: **13.2 %**

- table nameすらかなり間違えている。
- encoderにいれたScheme情報が長すぎた?
 - より関連するSchemeに絞る必要がある?



備考 (Resdsql)



Li, Haoyang, et al. "Resdsql: Decoupling schema linking and skeleton parsing for text-to-sql." *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol. 37. No. 11. 2023.

考察 (Spider+全体)

- SQL構造だけなら高精度
 - Geographyもvocab数が少なく、ほぼ構造出力だけだった
- Scheme情報の考慮が必要だと訓練データ数が少ない
 - 1dbあたり約30件の訓練データしかない

改善案:

- pretrainedモデルを導入し、queryからSchemeへの変換をサポート
 - ほとんどの論文でpretrainedモデルを使っている
- SQL構造にSchemeを当てはめる問題に変換する
 - テーブル名 (joinするか)の推論、スキーマの推論、条件の推論 ...
 - 解く方法は？