Chatbot baseado em rede neural Long Short-Term Memory (LSTM): um estudo de caso baseado no livro William Shakespeare

Vinicius Dalla Corte[1], Vagner Kaefer Dos Santos[1], Dalcimar Casanova[1]

[1]Programa de Pós-Graduação em Engenharia Elétrica (PPGEE) – Universidade Tecnológica Federal do Paraná (UTFPR)

Via do Conhecimento, Km 1, 85.503-390 - Pato Branco - PR - Brazil

dallacorte.eng@gmail.com, vagner@kaefer.eng.br, dalcimar@utfpr.edu.br

Abstract. *Chatbots have been increasingly used these days for presenting advanced solutions in the field of artificial intelligence, being able to offer great advantages in commercial applications. The purpose of this article is to describe the operation and implementation of a chatbot that uses a Long Short-Term Memory (LSTM) network. In this work were used Machine Learning and Natural Language Processing techniques to demonstrate the LSTM network that can understand the writing of the Portuguese language written in a short story book. After training, chatbot receives a phrase and continues its writing up to a predefined character limit. Better results were obtained with nearly one hundred to four hundred training times, with a correct percentage of up to 87% in relation to the correct writing of formal grammar.*

Resumo. *Chatbots vêm sendo cada vez mais utilizados nos dias de hoje por apresentarem soluções avançadas no campo da inteligência artificial, sendo capazes de oferecer grandes vantagens em aplicações comerciais. O presente artigo tem como objetivo descrever o funcionamento e implementação de um chatbot que utiliza uma rede Long Short-Term Memory (LSTM). No trabalho utilizou-se técnicas de Aprendizado de Máquina e Processamento de Linguagem Natural para demonstrar a rede LSTM que consegue compreender a escrita da língua portuguesa redigido em um livro de contos. Após o treinamento, o chatbot recebe uma frase e continua a sua escrita até um limite de caracteres pré-definido. Obteve-se melhores resultados com cerca de cem até quatrocentas épocas de treinamento, apresentando uma porcentagem de acerto de até 87% em relação à escrita correta da gramática formal.*

1. Introdução

Chatbot é um programa de computador que busca comunicar-se com usuários em linguagem natural, por meio da troca de entrada e saída de mensagens em diferentes formas (via texto, voz ou por ambas formas) [Abushawar e Atwell 2015]. Esses programas de diálogo já vêm sendo utilizados em diversas aplicações e cada vez mais despertam o interesse comercial de empresas, seja para automatizar o suporte técnico ou

também permitir uma interação homem-máquina em aplicativos de uma forma mais natural ou para substituir menus por entradas de texto ou voz [Jurafsky and Martin 2009].

A primeira aplicação de chatbot foi desenvolvido por Joseph Weinzenbaum no ano de 1966 no MIT, nomeado como ELIZA. Utilizava-se de correspondências entre palavras-chave para simular uma consulta a uma psicóloga por meio de respostas fornecidas a usuários. Com o avanço das interfaces gráficas e estudos na área de inteligência artificial arquiteturas de chatbots foram criadas e otimizadas, desde as mais simples até arquiteturas complexas com níveis avançados de aprendizado de máquina e inteligência artificial [AbuShawar e Atwell 2016].

Os modelos mais simples de chatbots são compostos geralmente por árvores de decisão e utilizam-se de menus nas quais o usuário escolhe certa opção (entrada) e o programa retorna uma resposta final (saída). Um segundo modelo de chatbot são baseados em reconhecimentos de palavras-chave, que identificam palavras escritas pelo usuário para retornar certa resposta, errando em alguns casos. Algumas versões híbridas combinam estas duas versões de bots [Comarella e Café 2008]. O terceiro modelo, foco deste trabalho, são os chatbots contextuais. Estes agentes conversacionais fazem o uso de redes neurais em conjunto com técnicas de aprendizado de máquina e inteligência artificial, processando informações de forma recorrente, de modo a serem capazes de aprender dependências ao longo do tempo [Maeda e Moraes 2017].

