

**Título do trabalho a ser apresentado à  
CPG para a dissertação/tese**

Nome completo do Autor

Dissertação/Tese apresentada  
ao  
Instituto de Matemática e Estatística  
da  
Universidade de São Paulo  
para  
obtenção do título  
de  
Mestre/Doutor em Ciências

Programa: Nome do Programa

Orientador: Prof. Dr. Nome do Orientador

Coorientador: Prof. Dr. Nome do Coorientador

Durante o desenvolvimento deste trabalho o autor recebeu auxílio financeiro da  
CAPES/CNPq/FAPESP

São Paulo, fevereiro de 2011

## **Título do trabalho a ser apresentado à CPG para a dissertação/tese**

Esta é a versão original da dissertação/tese elaborada pelo  
candidato (Nome Completo do Aluno), tal como  
submetida à Comissão Julgadora.

## **Título do trabalho a ser apresentado à CPG para a dissertação/tese**

Esta versão da dissertação/tese contém as correções e alterações sugeridas pela Comissão Julgadora durante a defesa da versão original do trabalho, realizada em 14/12/2010. Uma cópia da versão original está disponível no Instituto de Matemática e Estatística da Universidade de São Paulo.

Comissão Julgadora:

- Profa. Dra. Nome Completo (orientadora) - IME-USP [sem ponto final]
- Prof. Dr. Nome Completo - IME-USP [sem ponto final]
- Prof. Dr. Nome Completo - IMPA [sem ponto final]

## Agradecimientos

[illegible]



## Resumo

SOBRENOME, A. B. C. **Título do trabalho em português**. 2010. 120 f. Tese (Doutorado) - Instituto de Matemática e Estatística, Universidade de São Paulo, São Paulo, 2010.

[illegible]

**Palavras-chave:** palavra-chave1, palavra-chave2, palavra-chave3.



## Abstract

SOBRENOME, A. B. C. **Título do trabalho em inglês**. 2010. 120 f. Tese (Doutorado) - Instituto de Matemática e Estatística, Universidade de São Paulo, São Paulo, 2010.

[illegible]

**Keywords:** keyword1, keyword2, keyword3.





# Sumário

<b>Lista de Abreviaturas</b>	<b>ix</b>
<b>Lista de Símbolos</b>	<b>xi</b>
<b>Lista de Figuras</b>	<b>xiii</b>
<b>Lista de Tabelas</b>	<b>xv</b>
<b>1 Introdução</b>	<b>1</b>
1.1 Considerações Preliminares . . . . .	5
1.2 Objetivos . . . . .	6
1.3 Contribuições . . . . .	6
1.4 Organização do Trabalho . . . . .	7
<b>2 Trabalhos relacionados</b>	<b>9</b>
2.1 Question answering problem . . . . .	9
2.2 Redes neurais no problema de perguntas e respostas - verificar se utilizarei esta tradução . . . . .	9
2.3 Problema de perguntas e respostas entre linguagem natural e linguagem estruturada . . . . .	10
<b>3 Problema</b>	<b>11</b>
3.1 Problema de pares de perguntas e códigos-fontes do <i>StackOverFlow</i> . . .	11
3.1.1 Detalhes do dataset . . . . .	14
3.2 Arquitetura proposta . . . . .	14
<b>4 Resultados preliminares</b>	<b>15</b>
4.1 Resultados . . . . .	15
4.2 Considerações finais . . . . .	15
<b>5 Cronograma</b>	<b>17</b>
5.1 Próximos passos . . . . .	17
<b>A Sequências</b>	<b>19</b>

**Referências Bibliográficas**

**21**

## Lista de Abreviaturas

CFT	Transformada contínua de Fourier ( <i>Continuous Fourier Transform</i> )
DFT	Transformada discreta de Fourier ( <i>Discrete Fourier Transform</i> )
EIIP	Potencial de interação elétron-íon ( <i>Electron-Ion Interaction Potentials</i> )
STFT	Tranformada de Fourier de tempo reduzido ( <i>Short-Time Fourier Transform</i> )



## Lista de Símbolos

- $\omega$  Frequência angular
- $\psi$  Função de análise *wavelet*
- $\Psi$  Transformada de Fourier de  $\psi$



