UNIVERSIdade FEEVALE

GABRIEL MOUTINHO DA SILVA

DESENVOLVIMENTO DE UM *CHABOT* PARA PORTUGUÊS BRASILEIRO BASEADO NA ARQUIETURA *SEQUENCE-TO-SEQUENCE*

Novo Hamburgo

2018

GABRIEL MOUTINHO DA SILVA

DESENVOLVIMENTO DE UM *CHABOT* PARA PORTUGUÊS BRASILEIRO BASEADO NA ARQUIETURA *SEQUENCE-TO-SEQUENCE*

Trabalho de Conclusão de Curso

apresentado como requisito parcial

à obtenção do grau de Bacharel em

Sistemas de Informação pela

Universidade Feevale

Orientador: prof. Ms. Gabriel da Silva Simões

Novo Hamburgo

2018

Resumo

A automação de tarefas é um dos principais objetivos da computação em geral, tendo em vista a redução de custos e melhoria na efetividade dos processos. O desejo de automatizar as tarefas ligadas ao atendimento de clientes, suporte técnico, assim como o desejo de criar agentes que tenham a capacidade sustentar conversas de maneira natural com seres humanos serve de impulso para o avanço das técnicas de Processamento de Linguagem Natural (PLN).

Este trabalho se propõe a realizar o treinamento de dois *chatbots* na língua Português-Brasileiro baseado em *Deep Learning* a partir de um *corpus* de acesso público e domínio aberto e um *corpus* específico de conversas, utilizando as técnicas já consolidadas de *Sequence-to-Sequence* (Seq2Seq) e *Word Embeddings*, combinando-as com a técnicasde PLN para pré-processamento de textos. Tendo como objetivo a comparação de resultados entre estes modelos com o estado da arte para agentes conversacionais de domínio aberto em Português-Brasileiro.

Palavras-chave: Chatbots. Processamento de Linguagem Natural. Aprendizado Profundo. Word Embeddings. Sequence-to-Sequence.

Abstract

Automation of tasks is one of the main objectives of computing in general, with the objective of reducing costs and improving effectivity processes. The desire to automate tasks related to customer service, technical support, besides the wish to create agents who have a sustained ability to chat in a natural way with human service encourages new advances of techniques of Natural Language Processing. This work aims to create a open domain Deep Learning based Chatbot for Brazilian Portuguese, using the already consolidated techniques of Sequence-to-Sequence (Seq2Seq) and Word Embeddings, combining it with pre-processing techniques from NLP. Having the objective of comparing the results of the models created aiming for the state-of-art results for open domain chatbots in Brazilian Portuguese.

Key-words: Chatbots. Natural Language Processing. Deep Learning. Word Embeddings. Sequence-to-Sequence.

Lista de Figuras

[Figura 1 Os estágios da análise em PLN 17](#_Toc518564859)

[Figura 2 A Arquitetura CBOW 21](#_Toc518564860)

[Figura 3 A Arquitetura SKIP-GRAM. 22](#_Toc518564861)

[Figura 4 Algoritmos de Machine Learning 31](#_Toc518564862)

[Figura 5 Aprendizado Supervisionado 32](#_Toc518564863)

[Figura 6 Aprendizado Não-Supervisionado 33](#_Toc518564864)

[Figura 7 Aprendizado por Reforço 35](#_Toc518564865)

[Figura 8 Modelo de neurônio artificial por Haykin 36](#_Toc518564866)

[Figura 9 Perceptron 38](#_Toc518564867)

[Figura 10 - Multi-Layer Perceptron 38](file:///C:\Users\gsilva\Desktop\00-TCC2-Gabriel-V17-Final.docx#_Toc518564868)

[Figura 11 Deep Neural Network 40](#_Toc518564869)

[Figura 12 Arquitetura CNN 42](#_Toc518564870)

[Figura 13 Sequências nas RNN 43](#_Toc518564871)

[Figura 14 Modelo Sequence-to-Sequence 45](#_Toc518564872)

[Figura 15 Arquitetura Chabot proposto por Maeda e Moraes 51](#_Toc518564873)

[Figura 16 Processo desenvolvimento do trabalho 54](#_Toc518564874)

[Figura 17 Arquitetura do Chatbot 59](#_Toc518564875)

[Figura 18 Trecho conversa Chatbot Legendas 62](#_Toc518564876)

[Figura 19 Trecho conversa Chatbot Wit³ 63](#_Toc518564877)

[Figura 20 Respostas Pós-interação Wit³ 63](#_Toc518564878)

[Figura 21 Respostas Pós-interação legendas 64](#_Toc518564879)

[Figura 22 Questionário Pré-interação- Resposta 1 74](#_Toc518564880)

[Figura 23 Questionário Pré-interação- Resposta 2 74](#_Toc518564881)

[Figura 24 Questionário Pré-interação- Resposta 3 75](#_Toc518564882)

[Figura 25 Questionário Pré-interação- Resposta 4 75](#_Toc518564883)

[Figura 26 Questionário Pré-interação- Resposta 5 76](#_Toc518564884)

Lista de Tabelas

[Tabela 1 Exemplo de Relações Semânticas Encontradas por um Word Embedding 19](#_Toc518555478)

[Tabela 2 Exemplo de Tags POS Tagger PALAVRAS 24](#_Toc518555479)

[Tabela 3 Áreas de Conhecimento e sua influência no Aprendizado de Máquina 28](#_Toc518555480)

[Tabela 4 Dependências do Projeto 56](#_Toc518555481)

[Tabela 5 Hiperparâmetros chatbot legendas 59](#_Toc518555482)

[Tabela 6 Hiperparâmetros chatbot Wit³ 60](#_Toc518555483)

Lista de Abreviaturas e Siglas

|  |  |
| --- | --- |
| AM | Aprendizado de Máquina |
| AIML | Artificial Intelligence Markup Language |
| ALICE | Artificial Linguistic Internet Computer Entity |
| ASR | Automated Speech Recognition |
| BOW | Bag-of-Words |
| CBOW | Continuous Bag-of-Words |
| DL | Deep Learning |
| DNN | Deep Neural Networks |
| IA | Inteligência Artificial |
| LSTM | Long Short-Term Memory |
| MLP | Multi-Layer Perceptron |
| NLU | Natural Language Understanding |
| NLTK | Natural Language Toolkit |
| PLN | Processamento de Linguagem Natural |
| POS | Part-of-Speech |
| SLU | Spoken Language Understanding |
| VPA | Virtual Personal Assistants |

Sumário

[INTRODUÇÃO 11](#_Toc518555484)

[1.1 Objetivos 13](#_Toc518555485)

[1.2 Estrutura do Trabalho 14](#_Toc518555486)

[1 PROCESSAMENTO DE LINGUAGEM NATURAL 16](#_Toc518555487)

[1.1 Word Embeddings 19](#_Toc518555488)

[1.1.1 CBOW 21](#_Toc518555489)

[1.1.2 SKIP-GRAM 21](#_Toc518555490)

[1.2 Part-of-Speech Tagging 23](#_Toc518555491)

[1.3 Stemming 25](#_Toc518555492)

[1.4 Lemmatization 26](#_Toc518555493)

[1.5 Tokenization 26](#_Toc518555494)

[2 APRENDIZADO DE MÁQUINA 28](#_Toc518555495)

[2.1 Aprendizado Supervisionado 31](#_Toc518555496)

[2.2 Aprendizado Não-Supervisionado 32](#_Toc518555497)

[2.3 Aprendizado Semi-Supervisionado 33](#_Toc518555498)

[2.4 Aprendizado por Reforço 34](#_Toc518555499)

[2.5 Redes Neurais Artificiais 35](#_Toc518555500)

[2.6 Deep Learning 39](#_Toc518555501)

[2.6.1 Redes Neurais Convolucionais 41](#_Toc518555502)

[2.6.2 Redes Neurais Recorrentes 42](#_Toc518555503)

[2.6.3 Sequence-to-Sequence 43](#_Toc518555504)

[3 CHATBOTS 46](#_Toc518555505)

[3.1 Simbólico 47](#_Toc518555506)

[3.2 Modelo Baseado em Recuperação de Informações 47](#_Toc518555507)

[3.3 Probabilístico 48](#_Toc518555508)

[3.4 Aplicações 48](#_Toc518555509)

[4 TRABALHOS RELACIONADOS 50](#_Toc518555510)

[5 METODOLOGIA 52](#_Toc518555511)

[5.1 Tipo de pesquisa 52](#_Toc518555512)

[5.2 Design dos experimentos 53](#_Toc518555513)

[5.3 Design da avaliação dos resultados 53](#_Toc518555514)

[6 DESENVOLVIMENTO DOS CHATBOTS 54](#_Toc518555515)

[6.1 Aspectos de Implementação 54](#_Toc518555516)

[6.2 Definição e Pré-processamento dos *Corpora* 56](#_Toc518555517)

[6.3 Definição das Word Embeddings 58](#_Toc518555518)

[6.4 Arquitetura do Chatbot 59](#_Toc518555519)

[7 AVALIAÇÃO DE RESULTADOS 62](#_Toc518555520)

[CONCLUSÃO 65](#_Toc518555521)

[REFERÊNCIAS BIBLIOGRÁFICAS 67](#_Toc518555522)

[APÊNDICE A – Questionário pré-interação 72](#_Toc518555523)

[APÊNDICE B – Questionário Pós-interação 73](#_Toc518555524)

[APÊNDICE C – Respostas Questionário pré-interação 74](#_Toc518555525)

# INTRODUÇÃO

Os avanços na área de Inteligência artificial têm impulsionado diversas novas soluções para problemas de outras áreas, desenvolvendo soluções com capacidade de aprendizado, aumentando a precisão destes algoritmos, além de possibilitar um aumento na autonomia e tomada de decisão. Segundo Goodfellow, Bengio e Courville (2016) a área de Inteligência artificial é um campo próspero e em pleno desenvolvimento, com muitas aplicações práticas já adotadas na indústria e novas tecnologias sendo desenvolvidas pela comunidade científica. Hoje em dia podemos ver softwares inteligentes automatizando o trabalho manual em diversos campos, compreendendo fala, imagens e até mesmo auxiliando no diagnóstico médico e apoio em áreas de pesquisa.

Uma das principais técnicas que possibilitaram este aumento na eficiência e consequentemente utilização de algoritmos de inteligência artificial é o *Deep Learning* ou Aprendizagem Profunda segundo Cahn (2017) as redes neurais baseadas em *Deep Learning* diferem das redes neurais tradicionais por usarem mais camadas ocultas e a cada camada oculta lidando com aspectos mais complexos.

A área de Processamento de Linguagem Natural (PLN, do inglês *Natural Language Processing*) se beneficiou amplamente destas técnicas para a extração de conhecimento de estruturas complexas como a fala e texto, as quais são duas das áreas com grande potencial de desenvolvimento de novas técnicas para aplicação na indústria. Comarella e Café (2008) descrevem que o desenvolvimento de programas e agentes capazes de analisar e interpretar a linguagem humana ao nível de ser possível gerar uma conversação coerente é um dos principais desafios atuais da área de PLN (SPANHOL, 2017, p.31 apud COMARELLA; CAFÉ, 2008).

O PLN possui uma gama de métodos para extração semântica de informação e significado a partir da linguagem falada ou escrita. Estas técnicas podem identificar intenções a partir de texto ou fala não normatizada que podem conter diversos problemas como ruído, erros gramaticais, interrupções, defluências e autocorreções (CAHN, 2017, p.10).

Entre estas técnicas estão os “*Chatterbots*, também chamados de *chatbots* ou agentes conversacionais projetados para emular respostas humanas comuns durante uma conversação” (MORAES, MACHADO, 2016 apud KULIGOWSKA, LASEK, 2011, p. 1347). Um dos grandes problemas no desenvolvimento de *chatbots* sempre foi a dificuldade da uma máquina gerar uma conversação natural com um ser humano. Devido a imprevisibilidade, abstração e contexto a tarefa de modelar este conhecimento de forma manual é inviável e acaba por gerar modelos restritos ao domínio ao qual foram criados. Conforme Moraes e Machado (2016) as bases de conhecimento devem conter um grande número de padrões de perguntas e respostas para simular um diálogo natural, o que acaba por criar um modelo inflexível a partir de um grande esforço.

Os agentes conversacionais são definidos como programas de computador que simulam a conversação humana, tendo como objetivo responder perguntas de tal forma que o usuário tenha a sensação de estar conversando com uma pessoa (SPANHOL, 2017 apud LEONHARDT, 2005).

Conforme descrito por Moraes e Machado (2016), um dos problemas desta tecnologia, apesar de sua popularidade é que estes sistemas em sua maioria utilizam apenas técnicas triviais de Processamento de Linguagem Natural. Os *chatbots* também possuem certa limitação quando suas bases de conhecimento são escritas na linguagem AIML, baseada na correspondência de padrões. Isso torna estes sistemas limitados a sua de base de conhecimento e sem capacidade de atender ao usuário caso este saia do escopo delimitado. Desta forma a necessidade de sistemas que se adaptem independente ao domínio de treinamento e perguntas mapeadas continuam não sendo plenamente atendidas, pois os modelos gerados possuem limitações ao ultrapassar seus domínios. Uma alternativa a estes modelos seria a técnica de Aprendizado por Sequência (do inglês: *Sequence Learning*) muito utilizada em trabalhos com dados sequenciais (textos, fala, vídeo) aonde a rede neural retém a informações de sequências processadas anteriormente. Apesar da técnica de Aprendizado por Sequência também possuir limitações, os agentes criados possuem maior capacidade de interação com os usuários mesmo em casos em que as perguntas não foram mapeadas. Em relação a técnica de *Sequence to Sequence*, Vinyals e Le (2015) relatam que a sua maior vantagem como *framework* é demandar menor esforço no desenvolvimento de recursos orientados a especificidade de um domínio de conhecimento, onde a técnica demonstra resultados que se equiparam ou até mesmo superam o estado da arte. Este avanço permite aos pesquisadores trabalharem em tarefas para qual o domínio de conhecimento não está prontamente disponível ou em tarefas nas quais são inviabilizam a criação manual de regras.

O grande potencial das redes neurais é a capacidade de mapear estruturas complexas para outras estruturas complexas (VINYALS, LE, 2015). Desta forma as etapas de PLN são fortemente beneficiadas, gerando um ganho expressivo no treinamento de sistemas mais flexíveis e generalistas. Sendo assim o modelo é capaz de extrair conhecimento tanto de um *dataset* relacionado a um domínio especifico, quanto de um grande *dataset* de domínio geral com ruídos como por exemplo um *dataset* de legendas de filmes (VINYALS, LE, 2015, p. 7).

Segundo Madsen e Adamatti (2011) a principal característica das redes neurais são a capacidade de aprendizado a partir de exemplos, podendo ser treinada com dados do mundo real com ruídos e casos de informações faltantes ou incorretas (SPANHOL, 2017, p. 26 apud MADSEN, ADMANTTI, 2011).

Considerando a menor disponibilidade de *datasets* de perguntas e respostas ou de conversação o desenvolvimento de *chatbots* para a língua Português-Brasileiro é caracterizado por desafios maiores na construção do vocabulário a ser utilizado pelo modelo (MAEDA; MORAES, 2017). Quando alimentado com um *dataset* que consiga sintetizar o conhecimento de uma língua o modelo *Sequence-to-Sequence* (seq2seq) se torna capaz de gerar conversas simples e básicas, assim como realizar extração de conhecimento a partir de um *dataset* aberto de legendas de filmes (VINYALS, LE, 2015, p. 7).

A técnica de *Part-of-Speech Tagging* já é utilizada em outros campos, e em recentes trabalhos foi aplicada no treinamento de *chatbots*. Conforme descrito por Shawar e Atwell (2005), a técnica de *Part-of-Speech* analisa cada palavra em um texto e destaca palavras específicas em classes e categorias sintáticas as quais aparecem mais frequente que outros no texto. Sendo assim o presente trabalho se propõe a criar um agente conversacional utilizando técnicas de Deep Learning, por meio da arquitetura *Sequence-to-Sequence*, Processamento de Linguagem Natural e *Word Embeddings*

Os *chatbots* serão responsáveis por uma grande revolução no modo como os sistemas interagem com o usuário final, neste sentido todas as iniciativas em aprimorar a capacidade de interação destes agentes com o usuário são de extrema valia e devem ser estudadas de forma a se atingir os melhores resultados (MAEDA; MORAES, 2017).

## Objetivos

O objetivo geral deste trabalho é a criação de um agente conversacional de domínio aberto, na língua Português Brasileiro, utilizando a arquitetura *sequence-to-sequence* e técnicas de Processamento de Linguagem Natural.

A partir disso, são relacionados os seguintes objetivos específicos:

Objetivos Específicos

* Pesquisar a aplicação de redes neurais no processamento de linguagem natural;
* Pesquisar sobre os tipos existentes de *Chatbots*.
* Pesquisar o referencial teórico a respeito de *Sequence-to-Sequence*
* Pesquisar o referencial teórico a respeito de *Word Embeddings;*
* Pesquisar sobre a técnicas de Processamento de Linguagem Natural aplicadas a construção e agentes conversacionais;
* Pesquisar os *Corpora* ou *Corpus* disponíveis para a língua Português Brasileiro.
* Desenvolver um *Chatbot* utilizando a arquitetura *Sequence-to-Sequence* empregando as técnicas de *Word Embeddings* e Processamento de Linguagem Natural;
* Comparar os resultados obtidos utilizando diferentes abordagens de treinamento e *corpora*;
* Realizar experimentos com *corpora* não orientados a diálogos.
* Avaliar os resultados obtidos em comparação ao estado da arte em Português Brasileiro.