O objetivo, portanto, é criar um chatbot capaz de realizar uma conversação funcional entre humano e máquina. Dentro desse terceiro modelo existem dois diferentes tipos de chatbots são encontrados na literatura. O primeiro é baseado em palavras chaves, ou seja, dada uma sequência anterior de palavras apresentadas durante a conversação, tenta-se prever a próxima palavra de forma a continuar a comunicação no mesmo contexto. O segundo tipo é baseado em caracteres, ou seja, dada uma sequência de caracteres o método inteligente tenta prever qual será o próximo caractere a fim de formar uma palavra completa.

Obviamente o primeiro tipo de chatbot tem uma facilidade maior de manter o contexto, uma vez que as palavras fazem mais sentido que simples caracteres soltos. Todavia, o aprendizado é mais complicado pois existe uma variedade maior de palavras do que de letras, tornando o primeiro modelo muito mais esparso no ponto de vista das entradas apresentadas.

O foco deste trabalho é implementar o segundo tipo de chatbot (i.e. baseado em caracteres) com objetivo de aprender a grafia e alguma forma de comunicação contextual primitiva e concomitantemente compreender melhor o problema e as formas de resolução presentes na literatura. Futuramente espera-se implementar o primeiro tipo (i.e. baseado em palavras) cujo resultado da conversação espera-se ser mais natural.

2. Arquitetura do Agente Conversacional

Para criar um chatbot capaz aprender e escrever a grafia correta de palavras, a forma mais eficaz é através do treinamento baseado em caracteres de um texto. Posteriormente o mesmo deverá ser capaz de gerar a escrita correta das palavras aprendidas, e se possível, manter o contexto do diálogo. Nesse sentido buscou-se na

literatura uma rede baseada em inteligência artificial que fosse capaz de atender estes requisitos.

Atualmente ainda há uma grande dificuldade de se trabalhar com linguagens naturais pelo fato de existirem diversos fatores momentâneos, sentimentais, ambiguidades e duplos sentidos que dificultam a aprendizagem das máquinas, pois elas precisam entender não somente uma frase específica, mas ter uma forma de entender o enredo do que está acontecendo. Na literatura ainda são poucos trabalhos relacionados com chatbots em português que compreendem problemas complexos no domínio sequencial e temporal de um diálogo. Segundo [Sutskever et al. 2014] é necessário um grande trabalho de engenharia para resolver a limitação do mapeamento e modelagem de uma base de diálogo.

A insuficiência da utilização de redes clássicas como perceptron, perceptron multicamadas ou convolucional se deu pelo fato de não conseguirem levar em consideração uma sequência de fatos ao longo do tempo, em específico o contexto de caracteres em um texto, pois os mesmos não trabalham de forma recorrente. Redes neurais recorrentes (RNNs) conseguem trabalhar muito bem com sequências (textos, sinais médicos ou áudios) “aprendendo” a dependência em que os dados ocorrem e avaliando-os ao longo do tempo. Esta capacidade se dá pelo fato de existir uma conexão entre suas camadas, permitindo a rede “lembrar-se” de certas informações, atuando bem em problemas de memória a curto prazo. Porém possuem um problema de aprendizado em problemas complexos, dificultando a conexão de informações à medida que precisam se recordar da informação em períodos de tempo muito antigos [Barreto 2002].

Por fim, buscou-se na literatura um modelo de rede neural que seja capaz de lidar com dependências de longo prazo, neste caso longas sequências de caracteres. *Redes Long Short-Term Memory (LSTM)* que são o modelo de escolha deste trabalho, permitem utilizar técnicas de modo a compreender longos textos em linguagem natural, obtendo resultados satisfatórios.

2.1. Redes Long Short-Term Memory (LSTM)

Redes de memória a longo prazo – usualmente conhecidas como “LSTMs” – foram desenvolvidas inicialmente por [Hochreiter & Schmidhuber 1997] com o objetivo de resolver o problema de dependência de longo prazo das redes RNNs. Este problema (conhecido também como *Exploding Gradient* ou *Vanishing Gradient*) faz com que informações importantes se percam quando a rede lida com grandes sequências de dados. Portanto, as redes LSTM trabalham para recordar a informação por longos períodos de tempo, muito maiores que a RNNs, mesmo com certa limitação [Penha et al. 2018].