## Lista de Figuras

1.1	Perguntas relacionadas a pesquisa. Google and the Google logo are registered trademarks of Google LLC, used with permission. . . . .	4
3.1	Total de perguntas no <i>StackOverFlow</i> até 20 de março de 2019. Google and the Google logo are registered trademarks of Google LLC, used with permission. . . . .	12
3.2	Total de perguntas sobre Python e SQL no <i>StackOverFlow</i> até 20 de março de 2019. Google and the Google logo are registered trademarks of Google LLC, used with permission. . . . .	12





## Lista de Tabelas

3.1	Exemplo de pergunta <i>how-to</i> no <i>StackOverFlow</i> . Referência: <a href="https://stackoverflow.com/questions/19675761/how-to-sort-a-list-using-python">https://stackoverflow.com/questions/19675761/how-to-sort-a-list-using-python</a> . . . . .	13
A.1	Exemplo de tabela. . . . .	20



# Capítulo 1

## Introdução

O avanço e descoberta de novas tecnologias faz parte da história da humanidade. Em economia cunha-se o termo *General Purpose Technology*, que são avanços tecnológicos que afetam toda uma economia. Entre estes avanços tecnológicos temos o motor a vapor, posteriormente substituído pelo motor a combustão interna. O computador representou um avanço tecnológico. A internet representou um grande salto na maneira como são feitas as comunicações e transações financeiras hoje. E atualmente, inteligência artificial está caminhando para romper esta barreira e impactar toda a economia .

As primeiras redes neurais foram concebidas em 1945, o famoso perceptron. Mas desde 1945 e até os dias atuais, a área de inteligência artificial passou por vários momentos. Dentre eles momentos de euforia, no qual pesquisadores e entusiastas já previam robôs substituindo o ser humano e que todas as tarefas seriam feitas pelos computadores. E momentos de pouco investimento, chamado *AI Winter*, devido as muitas expectativas e poucos resultados. Porém demorou pelo menos 50 anos para termos os primeiros indícios do potencial desta área. Inteligência artificial abrange várias áreas. E a área que mais se destaca atualmente é a área de aprendizagem de máquina. Mais especificamente aprendizagem de máquina profunda.

Alan Turing após conceber o primeiro computador em 1930 (ver data e se está correto), disse que o computador poderia resolver qualquer problema desde que fosse possível passar as regras. O computador resolve facilmente operações aritméticas e até mesmo consegue ganhar de um campeão mundial de xadrez (1997). Porém, tarefas que são simples para o ser humano como reconhecer faces, falas ou traduzir textos, são tare-

adicionar referências

corrigir este parágrafo. Adicionar datas certas

fas muito complexas para o computador, pois o ser humano não consegue traduzir estas ações em regras ou fórmulas para o computador.

Porém, estes desafios de reconhecimento de imagens e tradução de textos foram superados pela máquina em 2015. Uma competição ImageNET no qual vários pesquisadores competem entre si para reconhecer milhares de imagem do banco imagenet. Um grupo do Canadá apresentou o CNN, uma rede neural convolucional profunda no qual obteve 70% de acurácia.

No ano posterior, ele obteve 97%. Este desafio foi dado como superado. Não foi somente reconhecimento de imagens que o computador igualou e superou o ser humano. A IBM desenvolveu um computador que ganhou do ser humano no *Jeopardy*. O Google desenvolveu o *Alpha Go* que superou o campeão mundial de Go.

Todas estas conquistas feitas por pesquisadores e especialistas da área de inteligência artificial, foram feitas utilizando redes neurais profundas, mais especificamente redes neurais convolucionais e recorrentes. Tanto as redes convolucionais e redes recorrentes já eram conhecidas desde a década de 90. Com algumas exceções e outros detalhes, a maioria dos algoritmos já haviam sido desenvolvidas na década de 80 e 90. Porém só em 2015 o primeiro resultado veio a tona.

Neste período de 20 a 30 anos, ocorreram evoluções em outras áreas da computação que permitiu o salto em aprendizagem de máquina. Os principais fatores que contribuíram foram:

- Aumento do processamento de CPU e GPU. Temos também pesquisas avançando em processadores específicos para a área de inteligência artificial como os TPUs
- Aumento da capacidade de armazenagem
- Acesso e democratização da internet, que permitiu gerar um grande volume de dados
- Avanços na pesquisa e no uso de arquiteturas de redes neurais mais eficientes para determinados tipos de problemas

Há outros fatores, mas estes fatores foram cruciais para o advento e o salto da aprendizagem de máquina atual. E atualmente cada vez mais empresas e governos estão investindo dinheiro na área de inteligência artificial. Há demandas para uso na área agrícola, investimentos, saúde, comércio. Empresas como o Google, IBM, Facebook, Apple e Amazon estão investindo bilhões de dólares nesta área. E uma das áreas que de aprendizagem de máquina é o processamento de linguagem natural.