## Estrutura do Trabalho

O presente trabalho está estruturado em 7 capítulos. O capítulo 1 faz uma introdução ao Processamento de Linguagem Natural (do inglês *Natural Language Processing*), além do embasamento sobre Processamento de Linguagem Natural foi realizada uma breve introdução sobre os tipos de técnicas existentes, técnicas relevantes para a construção de *chatbots,* algoritmos utilizados para o pré-processamento de *datasets*.

O capítulo 2 é dedicado a área de Aprendizado de Máquina (do inglês *Machine Learning*) descrevendo brevemente seu histórico, principais arquiteturas e técnicas relacionadas a este trabalho, também explanando sobre as principais subáreas e áreas relacionadas como os tipos de Aprendizado de Máquina, Aprendizado Profundo ou *Deep Learning,* Redes Neurais Artificiais.

No capítulo 3 são apresentados o histórico, estudos e definições e conceitos a respeito dos *chatbots*, assim como seus diferentes tipos. O capítulo 4 trata sobre os trabalhos relacionados. O capítulo 5 trata sobre as metodologias utilizadas neste trabalho. No capítulo 6 é apresentado o processo de desenvolvimento dos *chatbots*, os aspectos de implementação, arquitetura utilizada, bibliotecas e parâmetros de treinamento dos modelos gerados. Durante o capítulo 7 são avaliados os resultados obtidos a partir dos *chatbots* criados durante o desenvolvimento deste trabalho.

# PROCESSAMENTO DE LINGUAGEM NATURAL

A comunicação conforme Russel e Norvig (1995, p. 888) é o processo de troca intencional de informações e intenções, realizado através de um sistema compartilhado de sinais. A maioria dos animais, incluindo os seres humanos utiliza estes sinais para transmitir mensagens importantes

Diversas áreas do conhecimento se propõem a estudar com diferentes abordagens a comunicação humana, o Processamento de Linguagem Natural (PLN) empresta conceitos do campo da Linguística e Linguística Teórica. O PLN é um campo orientado a utilização de técnicas computacionais para analisar e representar textos que ocorram em naturalmente a fim de obter o processamento de linguagem semelhante ao nível de um ser humano no seu dia a dia (LIDDY. 2001). O PLN também pode ser definido como um campo da Ciência Linguística capaz de caracterizar a explicar a multiplicidade de observações linguísticas a nosso redor como por exemplo conversas, escrita e outras mídias (MANNING; SCHÜTZE, 1999).

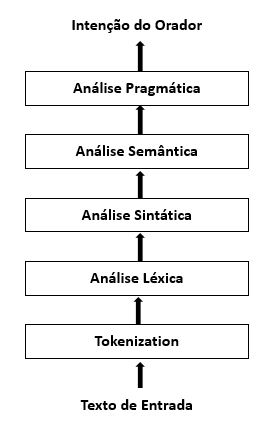
Em seu trabalho Cahn (2017, p.10) introduz os dois principais objetivos do PLN o processamento do output de sistemas de ASR (do inglês *Automated Speech Recognition*) para a produção de uma representação estruturada em forma de texto do idioma falado SLU (do inglês *Spoken Language Understanding)*, ainda segundo o autor, este hoje presente em diversas soluções tecnológicas como assistentes pessoais como a Siri da Apple, Google Now¹, entre outros. Cahn (2017, p.10) descreve que quando o input passado é textual se trata da NLU (do inglês *Natural Language Understanding*) onde são realizados diversos processos para extração de intenção e significado, com intuito de recuperar a real intenção do interlocutor. Além das áreas do PLN citadas por Cahn (2017) também cabe citar outras como Linguística de *corpus*, Representação do Conhecimento, Recuperação de Informações e Web Semântica. É importante ressaltar conforme Liddy (2001) que um sistema completo de NLU deve possuir as seguintes capacidades:

* Parafrasear um texto de entrada
* Traduzir o texto de entrada em outra linguagem
* Responder perguntas sobre o conteúdo do texto de entrada
* Extrair inferências do texto de entrada

Neste trabalho serão abordadas questões relacionadas a área de NLU, trazendo conceitos e técnicas do PLN aplicadas ao desenvolvimento de um agente conversacional ou *chatbot.* A Figura 1 ilustra um pipeline de técnicas e aplicadas para extração da intenção de uma frase, processo comumente utilizado no NLU. Um dos desafios do NLU é relacionado à sua aplicação em geral, quando exposto a erros gramaticais, defluências, interrupções e autocorreções (Cahn, 2017, p.10). Segundo Indurkhya e Damerau (2010) tradicionalmente a PLN aplicado a NLU realiza o processo de análise de linguagem decompondo em um número de estágios emprestados das distinções linguísticas entre análise Sintaxe, Semântica e Pragmática.

A análise de uma sentença passa por uma primeira etapa de análise em termos de sintaxe, este processo fornece a ordem e estrutura de forma mais acessível do que uma análise semântica ou em sentido literal, seguido pelo estágio de análise pragmática para determinar o significado do enunciado ou texto (INDURKHYA; DAMERAU, 2010, p. 4). Ainda segundo Indurkhya e Damerau (2010) a divisão da análise em três partes, sintaxe, semântica e pragmática serve apenas como um ponto de partida, quando estas técnicas são aplicas em cenários reais de PLN, sendo necessárias à aplicação de diferentes técnicas englobadas entre as análises e a utilização de uma granularidade mais baixa, como evidenciado em diversos trabalhos da área.

Figura Os estágios da análise em PLN



Fonte: Adaptado de Indurkhya e Damerau (2010)

De acordo com Russel e Norvig (1995, p. 860) “existem duas razões principais pelas quais queremos que nossos agentes computacionais possam processar linguagens naturais, primeiro, comunicar-se com humanos, [...] e segundo, adquirir informações de linguagem escrita. ” Isto fica evidenciado uma vez que a existem diversas iniciativas de pesquisa para modelagens de agentes conversacionais e representação de conhecimento. A capacidade destes agentes de reconhecer a linguagem natural é um ponto crucial para o alcance dos seus objetivos, Russel e Norvig (1995, p.861) relatam que o maior problema a se lidar quando tralhamos com Linguagem Natural é que estes modelos são extensos e mudam constantemente. A melhor estratégia segundo os autores seria a aproximação, sendo realizada por meio de uma distribuição probabilística. Liddy (2001) em seu trabalho cita diversos tipos de abordagens existentes na área de PLN, estas se diferenciando quando aplicadas a NLU ou SLU.

* Abordagem Simbólica

A abordagem simbólica é baseada na representação explicita da linguagem através de esquemas de representação e algoritmos de associação. Um exemplo de aplicação são os sistemas baseados em regras ou *lexicons*, um tipo de representação de palavras de uma linguagem. Entre suas principais aplicações estão extração de informação, categorização de texto, resolução de ambiguidade e aquisição lexical (LIDDY, 2001)

* Abordagem Conexionista

Abordagem conexionista também desenvolve modelos generalizados como a abordagem probabilística, sua principal diferença em relação ao modelo probabilístico é a utilização de várias teorias da representação, permitindo deste modo a inferência e manipulação dos modelos gerados através de formulas lógicas, sua aplicação está ligada a tarefas de PLN, como a análise sintática, tarefas de tradução de domínio limitado e recuperação associativa (LIDDY, 2001)

Segundo Manning e Schütze (1999, p. 24) o desenvolvimento da área da linguística e a pesquisa neste campo foi fortemente impactada pela disponibilidade de recursos textuais, como *corpus* e *corpora* por exemplo, ainda segundo os autores a definição de um *corpus* (do latim: corpo) é um conjunto de textos, e por sua vez um *corpora* é o conjunto de *corpus* (MANNING, SCHÜTZE, 1999, p. 37).

Apesar do positivo impacto das novas técnicas de PLN orientadas a abordagem probabilística a pouca diversidade nos *corpora* existentes para língua Português-Brasileiro afeta a pesquisa de agentes conversacionais utilizando abordagens probabilísticas. Neste trabalho serão utilizados os *word embeddings* uma técnica de representação de uma língua através de técnicas da abordagem probabilística como as Cadeias Ocultas de Markov (do inglês, *Hidden Markov Models*) (MIKOLOV et al, 2003), os *word embeddings* serão construídos a partir de um *corpus* de acesso público e utilizados para o treinamento de um agente conversacional construído com técnicas de RNA e seguindo o modelo *sequence-to-sequence* de (VINYALS; LE, 2015).

## Word Embeddings

O termo *Word embeddings* foi cunhado por Bengio et. al (2003) inspirado pela teoria da linguística Semântica Distribucional (do inglês *Distributional Semantics*) como um importante recurso na área de PLN. A representação distribucional de palavras foi proposta por Rumelhart (1986) e se tornou um recurso amplamente utilizado na pesquisa da área de PLN (MIKOLOV; LE, 2013 apud RUMERHART, 1986). A principal vantagem deste tipo de representação é a possibilidade de generalização de padrões de forma mais robusta, fornecendo assim uma maneira mais fácil de se lidar com entradas não esperadas e mesmo assim manter um nível de coerências semântica.

As “*Word Embeddings* são o mapeamento de palavras para vetores matemáticos” (Cahn, 2017, p. 19). Estes vetores mapeados possuem a capacidade de representar a relação entre palavras e até mesmo frases de uma língua, conforme citado por Levy, Goldberg e Dagan (2015, p.211) embora uma compreensão profunda, a nível de uma pessoa fluente seja ilusório, muitos destes métodos implementados são bem-sucedidos ao alcançar um nível satisfatório na recuperação de aspectos de semelhanças entre palavras. A Tabela 1 demonstra inferências e relações que podem ser extraídas a partir de uma *word embedding* treinada a partir de um *corpus* que expresse significantemente a linguagem estudada.

Tabela 1 Exemplo de Relações Semânticas Encontradas por um Word Embedding

|  |  |  |  |
| --- | --- | --- | --- |
| Relação | Exemplo 1 | Exemplo 2 | Exemplo 3 |
| França – Paris | Itália: Roma | Japão: Tóquio | Flórida: Tallahassee |
| Miami – Florida | Baltimore: Maryland | Dallas: Texas | Kona: Hawaii |
| Einstein – Cientista | Messi: Meio Campo | Mozart: Violinista | Picasso: Pintor |
| Sarkozy – França | Berlusconi: Italia | Merkel: Alemanha | Koizumi: Japão |
| Cobre – Cu | Zinco: Zn | Ouro: Au | Urânio: Plutônio |
| Berlusconi – Silvio | Sarkozy: Nicolas | Putin: Medvedev | Obama: Barack |
| Microsoft – Windows | Google: Android | IBM: Linux | Apple: Iphone |
| Microsoft – Ballmer | Google: Yahoo | IBM: McNealy | Apple: Jobs |
| Japão – Sushi | Alemanha: Bratwurst | França: Tapas | USA: Pizza |

Fonte: Adaptado de Mikolov et al. (2013)

Diversas abordagens baseadas na geração de *word embeddings* a partir de redes neurais foram propostas por diversos autores” (LEVY; GOLDBERG; DAGAN, 2015 apud BENGIO et al; COLLOBERT; WESTON, 2008, p. 211), estas abordagens utilizam-se de conceitos de redes neurais artificiais e aprendizado não supervisionado para a criação das matrizes de vetores sobre determinada língua. Os *word embeddings* no modo que são criados geram em sua representação uma forma de análise semântica, uma vez que vetores próximos tem relação um com o outro. Foi uma nova alternativa a modelos anteriores como os autores citam “A ideia de descobrir as similaridades entre palavras para se obter generalização de uma sequência para outra não é nova. Por exemplo é explorada na abordagem que se baseia em aprendizado em agrupamento de palavras. ” (BENGIO et al., 2003, p.1140-1141 apud BROWN et al., 1992, PEREIRA et al., 1993, NIESLER et al., 1998, BAKER e MCCALLUM, 1998).

A popularização da utilização dos *word embeddings* ocorreu através dos anos, sendo fortemente creditada ao trabalho de Mikolov et al. (2013) que criou o sistema word2vec, assim como Pennington et al. (2014) criador do GloVe, ambos sistemas que possibilitam a criação e utilização de forma mais simples destes recursos. Em diversas publicações foram constatadas a eficiência do modelo de *word embeddings* aplicado à PLN. “Foram conduzidos experimentos que comparam os *embeddings* gerados pelo word2vec aos métodos tradicionais de distribuição de palavras. Os resultados sugerem que os novos métodos superam uma margem não trivial em tarefas orientadas a similaridade. ” (LEVY; GOLDBERG; DAGAN, 2015 apud BARONI et al; 2014, p. 211).

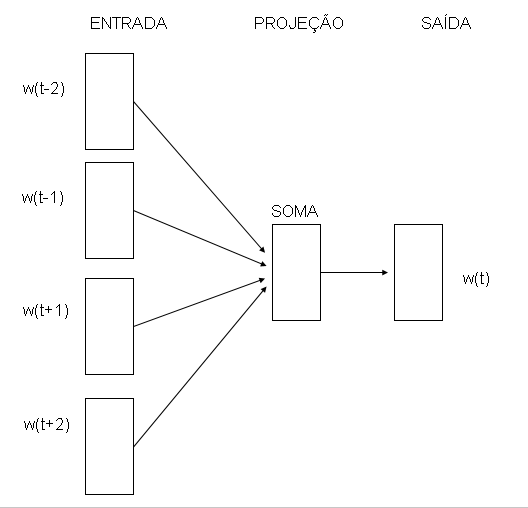
### CBOW

O ponto de partida desta sessão é a arquitetura a *bag-of-words* (BOW) a qual foi a primeira arquitetura base para o modelo *continuous bag-of-words* (CBOW), foi elaborada por Mikolov et al (2013) em seu trabalho “*Efficient estimation of word representations in vector space*” aonde foram propostos dois modelos de arquitetura para criação de *word embeddings*.

“No modelo CBOW, o objetivo de treinamento do modelo CBOW é combinar as representações de palavras adjacentes para prever a palavra no meio” (MIKOLOV; LE; SUTSKEVER, 2013, p.3), processo ilustrado pela Figura 2. O modelo CBOW ao contrário do modelo BOW cria uma representação distribuída continua do contexto observado.

Desta forma a matriz de peso entre a entrada e a camada de projeção é compartilhada entre todas as posições de palavras na mesma maneira que a arquitetura NNLM (MIKOLOV et al 2013). A arquitetura CBOW é mais leve e rápida devido as otimizações em relação à NLNN e BOW e, portanto, é mais adequada na utilização do processamento de grandes *datasets*.

Figura A Arquitetura CBOW



Fonte: Adaptado de Mikolov et al. (2013)

### SKIP-GRAM

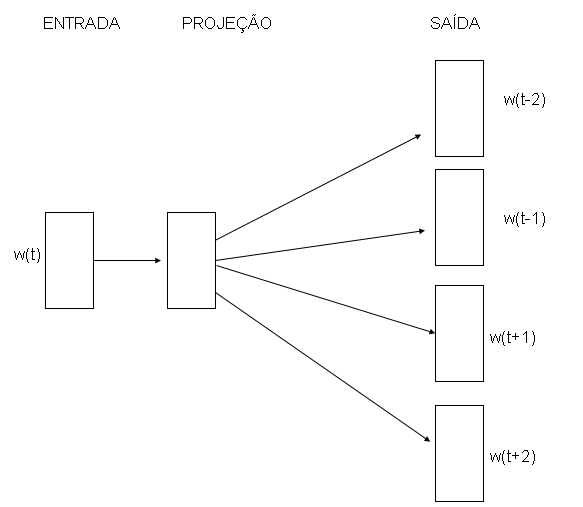
A segunda arquitetura proposta por Mikolov et al (2013) é a *Continuous* Skip-Gram, similar ao modelo CBOW, porém ao invés de tentar predizer a próxima palavra baseada no contexto ela tenta maximizar a classificação baseada em outra palavra da mesma sentença, processo ilustrado pela Figura 3.

“No modelo *skip-gram*, cada palavra w ∈ W é associada a um vetor vw ∈ Rd e similarmente cada contexto c ∈ C é representado como um vetor vc ∈ Rd, onde W é o vocabulário de palavras, C é o vocabulário de contextos e d é a dimensionalidade de incorporação” (LEVY; GOLDBERG, 2014, p. 2). O modelo *skip-gram* tenta predizer a próxima palavra de uma sentença considerando a série de palavras anteriores e posteriores a palavra atual.

A principal descoberta deste modelo segundo Mikolov et al (2013) é de que quando expandindo o alcance desta série de palavras é aumentada a qualidade dos vetores de palavras geradas, porém também elevando seu custo computacional.“A complexidade de treinamento dessa arquitetura é proporcional a Q = C × (D + D × log2 (V)) onde C é a distância máxima das palavras. ” (MIKOLOV et al, 2013, p. 4).

Deste modo para quanto maior o número da faixa de alcance utilizada maior o número de elementos a serem classificados. Quando treinados em grandes *datasets* estes modelos (CBOW e *skip-gram*) capturam uma substancial quantidade de relações semânticas (MIKOLOV; LE; SUTSKEVER, 2013).