A arquitetura de uma LSTM padrão utiliza um mecanismo denominado células de memória que permite que a rede consiga recordar a informação mesmo depois de diversas iterações. Consiste também de válvulas de três unidades multiplicativas: válvula de entrada, válvula de saída e válvula de esquecimento. Estas válvulas têm a função de manter o fluxo do erro constante através de unidades especiais chamadas “*gates*” (portões) permitindo os ajustes de pesos da mesma forma que o truncamento da sequência quando a informação não é necessária, simbolizando um esquecimento [Nelson 2017].

Segundo [Nelson 2017] desde sua criação, este método foi ramificado em diversas variações. Entretanto, quando avaliadas em relação à original apresentado por [Greff et al. 2015] conforme Figura 1, nenhum modelo foi capaz de apresentar melhoria considerável em termos de resultados.

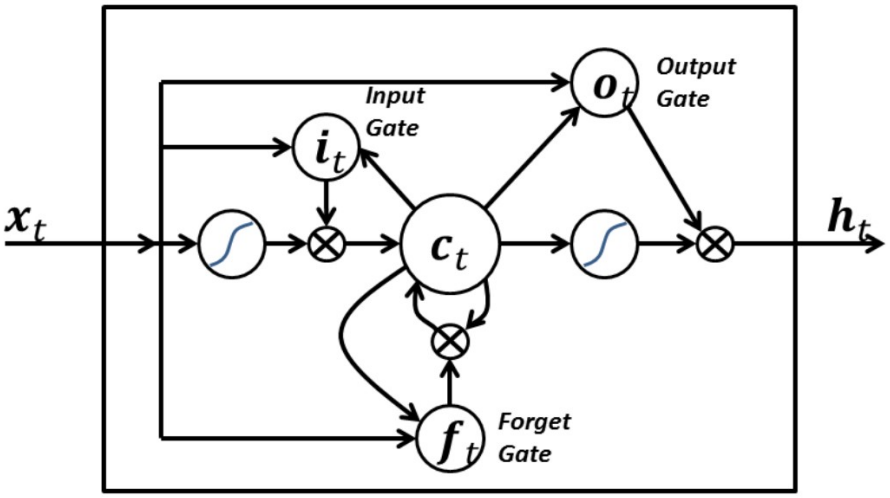

Figura 1. Redes Long short-term memory [Greff et al., 2015]

Internamente à rede LSTM cada elemento apresenta sua função, de modo que:

- Cell State: Possui a função mais importante no núcleo da LSTM. Permite através de interações lineares que há informações flua através da rede ao longo do tempo.
- Forget Gate: Tem a atribuição de decidir se informação que vem do estado anterior vai ser descartada ou mantida.
- Input Gate: Possui a capacidade de decidir qual informação que vem da entrada vai ser inserida e combinado com as anteriores.
- Output Gate: Determina quais partes serão enviadas à saída.

3. Chatbot baseado em rede neural recorrente LSTM

O primeiro passo necessário para constituir o chatbot utilizando uma rede LSTM consiste em mapear a base de dados de treinamento. Este processo teve o intuito de analisar todas as letras contidas na base e transformá-las uma a uma em um número inteiro, processo este que gerou um mapeamento (dicionário).

A rede neural não interpreta letras, ela recebe somente a sequência de números que serão aprendidos. Todos os processos de treinamento, testes e resultados foram realizados com números inteiros positivos e somente na etapa final é que esses números foram transformados novamente em letras, com a utilização do dicionário. A Figura 2 exibe um exemplo da tradução para números e sua função inversa (utilizada no final de todo processo) para letras.

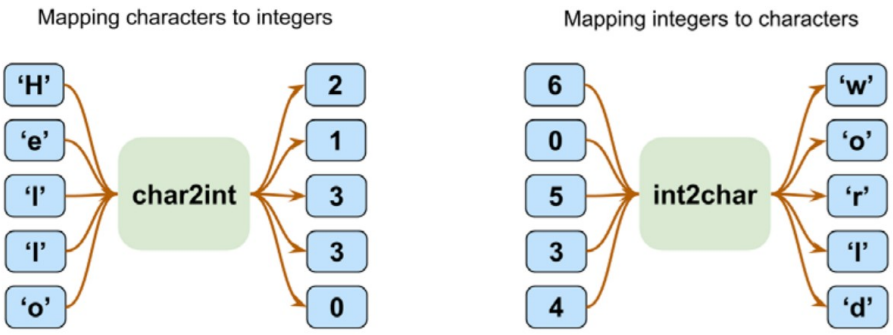

Figura 2. Exemplo de tradução de letras para números e função inversa [Raschka e Mirjalili et al., 2017]