A área de processamento de linguagem natural pesquisa e trabalha com diversos problemas. Entre eles temos a tradução de textos, geração de textos e um problema bastante comum é a solução de perguntas e respostas. O problema da pergunta e respostas consiste basicamente em dado uma pergunta e um conjunto de possíveis respostas, o algoritmo deve selecionar qual ou quais respostas respondem aquela pergunta.

verificar as  
areas de N

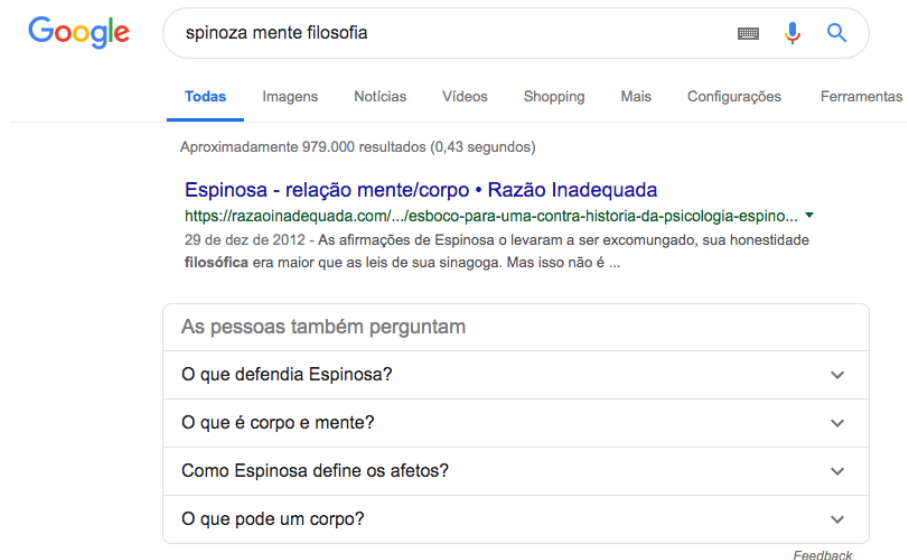
Normalmente este problema consiste em perguntas e respostas de no máximo 1 parágrafo. É um desafio hoje criar modelos que dado uma pergunta, consiga responder com um artigo. Ou dado um livro ou artigo, responder perguntas a respeito. Tanto que o próprio Google anunciou em janeiro de 2019, uma competição no qual deve-se criar um modelo capaz de responder ou não a perguntas, extraindo informação de artigos do Wikipedia. <https://ai.googleblog.com/2019/01/natural-questions-new-corpus-and.html>

adicionar re  
ferências

A competição tem uma base aberta para os interessados. É possível conferir o ranking também das arquiteturas propostas pelos pesquisadores: <https://ai.google.com/research/NaturalQuestions>

A partir de uma base de conhecimento, FAQ, empresas e governos podem utilizar um modelo treinado neste contexto para responder perguntas e dúvidas de usuários. Ao fazer pesquisa no *Google*, ele exibe perguntas e respostas sobre determinados tópicos pertinentes ao assunto buscado. Normalmente referem-se a dúvidas comuns de usuários. Exemplo está na figura 1.1.

Este problema de perguntas e respostas consiste em criar um modelo capaz de responder a dúvidas de usuários. A maioria das pesquisas atuais trabalham com linguagens naturais. Então, o usuário faz uma pergunta numa linguagem natural e o modelo responde com um texto em linguagem natural. Um outro desafio é dado uma pergunta em linguagem natural, criar um modelo capaz de responder em linguagem estruturada como XML, código fonte de linguagens como Java e/ou Python.



**Figura 1.1:** Perguntas relacionadas a pesquisa. Google and the Google logo are registered trademarks of Google LLC, used with permission.

Antes de obter o modelo, é necessário definir uma arquitetura em rede neural e ter os dados para treinar. Com relação a arquitetura, pode-se utilizar, inicialmente, alguma das diversas arquiteturas propostas para o problema de perguntas e respostas em linguagem natural.