Figura A Arquitetura SKIP-GRAM.



Fonte: Adaptado de Mikolov et al. (2013)

## Part-of-Speech Tagging

“O *Part-of-Speech Tagging* (POS) é uma técnica que aplica um rótulo gramatical a cada palavra dado uma sentença de entrada" (CAHN, 2017, p. 13-14), no processo de POS *Tagging* são geradas as POS *Annotation*, aonde para cada palavra da frase de entrada é atribuída sua classe gramatical (substantivo, verbo, adjetivo, etc.) a partir de sua definição e contexto da frase.

O POS, também conhecido como categorias sintáticas, é aplicado para extrair informações importantes sobre uma palavra e seus vizinhos em uma frase (JURAFSKY; MARTIN, 2008). A partir do processo de POS Tagging são aplicadas as *POS Annotations*, marcações ou *tags* em cada palavra de uma sentença em ordem de identificar corretamente a classe gramatical das palavras no texto considerando seu contexto.

As POS *Annotations* são uteis também para a realização de generalizações podendo ser agrupadas em categorias léxicas para a formação de categorias sintáticas, como por frase nominal, verbal, etc. A combinação destas categorias sintáticas em árvores que representam a estrutura de frases em categorias (RUSSEL; NORVIG, 1995, p. 888).

O processo de aplicação das *part-of-speech annotations* se dá na etapa de pré-processamento, contido nas etapas de análise léxica e sintática, este processo pode ser realizado tanto manualmente quanto automaticamente, por programas chamados de POS *Taggers*. Os POS *Taggers* são desenvolvidos para atender as regras de uma linguagem especifica e são aplicados a sentença contidas em um *dataset* ou *corpora*. Segundo Indurkhya e Damerau (2010) alguns autores realizam uma validação manual das *annotations* aplicadas pelos *Taggers*.

A técnica de POS é aplicada em diversos cenários, podendo auxiliar em tarefas de extração de informação (do inglês, *Information Retrieval*), algoritmos de *stemming*, sumarização de textos e até mesmo em sistemas de *text-to-speech.*

Este processo traz diversos ganhos uma vez que sinalizando a classe gramatical de uma palavra elimina ou reduz duplicidade de sentidos, melhorando a extração de significado nas etapas posteriores do PLN. Um dos *taggers* mais utilizados para criação das POS *Annotations* em textos em português é o PALAVRAS, a Tabela 2 mostra as *tags* utilizadas por este programa.

Tabela 2 Exemplo de Tags POS Tagger PALAVRAS

|  |  |  |
| --- | --- | --- |
| Nome | Classe gramatical | Indicações de subclasse |
| N | Substantivo | Prop |
| PROP | Nome próprio | kc (nomes próprios que incluem o caracter &) |
| SPEC | Especificador (pronome ou adjetivo) | rel, interr, dem, quant, diff |
| DET | Determinante (artigo, pronome ou adjetivo) | artd, arti, quant, dem, poss, diff, rel, ident, interr |
| PERS | Pronome pessoal | refl, obj, coll, reci |
| ADJ | Adjetivo | NUMord, prop, KOMP, n, mente |
| ADV | Advérbio | rel, quant, kc, ks, prp, interr, co-vfin, prop, dem, co-fmc, parkc-2\* |
| V | Verbo | n, prop, fmc, quant |
| NUM | Numeral | card, prop |
| KS | Conjunção subordinativa |  |
| KC | Conjunção coordenativa | co-fmc, co-vfin, co-inf, co-pcv, co-prparg, co-ger, co-advl, co-acc, co-sc, co-pred, co-app, co-subj, co-postnom, parkc-1\*, parkc-2\* |
| PRP | Preposição |  |
| IN | Interjeição |  |
| EC | Prefixo |  |

Fonte: Adaptado de Linguateca (2015)

## Stemming

A técnica de *Stemming* ou Stemização foi proposta por LOVINS (1968) a qual definiu o algoritmo como um meio de agrupar palavras a uma mesma raiz gramatical ou termo indexador, de modo a melhorar o desempenho de tarefas relacionas a recuperação de informação em PLN. Neste primeiro trabalho foi criado um algoritmo para à língua Inglesa e proposto a criação de iniciativas para outras linguagens, tendo em vista que esta técnica não seria restrita apenas ao inglês.

O mapeamento de palavras pelo *Stemmer,* algoritmo de *Stemming*, ocorre em iterações de palavra por palavra, não analisando palavras próximas ou sentido da frase em que está contida, o algoritmo remove sufixos e prefixos da palavra e a agrupa pelo seu termo de indexação (LOVINS, 1968).

De acordo com os autores Indurkhya e Damerau (2010) o *Stemming* está entre as técnicas de pré-processamento mais utilizadas em PLN, sendo principalmente aplicada a algoritmos de *Information Retrieval* (IR). Um dos trabalhos que mais impactou esta área foi o *stemmer* desenvolvido por Porter (1980) o qual foi amplamente utilizado processo de stemming é utilizado para reduzir palavras flexionadas a seu tronco (*stem*) um exemplo demonstrado pelos autores Indurkhya e Damerau (2010) seria a unificação de diferentes variações morfológicas de uma palavra através de sua raiz gramatical, por exemplo as palavras **Computação** e **Computador** seriam representadas por **Comput.**

Este tipo de abordagem é comumente aplicado na indexação de documentos, aonde múltiplos termos de pesquisa podem levar a um mesmo recurso contendo as informações desejadas, independente da utilização exata do termo que consta no documento. Este agrupamento de palavras é de longe o procedimento de normalização mais utilizado em sistemas de IR aumentando a eficiência da recuperação de informação quando os *stems* ou termos indexadores agrupam corretamente palavras e expressões que carregam o mesmo sentido ou ideia (INDURKHIA; DAMERAU, 2010).

Devido a capacidade de representação das *word embeddings* e do modelo *sequence-to-sequence,* as técnicas de *stemming* não são utilizadas na criação de *chatbots* pois geram a perda de estrutura sintática e semântica aprendidas pelo agente conversacional quando treinados com textos contendo todas as flexões e variações gramaticais.

## Lemmatization

A técnica de *Lemmatization ou* Lematização é similar a já mencionada técnica de *stemming*, onde o algoritmo de lemmatization busca agrupar diferentes flexões e variações de uma palavra por meio de um *lemma ou Lexeme,* mas diferente do stemming esta técnica considera as palavras vizinhas ou até mesmo o contexto inteiro para a aplicação da tag. Deste modo um lemmatizer aplica os respectivos *lemmas* para cada palavra considerando a intenção da oração.

De acordo com Indurkhya e Damerau (2010) lemmatization pode ser utilizada em diversas etapas do PLN como análise léxica, morfossintática, semântica, *Machine Translation, Information Retrieval* sendo também aplicada ao pré-processamento de textos.

A utilização de técnicas de *lemmatization* traz benefícios quando aplicadas a problemas de IR, porém em alguns casos esta pode gerar problemas devido a perda de informações durante a conversão das palavras para seu respectivo *lemma*(MANNING; SCHÜTZE, 1999).

Deste modo considerando as capacidades da técnica de *lemmatization* foi constatado que sua aplicação não é indicada no pré-processamento de *corpora* para utilização por *chatbots* baseados em *Deep Lerning,* uma vez que o modelo criado a partir destas arquiteturas possui a capacidade de aprender a reproduzir verbos flexionados corretamente, quando treinado com um *dataset* representativo o suficiente, como demonstrado até mesmo em sua primeira aplicação durante o trabalho de Vinyals e Le (2015).

## Tokenization

A Tokenização (do inglês *Tokenization*) ou *word segmentation* é o processo de separar em uma sentença em elementos individuais cada uma de suas partes, por exemplo palavras, pontuações, números, datas, etc. (CAHN, 2017). Esta técnica separa os elementos de uma sentença em unidades chamadas de *tokens,* utilizando geralmente validações criadas em forma de funções para desconsiderar elementos indesejados como por exemplo espaço em branco a maioria dos programas de *Tokenization* permitem a escrita de regras customizadas, podendo remover pontuações ou utilizas como *tokens* (JOST, 2015; MANNING; SCHÜTZE, 1999).

A técnica de *Tokenization* está presente na execução de diversos outros tipos de algoritmos, sendo uma etapa de algoritmos como *Part-of-Speech Tagging*, criação de *word embeddings, Text-to-Speech,* processos de segmentação, análise de textos, entre outros sendo essa uma das técnicas mais utilizadas no pré-processamento de textos (CAHN, 2017). Segundo Manning e Schütze (1999) para a escrita de um tokenizador ou *tokenizer* é necessário um grande esforço para criação de regras manuais, necessitando de conhecimento da linguagem para a qual o tokenizador foi feito.

Conforme Indurkhya e Damerau (2010) o processo de Tokenização é um passo crucial para o fluxo de processamento da PLN, ilustrado pela Figura 1 Os estágios da análise em PLN, uma vez que os dados de entrada dos algoritmos de PLN não costumam ser normalizados em textos curtos, com correta formatação, sentenças bem delimitadas, etc. Por este motivo a técnica de Tokenização tem um impacto enorme em outras técnicas, sendo capaz de normalizar um texto de forma que outros tipos de análises e algoritmos sejam executados a partir de dados normalizados.

Durante a pesquisa de referencial teórico deste trabalho foram encontradas diversas técnicas que se apoiam na Tokenização e algoritmos que utilizam a Tokenização como etapa inicial de preparação dos dados para processamento. A Tokenização está contida nos processos de criação de *word embeddings* (MIKOLOV et al, 2013) e presente em diversos trabalhos que aplicam PLN para construção de *corpus* ou recursos linguísticos.

# APRENDIZADO DE MÁQUINA

O Aprendizado de Máquina (AM, do inglês *Machine Learning*) é uma das áreas da computação que mais inspira a reflexão de como os computadores podem evoluir por meio do aprendizado por experiências e como a sua utilização seria afetada pela capacidade de aprendizado.

De acordo com Mitchell (1997) se pudéssemos entender como programar um sistema para aprender e aprimorar-se automaticamente por meio de sua experiência o impacto seria dramático. O responsável por cunhar o termo *Machine Learning* foi Arthur Lee Samuel (SAMUEL, 1959) aonde no primeiro trabalho a utilizar a expressão demonstrou a construção de um programa capaz de aprender e se aprimorar por meio da experiência a sua habilidade em jogar o jogo damas. O potencial do programa era que por meio do conhecimento das regras do jogo, um senso de direção e uma incompleta lista de parâmetros com pouco tempo de treinamento o programa se tornava mais habilidoso que seu autor no jogo de damas (SAMUEL, 1959).

Apesar do conhecimento desta área ainda não possuir as técnicas e tecnologias necessário para a criação de uma inteligência artificial generalista, que obtivesse a capacidade de aprender e aprimorar habilidades em qualquer domínio, atualmente é possível empregar diversos algoritmos e técnicas para um aprendizado efetivo de certos tipos de tarefas (MITCHELL, p. 1, 1997).

O aprendizado de máquina essencialmente é um campo multidisciplinar, herdando conceitos e resultados de diversos outros campos conforme a Tabela 3.

Tabela 3 Áreas de Conhecimento e sua influência no Aprendizado de Máquina

|  |  |
| --- | --- |
| Inteligência Artificial | Aprendizado simbólico de conceitos. Aprendizado de máquina com um problema de busca. Aprendizagem como uma abordagem para melhorar a resolução de problemas. Usando conhecimento prévio junto com dados de treinamento para guiar o aprendizado. |
| Métodos Bayesianos | Teorema de Bayes como base para calcular probabilidades de hipóteses. O classificador *Naive* Bayes. Algoritmos para estimar valores de variáveis ​​não observadas. |
| Teoria da Complexidade Computacional | Limites teóricos sobre a complexidade inerente de diferentes tarefas de aprendizagem, medidas em termos de esforço computacional, número de exemplos de treinamento |
| Teoria do Controle | Procedimentos que aprendem a controlar processos para otimizar objetivos pré-definidos e que aprendem a prever o próximo estado do processo que estão controlando. |
| Teoria da Informação | Medidas de entropia e conteúdo informativo. Comprimento de descrição mínima como abordagem para aprendizagem. Códigos otimizados e sua relação com sequências de treinamento ideais para codificação de uma hipótese. |
| Filosofia | A navalha de Occam, sugerindo que a hipótese mais simples é a melhor. Análise da justificativa para generalização além dos dados observados. |
| Psicologia e Neurobiologia | A lei de poder da prática, que afirma que, em uma gama muito ampla de problemas de aprendizagem, o tempo de resposta das pessoas melhora com a prática de acordo com uma lei de poder. Estudos neurobiológicos motivando modelos de redes neurais artificiais de aprendizagem. |
| Estatística | Caracterização de erros (por exemplo, viés e variância) que ocorrem ao estimar a precisão de uma hipótese com base em uma amostra limitada de dados. Intervalos de confiança, testes estatísticos. |

Fonte: Adaptado de Mitchell. (1997)

O conceito central do AM é a utilização de experiências anteriores para o aprimoramento do programa que a implementa. No primeiro trabalho relacionado a área de AM Mitchell (1997) define que um computador ou programa é dito aprender a partir de suas experiências anteriores, em relação a uma tarefa pré-determinada aonde através do acréscimo de sua experiência é aumentada sua capacidade de resolução de problemas.

Entre as técnicas e diferentes abordagens existentes entre as subáreas do AM todas elas estão direcionadas a melhoria da *performance* em uma determinada tarefa. Para que os programas de AM aumentem sua *performance* são utilizados algoritmos de treinamento para extrair conhecimento na resolução de problemas de um determinado domínio a partir de um grande conjunto de dados sobre este domínio, conhecido como *dataset*. O treinamento pode ocorrer utilizando diversas diferentes técnicas tanto para modelagem do sistema de aprendizado como para processamento dos dados a serem utilizados para treinamento, sendo que a quantidade de dados e o modo que estes serão pré-processados e tratados tem um impacto essencial no resultado final.

Entre as áreas do AM existem o Aprendizado Supervisionado, Aprendizado Não Supervisionado, Aprendizado Por Reforço e Aprendizado Semi-Supervisionado, sendo que o foco de todos os tipos de AM está em extrair padrões e *insights* da experiência contida nos dados dos *datasets* de treino. Determinados tipos de AM podem aprender inclusive com os erros obtidos durante e depois da fase de treinamento do algoritmo.

As tarefas abordadas pelo AM geralmente são tarefas complexas demais para serem resolvidas via algoritmos clássicos com regras fixas, criadas manualmente, um bom desempenho na realização de uma determinada tarefa é o objetivo do treinamento no AM, sendo avaliada por meio de métricas para avaliação do desempenho do algoritmo. Conforme a Figura 4 pode-se verificar diversos algoritmos de AM agrupados por categorias, demonstrando a variedade de técnicas que são encontradas na literatura científica.

Figura Algoritmos de Machine Learning



Fonte: Adaptado de Machine Learning Porto Alegre. (2017)

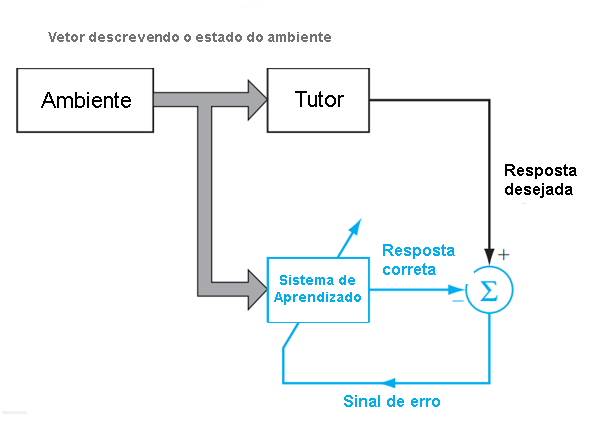
Apesar da crescente adoção de soluções utilizando conceitos e tecnologias de AM é observado que “Ainda que AM seja uma ferramenta poderosa para a aquisição automática de conhecimento, deve ser observado que não existe um único algoritmo que apresente o melhor desempenho para todos os problemas. ” (MONARD; BARANAUSKAS, 2003, p. 39). É importante desta forma ressaltar que esta tecnologia amadureceu muito através dos anos, já possuindo aplicações comerciais que utilizavam AM na década de 90, as quais já naquela época superavam métodos tradicionais de programação em determinadas áreas, como por exemplo reconhecimento de fala (MITCHELL, p.1 ,1997).

## Aprendizado Supervisionado

No Aprendizado Supervisionado (do inglês: *Supervised Learning*) todos os dados de treinamento são rotulados, assim são criados modelos estatísticos para prever as próximas ocorrências de determinados padrões. O modelo de treinamento supervisionado se utiliza de informações rotuladas, geralmente um problema conhecido aonde dado um conjunto de dados sabemos a saída correta, preparando assim o algoritmo a gerar uma saída semelhante aquelas encontradas no *dataset* de treinamento.

“No aprendizado supervisionado é fornecido ao algoritmo de aprendizado, ou indutor, um conjunto de exemplos de treinamento para os quais o rótulo da classe associada é conhecido. ” (MONARD; BARANAUSKAS, 2003, p. 40). A Figura 5 ilustra o funcionamento do AM supervisionado. "Conceitualmente podemos considerar o tutor em posse do conhecimento do ambiente, com este conhecimento sendo representado como um vetor de exemplos entrada-saída. ” (HAYKIN, 2009, p. 34) O tutor ou indutor é responsável por apontar os erros cometidos pelo modelo durante o treinamento, deste modo o sistema pode aprender também com os erros de classificação, ajustando os parâmetros na busca de um melhor resultado em suas predições.

