Avaliação da aplicabilidade do SQLNet na conversão de consultas em linguagem natural para consultas SQL a uma Base Integrada Corporativa

Henrique Bueno Rodrigues (henriquebueno@id.uff.br)

Universidade Federal Fluminense

Instituto de Computação - Pós-Graduação

**Resumo**

Pendente

**1. Introdução**

Apesar do surgimento de diversas tecnologias de bancos de dados não convencionais, por exemplo, NoSQL [] e Hadoop [], sistemas de corporações pequenas, médias e grandes ainda possuem enorme acoplamento e dependência de bancos de dados relacionais, tais como Oracle [] e SQL Server [].

Em paralelo a isso, surge uma demanda enorme de Democratização da Informação [] por parte de usuários corporativos que desejam explorar seus dados além das limitadas interfaces de sistemas e que não conhecem a linguagem SQL.

Nesse contexto, o problema conhecido como Natural-Language-to-SQL (NL2SQL), que é a tarefa de converter uma consulta em linguagem natural para SQL, tem sua relevância ampliada e recebe maior atenção de pesquisadores. Um exemplo disso é o ranking WikiSQL [] que tem recebido com uma frequência alta submissões de trabalhos.

Um dos artigos que evoluiu os resultados sobre o dataset WikiSQL é o SQLNet [], que propôs a geração de consultas SQL a partir de consultas em linguagem natural sem o uso de aprendizado por reforço.

A proposta desse trabalho é avaliar a aplicação da solução apresentada em SQLNet na geração de consultas SQL a partir de consultas em linguagem natural sobre uma Base de Dados Integrada Corporativa (BDIC). Seguem algumas características dessa base:

* Possui 59 tabelas;
* A tabela com maior quantidade de colunas tem 53 colunas;
* A tabela com menor quantidade de colunas tem 2 colunas;
* A média de colunas por tabela é 12.

As próximas seções detalharão as atividades que foram desempenhadas para alcançar o objetivo apresentado nos parágrafos anteriores. A seção 2 apresenta uma avaliação do WikiSQL. A seção 3 apresenta uma avaliação do SQLNet. A seção 4 apresenta as atividades desempenhadas para a criação de um dataset de consultas para BDIC. A seção 5 avaliou uma rede neural de tradução de textos em português para inglês. A seção 6 apresenta a conclusão e ideias de trabalhos futuros.

**2. WikiSQL**

O trabalho [WikiSQL] apresenta SEQ2SQL, uma rede neural profunda que tem o objetivo de traduzir perguntas em linguagem natural para consultas SQL usando aprendizado por reforço.

Uma outra contribuição desse trabalho é o *dataset* WikiSQL que tem 80654 registros com pares de perguntas em linguagem natural e consultas em SQL. Esse dataset foi anotado manualmente através de *crowdsourcing* utilizando a infraestrutura *Amazon Mechanical Turk* [turk].

Diversos pesquisadores se interessaram em avaliar o desempenho de seus modelos no WikiSQL. A Tabela 1 [GithubWikiSQL] apresenta uma lista de modelos e seus resultados sobre o *dataset* em questão.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Model** | **Acurácia da forma lógica na base de desenvolvimento** | **Acurácia da execução na base de desenvolvimento** | **Acurácia da forma lógica na base de teste** | **Acurácia da execução na base de teste** |
|
| [**Coarse2Fine (Dong 2018)**](https://arxiv.org/abs/1805.04793) | 72.5 | 79.0 | 71.7 | 78.5 |
|
| [**TypeSQL (Yu 2018)**](https://arxiv.org/abs/1804.09769) | - | 74.5 | - | 73.5 |
|
| [**PT-MAML (Huang 2018)**](https://arxiv.org/abs/1803.02400) | 63.1 | 68.3 | 62.8 | 68.0 |
|
| [**SQLNet (Xu 2017)**](https://arxiv.org/abs/1711.04436) | - | 69.8 | - | 68.0 |
|
| [**Wang 2017**](https://www.microsoft.com/en-us/research/publication/pointing-sql-queries-text/) | 62.0 | 67.1 | 61.5 | 66.8 |
|
| **Seq2SQL (Zhong 2017)** | 49.5 | 60.8 | 48.3 | 59.4 |
|
| [**Baseline (Zhong 2017)**](https://arxiv.org/abs/1709.00103) | 23.3 | 37.0 | 23.4 | 35.9 |
|

Tabela 1. Lista de modelos e seus resultados para o *dataset* WikiSQL

Foram considerados dois pontos na avaliação dos modelos: (a) forma lógica do SQL gerado (colunas 2 e 4) e (b) resultado da execução da query SQL gerada (colunas 3 e 5).