Já em relação aos dados para o treinamento do modelo, é necessário criar uma base de dados de perguntas em linguagem natural e respostas numa linguagem estruturada. Uma possível fonte para este problema são os fóruns de programação. Junto com a internet, nasceu os fóruns. E os fóruns de programação fazem parte do cotidiano de todo programador. Diariamente vários programadores fazem e respondem perguntas sobre programação. E o site mais famoso e utilizado atualmente para perguntar e responder dúvidas de programação é o *StackOverflow*.

Neste site, o usuário publica uma pergunta sobre programação. A pergunta feita pode ser simples e de uma linha só como, por exemplo, *como ordenar um vetor de números em Python* ou pode ser uma pergunta mais longa que remete a um problema ou erro. Por exemplo, o usuário pode publicar um trecho de um programa e perguntar o motivo dele não funcionar. Os usuários do fórum respondem com uma possível solução para a pergunta e/ou problema. Caso esta solução consiga resolver o problema, o usuário que criou a pergunta pode marcar esta resposta como aceita. Além disso, os usuários do site

votam para as perguntas e respostas que eles julgaram mais relevantes.

Desde 2014, as perguntas e respostas feitas no site *StackOverFlow* estão disponíveis no site <https://archive.org/details/stackexchange>. Com estes dados disponíveis, várias pesquisas foram feitas . E uma delas é a pesquisa feita pelo [Yao et al. \(2018\)](#), no qual foram extraídas as perguntas e as respectivas respostas. A diferença desta pesquisa em relação as outras é que foram extraídas perguntas que tivessem uma resposta curta e direta. As perguntas que mais se encaixam nisso são as perguntas do tipo *How-to*. E as respostas também são curtas e diretas, e foram extraídas somente os códigos-fontes que respondiam a pergunta feito pelo usuário. Neste caso, foram extraídas os códigos-fontes das respostas aceitas pelo usuário que fez a pergunta.

Adicionar referências

Ao final desta pesquisa uma base de dados contendo pares de perguntas e códigos-fontes foi criada. Esta base foi disponibilizada ao final de 2018 [Yao et al. \(2018\)](#). [Yao et al.](#) utilizou uma rede neural recorrente bidirecional, no qual conseguiu obter uma precisão de 80% com uma acurácia e recall de 84% e 87%. Os próprios autores citam possíveis caminhos que podem ser explorados para melhorar estes números. E um deles é o uso de redes convolucionais que mostraram bons resultados na aprendizagem de representação de textos.

Neste sentido, o presente trabalho busca verificar o comportamento de uma arquitetura de redes neurais convolucionais combinada com uma rede recorrente no problema de pares de perguntas e códigos-fontes.

## 1.1 Considerações Preliminares

Neste trabalho, a arquitetura proposta que é composta pela combinação de redes convolucionais combinada com redes recorrentes é uma adaptação da arquitetura proposta por [Feng et al. \(2015\)](#). Esta arquitetura de rede neural mostrou bons estudos no problema de perguntas e respostas em linguagem natural.

Inicialmente, serão utilizados os dados disponibilizados pelo [Yao et al.](#). E para os resultados preliminares, serão levados em consideração os pares de perguntas e códigos fontes apenas em Python.



Para ter uma base de comparação, o resultado da arquitetura de redes neurais que combina rede convolucional e rede recorrente será comparado com os resultados de uma rede neural convolucional e outro somente rede neural recorrente.

## 1.2 Objetivos

O objetivo do presente trabalho é avaliar o desempenho de uma arquitetura de rede neural composta por uma rede neural convolucional e rede neural recorrente num problema de pares de perguntas e códigos-fontes. Neste estudo, busca responder a seguinte pergunta:

- Qual o desempenho de uma arquitetura de rede neural composta pela combinação de uma rede neural convolucional e rede neural recorrente no problema de pares de perguntas e códigos-fontes?

Ao analisar o desempenho, o principal objetivo é saber se o modelo obtido através do treinamento nos dados do *StackOverflow* é capaz de aprender. E aprender, significa extrair o conhecimento dos dados disponibilizados no treinamento e ser capaz de generalizar. Caso o modelo consiga generalizar, há a possibilidade de aplicabilidade prática.