Figura Aprendizado Supervisionado



Fonte: Adaptado de Haykin. (2009)

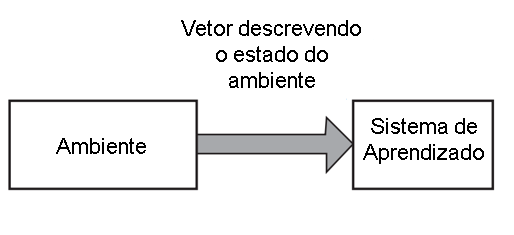
## Aprendizado Não-Supervisionado

“No Aprendizado Não-Supervisionado (do inglês: *Unsupervised Learning*), ou Aprendizagem auto organizada, não há professor externo ou crítico para supervisionar o processo de aprendizagem. ” (HAYKIN, 2009, p. 37). Nesta abordagem dada uma massa de dados não rotulados o sistema deverá extrair informações, padrões ou agrupar estes dados por proximidade.

“Uma vez que a rede esteja sintonizada com as regularidades estatísticas dos dados de entrada, a rede desenvolve a capacidade de formar representações internas para codificar características da entrada e, assim, criar novas classes automaticamente” (HAYKIN, 2009 apud BECKER, 1991, p. 37). Esses agrupamentos são formados conforme o algoritmo descobre semelhanças e diferenças entre os conjuntos de dados analisados, sendo assim possível derivar conclusões com o conhecimento gerado.

Esta abordagem possibilita a análise de casos aonde não se sabe qual a relação entre as variáveis do conjunto, podendo ser extraído conhecimento para replicá-la. Pode também ser utilizada para a simplificação da análise de um problema maior em problemas menores, aonde o algoritmo se concentra apenas nos atributos mais relevantes para a análise do problema. “Já no aprendizado não-supervisionado, o indutor analisa os exemplos fornecidos e tenta determinar se alguns deles podem ser agrupados de alguma maneira” (MONARD; BARANAUSKAS, 2003, p. 40), processo que pode ser visto na Figura 6.

Figura Aprendizado Não-Supervisionado



Fonte: Adaptado de Haykin. (2009)

## Aprendizado Semi-Supervisionado

O Aprendizado Semi-Supervisionado (do inglês: *Semisupervised Learning*) utiliza dados rotulados e não rotulados em seu treinamento (HAYKIN, 2009). “Essa nova forma de aprendizado, denominada aprendizado semi-supervisionado, é bastante útil quando apenas um pequeno número de exemplos rotulados encontra-se disponível, o que ocorre na maioria dos casos ” (SANCHES, 2003, p. 19).

É importante ressaltar que o aprendizado semi-supervisionado pode ser utilizado tanto em tarefas de classificação quanto em tarefas de *clustering* (SANCHES, 2003, p. 19). A principal motivação para a criação e utilização desta técnica de aprendizado é pelo fato da disponibilidade de informações rotuladas quando comparadas as não rotuladas, somado ao custo de rotulação dos dados na maioria dos casos estudados por pesquisadores.

Outra motivação seria a baixa assertividade de classificadores treinados por apenas uma pequena massa de dados rotulada (SANCHES, 2003). "A ideia do aprendizado semi-supervisionado é então utilizar os exemplos rotulados para se obter informações sobre o problema e utilizá-las para guiar o processo de aprendizado a partir dos exemplos não rotulados." (SANCHES, 2003 apud BRUCE, 2001, p. 50)

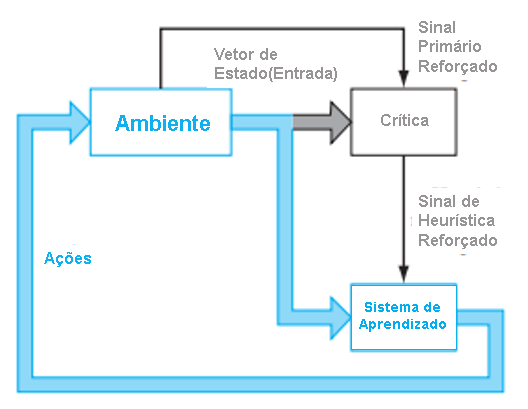
## Aprendizado por Reforço

“Na Aprendizagem por Reforço (do inglês *Reinforcement Learning*), a aprendizagem de um mapeamento de entrada-saída é realizada através da interação contínua com o ambiente, a fim de minimizar um índice escalar de desempenho” (HAYKIN, 2009, p. 36). O Aprendizado por Reforço é considerado desde os anos iniciais uma técnica com grande potencial, tendo em vista sua tentativa de aproximação aos paradigmas de aprendizado humanos (MITCHELL, 1997).

“O sistema é projetado para aprender sob reforço atrasado, o que significa que o sistema observa uma sequência temporal de estímulos também recebidos do ambiente, que eventualmente resultam na geração de um sinal de reforço heurístico” (HAYKIN, 2009, p. 36). O Aprendizado por reforço trata o problema de encontrar ações adequadas dada uma situação em ordem de maximizar a recompensa ganha. Neste tipo de aprendizado a rede não recebe os resultados desejados, como no Aprendizado Supervisionado, neste caso o algoritmo deve descobrir os resultados esperados por meio de tentativa e erro (BISHOP, 2006, apud SUTTON e BARTO, 1998, p. 3).

A principal intenção desta técnica, demonstrada pela Figura 7, é de analisar o impacto dos estágios iniciais de uma sequência de etapas na solução de um problema, não descartando sequências que possam parecer ineficientes na solução do problema em seus primeiros passos. “O objetivo da aprendizagem por reforço é minimizar a função custo-a-passo, definida como a expectativa do custo acumulado de ação em uma sequência de etapas, em vez de simplesmente o custo imediato. ” (HAYKIN, 2009, p. 37).

Figura Aprendizado por Reforço



Fonte: Adaptado de Haykin. (2009)

.

## Redes Neurais Artificiais

Segundo Haykin (2009) “uma rede neural é um processador paralelo distribuído formado por simples unidades com uma propensão natural para armazenar conhecimento experiencial e torná-lo disponível para uso.”. Ainda conforme o autor as Redes Neurais Artificiais (RNA, do inglês: *Artificial Neural Networks*) mantém o seu conhecimento através das forças de conexão interna, os pesos sinápticos.

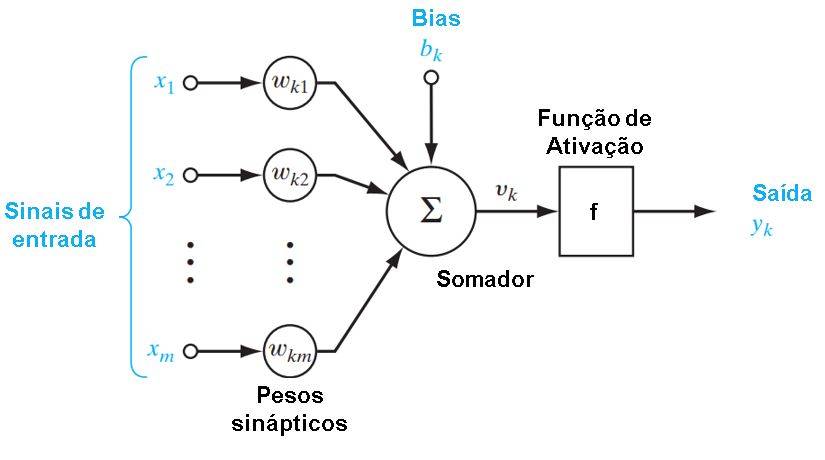
As RNA tentam simular o comportamento do neurônio biológico, inspiradas na estrutura e no funcionamento do cérebro humano. As redes neurais são compostas por nós ou unidades e ligados por diferentes conexões. Uma conexão entre uma unidade e outra serve para propagar a ativação de um neurônio a outro. Cada conexão ativada possui um peso associado que determina a força e sinal da conexão (RUSSEL; NORVIG, 1995). As principais capacidades destes modelos segundo (GOLDSCHMIDT, 2010) são que “Estes modelos têm capacidade de adquirir, armazenar e utilizar conhecimento experimental e buscam simular computacionalmente habilidades humanas tais como aprendizado, generalização, associação e abstração. ”.

“As primeiras informações sobre neurocomputação surgiram em 1943, em artigos do neurofisiologista Warren McCulloch, do MIT, e do matemático Walter Pitts da Universidade de Illinois” (GOLDSCHMIDT, 2010) estes autores são os responsáveis pelo primeiro modelo de neurônio artificial, o qual pode ser observado na Figura 8.

Haykin (2009) define um neurônio artificial como uma unidade de processamento de informações fundamental para a operação de uma rede neural, possuindo as quatro características básicas conforme visto na Figura 8.

* As **sinapses,** são um conjunto de conexões no qual cada conexão é caracterizada por um peso, representadas na Figura 8 por **wk1, wk2, wkm**. Um sinal ao entrar por uma sinapse é multiplicado pelo peso sináptico da conexão da sinapse com o neurônio.
* Um **somador** para somar os sinais de entrada, ponderados pelas respectivas forças sinápticas do neurônio; as operações descritas aqui constituem um combinador linear, representado por ∑.
* Uma **Função de Ativação** para limitar a amplitude de saída do neurônio e introduzir não-linearidade, representado por **f**.
* O **Bias** é responsável por aumentar ou abaixar o valor de entrada da função de ativação, é um elemento responsável por aumentar o grau de liberdade dos ajustes dos pesos sinápticos, representado por **bk**.

Figura Modelo de neurônio artificial por Haykin

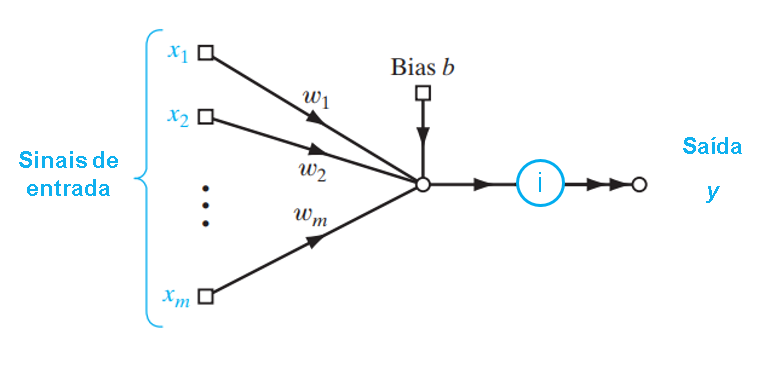


Fonte: Adaptado de Haykin. (2009)

O modelo do neurônio artificial de McCulloch–Pitts inspirou a criação de outros modelos de neurônios artificiais conforme citado por Spanhol “Madsen e Adamatti (2011) constatam que o modelo inspirou outros padrões neuronais, como de Redes *Perceptron”* (SPANHOL, 2017, p.27 apud MADSEN; ADAMATTI, 2011). O comportamento inteligente de uma RNA é formado através das conexões entre os neurônios de uma RNA e seu processo de aprendizagem ocorre através do ajuste dos pesos de conexão.

O algoritmo de aprendizado do *Perceptron* é motivado pela minimização da função de erro, aonde são percorridos os dados de treinamento e para cada padrão avaliado é feita uma verificação, se for classificado corretamente permanece inalterado, caso classifique incorretamente é aplicado o algoritmo de aprendizado para a correção de erros (BISHOP, 2006). “Rosenblatt demonstrou que, se fossem acrescentadas sinapses ajustáveis, as RNAs de McCulloch e Pitts poderiam ser treinadas para classificar padrões em classes linearmente separáveis, convergindo em um número limitado de passos (POMMERANZENBAUM, 2014, p. 10). A utilização do gradiente descendente é utilizada para a busca do vetor de pesos sinápticos que melhor se ajuste ao conjunto de dados de treinamento da RNA. “O exemplo mais antigo de modelo de rede neural é o modelo *perceptron*. Tem como objetivo classificar corretamente o conjunto de estímulos aplicados externamente à rede em uma de duas classes. ” (GOLDSCHMIDT, 2010, p.88). O *Perceptron* é considerado como o modelo mais simples de RNA *feedforward* existente, sendo um classificador linear que mapeia suas entradas para o estado final de verdadeiro ou falso, a Figura 9 demonstra o funcionamento da arquitetura *Perceptron* aonde x1, x2, xm são os sinais de entrada, w1, w2, wm são os pesos sinápticos b é o *Bias e i*  é o limitador, a função responsável por aplicar a regra para conversão do valor cálculo através da função de ativação e bias para um valor binário verdadeiro ou falso. Segundo Mitchell (1997) redes perceptron podem representar todos os tipos de valores booleanos primitivos AND, OR, NAND, NOR.

Figura Perceptron



Fonte: Adaptado de Haykin (2009).

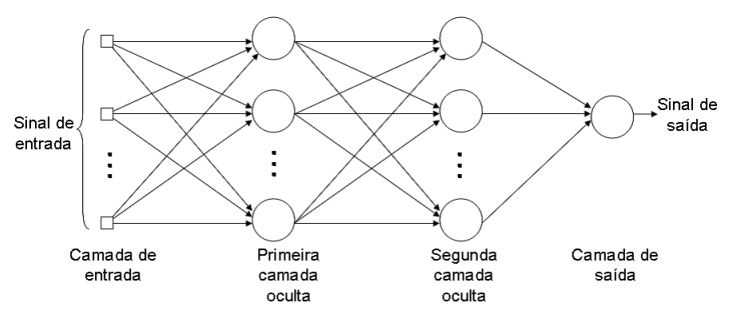
“As redes MLP (*Multi-Layer Perceptron*), portanto, representam uma generalização do *Perceptron* de camada única, possuindo uma ou mais camadas intermediárias (escondidas) de processamento” (POMMERANZENBAUM, 2014, p. 10).

Figura - Multi-Layer Perceptron

Fonte: Adaptado de Pommeranzenbaum (2014).

As RNA do tipo MLP, que pode ser observado na Figura 10, utilizam o popular algoritmo de treinamento *Backpropagation*, o qual utiliza pares de entrada e saída para por meio da correção de erros, ajustar os pesos sinápticos da rede com o intuído de reduzir os erros (GOLDSCHMIDT, 2010), ainda segundo Goldschmidt (2010) durante o aprendizado, ele processa em duas fases: a propagação e a retro-propagação. Na propagação o sinal flui da esquerda para a direita e na retro-propagação, com o erro calculado, o algoritmo ajusta os pesos das conexões da direita para a esquerda.

Segundo Haykin (2009) os maiores benefícios das RNA são: massiva arquitetura paralela, capacidade de generalização, não-linearidade, mapeamento de entrada-saída, adaptabilidade, resposta evidencial, informação contextual, tolerância a erros, Uniformidade de Análise e sua arquitetura*.*

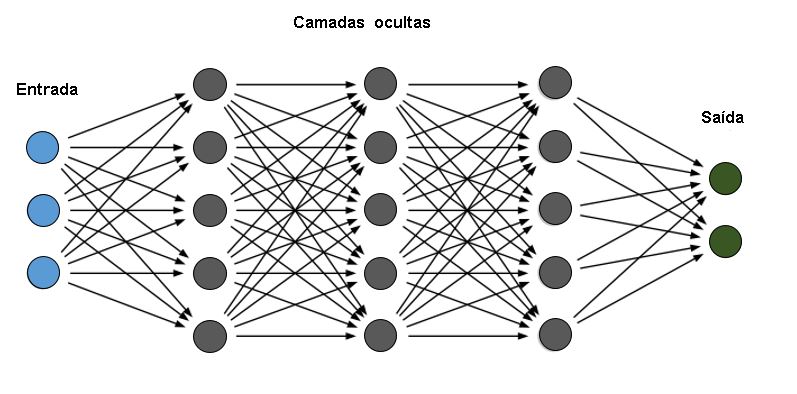
## Deep Learning

O Aprendizado Profundo (DL, do inglês *Deep Learning*) é uma subárea do AM conforme Deng e Yu (2014), citados por Jost (2015), o DL pode ser considerado como uma subárea devido a:

* Utilização de várias camadas de informação não-lineares para extração de características, transformação e análise de padrões.
* Uso de algoritmos de aprendizado de múltiplos níveis de representação para identificar relações entre os dados através de modelos estatísticos, normalmente com o uso de redes neurais.

O principal diferencial do DL em relação ao AM é a utilização de RNAs com maior número de camadas ocultas, conforme ilustrado pela Figura 11, e sua capacidade de processar dados não normalizados e/ou em sua forma bruta, como imagens ou texto, diferentemente de algoritmos de AM tradicionais, que necessitam de modelagem dos dados para treinamento, com etapas de normalização e seleção de *features* ou características relevantes para a solução do problema.

Figura Deep Neural Network



Fonte: Autor (2018).

O *Deep Learning* se diferencia pela aprendizagem de representação que é um conjunto de métodos que permite que uma máquina seja alimentada com dados brutos e descubra automaticamente as representações necessárias para detecção ou classificação destes dados (LECUN; BENGIO; HINTON, 2015). Desta forma o algoritmo de DL realiza a seleção de *features* automaticamente a partir do conjunto de dados de treinamento, desta forma é gerado um grande impacto positivo no processo de construção de aplicações que utilizam DL devido ao fato de os profissionais de ciência de dados se destinarem exclusivamente aos aspectos técnicos do problema e não a aspectos de conhecimento específico da área abordada.