O modelo “Wang 2017” utilizou o conteúdo das tabelas para o treinamento do modelo. Esse é um ponto de atenção uma vez que pode violar restrições de sigilosidade das informações armazenadas nas bases de dados. Outro problema no uso do conteúdo das tabelas é a escalabilidade, uma vez que tabelas muito grandes podem aumentar consideravelmente o tempo da etapa de treinamento.

O modelo SEQ2SQL, que foi apresentado em conjunto com o *dataset* WikiSQL, gerou os primeiros resultados do ranking. Em pouco tempo diversos trabalhos evoluíram os resultados e atualmente o modelo “Coarse2Fine” apresenta a melhor acurácia de execução para a base de teste (71.7%).

Um ponto de destaque do WikiSQL é que as tabelas citadas no dataset são aleatoriamente distribuídas nas bases de treinamento, desenvolvimento e teste, de tal forma que cada tabela só aparece em uma única base (treinamento, desenvolvimento ou teste). Essa característica representa um desafio considerável para os modelos, pois demanda que os modelos seja capazes de generalizar a capacidade de gerar SQLs, independente dos *schemas* dos modelos.

**2.1. Avaliação do WikiSQL**

Para avaliar o WikiSQL foram realizadas as seguintes etapas:

* Estrutura do WikiSQL: o dataset foi disponibilizado nos formatos jsonl e db (SQLite3). A Figura 1 apresenta um exemplo de registro do *dataset* WikiSQL.

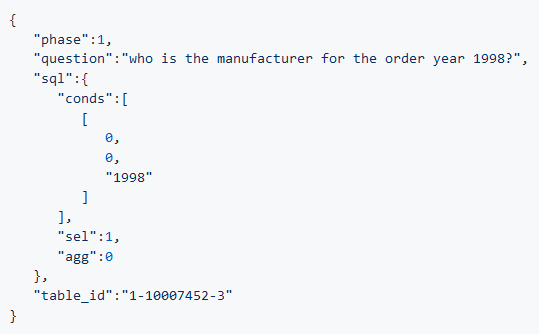


Figura 1. Exemplo de um registro do WikiSQL

O campo *phase* representa uma informação interna do processo de geração do conjunto de dados. O campo *question* representa a pergunta em linguagem natural. O campo *sql* representa a consulta em *sql*. O campo *sql* é dividido nos campos *conds*, *sel* e *agg*. O campo *conds* tem 3 valores, onde o primeiro indica o índice da coluna da restrição, o segundo indica o índice do operador e o terceiro indica o valor a ser comparado com o operando. O campo *sel* indica o índice da coluna que será selecionado. O campo *agg* representa o índice do operador de agregação, por exemplo, *sum* e *avg*. Por fim, o campo *table\_id* identifica a tabela que será consultada.

* Execução do SEQ2SQL: A rede neural SEQ2SQL é um modelo inicial para a avaliação do *dataset*. Esse modelo foi implementado usando Python 3 e a biblioteca PyTorch [PyTorch].

**3. SQLNet**

O trabalho [SQLNet] propôs uma solução para o *dataset* WikiSQL. Diferentemente de outros modelos de parsing semântico que são projetados para serem agnósticos em relação à gramática de saída, a ideia básica do SQLNet é utilizar um *sketch* para a gramática do SQL. Dessa forma, o principal trabalho é preencher as lacunas do *sketch* ao invés de ter que prever tanto a gramática de saída quanto o conteúdo. A figura 2 ilustra o *sketch* utilizado.



Figura 2. Modelo de sketch e dependências entre os componentes

Diferentemente do modelo SEQ2SQL, esse modelo não utilizou a técnica de aprendizado por reforço e obteve resultados superiores (próximos a 70%).