## 1.3 Contribuições

As principais contribuições deste trabalho são as seguintes:

- Avaliação do desempenho da aplicação de uma arquitetura de rede neural composta pela combinação de redes neurais convolucionais e redes recorrentes no problema de pares de perguntas e códigos-fontes.
- Disponibilização do código fonte adaptado para uso futuro por outros pesquisadores e/ou interessados.

## 1.4 Organização do Trabalho

No Capítulo 2, apresentamos os conceitos e trabalhos relacionados ao problema de perguntas e respostas em processamento de linguagem natural. No capítulo 3, discutimos o problema de pesquisa proposto para o presente trabalho, a arquitetura proposta e apontamos detalhes dos dados utilizados pertinentes a pesquisa. No capítulo 4, exibimos os resultados preliminares da arquitetura proposta utilizando os dados do Yao *et al.* (2018). E também discutimos as principais dificuldades encontradas para adaptar a arquitetura do Feng *et al.* (2015) para resolver problema de pares de perguntas e códigos-fontes. Já no capítulo 5, temos as considerações finais, proposta do cronograma e os próximos passos da pesquisa.



## Capítulo 2

### Trabalhos relacionados

Este capítulo vai explicar sobre o problema de perguntas e respostas. Uso de redes neurais como CNN e LSTM para resolver este problema. Irá abordar também o problema de perguntas e respostas no caso de linguagem natural e linguagem estruturada (código-fonte).

#### 2.1 Question answering problem

Maioria dos modelos que tentam resolver este problema concentram-se em perguntas e respostas de no máximo 1 parágrafo. Este é um desafio ainda a ser superado. Atualmente não há modelos que conseguem a partir de uma pergunta ou um tema responder com um artigo.

E um outro desafio proposto aqui é dado uma pergunta em uma linguagem natural, responder com um texto estruturado, código fonte em Python.

#### 2.2 Redes neurais no problema de perguntas e respostas - verificar se utilizarei esta tradução

O problema de perguntas e respostas é muito comum na área de processamento de linguagens naturais. Há vários datasets públicos a disposição para que pesquisadores proponham solução para este problema. E redes neurais desponta como o modelo mais utilizado para este tipo de problema atualmente. O mais comum é o uso de redes neurais

recorrentes e convolucionais.

## **2.3 Problema de perguntas e respostas entre linguagem natural e linguagem estruturada**

A maioria dos modelos citados na seção anterior tentam solucionar o problema de perguntas e respostas em linguagem natural. Porém, com o advento do Big Code, repositórios open source, outros problemas que envolvem uma complexidade maior surgem. Por exemplo, dado uma sentença em linguagem natural, uma pergunta, qual a resposta poderíamos ter em uma linguagem estrutural como Java, XML ou Python? Ou até mesmo SQL? A partir de análises de commits do github, podemos aferir a partir de uma história de usuário, quais alterações são necessárias para concluir aquela tarefa, por exemplo. No nosso caso, a partir do database do stackoverflow, queremos saber a partir de uma pergunta, qual o código fonte é a solução.