A área do DL possui diferentes abordagens que buscam identificar e selecionar características ou *features* dos dados, possibilitando diferentes formas de representações destes, reduzindo também sua dimensionalidade (JOST, 2015 apud ARNOLD, 2013). Devido aos algoritmos de DL serem alimentados por dados em sua forma bruta são gerados durante o processo de aprendizado formas de representação para os dados de entrada, deste modo é diminuída a dimensionalidade dos dados processados assim como otimizado os recursos computacionais utilizados na construção dos modelos.

A cada camada a granularidade dos aspectos analisados se torna mais fina gerando uma análise mais completa e resultados próximos aos de um ser humano em tarefas triviais. Ainda segunda Cahn (2017, p.17). A utilização do DL é amplamente difundida no aprendizado não-supervisionado devido a sua capacidade de suas camadas lidarem com problemas cada vez mais complexos. O Deep Learning é aplicado também ao AM supervisionado para atividades como as de classificação por exemplo, para se obter o melhor resultado a partir dos dados de treinamento (LECUN; BENGIO; HINTON, 2015).

As aplicações de Deep Learning vêm definindo novos padrões de qualidade em diversas áreas, ultrapassando os resultados alcançados com técnicas de AM, redefinindo o estado da arte em tarefas tais PLN, Chatbots de domínio aberto, aprendizado de sequências, criação de sistemas de *Part-of-Speech* *Tagging* a partir de dados rotulados, modelos generativos, reconhecimento facial, de objetos e fala (FERREIRA, 2017, INDURKHYA; DAMERAU, 2010, LECUN; BENGIO; HINTON, 2015, VINYALS; LE, 2015, WANG; RAJ, 2017).

### Redes Neurais Convolucionais

As Redes Neurais Convolucionais (CNN, do inglês *Convolutional Neural Network*) foram aplicadas pela primeira vez com sucesso em 1989 no treinamento de DNN capazes de reconhecer imagens com um bom nível de generalização, para atingir este objetivo Lecun et al (1989) criou uma arquitetura de rede baseada no *Multi-Layer Perceptron* treinada com algoritmo de *backprogation*.

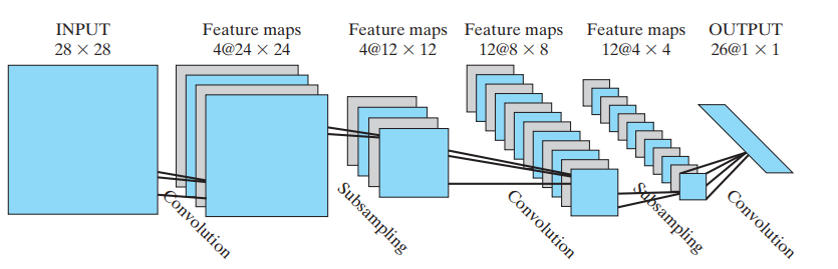
Segundo Bishop (2006) as redes convolucionais são amplamente aplicadas a detecções de dados a partir de imagens devido a sua capacidade de criar modelos que possuam características invariantes, que mesmo aplicando uma transformação sobre os dados, isto se traduz na rede por meio da construção de propriedades de invariância estrutural dentro do modelo proposto. Uma rede convolucional é formada por camadas onde as unidades da rede são organizadas em planos chamados de *feature map,* a qual recebe como entrada apenas uma pequena amostra da variável analisada, quando aplicada a imagens isto se traduziria por receber apenas uma pequena região de *pixels* com cada plano analisando parte da imagem compartilhando os pesos sinápticos (BISHOP, 2006), ilustrado pela Figura 12.

As redes neurais convolucionais são formadas por três mecanismos

* *local receptive fields*
* Compartilhamento de pesos
* Sub-sampling

Cada unidade de um *feature map* recebe entradas igualmente divididas de sub-regiões de uma imagem, deste modo todas as unidades buscam pelo mesmo padrão em diferentes localizações da imagem, usando pesos compartilhados, deste modo é gerada a invariância necessária para a generalização de atividades reconhecimento (BISHOP, 2006). Cada camada oculta alterna entre realizar a etapa de convolução e *sub-sampling* seguindo a ideia de células simples seguidas de células complexas (HAKYN, 2009 apud HUBEL; WIESEL, 1962).

Figura Arquitetura CNN



Fonte: Haykin (2009)

As CNN são tradicionalmente aplicadas a atividades de reconhecimento de padrões em imagens (BISHOP, 2006; LECUN et al 1989; KRIZHEVSKY; SUTSKEVER; HINTON, 2012) mas recentemente houveram iniciativas na aplicação de CNN em atividades de classificação em textos como por exemplo mineração de opiniões, análise de sentimentos e agentes conversacionais baseados em IR (JOST, 2015; KALCHBRENNER; GREFENSTETTE; BLUNSOM, 2014; PRAKASH; BROCKETT; AGRAWAL, 2016).

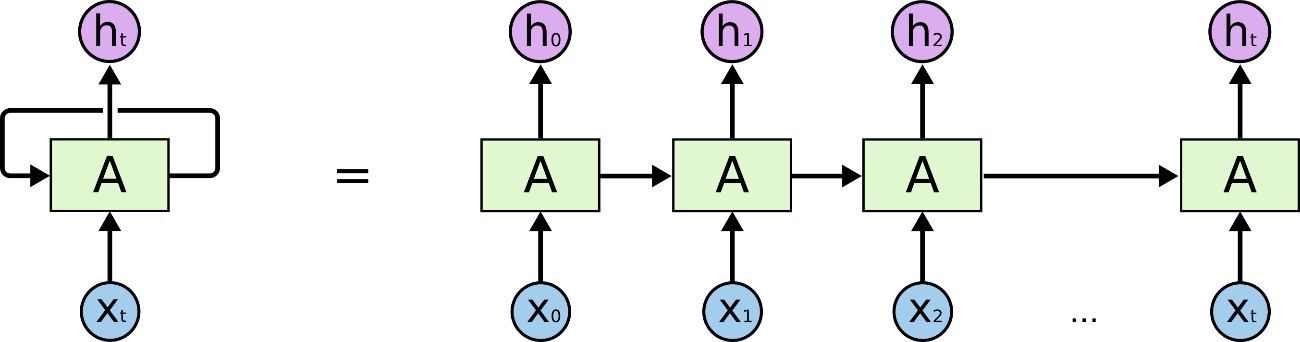
### Redes Neurais Recorrentes

As RNA possuem capacidades de processamento de dados, extração de padrões e generalização sendo empregadas a diversos problemas, uma das principais deficiências das RNA clássicas é a incapacidade de tratamento de dados temporais, as Redes Neurais Recorrentes (do inglês: *Recurrent Neural Networks*, RNN) surgiram para suprir esta demanda por RNA capazes de tratar séries temporais de dados.

RNN constituem uma ampla classe de RNAs cujo estado depende tanto do estado atual como de estados anteriores, são amplamente utilizadas em análise de dados sequenciais devido a capacidade destas redes de analisarem sequências complexas. Através de diversas camadas ocultas, uma para cada intervalo de tempo, as RNN compartilham os parâmetros entre as camadas de forma a disponibilizar um histórico dos dados de todos os elementos antigos da sequência (SUTSKEVER et al, 2013, LECUN; BENGIO; HINTON, 2015).

As RNN possuem *loops* que permitem persistir os passos anteriores de uma sequência (OLAH, 2015), conforme ilustrado pela Figura 13 na qual A é uma RNN Xt é a entrada da rede e Ht é saída, a cada iteração do *loop* é passado como *input* um elemento X e a sua saída H se torna a entrada do próxima iteração.

Figura Sequências nas RNN



Fonte: OLAH (2015).

Conforme Sutskever et al (2013) pelo número de camadas ocultas geradas pelas RNN elas se tornam especialmente difíceis de treinar devido a retropropagação ou *backpropagation* do gradiente causar o aumento ou diminuição do gradiente a níveis extremos (LECUN; BENGIO; HINTON, 2015).

AS RNN são muito boas em prever o próximo caractere no texto ou a próxima palavra em uma sequência, mas também podem ser usados para tarefas mais complexas como reconhecimento de fala, PLN, tradução, sinais analógicos e digitais (LECUN; BENGIO; HINTON, 2015; OLAH, 2015).

As *Long Short-Term Memory* (LSTM) são um tipo de RNN utilizada em conjunto com algoritmo de aprendizado baseado em gradiente. As redes LSTM foram projetadas para serem tolerantes a ruídos, capaz de ligar intervalos de tempo em mais de 1000 passos sem perda de recursos de longo prazo (HOCHREITER; SCHMIDHUBER, 1997).

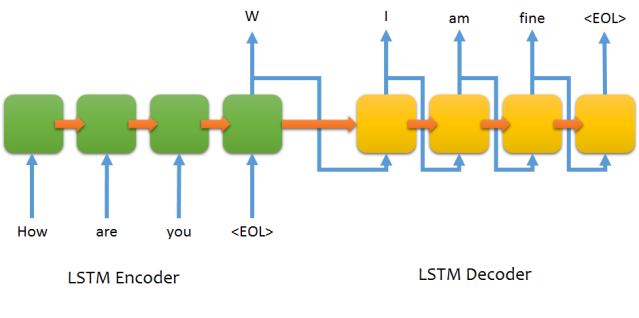
### Sequence-to-Sequence

O modelo *sequence-to-sequence* (Seq2Seq) é uma técnica de *Deep Learning* que aplica conceitos de aprendizado supervisionado possibilitando o mapeamento de uma sequência ou sequências de entrada para uma sequência de saída, tendo em seu principal diferencial a flexibilização da dimensionalidade das sequências de entrada e saída (SUTSKEVER; VINYALS; LE, 2014).

“Os modelos de sequência a sequência existentes com atenção concentraram-se em gerar a sequência de destino alinhando cada *token* de saída gerado a outro *token* na sequência de entrada. ”. Ao contrário de abordagens anteriores como *Deep Neural Networks* (DNN) onde as sequências deveriam obrigatoriamente possuir uma dimensionalidade fixa o modeloSeq2Seq permite diferentes dimensões para cada sequência possibilitando assim ao modelo maior capacidade de aprendizado sobre dados sequenciais encontrados em textos, fala e vídeos por exemplo. No caso da aplicação da arquitetura *sequence-to-sequence* aplicada a criação de *Chatbots* são passados como entrada para o algoritmo uma coleção de textos com um grande número de amostras de conversações, a rede aprende a partir de texto puro e sua saída também é texto montando uma frase tentando se aproximar de uma resposta válida predizendo a resposta da pergunta feita.

A arquitetura do modelo utiliza de redes neurais recorrentes, podendo ser utilizadas RNN clássicas, GRU ou LSTM, neste modelo duas RNA LSTM uma agem como codificador e outra como decodificador este modelo é conhecido como *Encoder-Decoder*. Um exemplo de mapeamento de sequências para sequências pode ser visto conforme ilustrado pela Figura 14, aonde é demonstrada a arquitetura *Encoder-Decoder* utilizada pelo modelo Seq2Seq formada por duas redes LSTM onde o modelo gera a sequência de saída através da geração da previsão do próximo passo considerando a sequência de entrada e saída gerada até o momento (JAITLY et al 2016).

Figura Modelo Sequence-to-Sequence



Fonte: Cahn (2017) apud RAM (2016)

O modelo Seq2Seq foi aplicado com sucesso ao longo dos anos em diversas tarefas como:

* Tradução
* Reconhecimento de Fala
* Captura de Imagens
* Rotulação de imagens
* Modelagem de diálogo

Segundo Cahn (2017) embora seja importante ressaltar a qualidade de respostas geradas por um *chatbot* treinado com Seq2Seq o modelo não é capaz de gerar interações a nível de um ser humano, fornecendo respostas diversas sendo dependente da variedade de frases em seu *dataset* de treinamento.

# CHATBOTS

Em de 1950, Alan Turing publicou um artigo “*Computing Machinery and Intelligence*”, o qual descreve um teste para avaliar a capacidade de uma máquina ou computador de pensar, este teste na época chamado “*Imitation Game*” hoje é conhecido como Teste de Turing. O Teste de Turing se baseia na impossibilidade de distinção entre um ser humano e uma máquina, o computador seria aprovado caso em um teste às cegas realizado com diferentes avaliadores humanos esse fosse considerado uma pessoa.

Esta proposta por muitos anos foi utilizada como objetivo e referência para alcançar um modelo de programa que fosse considerado inteligente. Diversas iniciativas surgiram baseadas neste objetivo, uma delas foi a de Joseph Weizenbaum (1966) citado por Cahn (2017, p. 4) que criou o primeiro *chatbot* ELIZA, que obteve em sua época os melhores resultados a imitar um humano em uma conversação. Segundo Jurafsky e Martin (2009) citados por Maeda e Moraes (2017) “Agentes conversacionais, também conhecidos como sistemas de diálogo, são programas que se comunicam com os usuários em linguagem natural (por texto, por voz ou por ambas as formas) ”. Dada uma frase de entrada, ELIZA idêntica palavras-chave e as relaciona com padrões de uma base pré-programada de regras para a geração de respostas apropriadas. ELIZA foi um dos primeiros programas a utilizar-se de técnicas de correspondência de padrões para simular um humano em uma conversação.

Os *chatbots* possuem diversas nomenclaturas, sendo também chamados de *Chatterbots,* agentes conversacionais, bots entre outras. “Ainda assim pesquisadores do ramo da IA devotaram pequeno esforço para passar no Teste de Turing, acreditando que mais importante que duplicar um exemplar da inteligência humana é estudar seus princípios fundamentais. ” (RUSSEL; NORVIG, 1995, p. 3). Isto se torna ainda mais evidente uma vez que através dos recentes desenvolvimentos da capacidade computacional e nos modelos de RNA e AM mostram que embora em muitos pontos os computadores podem ultrapassar a capacidade humana de análise, existem muitos outros os quais ainda são inferiores. Indicando que o estudo de uma forma isolada de inteligência não refletiria em uma conclusão definitiva a respeito do nível de inteligência.

Comarella e Café (2008, p. 55) citado por Spanhol (2017, p. 31) comentam que “dentre os grandes desafios da computação está o de se criar meios para tornar a comunicação homem-máquina mais natural e intuitiva [...] desenvolvendo programas capazes de ‘compreender’ a linguagem humana”.

Atualmente é percebido um crescimento no interesse corporativo em *chatbots* conforme evidenciado por Maeda e Moraes (2017, p.11) as capacidades de comunicação em linguagem natural atraem o interesse de empresas na utilização destes agentes como facilitadores em diversos campos de atuação como atendimento ao cliente quanto a uma alternativa aos aplicativos móveis convencionais.

## Simbólico

Os *chatbots* simbólicos ou agentes orientados a tarefas, são aqueles em que suas regras devem ser codificadas manualmente e ficam armazenadas em documentos, como os das primeiras gerações que utilizam a linguagem AIML (do inglês *Artificial Intelligence Markup Language).*

O AIML possibilita a codificação de regras para correspondência de padrões, possibilitando a criação de respostas para diferentes situações. Quando o *chatbot* recebe uma entrada que corresponde a uma *tag* <pattern>, envia a mensagem armazenada na *tag* <template> como uma resposta, o <pattern> pode ser uma frase "Qual é o seu nome?" ou um padrão "Meu nome é \*", onde o asterisco é uma expressão regular. Normalmente, esses pares <pattern> <template> são feitos à mão (CAHN, 2017, p. 17).

A principal responsabilidade do desenvolvedor ao criar um chatbot utilizando este modelo é a escolha do algoritmo de casamento de padrões, tais como:

* ELIZA Incremental Parsing
* Artificial Linguistic Internet Computer Entity (ALICE)/AIML Direct Match Technique
* VPBot Keyword Set Matching Technique
* ViDi One-Match and All-Match Categories (OMAMC) Technique

O principal problema deste modelo é a necessidade de especificação de todas as regras em seus documentos gerando respostas padrões quando questionado sobre algo que não esteja cadastrado em sua base de conhecimento.

## Modelo Baseado em Recuperação de Informações

O Modelo Baseado em Recuperação de Informações (do inglês *Information Retrieval Based Models*) foi criado baseado na disponibilidade de dados de conversas em plataformas *online,* aonde um grande número de conversas é armazenado para análise, sendo armazenados em pares.

“O princípio por trás dos sistemas baseados em recuperação de informações é que dado uma sentença, ela é combinada com um conjunto de <status> ou <respostas> para determinar uma resposta apropriada. ” (CAHN, 2017, p.18).

O desenvolvedor no caso de escolha desta abordagem deve escolher se seu algoritmo fará a comparação via status ou resposta.

## Probabilístico

Os *chatbots* probabilísticos ou generativos são derivados das técnicas de *Statistical Machine Translation Generative Models* (SMT) aonde as respostas são geradas em tempo real para o usuário, “são utilizadas técnicas de AM: classificadores são treinados com base em diálogos reais e usados para gerar respostas para usuários ‘traduzindo’ frases de entrada em respostas. ” (CAHN, 2017, p.20).

