**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

**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.

Esse trabalho está dividido em 4 partes.

Para a realização dos objetivos listados anteriormente são propostas as atividades abaixo:

Passo 1: https://github.com/salesforce/WikiSQL

Passo 2: https://github.com/xiaojunxu/SQLNet

Passo 3: Preparar uma base de dados para treinamento no contexto do SVIS e BDIC.

Passo 4: Avaliar a aplicação das técnicas propostas no artigo no problema de conversão de consultas em linguagem natural para consultas sobre o metamodelo do SVIS, permitindo que usuários expressem suas consultas em linguagem natural.

**Avaliação WikiSQL**

Rodei Seq2SQL. Entendi a estrutura do WikiSQL.

Cada tabela só aparece na base de treinamento, desenvolvimento ou de teste: The queries in WikiSQL span over a large number of tables and hence presents an unique challenge: the model must be able to not only generalize to new queries, but to new table schema. The tables, their paraphrases, and SQL queries are randomly slotted into train, dev, and test splits, such that each table is present in exactly one split.

We collect WikiSQL by crowd-sourcing on Amazon Mechanical Turk in two phases.

**Avaliação SQLNet**

Código para rodar com CPU estava com erro.

Não estava identificando que não havia GPU, apesar do código estar previsto para isso.

Python 2.7 dificultou o processo.

Aparentemente não houve testes para versão com CPU.

Ajustes foram feitos na semana 28/05 até 03/06.

Não disponibilizou o modelo treinado. Criei uma issue no git do projeto solicitando o modelo treinado.

<CASO EU CONSIGA O MODELO, PRECISAREI FORMATAR MEUS SCHEMAS E MINHAS PERGUNTAS PARA O FORMATO DE ENTRADA ESPERADO PELO TEST.PY>

**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 1 ilustra o formato padrão proposto.

Figura 1. 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 1), é possível escrever um algoritmo para gerar um conjunto de consultas. O procedimento descrito na Figura 2 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 2. 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.

**Avaliação de uma rede neural para tradução**

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

**Conclusão e Trabalhos Futuros**

Investigar outros artigos que tratam do tema e já obtiveram melhores resultados.

Evoluir o Dataset.

**Referências**

[] WikiSQL

[] SQLNet

[] Seq2Seq: https://papers.nips.cc/paper/5346-sequence-to-sequence-learning-with-neural-networks.pdf e http://emnlp2014.org/papers/pdf/EMNLP2014179.pdf

[] Data Scientist Report 2017

[] Amazon

[] Amazon Mechanical Turk

[] https://github.com/salesforce/WikiSQL

[] https://github.com/xiaojunxu/SQLNet

[] NoSQL

[] Hadoop

[] Oracle

[] SQL Server

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

[]

[]