**3.1. Avaliação do SQLNet**

O código do SQLNet foi disponibilizado pelos autores no GitHub [SQLNetGITHUB]. O SQLNet foi desenvolvido em Python 2.7 [Python 2.7] utilizando a biblioteca PyTorch [PyTorch]. Seguem alguns pontos observados durante o processo de execução do código disponibilizado:

* Analisando o código observei que ele estava preparado para rodar com CPU e GPU. Entretanto, ao rodar o treinamento com CPU, ocorreram alguns erros que aparentemente eram simples e que, teoricamente, evidenciaram que o código não foi testado com CPU.
* O código de verificação da disponibilidade de GPU no hardware alvo da execução não estava funcionando. Quando tentei fazer o treinamento, ele não identificava GPU e mesmo assim passava pelo trecho de código previsto para hardware com GPU.
* O software foi desenvolvido em Python 2.7. Isso gerou algumas dependências com outras versões de bibliotecas que utilizam Python 3.
* Os autores disponibilizaram o código, porém não liberaram o modelo treinado. Entrei em contato com os autores solicitando o modelo treinado, mas ainda não obtive resposta (12/06/2018).

O artigo que apresenta o SQLNet diz que o modelo foi treinado em 200 épocas. Dessa forma, como o modelo não foi disponibilizado pelos autores, executei o processo de treinamento para esse período.

Executar o treinamento para 200 épocas.

Converter as consultas de interesse para inglês.

Criar os arquivos que representam o schema da base.

Executar o teste.

Exibir possíveis resultados do SQLNet.

**4. Proposta de criação de um dataset**

Conforme o relatório *Data Scientist Report 2017* [], para 53% dos entrevistados, a atividade que mais consome tempo dos cientistas de dados é coletar, rotular, limpar e organizar dados. Para que técnicas de aprendizado de máquina sejam aplicadas com eficácia, é primordial que os dados que serão utilizados de entrada sejam tratados e representem o modelo que se deseja prever.

Para os problemas da classe NL2SQL essa questão não é diferente. Dessa forma, para que um modelo de aprendizado de máquina consiga converter consultas em linguagem natural escritas em português para consultas em SQL, é fundamental que exista um conjunto de dados expressivo que ilustre diferentes cenários.

A base utilizada nesse trabalho se chama Base de Dados Integrada Corporativa (BDIC). A tabela abaixo ilustra a estrutura dessa base.

|  |  |  |  |
| --- | --- | --- | --- |
| **TABELA** | **COLUNA** | **TIPO DA COLUNA** | **OPERADOR** |
| País | Código | Número | EQUAL |
| País | Código | Número | NOT\_EQUAL |
| País | Código | Número | GREATER |
| País | Código | Número | LESS |
| ... | ... | ... | ... |
| País | Sigla | Texto | EQUAL |
| País | Sigla | Texto | NOT\_EQUAL |
| País | Sigla | Texto | LIKE |
| ... | ... | ... | ... |

Tabela 1: Representação das tabelas da BDIC

Cada linha da Tabela 1 é uma combinação <tabela, coluna, tipo coluna, operador> que representa um operador (EQUAL, NOT\_EQUAL, LESS, GREATER, LESS\_OR\_EQUAL, GREATER\_OR\_EQUAL, BETWEEN, NULL, NOT NULL, LIKE e NOT LIKE) suportado pelo tipo de uma coluna de uma determinada tabela. Assim, é possível gerar conjuntos de consultas SQL para a BDIC.

Para a geração do dataset desse trabalho, foi proposto um formato padrão para as consultas. A Figura 3 ilustra o formato padrão proposto.

Figura 3. Formato das consultas do dataset

SELECT COLUNA (, COLUNA)\*

FROM TABELA

WHERE COLUNA OP VALOR (AND COLUNA OP VALOR)\*

Onde os tokens COLUNA, TABELA, OP e VALOR representam:

* TABELA: token que identifica uma tabela. Não depende de nenhum outro token.
* COLUNA: token que identifica uma coluna. Depende de TABELA.
* OP: token que identifica um operador. Depende de COLUNA.
* VALOR: token que identifica um valor. Depende de COLUNA e OPERADOR.

A partir da descrição das tabelas da BDIC (Tabela 1) e o formato padrão das consultas SQL (Figura 3), é possível escrever um algoritmo para gerar um conjunto de consultas. O procedimento descrito na Figura 4 apresenta o pseudo-código da geração de um conjunto de consultas a serem executadas sobre as tabelas da BDIC.