Este modelo se destaca devido sua capacidade de responder a perguntas que nunca recebeu antes com base em uma abordagem probabilística, onde por meio de seu treinamento com técnicas de AM se torna capaz de gerar respostas com base na sentença de entrada. Segundo Cahn (2017) é utilizado o algoritmo Seq2Seq para mapeamento de sentenças de entrada para sentenças de saída, este algoritmo utiliza redes neurais recorrentes para realizar este processo, sendo atualmente a prática que alcança os melhores resultados.

A utilização das RNA no processo possibilita a inclusão de dados não estruturados no treinamento do *chatbot*. O modelo Seq2Seq utiliza duas RNA recorrentes, o chamado modelo *Encoder-Decoder* no qual, primeiro a rede lê a sentença e segundo gera a resposta.

## Aplicações

As aplicações dos *chatbots* incluem diversas atividades as quais necessitam inteligência para sua execução. A inteligência artificial empregada neste sentido é utilizada para que os algoritmos se comportem como um ser humano em uma conversação. Diversas iniciativas visam utilizar os chatbots para atendimento ao cliente, suporte técnico, apoio ao aprendizado (MORAES; MACHADO, 2016), entre muitas outras possibilidades.

Um ponto importante também abordado por estas tecnologias é a representação de conhecimento por interfaces mais simples e amigáveis os sistemas podem agir como uma interface para o conhecimento estruturado e não estruturado, obtido através dos processos de treinamento.

Outra utilização comum para estes agentes, como abordado por Cahn (2017) são os *Virtual Personal Assistants* (VPAs, ou Assistente Pessoal Virtual), tais como exemplos já citados como a assistente do sistema operacional IOS Siri e o assistente do sistema operacional Android Google Now. “Entre 2007 e 2015, os *chatbots* estavam participando de um terço a metade de todas as interações online (RADZIWILL; BENTON, 2017 apud TSVETKOVA et al., 2016) ”, também cresce sua utilização em serviços de mensageria como Telegram, Slack entre outros.

# TRABALHOS RELACIONADOS

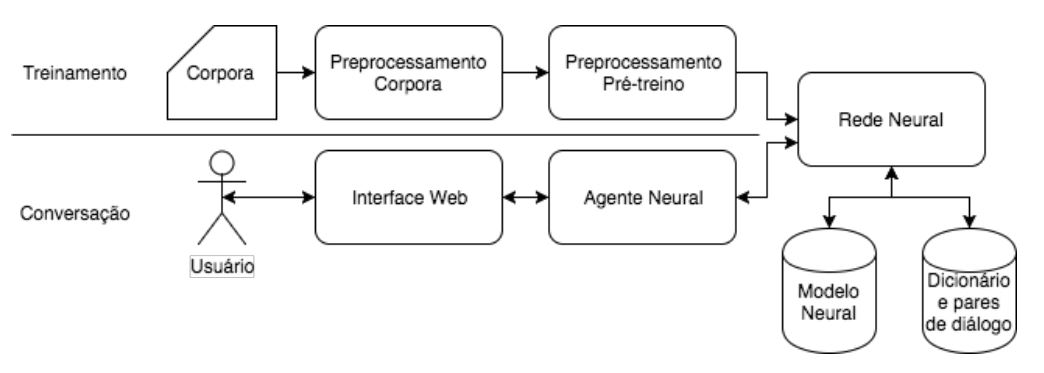
Neste capítulo comentamos trabalhos relacionados à construção de *chatbots* ou agentes conversacionais baseados em *Deep Learning*, assim como *chatbots* baseados em *Deep Learning* para a língua Português-Brasileiro. É importante ressaltar que a utilização do *Deep Learning* aplicado à construção de chatbots ainda é um campo com poucos trabalhos desenvolvidos quando observado a língua Português-Brasileiro como objetivo.

Diversos trabalhos envolvendo tais tecnologias são vistos aplicados a língua inglesa como, por exemplo, Vynials e Le (2015) com as primeiras implementações utilizando o modelo seq2seq e em trabalhos mais recentes por exemplo Kottur, Wang e Carvalho (2017) , onde realizam uma comparação entre diferentes arquiteturas de redes neurais disponíveis para a elaboração de um novo modelo de *chatbot,* o qual se adapta às informações da pessoa que está conversando assim como informações do contexto da conversa.

Durante o levantamento de trabalhos que propõem o desenvolvimento de *chatbots* para a língua Português-Brasileiro foi verificado que a maioria das abordagens de PLN utilizadas na criação de tais agentes utilizava a abordagem simbólica da PLN, sendo criados modelos utilizando as técnicas de orientação a regras ou a recuperação de informações, como por exemplo o trabalho de Moraes e Machado (2016) no qual são aplicadas técnicas de *Deep Learning* na construção da base de conhecimento do agente, mas não na sua arquitetura.

O único trabalho encontrado que utilizava as tecnologias de *Deep Learning* e RNA foi o chatbot criado por Maeda e Moraes (2017), Figura 15, o qual aplicou o modelo seq2seq de Vynials e Le (2015) na criação de um agente utilizando dois *datasets,* um extraído das legendas de filmes do Star Wars e outro de um *dataset* de conversação. Os principais desafios citados por Maeda e Moraes (2015) são a falta de *datasets* abertos criados apenas com históricos de conversação, algo que tornaria melhor os resultados obtidos.

Figura Arquitetura Chabot proposto por Maeda e Moraes



Fonte: Adaptado de Maeda e Moraes (2017).

Quanto aos trabalhos futuros, Maeda e Moraes (2017) citam a possibilidade de obtenção de melhores resultados na criação de um chatbot desenvolvido com *Deep Learning* a utilização de um *pipeline* mais robusto de técnicas de PLN no pré-processamento dos *word embeddings* gerados, assim como mais testes com os hiperparâmetros da RNA gerada. Nesta visão considerando o escopo deste trabalho em relação a utilização de técnicas de PLN, e uma comparação na efetividade de um *word embedding* gerado a partir de um *corpus* com anotações sintáticas ou não, este estudo busca dar continuidade ao trabalho iniciado por Maeda e Moraes (2017). Uma vez que serão aplicadas a mesma arquitetura de redes neurais, o modelo seq2seq, assim como aplicados conceitos de PLN e anotações sintáticas no treinamento do modelo a ser gerado. O principal diferencial deste estudo em relação a trabalhos anteriores é a aplicação de técnicas de pré-processamento baseadas em PLN e a utilização de *corpus* já existentes para o idioma Português Brasileiro.

# METODOLOGIA

Nesta seção serão apresentados os procedimentos metodológicos que possibilitem o cumprimento dos objetivos propostos neste estudo.

## Tipo de pesquisa

Neste trabalho será realizado tanto um estudo teórico quanto uma experimentação prática, divido entre uma etapa de pesquisa de conceitos para embasamento teórico do trabalho e uma etapa para aplicação prática dos conceitos estudados. A etapa teórica estabelecerá o embasamento necessário para a compreensão dos assuntos estudados, assim como ressaltar as diferenças de abordagem nas técnicas a serem comparadas. Na segunda etapa do trabalho serão implementados os algoritmos de aprendizagem de máquina e treinados os modelos a serem comparados, sendo em uma última etapa realizada a comparação dos resultados gerados pelos dois modelos.

Conforme Prodanov e Freitas (2013) a pesquisa bibliográfica é um procedimento para a partir de materiais já publicados o autor da pesquisa se familiarizar com a área de estudo. A pesquisa bibliográfica irá contextualizar as técnicas e problemas conhecidos da área para um melhor embasamento no momento de aplicação prática dos conceitos, ao desenvolver os *chatbots* propostos. O estudo buscará as informações a respeito das técnicas relacionadas a construção de modelos de *Sequence-to-Sequence Learning* e *Word Embeddings* para o desenvolvimento da solução de forma mais eficaz, buscando alinhamento com procedimentos de pesquisa adequados ao problema proposto, “A investigação científica depende de um conjunto de procedimentos intelectuais e técnicos para que seus objetivos sejam atingidos: os métodos científicos. ” (PRODANOV; FREITAS, 2013, p. 126).

Outro foco da pesquisa teórica será a busca por argumentos e benefícios do uso de técnicas derivadas do Processamento de Linguagem Natural aplicado ao pré-processamento de *datasets* para o treinamento de agentes conversacionais, as estratégias para desenvolvimento de *chatbots* na língua Português-Brasileiro assim como o estado da arte atual.

Considerando que o objetivo principal deste trabalho será a criação e avaliação de um *chatbot* na língua Português Brasileiro, será realizada uma avaliação de resultados do modelo criado através da aplicação de questionários para mensurar o sucesso do modelo treinado. Deste modo a pesquisa é caracterizada por possuir o caráter de uma metodologia de pesquisa aplicada, utilizando-se de técnicas de pesquisa quantitativa, uma vez que “requer o uso de recursos e técnicas de estatística, procurando traduzir em números os conhecimentos gerados pelo pesquisador” (PRODANOV; FREITAS, 2013 p. 128) e procedimentos de pesquisa experimental aonde “Determinamos um objeto de estudo, selecionamos as variáveis e definimos as formas de controle e de observação dos efeitos. ” (PRODANOV; FREITAS, 2013 p. 128), no qual serão definidas as variáveis a serem comparadas entre os diferentes modelos de *chatbots* desenvolvidos.

## Design dos experimentos

Serão treinados diversos *chatbots* baseados na arquitetura *sequence-to-sequence* durante o decorrer deste trabalho em busca da criação de dois agentes, um treinado a partir de um *corpus* de domínio aberto e acesso público e outro com *corpus* especifico de conversação em Português Brasileiro. Caso este recurso não seja encontrado será gerado um *dataset* para treinamento com dados específicos de conversação, extraídos a partir das legendas de filmes ou de séries de televisão. O *chatbot* em sua arquitetura utilizará técnicas presentes no PLN para o tratamento dos dados e técnicas de AM para a realização do seu treinamento e criação da rede neural seq2seq.

## Design da avaliação dos resultados

O modelo utilizado para avaliação dos resultados obtidos é baseado na metodologia descrita nos trabalhos de Maeda e Moraes (2017) e de Souza e Moraes (2015) onde são aplicados questionários pré-interação com o *chatbot* e aplicado um questionário pós-interação com o intuito de mensurar a qualidade da interação dos agentes criados bem como o conhecimento do público de testes sobre o assunto.

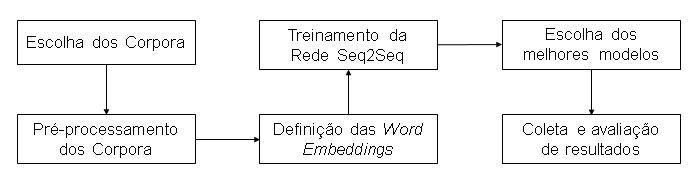
# DESENVOLVIMENTO DOS CHATBOTS

Para o desenvolvimento dos agentes fez-se necessária a elaboração e execução das seguintes etapas:

* Escolha do *dataset* ou *corpus* a utilizar;
* Pré-processamento dos *corpora* escolhidos;
* Escolha das *word embeddings* a utilizar;
* Escolha do algoritmo seq2seq a ser utilizado no trabalho;
* Escolha dos parâmetros para treinamento da rede seq2seq;
* Coleta e análise dos resultados alcançados pelos modelos;
* Escolha dos modelos a serem avaliadas pelos usuários de teste.

O processo de desenvolvimento das etapas deste trabalho, ilustrado pela Figura 16, ocorreu após a etapa de fundamentação teórica, onde através dos conhecimentos adquiridos foram desenvolvidos os recursos necessários para a criação do *chatbot* seq2seq.

Figura Processo desenvolvimento do trabalho



Fonte: Autor (2018).

## Aspectos de Implementação

A linguagem utilizada para o desenvolvimento de scripts de pré-processamento e dos algoritmos de AM foi a linguagem Python. Segundo Pedregosa et al (2011) a linguagem Python se estabeleceu como uma das mais populares para a computação científica devido ao seu grande número de bibliotecas e ecossistema tem grande adoção por parte da academia.

O Python é utilizado para diversos fins, possuindo *frameworks* e *libraries* que possibilitam desenvolvimento de diversas aplicações, tais como as utilizadas neste trabalho nas seguintes tarefas:

* Pré-processamento de texto: através da biblioteca NLTK;
* Criação de Word Embeddings: através da biblioteca Gensim;
* Rede Neural Seq2seq: através do *framework* Tensorflow;
* Interface Web: através do framework Django;

A IDE utilizada no desenvolvimento foi o Visual Studio Code (versão 1.23). Para facilitar a realização do pré-processamento nos *corpora* foi escolhida a biblioteca *Natural Language Toolkit* (NLTK), disponível para a linguagem Python, que visa reunir e agrupar uma grande variedade de algoritmos que implementam técnicas de PLN como por exemplo algoritmos de *tokenization, stemming, lemmatization, part-of-speech taggers*, ferramentas para *parse* e análise semântica.

Para os experimentos com as *Word Embeddings* em Português Brasileiro foi utilizada a biblioteca Gensim desenvolvida por Rehurek (2011) em sua tese de doutorado. Gensim é um framework para *vector space modelling,* e contém diversas implementações de algoritmos, para os testes de criações de *word embeddings* a partir do *corpus* Wit³ foi utilizada a implementação do algoritmo word2vec contida neste *framework*.

O *framework* Tensorflow foi utilizado para a modelagem da arquitetura de DNN *sequence-to-sequence* em ordem de criar e treinar os agentes conversacionais propostos por este trabalho. O Tensorflow é uma biblioteca de código aberto criado pelo Google *Brain Team* e mantida em conjunto com a comunidade de software livre. Segundo Fernandes e Da Silva (2017), sua vasta documentação aliada ao suporte multi-GPU e grande número de usuários e interfaces de alto e baixo nível são os principais pontos fortes do *framework*.

Segundo Abadi et al (2016) o framework Tensorflow suporta treinamento e inferência em larga escala, buscando a utilização de recursos de forma eficiente, podendo ser utilizado em diferentes infraestruturas como *datacenters* com computação distribuída, processamento através da GPU. Os modelos gerados por este *framework* são também flexíveis a ponto de rodarem em diferentes dispositivos como computadores, servidores e até mesmo celulares.

Ainda segundo Abadi et al (2016) uma aplicação criada pelo tensorflow pode ser descrita como um grafo composto por nós, no qual este grafo representa o fluxo dos dados atualizando seus estados e compartilhando entre os nós e diferentes execuções no caso de execução paralela.

Devido a utilização de uma técnica que necessita de grande poder computacional para sua execução foi utilizada a versão tensorflow-gpu para o treinamento da rede neural foram utilizadas as dependências listadas na Tabela 4 Dependências do Projeto.

Tabela 4 Dependências do Projeto

|  |  |
| --- | --- |
| cuda | 8.0 |
| cudnn | 5.1 |
| python | 3.5 |
| tensorflow-gpu | 1.4 |
| numpy | 1.14.3 |
| django | 1.10 |
| channels | 2.1.1 |
| redis | 2.10.6 |
| asgi\_redis | 1.0 |

## Definição e Pré-processamento dos *Corpora*

Tendo em vista a necessidade de uma grande massa de dados para o treinamento do modelo *sequence-to-sequence*, característica marcante especialmente as técnicas derivadas do *Deep Learning*, foram pesquisados *corpora* abertos disponíveis na língua Português Brasileiro. Uma dificuldade encontrada durante esta fase inicial da pesquisa, em comum a Maeda e Moraes (2017), foi a escassez de *corpora* específicos de para o Português Brasileiro. A disponibilidade de *corpora* orientados especificamente ao diálogo é um ponto extremamente importante, uma vez que a rede treinada aprende de exemplos contidos nos dados de treinamento (VYNIALS, LE, 2015).

Para as pesquisas de *corpora* disponíveis foi consultado o maior repositório conhecido para a língua portuguesa o site Linguateca (2015) que visa facilitar o acesso de pesquisadores da língua portuguesa a recursos já existentes. Através da busca foi selecionado o *corpus* CetenFolha, construído a partir de todas as publicações feitas pelo jornal Folha de São Paulo durante o ano de 1994, contendo 1.597.807 frases, este *corpus* foi utilizado apenas inicialmente no trabalho, porém após problemas na normalização dos dados e no fato do *corpus* ser estruturado por parágrafos e não frases foi decidido a utilização de outro recurso.

Após ser descartada a utilização do *corpus* CetenFolha foi encontrado o *corpus* baseado nas legendas de palestras do evento Ted (MAURO; CHRISTIAN; MARCELLO, 2012), este *corpus* se mostrou com qualidade superior de seus dados em relação ao CetenFolha, uma vez que apesar do menor número de dados o número de ruídos no *dataset* era menor e o número médio de palavras nas frases era menor. O *corpus* WIT³ (MAURO; CHRISTIAN; MARCELLO, 2012) foi criado a partir de dados de 900 palestras coletadas ao final de 2011, a última atualização deste *corpus* foi em abril de 2016 e já possui 109 linguagens disponíveis, para este trabalho foi utilizado a versão do *corpus* para a língua portuguesa.

Para o pré-processamento do *corpus* Wit³ foi criado um *script* com a linguagem Python e biblioteca *Natural Language Tookit* (NLTK) para a remoção *tags* XML, pontuações e espaços em branco em excesso. Também durante esta etapa foi convertido o *corpus* do formato de codificação Latin-1 para UTF-8 com a intenção de facilitar a manipulação do arquivo por bibliotecas externas como o NLTK e biblioteca unidecode, foram removidos todas as pontuações do arquivo, exceto ponto de interrogação, removidos endereços de email contidos no *corpus*, substituídos os números por 0, removidos links para páginas de internet, textos entre aspas simples e duplas para esta etapa foram utilizados principalmente a biblioteca NLTK já citada e expressões regulares também conhecidos como regex. Nesta etapa também foram aplicados o processo de *lowercase* e tokenização.