## Capítulo 3

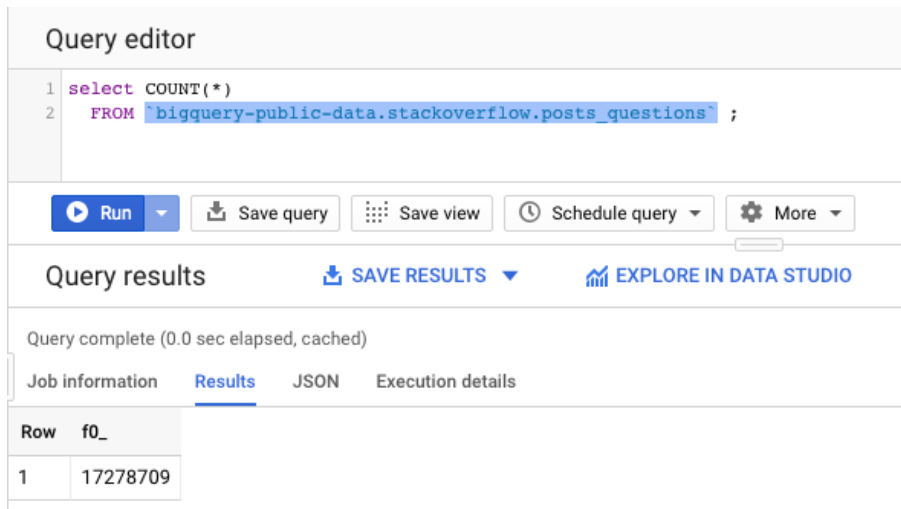
### Problema

Conforme o capítulo 2, há várias pesquisas sobre o problema de perguntas e respostas no processamento de linguagem natural. Porém, como observado pelo trabalho da Yao *et al.* (2018) e apontado na seção 2.3, não há muita pesquisa sobre o problema de pares de perguntas e códigos-fontes. A partir de 2014, o StackOverFlow passou a disponibilizar a base de dados livremente. Diversas pesquisas foram feitas utilizando esta base de dados. E uma frente explorada pela Yao *et al.* (2018) e outros pesquisadores, é tentar resolver o problema de pergunta e resposta utilizando os dados do *StackOverFlow*.

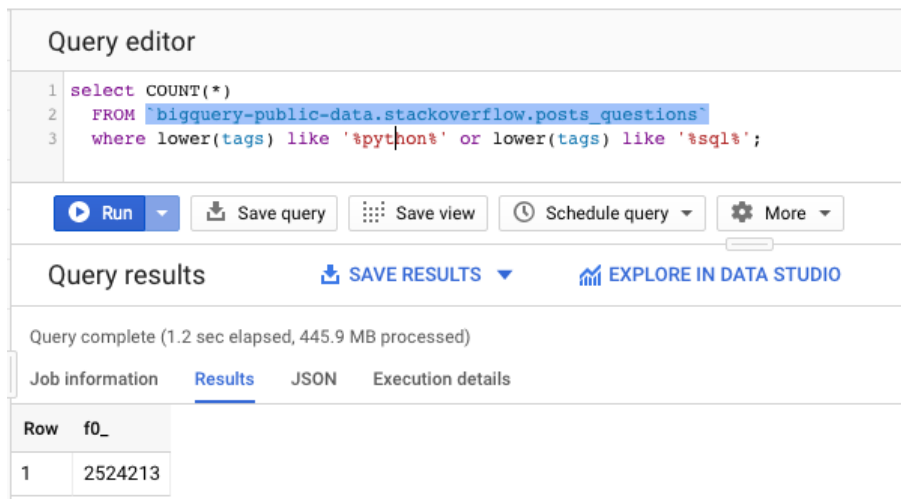
### 3.1 Problema de pares de perguntas e códigos-fontes do *StackOverFlow*

O problema de pares de perguntas e códigos-fontes já foi discutido na seção 2.3. E um dos obstáculos mencionados, é a obtenção de dados para o treinamento de modelos supervisionados. Desde 2014, o site *StackOverFlow* disponibiliza sua base de perguntas e respostas através do site *Stack Exchange* (2019). Estes dados também podem ser consultados através do site *BigQuery* do Google (2019) também. Numa consulta rápida, até 20 de março de 2019, havia mais de 17 milhões de perguntas em diversos tópicos de programação. Dúvidas sobre Python e SQL totalizam juntas mais de 2 milhões e meio.

Somente com estes números, vê-se o potencial do *StackOverFlow* como fonte de perguntas em linguagem natural e a sua solução em código fonte Yao *et al.* (2018). Porém, ao utilizar o *StackOverFlow* como fonte de repositório de pares de perguntas e códigos-



**Figura 3.1:** Total de perguntas no StackOverFlow até 20 de março de 2019. Google and the Google logo are registered trademarks of Google LLC, used with permission.



**Figura 3.2:** Total de perguntas sobre Python e SQL no StackOverFlow até 20 de março de 2019. Google and the Google logo are registered trademarks of Google LLC, used with permission.

fontes, é necessário levar em consideração quais assuntos é permitido fazer a pergunta. De acordo com o guia de boas práticas do [stackoverflow \(2019\)](#), são permitidas perguntas sobre:

- um problema específico de programação
- um algoritmo de programação
- ferramentas de programação
- problemas referentes a desenvolvimento de software

Analisando estes tipos de perguntas, uma pergunta sobre ferramentas de programação, dificilmente levará a uma resposta cuja solução seja um código fonte. E se analisar cada um dos tópicos, verifica-se que nem todas as dúvidas vão levar a uma resposta com código-fonte. Muitas vezes, a resposta é somente uma orientação e não necessariamente leva a solução da pergunta feita por um usuário.