Após a criação do dicionário de letras a próxima etapa é realizar o treinamento da rede neural. A preparação da base de treino é realizada da seguinte forma: todo o texto de entrada é percorrido e inserido na matriz de entrada x, ao mesmo tempo o termo seguinte da palavra inserida na matriz x é adicionado na matriz de respostas y. A Figura 3 exibe um exemplo de como este processo é realizado:

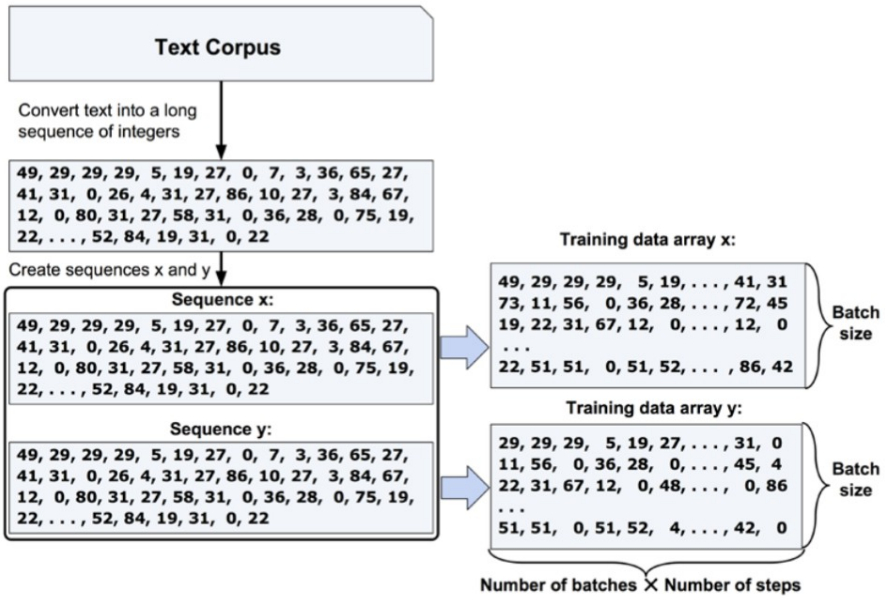

Figura 3. Exemplo de preparação das informações para o treinamento [Raschka e Mirjalili et al., 2017]

Normalmente os chatbots tradicionais recebem uma pergunta e geram certa resposta baseada em seu aprendizado. No caso deste trabalho e de chatbots que tem seu treinamento baseados em livros e histórias, o mesmo funciona mais como um historiador do que um dialogador. Ele recebe uma frase inicial e irá continuar a frase, gerando uma história baseada com o que aprendeu do livro de treinamento. Portanto, espera-se que o texto gerado pela inteligência artificial possua um engajamento com as terminologias e conteúdo do livro base.

Ao utilizar essa abordagem, podemos simplificar que possuímos somente uma entrada sequencial de dados. Como trabalhou-se com um livro, a matriz de entrada e resposta pode possuir valores elevados (i.e. tamanho total do livro). Para facilitar o gerenciamento do treinamento, onde o mesmo foi dividido em *batches* (lotes). Assim, a entrada foi dividida em matrizes e treinada em lotes, onde durante o treinamento de cada lote foi criado um *checkpoint* para continuação futura, em casos de qualquer erro o treinamento não precisa ser reiniciado de sua etapa inicial.

A quantidade de épocas de treinamento é uma das principais variáveis que devem ser configuradas nos algoritmos de treinamento, essa é a variável que define o alcance do nível de inteligência desejado.

Após todos os processos citados anteriormente serem efetuados, selecionou-se uma sequência aleatória de frases do livro de treinamento e em seguida a rede gerou certa continuação daquela frase até um limite de caracteres preestabelecidos.