Como já identificado por Maeda e Soares (2017) a língua portuguesa tem carência por *corpora* específicos de conversação, fato que dificulta a criação de *chatbots* baseados em DL. Devido a necessidade de utilizar um *corpus* com dados específicos de conversação foi necessária a criação deste recurso durante a pesquisa. Para a criação do *corpus* de Legendas foram baixados os todos arquivos de legenda disponíveis na língua Português Brasileiro sobre as 9 primeiras temporadas da série americana *The Middle* disponíveis no site OpenSubtitles, a qual foi escolhida por tratar de o cotidiano de uma família e por este motivo teriam mais conversas casuais.

O processo de criação deste *corpus* de legendas ocorreu através do *download* dos arquivos de legenda em formato .srt, conversão para o formato de codificação utf-8 e unificação em um só arquivo e posteriormente realizada a remoção das *timestamps*, informação de quando cada sentença deve aparecer, e remoção das últimas 3 linhas de cada arquivo, onde geralmente era exibido o nome dos autores de cada legenda, assim como a remoção de outras expressões que foram caracterizadas com ruído. Após o processo de normalização e unificação dos diferentes arquivos de legenda foi gerado um *corpus* contendo 584.401 sentenças de diálogos.

Ao final desta etapa do trabalho estava gerado uma versão normalizada do *corpus* Wit³, com a remoção de *tags* XML e pontuações e o *corpus* criado pelas legendas da série *The Middle,* unificado a partir de diferentes arquivos e processado para remoção de ruídos e pontuações.

## Definição das Word Embeddings

O objetivo inicial da etapa de definição das *word embeddings* era a criação deste recurso através dos *corpora* normalizado, porém após experimentos inicias com a biblioteca Gensim foi percebido que as *word embeddings* geradas possuíam um nível de relação semântica entre palavras muito baixo não passando em testes como o rei-rainha descrito por Mikolov et al (2013).

Por este motivo foi optado pela utilização de um conjunto de *word embeddings* gerados pelo trabalho de Hartmann et al (2017) onde foi selecionada a *word embeddings* no modelo skip-gram de 300 dimensões por ser similar as especificações utilizadas por Maeda e Soares (2017) e Vinyals e Le (2015) no desenvolvimento de seus agentes.

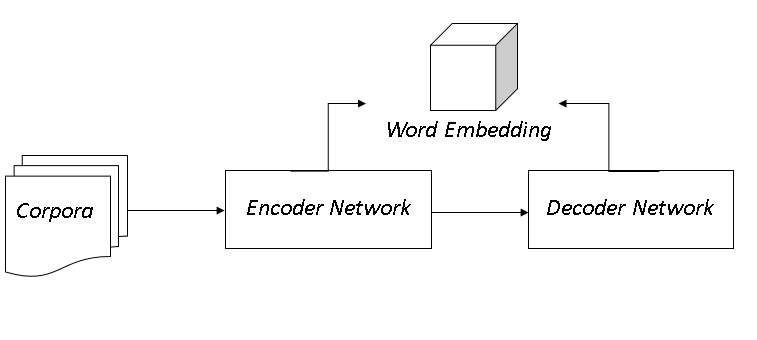
Para a criação de suas *word embeddings* Hartmann et al (2017) realizou um processo de pré-processamento utilizando técnica de Tokenização e remoção de palavras com menos que 5 ocorrências por um *token* ***unknown***, para endereços eletrônicos contidos foram substituídos pelo *token* **URL** e endereços de e-mail par ao *token* **EMAIL**. Ainda segundo os autores foram utilizados 714.286.638 *tokens* para a geração das *word embeddings*, treinadas em 31 diferentes modelos com os algoritmos GloVe, Word2Vec, Wang2Vec e FastText com as dimensões 50, 100, 300, 600 e 1.000.

A definição das *word embeddings* é uma etapa crucial na criação de agentes conversacionais baseados no modelo *sequence-to-sequence* uma vez que a *embedding layer* presente no modelo mapeia as palavras passadas ao modelo para o índice contido no vocabulário do modelo.

## Arquitetura do Chatbot

A arquitetura utilizada para o desenvolvimento do *chatbot* foi o modelo padrão da arquitetura *sequence-to-sequence* ilustrado pela Figura 17 e aplicado pelos trabalhos de Vinyals e Le (2015) e Maeda e Soares (2017). Foi utilizado a implementação em Python DeepQA¹ do modelo *sequence-to-sequence* disponível utilizando o *framework* tensorflow-gpu. A rede neural é responsável por carregar os tensores a partir do *corpus* normalizado e inicializar a *word embedding* conforme os parâmetros passados na Tabela 5 e Tabela 6.

Figura Arquitetura do Chatbot



Fonte: Autor (2018).

Para executar os experimentos foi utilizado um *notebook* modelo Dell Vostro 5470, com sistema operacional Windows 10, processador Intel Core I7 de 2,6 GHz e placa de vídeo Nvidea GeForce GT 740M a rede foi treinada por 30 épocas com média de 30 minutos por época para o *chatbot* baseado em legendas e 2 horas por época para o chatbot baseado no *corpus* Wit³.

Tabela 5 Hiperparâmetros chatbot legendas

|  |  |  |
| --- | --- | --- |
| hiddenSize | 254 | Número de *hidden units* em cada camada RNN |
| Corpus | LegendasTheMiddle | Arquivo contendo corpus em formato UTF-8 |
| maxLength | 10 | Tamanho máximo das sentenças a utilizar |
| filterVocab | 5 | Desconsiderar palavras que apareçam menos que 5 vezes |
| vocabularySize | 20.063 | Tamanho do vocabulário |
| initEmbeddings |  | Habilita Inicialização de *word embeddings* |
| embeddigSource | nilc\_skip\_300.bin | Arquivo das *word embeddings* |
| embeddingSize: | 300 | Dimensões das *word embeddings* inicializadas |
| watsonMode | False | Inverter a ordem das palavras nas frases ao passar para o modelo |
| autoEncode | False | Seq2seq arquitetura *autoencode* |
| QA | 98.365 | Número de pares pergunta-resposta |
| numEpochs | 30 | Número de máximo de épocas |

Fonte: Autor (2018).

Tabela 6 Hiperparâmetros chatbot Wit³

|  |  |  |
| --- | --- | --- |
| hiddenSize | 254 | Número de *hidden units* em cada camada RNN |
| Corpus | Wit3-Ted | Arquivo contendo *corpus* em formato UTF-8 |
| maxLength | 10 | Tamanho máximo das sentenças a utilizar |
| filterVocab | 5 | Desconsiderar palavras que apareçam menos que 5 vezes |
| vocabularySize | 40.000 | Tamanho do vocabulário |
| initEmbeddings |  | Habilita Inicialização de *word embeddings* |
| embeddigSource | nilc\_skip\_300.bin | Arquivo das *word embeddings* |
| embeddingSize: | 300 | Dimensões das *word embeddings* inicializadas |
| watsonMode | False | Inverter a ordem das palavras nas frases ao passar para o modelo |
| autoEncode | False | Seq2seq arquitetura *autoencode* |
| QA | 98.365 | Número de pares pergunta-resposta |
| numEpochs | 30 | Número de máximo de épocas |

Fonte: Autor (2018).

Foram gerados ao todo 11 modelos treinados com os *corpora* Wit³ e Legendas onde através de pequenas mudanças de parâmetros e na evolução do processo de construção do pré-processamento do*s corpora.* Nos primeiros modelos não era realizado nenhum tipo de pré-processamento o que resultava em um grande número de ruídos no modelo final, nos modelos seguintes foram removidos as vírgulas e pontuações exceto o ponto de interrogação além de remoção de *tokens* considerados como ruídos ou informações privadas como e-mails e números de telefones. Foram utilizados um script com 130 perguntas padrões para interação com cada um dos *chatbots* criados, durante esta etapa foi necessária a seleção manual devido as respostas geradas não serem capazes de serem caracterizadas como certos ou erradas de maneira automatizada.

Após a fase de treinamento da rede e escolha dos modelos para avaliação com o público em geral foram publicados a interface *web* para interação dos usuários com o *chatbot*, para isto foi utilizado o site desenvolvido com o framework *web* Django disponível junto ao código do projeto DeepQA, a interface web é responsável por receber a pergunta realizada pelo usuário, passar para o módulo do *chatbot* e retornar a resposta ao usuário seguindo assim o fluxo da conversa.

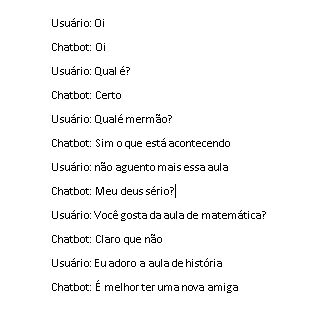
# AVALIAÇÃO DE RESULTADOS

Para a realização da coleta de resultados foram aplicados dois questionários para o público de teste, um questionário pré-interação com o *chatbot*, descrito no APÊNDICE A – Questionário pré-interação para mapeamento do perfil do público de teste e um questionário pós-interação APÊNDICE B – Questionário Pós-interação para avaliação da qualidade das respostas geradas pelos modelos.

Os questionários foram respondidos por 44 usuários, que interagiram com o *chatbot* através da interface *web.* A distribuição da faixa etária dos avaliadores pode ser consultada na Figura 22, 40,9% de 18 a 25 anos, 27,3% dos respondentes acima de 45 anos, 18,2% tinha de 25 a 35 anos, 9,1% menos que 18 anos e 4,5% dos usuários tinham de 35 a 45 anos. Em relação a escolaridade dos usuários entrevistados, ilustrado pela Figura 23, 31,8% possuíam Ensino Superior Incompleto, 22,7% Ensino Fundamental, 22,7 Ensino Superior Completo, 13,6% Ensino Médio, 9,1% Pós-graduados. A maioria dos acadêmicos entrevistados era da área de Ciências Exatas e Tecnológicas correspondendo a 66,7% seguido por 20% da área de Ciências Humanas.

Em relação ao conhecimento prévio sobre o tema 59% dos usuários de teste já conheciam o termo *chatbot* ou Agente Conversacional e destes 54,5% já haviam interagido com um, gráficos ilustrando as perguntas podem ser visualizados na Figura 25 e Figura 26. Foi percebido que os usuários que interagiram com o *chatbot* gerado por legendas que simularam conversas em um ambiente escolar recebiam respostas mais coerentes, como por exemplo o trecho apresentado na Figura 18, isso se deve pelo contexto das frases do *dataset* de treinamento.

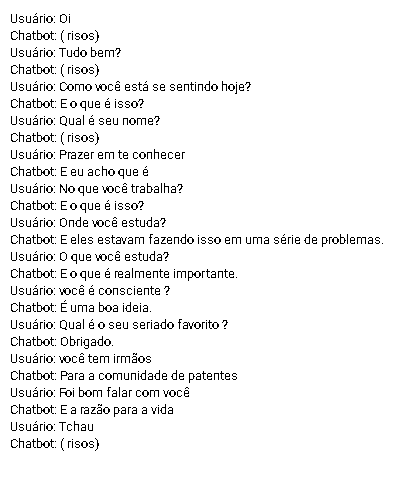
Figura Trecho conversa Chatbot Legendas



Fonte: Autor (2018).

Em relação ao *chatbot* baseado no *corpus* Wit³ as respostas independentes do contexto utilizado exibiam um padrão errático, o qual não conseguia manter coerência semântica na em frases isoladas ou em sequências como pode ser visto um exemplo de trecho de diálogo na Figura 19. Para entradas iguais os *chatbots* geravam sempre a mesma resposta, comum ao modelo clássico de *sequence-to-sequence* e mesmo comportamento do chatbot de Maeda e Moraes (2017).

Figura Trecho conversa Chatbot Wit³



Fonte: Autor (2018).

Os resultados dos questionários pós-interação foram realizados pelos 44 usuários de teste dos chatbots, os quais interagiram com ambos os *chatbots* baseado em legendas e baseado em *corpus.* Os resultados obtidos foram abaixo do estado da arte em Português Brasileiro do modelo de Maeda e Moraes (2017) e do modelo de Vinyals e Le (2015), sendo que a maioria das respostas obteve a avaliação “nenhuma” resposta coerente para o modelo gerado a partir do *corpus* Wit³, Figura 20 e “Poucos” para o *corpus* baseado em legendas, Figura 21.

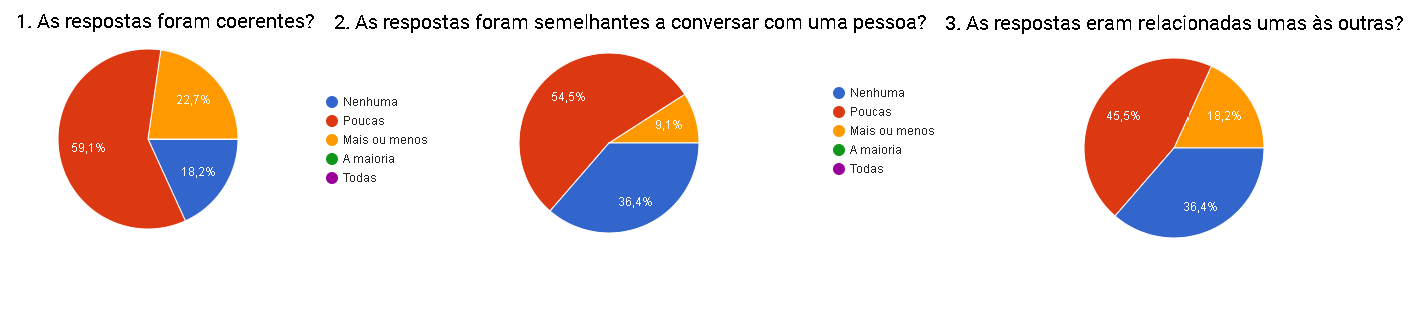
Figura Respostas Pós-interação Wit³



Fonte: Autor (2018).

O modelo criado a partir do *corpus* de legenda se com avaliações melhores possuindo 59,1% usuários o avaliando como pouco coerente comparado aos 14,3% do modelo derivado do *corpus* Wit³. Este fato pode demonstrar que mesmo com o número de sentenças de treinamento do *corpus* gerado a partir de legendas sendo expressivamente menor, a qualidade dos dados, específicos de conversação exerceram um impacto positivo no modelo, sendo que talvez o com um maior número de frases poderiam ser gerados um modelo mais robusto.

Figura Respostas Pós-interação legendas



Fonte: Autor (2018).

# CONCLUSÃO

O desenvolvimento de agentes conversacionais generativos ou baseados em *Deep Learning* com capacidade de gerar conversas de domínio aberto é um tópico aberto de pesquisa com muitos esforços direcionados ao seu estudo. Mesmo como um campo de pesquisa em aberto os *chatbots* generativos são aplicados atualmente a diversas tarefas quando utilizados em um domínio fechado como por exemplo educação, suporte técnico ou entretenimento. Há um crescente interesse na utilização destes agentes aplicados ao atendimento ao cliente e suporte técnico, automatizando as tarefas mais simples gerando assim maior economia as organizações que prestam estes serviços.

Este é o segundo trabalho a abordar a construção de *chatbots* baseados em *Deep Learning* para a língua Português Brasileiro. Para atingir os objetivos deste trabalho foram pesquisadas as principais técnicas de PLN aplicados no desenvolvimento de *chatbots*, os tipos de *chatbots* existentes, a arquitetura clássica da rede *sequence-to-sequence,* o processo de criação *word embeddings* e a sua utilização em *chatbots* construídos através da arquitetura *sequence-to-sequence*, a criação de recursos próprios de *dataset* visto a escassez de *corpus* específico de conversação e a pesquisa de *corpus* existentes para utilização no trabalho.

O resultado obtido por meio do *corpus* gerado por legendas da série americana *The Middle* se aproximou mais do resultado obtido pelo *chatbot* desenvolvido por Maeda e Soares (2017) ao usar o *corpus* baseado nos filmes de Star Wars e ao trabalho de Vinyals e Le (2015) ao realizar o treinamento a partir do histórico de um *chat* de suporte de TI. O nível de coerência semântica não seguiu o esperado, possivelmente devido a falhas na etapa de pré-processamento dos dados gerando uma rede neural com fortes *bias* ou viés a sentenças com maiores frequências nos *corpora*.

O problema de *bias* na rede tem maior criticidade no *chatbot* Wit³evidenciado por diversas sentenças que apareceram como respostas durante as interações com usuários como por exemplo o *token* “ (risos) ” ou “ (aplausos) ”. Estes *tokens* caso tivessem sido removidos antes do treinamento da rede, na etapa de pré-processamento, provavelmente ocasionariam em respostas mais coerentes do *chatbot*. Uma hipótese para a solução deste problema seria a mudança em seu pré-processamento, removendo textos entre parênteses e utilizar apenas frases terminando em sinal de interrogação e sua frase seguinte, esta estratégia possivelmente geraria uma melhora expressiva nos resultados obtidos, uma vez que os pares de pergunta e resposta geram melhores resultados como os alcançados através do *corpus* de legendas.