Levando isso em consideração, Yao *et al.* coletou a partir da base aberta de questões e respostas do [stackoverflow](#) (2019), somente perguntas do tipo *how-to* referentes as linguagens SQL e *Python*. Perguntas do tipo *how-to*, normalmente tem apenas uma resposta curta e direta cuja solução é um código-fonte. Exemplo:

<b>Pergunta:</b> how to sort a list using python?
<b>Descrição:</b> How I sort my python list below using name property, <code>[{'name':'Accra','url':'"}, {'name':'Dubai','url':'"}, {'name':'Birmingham','url':'"},  {'name':'Dhaka','url':'"}, {'name':'Abu Dhabi','url':'"}, {'name':'Lahore','url':'"},  {'name':'Karachi','url':'"}, {'name':'Kathmandu','url':'"}, {'name':'SEA','url':'"}]</code>
<b>Resposta aceita pelo usuário:</b>
<pre>In [26]: L = [{'name':'Accra','url':'"},{'name':'Dubai','url':'"},  {'name':'Birmingham','url':'"},{'name':'Dhaka','url':'"},  {'name':'Abu Dhabi','url':'"},{'name':'Lahore','url':'"},  {'name':'Karachi','url':'"},{'name':'Kathmandu','url':'"},  {'name':'SEA','url':'"}]</pre> <pre>In [27]: L.sort(key=lambda d:d['name'])</pre> <pre>In [28]: L Out[28]: [{'name': 'Abu Dhabi', 'url': '"}, {'name': 'Accra', 'url': '"},  {'name': 'Birmingham', 'url': '"}, {'name': 'Dhaka', 'url': '"}, {'name': 'Dubai', 'url':  "}, {'name': 'Karachi', 'url': '"}, {'name': 'Kathmandu', 'url': '"}, {'name': 'Lahore',  'url': '"}, {'name': 'SEA', 'url': '"}]</pre>

**Tabela 3.1:** Exemplo de pergunta *how-to* no StackOverFlow. Referência: <https://stackoverflow.com/questions/19675761/how-to-sort-a-list-using-python>

No exemplo da tabela 3.1, a resposta aceita<sup>1</sup> pelo usuário, tem apenas um código-fonte como solução. Entretanto, a resposta aceita pelo usuário pode conter mais de um código-fonte. E cada código-fonte pode ser uma possível solução para a questão.

Adicionar exemplo...

<sup>1</sup>Para uma resposta ser aceita no StackOverFlow



### 3.1.1 Detalhes do dataset

- Tamanho do dataset inicial - Dados sobre as perguntas, respostas (tamanho, quantidade de palavras etc.)

## 3.2 Arquitetura proposta

- Uso da rede LSTM/CNN - Utilizando cosine similarity - Uso do word2vec para pergunta (linguagem natural) e outro para código (linguagem estruturada) - Verificar também se é viável usar o artigo Deep contextualized word representations ELMo representation (ao invés de word2vec) (According to the article: "We show that these representations can be easily added to existing models and significantly improve the state of the art across six challenging NLP problems, including question answering, textual entailment and sentiment analysis." )

## Capítulo 4

### Resultados preliminares

Este capítulo vai exibir os resultados preliminares da arquitetura proposta no dataset escolhido até o momento.

#### 4.1 Resultados

- Lidando com overfitting (Uso de regularizacao ou early stopping) - Quantidade de hidden layers vai ser ajustada pela regularização - Ajuste dos parâmetros de acordo com o dataset

- Validacao do resultado - Uso do cross validation - Dado que o dataset é pequeno - Análise dos resultados do cross validation para saber se o modelo aprendeu (É suficiente o resultado do cross validation para saber se o modelo aprendeu?) - Resultados do cross validation: - Accuracy - Precision - Recall - F1 Score

#### 4.2 Considerações finais



## Capítulo 5

## Cronograma

## 5.1 Próximos passos

[illegible]

<sup>1</sup>Exemplo de referência para página Web: [www.vision.ime.usp.br/~jmena/stuff/tese-exemplo](http://www.vision.ime.usp.br/~jmena/stuff/tese-exemplo)



## Apêndice A

## Sequências

[illegible]