*Procedimento de criação de registros para dataset{*

*Escolha uma tabela T;*

*Colunas das cláusulas "select" e "where" serão independentes?*

*Sim{*

*1: Escolha a qtd k de colunas da cláusula select;*

*2: Escolha k colunas da tabela T. OBS: as qtds de colunas das tabelas variam;*

*3: Escolha Q a quantidade de restrições da cláusula "WHERE";*

*4: Escolha Q colunas da tabela T;*

*5: Para cada uma das Q colunas, escolha P operadores OBS: os valores serão fixos.*

*}*

*Não{*

*1: Igual;*

*2: Igual;*

*3': Escolha Q a quantidade de restrições da cláusula "WHERE" onde 1<=Q<=K;*

*4': Escolha Q colunas da tabela T onde esse conjunto seja subconjunto das colunas escolhidas em 2;*

*5: Igual.*

*}*

*}*

Figura 4. Procedimento para geração de consultas SQL para a BDIC

Para a criação do dataset é preciso que para cada uma das consultas SQL geradas, haja uma representação dela em linguagem natural. Apesar de ser possível gerar as consultas relacionadas em linguagem natural, essa abordagem torna o dataset extremamente pobre já que através dele não será possível criar modelos gerais que consiga compreender as diferentes formas de se escrever uma consulta em linguagem natural, mas ele ficará viciado na forma que as consultas em português foram geradas automaticamente.

Uma abordagem para a geração de consultas em linguagem natural para as consultas geradas em SQL é a estratégia colaborativa. Nesse formato, pessoas são remuneradas por cada colaboração. A *Amazon* disponibiliza uma infraestrutura para a publicação desses projetos de colaboração, chamada *Amazon Mechanical Turk* [].

O material produzido está disponibilizado em https://github.com/riquebueno/trabalhoIA2.

**5. Avaliação de uma API de tradução**

O trabalho [] apresenta uma API para tradução de textos baseada no Google Tradutor []. O objetivo é utilizar essa API para processar as perguntas em linguagem natural escritas em português, traduzi-las para inglês e posteriormente repassá-las para o modelo SQLNet.

Em função de limitações de tempo, não foi possível avançar nessa avaliação.

**6. Conclusão e Trabalhos Futuros**

Este trabalho apresentou uma visão geral do *dataset* WikiSQL e da rede neural SQLNet. Também foi discutida uma estratégia para a geração de um *dataset* específico para a BDIC.

Como próximos trabalhos, espera-se evoluir o *dataset* para o BDIC e avaliar evoluções dos modelos NL2SQL.

Considerando que é desejado utilizar esses modelos em um ambiente corporativo, em função de políticas de sigilosidade, o conteúdo das tabelas pode não estar disponibilizado para apoiar a geração do modelo de predição. Esse é um ponto a ser avaliado para próximas atividades.

**Referências**

[] Zhong, Victor, Caiming Xiong, and Richard Socher. "Seq2SQL: Generating Structured Queries from Natural Language using Reinforcement Learning." arXiv preprint arXiv:1709.00103 (2017).

[] Xu, Xiaojun, Chang Liu, and Dawn Song. "SQLNet: Generating Structured Queries From Natural Language Without Reinforcement Learning." arXiv preprint arXiv:1711.04436 (2017).

[] Sutskever, Ilya, Oriol Vinyals, and Quoc V. Le. "Sequence to sequence learning with neural networks." Advances in neural information processing systems. 2014.

[] Cho, Kyunghyun, et al. "Learning phrase representations using RNN encoder-decoder for statistical machine translation." arXiv preprint arXiv:1406.1078 (2014).

[] Site https://visit.crowdflower.com/WC-2017-Data-Science-Report\_LP.html acessado em 12/06/2018.

[] Site https://www.mturk.com acessado em 12/06/2018.

[] Site https://github.com/xiaojunxu/SQLNet acessado em 12/06/2018.

[] Site https://pt.wikipedia.org/wiki/NoSQL acessado em 12/06/2018.

[] Site http://hadoop.apache.org acessado em 12/06/2018.

[] Site https://www.oracle.com/br/database/index.html acessado em 12/06/2018.

[] Site https://www.microsoft.com/en-us/sql-server/sql-server-2017 acessado em 12/06/2018.

[] Site https://www.forbes.com/sites/bernardmarr/2017/07/24/what-is-data-democratization-a-super-simple-explanation-and-the-key-pros-and-cons acessado em 12/06/2018.

[] https://github.com/salesforce/WikiSQL acessado em 12/06/2018.

[] Site https://pytorch.org acessado em 12/06/2018.

[] https://github.com/matheuss/google-translate-api

[] https://translate.google.com