É importante ressaltar que os modelos criados não utilizaram o mesmo número de camadas ocultas que os trabalhos de Vynials e Le (2015) devido a limitações de poder computacional disponível para a execução deste trabalho, é possível que a geração de modelos com os mesmos parâmetros de *hidden units,* máximo de épocas e *learning rate* produzam resultados mais próximos ao estado da arte. Referente aos objetivos parciais do trabalho todos eles foram atendidos visto que eram etapas necessárias para construção dos agentes conversacionais, porém importante ressaltar que os modelos gerados ficaram abaixo dos resultados obtidos pelo estado da arte como dito anteriormente.

Outro ponto observado foi que o modelo de *chatbot* baseado em legendas mesmo treinado com aproximadamente um quinto da quantidade de frases do modelo baseado no corpus Wit³ se saiu ligeiramente superior, tanto nos testes preliminares quanto com os testes com o público em geral. Assim como o trabalho de Maeda e Soares (2017) uma das principais dificuldades foi a execução da pesquisa sem os recursos de *corpus* orientados exclusivamente a diálogos, como os resultados do trabalho apontam que a utilização de *corpus* de domínio aberto não gera modelos de qualidade similar esta questão se mostra um problema ainda maior.

Por fim dado a falta de trabalhos que abordem a utilização de técnicas de *Deep Learning* para a criação de *chatbots* torna-se necessário a criação de um *corpus* de conversas para sanar o principal impeditivo para a evolução deste tópico de pesquisa no Brasil e a geração de um maior número de agentes conversacionais baseados em *Deep Learning* para a língua portuguesa. Outro ponto interessante a ser abordado como trabalho futuro seria a utilização de outras variações do modelo seq2seq como por exemplo o modelo seq2seq com o mecanismo de *attention* que induz uma maior variância no modelo ou ainda técnicas baseadas em aprendizado por reforço.

# REFERÊNCIAS BIBLIOGRÁFICAS

ABADI, Martín et al. TensorFlow: A System for Large-Scale Machine Learning. In: **OSDI**. 2016. p. 265-283.

AFONSO, Susana et al. Floresta sintá (c) tica: a treebank for Portuguese. In: quot; In Manuel González Rodrigues; Carmen Paz Suarez Araujo (ed) Proceedings of the Third International Conference on Language Resources and Evaluation (LREC 2002) (Las Palmas de Gran Canaria Espanha 29-31 de Maio de 2002) Paris: ELRA. ELRA, 2002.

BATISTA, Gustavo Enrique de Almeida Prado et al. **Um ambiente para avaliação de algoritmos de aprendizado de máquina simbólico utilizando exemplos**. 1997. Tese de Doutorado. Universidade de São Paulo.

BENGIO, Yoshua et al. A neural probabilistic language model. **Journal of machine learning research**, v. 3, n. Feb, p. 1137-1155, 2003.

BISHOP, Christopher M. Pattern Recognition and Machine Learning. Cambridge, 2006. 758 p.

CAHN, Jack. **CHATBOT: Architecture, Design, & Development**. 2017. Tese de Doutorado. University of Pennsylvania.

OLAH, Christopher. Understanding LSTM Networks, 2015. Disponível em:<http://www. http://colah.github.io/posts/2015-08-Understanding-LSTMs/>. Acesso em: 05 de abr. 2018.

FERREIRA, Alessandro dos Santos. Redes Neurais Convolucionais Profundas na Detecção de Plantas Daninhas em Lavoura de Soja. 2017. Dissertação de Mestrado.

GOLDSCHMIDT, Ronaldo Ribeiro. Uma Introdução à Inteligência Computacional: fundamentos, ferramentas e aplicações. **Rio de Janeiro Brasil: IST-Rio**, p. 32, 2010.

GOODFELLOW, Ian; BENGIO, Yoshua; COURVILLE, Aaron. **Deep learning**. MIT press, 2016.

HARTMANN, Nathan et al. Portuguese Word Embeddings: Evaluating on Word Analogies and Natural Language Tasks. **arXiv preprint arXiv:1708.06025**, 2017.

HAYKIN, Simon S. **Neural networks and learning machines**. Pearson Prentice Hall, 2009, p. 938.

HOCHREITER, Sepp; SCHMIDHUBER, Jürgen. Long short-term memory. **Neural computation**, v. 9, n. 8, p. 1735-1780, 1997.

INDURKHYA, Nitin; DAMERAU, Fred J. (Ed.). **Handbook of natural language processing**. CRC Press, 2010.

JAITLY, Navdeep et al. An online sequence-to-sequence model using partial conditioning. In: **Advances in Neural Information Processing Systems**. 2016. p. 5067-5075.

JURAFSKY, Dan; MARTIN, James H. **Speech and language processing**. London:: Pearson, 2014.

JOST, Ingo. Aplicação de Deep Learning em dados refinados para Mineração de Opiniões. 2015.

KALCHBRENNER, Nal; GREFENSTETTE, Edward; BLUNSOM, Phil. A convolutional neural network for modelling sentences. **arXiv preprint arXiv:1404.2188**, 2014.

KOTTUR, Satwik; WANG, Xiaoyu; CARVALHO, Vitor R. Exploring Personalized Neural Conversational Models.

KRIZHEVSKY, Alex; SUTSKEVER, Ilya; HINTON, Geoffrey E. Imagenet classification with deep convolutional neural networks. In: **Advances in neural information processing systems**. 2012. p. 1097-1105.

LECUN, Yann; BENGIO, Yoshua; HINTON, Geoffrey. Deep learning. **nature**, v. 521, n. 7553, p. 436, 2015.

LECUN, Yann et al. Backpropagation applied to handwritten zip code recognition. **Neural computation**, v. 1, n. 4, p. 541-551, 1989.

LECUN, Yann et al. Gradient-based learning applied to document recognition. **Proceedings of the IEEE**, v. 86, n. 11, p. 2278-2324, 1998.

LEVY, Omer; GOLDBERG, Yoav. Dependency-based word embeddings. In: Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers). 2014. p. 302-308.

LEVY, Omer; GOLDBERG, Yoav. Dependency-based word embeddings. In: Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers). 2014. p. 302-308.

LEVY, Omer; GOLDBERG, Yoav; DAGAN, Ido. Improving distributional similarity with lessons learned from word embeddings. **Transactions of the Association for Computational Linguistics**, v. 3, p. 211-225, 2015.

LIDDY, Elizabeth D. Natural language processing. 2001.

LINGUATECA. Acesso a corpos de português: Projecto AC/DC: corpo NILC/São Carlos, 2015. Disponível em: < https://www.linguateca.pt/ >. Acesso em 1 de dezembro de 2017.

LOVINS, Julie Beth. Development of a stemming algorithm. **Mech. Translat. & Comp. Linguistics**, v. 11, n. 1-2, p. 22-31, 1968.

MAEDA, A. C.; MORAES, S. M. W. Chatbot baseado em Deep Learning: um Estudo para Língua Portuguesa. 2017.

MACHINE LEARNING PORTO ALEGRE. Chatbots e Deep Learning. Porto Alegre, 2017, (50 min). Disponível em:< https://www.youtube.com/watch?v=WFMCEGyBzjQ&t=95s >. Acesso em: 20 mar. 2018.

MANNING, Christopher D.; SCHÜTZE, Hinrich. **Foundations of statistical natural language processing**. MIT press, 1999.

MAURO, Cettolo; CHRISTIAN, Girardi; MARCELLO, Federico. Wit3: Web inventory of transcribed and translated talks. In: **Conference of European Association for Machine Translation**. 2012. p. 261-268.

MIKOLOV, Tomas; LE, Quoc V.; SUTSKEVER, Ilya. Exploiting similarities among languages for machine translation. **arXiv preprint arXiv:1309.4168**, 2013.

MIKOLOV, Tomas et al. Efficient estimation of word representations in vector space. arXiv preprint arXiv:1301.3781, 2013.

MITCHELL, Tom M. Machine Learning. McGraw Hill, 1997.

MONARD, Maria Carolina; BARANAUSKAS, José Augusto. Conceitos sobre aprendizado de máquina. **Sistemas inteligentes-Fundamentos e aplicações**, v. 1, n. 1, p. 32, 2003.

MORAES, Silvia; MACHADO, Rackel. Chatterbot for Education: a Study based on Formal Concept Analysis for Instructional Material Recommendation. In: **Brazilian Symposium on Computers in Education (Simpósio Brasileiro de Informática na Educação-SBIE)**. 2016. p. 1347.

POMMERANZENBAUM, Igor Ramalho. REDES NEURAIS ARTIFICIAIS NA PREDIC AO DAS PRINCIPAIS SÉRIES DO ÍNDICE IBOVESPA E SUAS APLICAC OES EM SISTEMAS AUTOMATIZADOS DE NEGOCIAC AO. 2014. Tese de Doutorado. Universidade Federal do Rio de Janeiro.

PORTER, Martin F. An algorithm for suffix stripping. **Program**, v. 14, n. 3, p. 130-137, 1980.

PRAKASH, Abhay; BROCKETT, Chris; AGRAWAL, Puneet. Emulating human conversations using convolutional neural network-based IR. **arXiv preprint arXiv:1606.07056**, 2016.

PRODANOV, Cleber Cristiano; DE FREITAS, Ernani Cesar. Metodologia do Trabalho Científico: Métodos e Técnicas da Pesquisa e do Trabalho Acadêmico-2ª Edição. Editora Feevale, 2013

RADZIWILL, Nicole M.; BENTON, Morgan C. Evaluating Quality of Chatbots and Intelligent Conversational Agents. **arXiv preprint arXiv:1704.04579**, 2017.

RUSSELL, Stuart; NORVIG, Peter; INTELLIGENCE, Artificial. A modern approach. **Artificial Intelligence. Prentice-Hall, Egnlewood Cliffs**, v. 25, p. 27, 1995.

ROCHA, Paulo Alexandre; SANTOS, Diana. CETEMPúblico: Um corpus de grandes dimensões de linguagem jornalística portuguesa. quot; In Maria das Graças Volpe Nunes (ed) V Encontro para o processamento computacional da língua portuguesa escrita e falada (PROPOR 2000) (Atibaia SP 19-22 de Novembro de 2000) São Paulo: ICMC/USP, 2000.

SAMUEL, Arthur L. Some studies in machine learning using the game of checkers. **IBM Journal of research and development**, v. 3, n. 3, p. 210-229, 1959.

SHAWAR, Abu; ATWELL, E. S. A chatbot system as a tool to animate a corpus. **ICAME Journal: International Computer Archive of Modern and Medieval English Journal**, v. 29, p. 5-24, 2005.

SOUZA, LS d; MORAES, Silvia Maria Wanderley. Construção automática de uma base AIML para chatbot: um estudo baseado na extração de informações a partir de FAQs. **Anais do XII ENIAC**, p. 137-141, 2015.

SPANHOL, Tamira Silva et al. Um estudo sobre a interação entre usuários e chatterbots. 2017.

SUTSKEVER, Ilya et al. On the importance of initialization and momentum in deep learning. In: **International conference on machine learning**. 2013. p. 1139-1147.

SUTSKEVER, Ilya; VINYALS, Oriol; LE, Quoc V. Sequence to sequence learning with neural networks. In: **Advances in neural information processing systems**. 2014. p. 3104-3112.

TAN, Pang-Ning; STEINBACH, Michael; KUMAR, VIPIN. Introduction to Data Mining. Pearson Addison-Wesley, 2005, p.792.

TURING, Alan M. Computing machinery and intelligence. Mind, v. 59, n. 236, p. 433-460, 1950.

VINYALS, Oriol; LE, Quoc. A neural conversational model. **arXiv preprint arXiv:1506.05869**, 2015.

WANG, Haohan; RAJ, Bhiksha; On the origin of deep learning. **arXiv preprint arXiv:1702.07800**, 2017.

# APÊNDICE A – Questionário pré-interação

Questionário aplicado com usuários de teste antes da interação com o *chatbot*

1. Qual sua idade?

( ) Menos de 18 anos

( ) 18 - 25 anos

( ) 25 - 35 anos

( ) 35 - 45 anos

( ) Acima de 45 anos

1. Escolaridade

( ) Ensino Fundamental

( ) Ensino Médio

( ) Ensino Superior Completo

( ) Ensino Superior Incompleto

( ) Pós-graduação

1. Qual sua área de estudo?

( ) Ciências Exatas e Tecnológicas

( ) Ciências Sociais Aplicadas

( ) Ciências Humanas

( ) Ciências da Saúde

1. Você já ouviu falar no termo Chatbot ou Agente Conversacional?

( ) Sim

( ) Não

1. Você já interagiu com algum Chatbot?

( ) Sim

( ) Não

# APÊNDICE B – Questionário Pós-interação

Questionário aplicado com usuários de teste após a interação com o *chatbot*

1. As respostas foram coerentes?

( ) Nenhuma

( ) Poucas

( ) Mais ou menos

( ) A maioria

( ) Todas

1. As respostas foram semelhantes a conversar com uma pessoa?

( ) Nenhuma

( ) Poucas

( ) Mais ou menos

( ) A maioria

( ) Todas

1. As respostas eram relacionadas umas às outras?

( ) Nenhuma

( ) Poucas

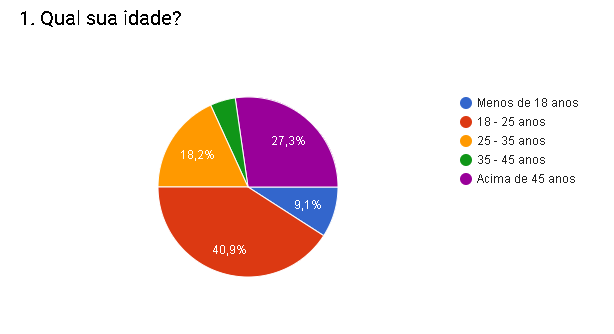
( ) Mais ou menos

( ) A maioria

( ) Todas

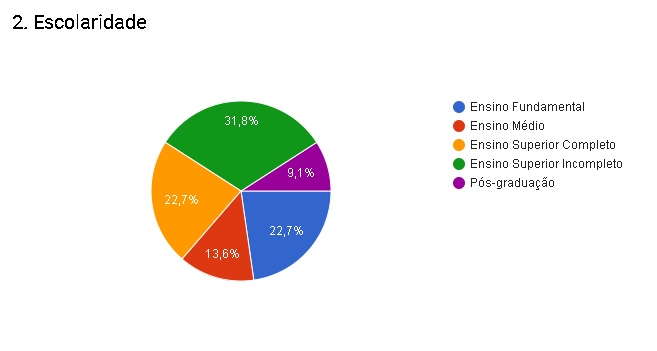
# APÊNDICE C – Respostas Questionário pré-interação

Figura Questionário Pré-interação- Resposta 1



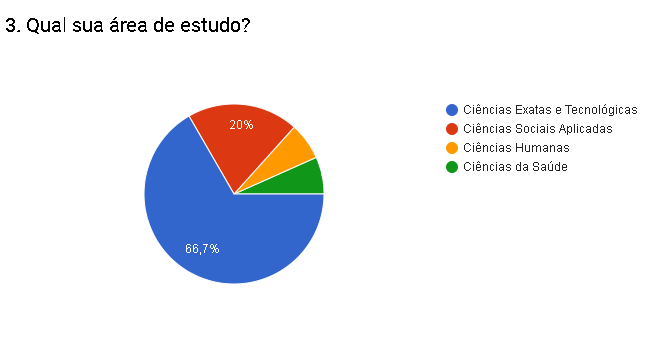
Fonte: Autor (2018).

Figura Questionário Pré-interação- Resposta 2



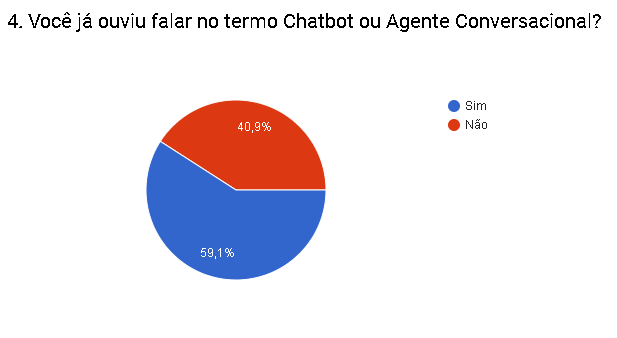
Fonte: Autor (2018).

Figura Questionário Pré-interação- Resposta 3



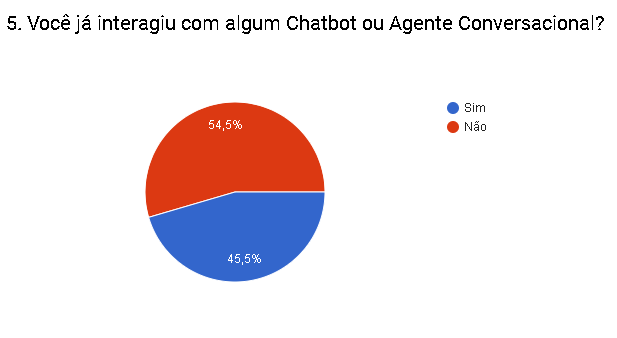
Fonte: Autor (2018).

Figura Questionário Pré-interação- Resposta 4



Fonte: Autor (2018).

Figura Questionário Pré-interação- Resposta 5



Fonte: Autor (2018).