4. Experimento

Como entrada optou-se por utilizar o livro “A tragédia de Hamlet, príncipe da Dinamarca” de William Shakespeare. Como o número de palavras é elevado, foi definido que a entrada seria dividida em 64 lotes, cada um com aproximadamente 2.500 termos. Juntamente, foram definidos que a rede neural teria seria treinada com 50, 150, 200, 300, 400 e 500 épocas de treinamento. Para padronização, foram fixadas as seguintes variáveis na implementação da rede neural:

- Tamanho do LSTM: 128
- Camadas da rede neural: 1
- Taxa de aprendizado: 0.001

O texto de treinamento continha ao total 77.061 palavras da língua portuguesa, totalizando assim 440.178 caracteres. O texto utilizado para treinamento não sofreu nenhum tipo otimização, utilizado o mesmo de forma bruta. Do total de caracteres, se fossem removidos os espaços entre as palavras, teríamos 369.013 caracteres para aprendizado. Após o treinamento, foi selecionada de forma aleatória uma frase do livro base, que foi utilizada como a entrada da IA treinada.

5. Resultados

A rede LSTM foi treinada com diversas épocas. Para cada treinamento se analisou os acertos e/ou erros ortográficos e erros sintáticos de uma saída contendo 3.000 caracteres. Essa contagem fez o uso do corretor ortográfico MS World, da Microsoft. A Tabela 1 apresenta os resultados obtidos para cada um dos cenários testados.

Tabela 1. Resultados do algoritmo LSTM implementado

Épocas	Palavras	Acertos	Erros Ortográficos	Erros Sintáticos	Porcentagem Acerto (%)
50	547	416	131	11	76.05
100	542	457	85	28	84.32
150	550	461	89	30	83.82
200	530	453	77	19	85.47
300	543	472	71	35	86.92
400	525	446	79	3	84.95
500	509	450	88	15	82.71

Observou-se que a rede LSTM tende a aumentar sua taxa de acerto conforme se utiliza um número maior de épocas. Todavia, aparentemente um número muito elevado pode levar a *overfitting* (sobre-ajuste) dos pesos, fenômeno pelo qual a rede decora os exemplos em vez de generalizar e aprender. Neste caso, para épocas maiores que 300 a rede não apresentou uma melhora significativa no desempenho, utilizando apenas processamento computacional maior.

O treinamento foi realizado com a sequência de caracteres, logo espera-se em princípio que a rede aprenda de forma satisfatória a gramática e a ortografia. Como pode ser visto nas duas saídas abaixo, a rede conseguiu manter uma boa escrita, escrevendo corretamente a grafia das palavras. Porém, as frases não transmitem uma sequência contextual coerente para humanos, o que já era esperado. Também há de se salientar que os caracteres de acentos, espaços e pontuações também são codificados e inseridos como treinamento da rede, logo as saídas abaixo apresentadas refletem fielmente o obtido pela metodologia. Os trechos de saída foram gerados com a rede LSTM e 300 ciclos de treinamento:

“– Nada. Ela entrou para olhos alguns pessoas. Na pessoa deles. A pessoa pessoal de alam e eles possava com a mensagem. Não tinha amigo pretesso da expressão de criser ao caso. Nenhum dareiro para o modo certo alguma coisa está parta da pessoa a papelme.”

“– A mesma caixa da seu arminho a minha corde, alguma corente como este dia até o pessoal. Ele ponsa sobre a mesma cara casa. No seu senhor este depoinde estavam espastos por deixados diferante, para a cabeça ansioso dela?”

“– O corpe desses pontos do pedido. Agoro, a cortina para acontecido, suficiente no caso no crime para com meia do que acha nar a caminhável que se a percerestar o continio esperando a montada.”

“– Numa discistão a permonhas. Ela pode sido para sentir as certas a mesma assastir a senhora. A chave dele estava caras, e olharema a mostrar olhar escritócio a chegada da mulher e altura ponta pensado que a pouca caminha de sou ou e de casa e como a mais assassinato.”

As palavras em sublinhado representam os termos onde a rede neural cometeu erros ortográficos. Porém, percebe-se que o erro foi de apenas um caractere na maioria dos casos.

Observa-se que não faz-se nenhuma diferenciação entre palavras com construções simples (e.g. artigos ‘a’ ‘e’) de construções complexas (e.g. substantivos ‘elefante’). Todavia, percebe-se que aparentemente a rede não tem uma tendência de gerar construções simples, preferindo as construções complexas, o que de certa forma podemos dizer que é um bom resultado. De forma complementar, a Figura 4 exibe um comparativo realizado entre acertos e erros variando-se o número de épocas de treinamento. Verificou-se um bom número de acertos em relação aos erros, sejam ortográficos ou sintáticos. Em relação especificamente aos erros sintáticos, acredita-se que o MS Word não foi capaz de identificar todos devido à falta de contexto nas frases geradas, ou seja, tal número pode estar subestimado e mais estudos necessitam ser realizados para uma melhor avaliação desse parâmetro.