<i>Limiar</i>	MGWT			AMI			<i>Spectrum</i> de Fourier			Características espectrais		
	<i>Sn</i>	<i>Sp</i>	<i>AC</i>	<i>Sn</i>	<i>Sp</i>	<i>AC</i>	<i>Sn</i>	<i>Sp</i>	<i>AC</i>	<i>Sn</i>	<i>Sp</i>	<i>AC</i>
1	1.00	0.16	0.08	1.00	0.16	0.08	1.00	0.16	0.08	1.00	0.16	0.08
2	1.00	0.16	0.09	1.00	0.16	0.09	1.00	0.16	0.09	1.00	0.16	0.09
2	1.00	0.16	0.10	1.00	0.16	0.10	1.00	0.16	0.10	1.00	0.16	0.10
4	1.00	0.16	0.10	1.00	0.16	0.10	1.00	0.16	0.10	1.00	0.16	0.10
5	1.00	0.16	0.11	1.00	0.16	0.11	1.00	0.16	0.11	1.00	0.16	0.11
6	1.00	0.16	0.12	1.00	0.16	0.12	1.00	0.16	0.12	1.00	0.16	0.12
7	1.00	0.17	0.12	1.00	0.17	0.12	1.00	0.17	0.12	1.00	0.17	0.13
8	1.00	0.17	0.13	1.00	0.17	0.13	1.00	0.17	0.13	1.00	0.17	0.13
9	1.00	0.17	0.14	1.00	0.17	0.14	1.00	0.17	0.14	1.00	0.17	0.14
10	1.00	0.17	0.15	1.00	0.17	0.15	1.00	0.17	0.15	1.00	0.17	0.15
11	1.00	0.17	0.15	1.00	0.17	0.15	1.00	0.17	0.15	1.00	0.17	0.15
12	1.00	0.18	0.16	1.00	0.18	0.16	1.00	0.18	0.16	1.00	0.18	0.16
13	1.00	0.18	0.17	1.00	0.18	0.17	1.00	0.18	0.17	1.00	0.18	0.17
14	1.00	0.18	0.17	1.00	0.18	0.17	1.00	0.18	0.17	1.00	0.18	0.17
15	1.00	0.18	0.18	1.00	0.18	0.18	1.00	0.18	0.18	1.00	0.18	0.18
16	1.00	0.18	0.19	1.00	0.18	0.19	1.00	0.18	0.19	1.00	0.18	0.19
17	1.00	0.19	0.19	1.00	0.19	0.19	1.00	0.19	0.19	1.00	0.19	0.19
17	1.00	0.19	0.20	1.00	0.19	0.20	1.00	0.19	0.20	1.00	0.19	0.20
19	1.00	0.19	0.21	1.00	0.19	0.21	1.00	0.19	0.21	1.00	0.19	0.21
20	1.00	0.19	0.22	1.00	0.19	0.22	1.00	0.19	0.22	1.00	0.19	0.22

Tabela A.1: *Exemplo de tabela.*

## Referências Bibliográficas

**Feng et al.(2015)** M. Feng, B. Xiang, M. R. Glass, L. Wang e B. Zhou. Applying deep learning to answer selection: A study and an open task. Em *2015 IEEE Workshop on Automatic Speech Recognition and Understanding (ASRU)*, páginas 813–820. doi: 10.1109/ASRU.2015.7404872. Citado na pág. 5, 7

**Google(2019)** Google. Google cloud bigquery, 2019. URL <https://cloud.google.com/bigquery/>. Citado na pág. 11

**Stack Exchange(2019)** Inc. Stack Exchange. Stack exchange data dump, 2019. URL <https://archive.org/details/stackexchange>. Citado na pág. 11

**stackoverflow(2019)** stackoverflow. What topics can i ask about here?, 2019. URL <https://stackoverflow.com/help/on-topic>. Citado na pág. 12, 13

**Yao et al.(2018)** Ziyu Yao, Daniel S. Weld, Wei-Peng Chen e Huan Sun. Staqc: A systematically mined question-code dataset from stack overflow. Em *Proceedings of the 2018 World Wide Web Conference, WWW '18*, páginas 1693–1703, Republic and Canton of Geneva, Switzerland. International World Wide Web Conferences Steering Committee. ISBN 978-1-4503-5639-8. doi: 10.1145/3178876.3186081. URL <https://doi.org/10.1145/3178876.3186081>. Citado na pág. 5, 7, 11, 13