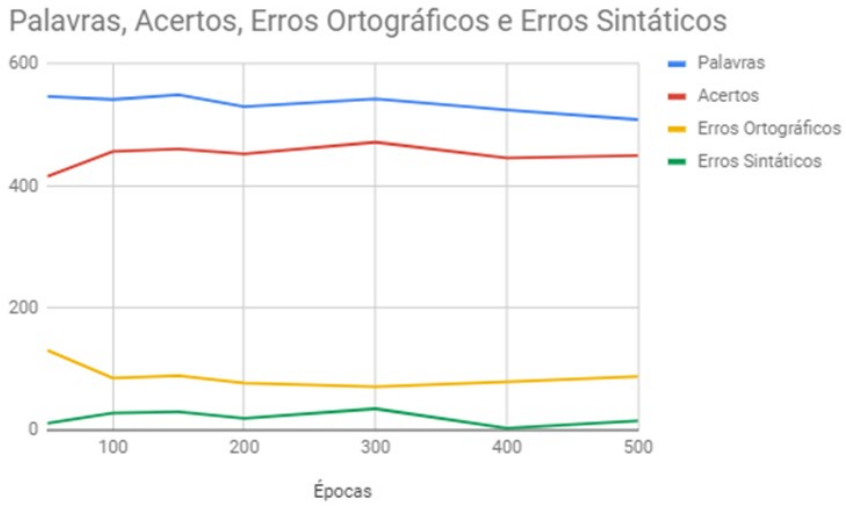

Figura 4. Comparativo exibindo número de palavras geradas, número de palavras corretas (acertos), número de erros ortográficos e sintáticos conforme o número de épocas de treinamento (eixo x).

6. Conclusão

Em primeira instância a rede neural não reproduziu uma história muito coerente. Em vez de uma continuação dos contos do livro de William Shakespeare, encontrou-se um conjunto de frases que seguem um ritmo contínuo, com pouca coerência.

Porém, ao analisar mais cuidadosamente as frases, é possível ver que a rede neural seguiu a correta escrita da língua (gramaticalmente). Outro fato interessante é que a rede conseguiu seguir as normas de língua portuguesa quase em sua totalidade, mesmo sem otimizações, a entrada era bruta, com acentos, espaços, caracteres especiais etc.

Obviamente esses resultados poderiam ser refinados a partir de mais testes, com diferentes parâmetros e aumentando-se o tamanho da entrada. Também há de se considerar que uma semântica mais consistente poderia ser obtida utilizando para treinamento palavras em vez de caracteres individuais. Todavia, o mesmo processo empregado neste trabalho pode ser utilizado para diversas aplicações, possibilitando a

criação de um chatbot que faz o uso de redes neurais recorrentes, o qual, além de aprender a responder questionamentos, pode possuir uma (ou mais) memórias e seguir um fluxo de informações e contextos.

Trabalhos futuros poderiam incluir mais de um livro, mesmo sendo de áreas diversas ou um comparativo com diferentes idiomas, utilizando palavras em vez de letras e então verificar como foi o aprendizado em um âmbito mais geral.

Referências

B. AbuShawar, E. Atwell. (2015) “ALICE Chatbot: Trials and Outputs”. Computación y Sistemas, vol. 19, no. 4, pp. 625–632.

Jurafsky, D. and Martin, J. H. (2009) “Speech and Language Processing”, Prentice-Hall, 2nd Edition.

Hicgreuterm Seep; SCHMIDHUBER, Jurgen. “Long short-term memory”. Neural computation, v. 9, n. 8, p. 1735-1780, 1997.

Bayan AbuShawar and Eric Atwell. (2016) “Automatic extraction of chatbot training data from natural dialogue corpora”. Workshop on Collecting and Generating Resources for Chatbots and Conversational Agents.

Comarella, Rafaela Lunardi; Café, Lígia Maria Arruda. (2008) “CHATTERBOT: conceito, características, tipologia e construção”. Informação & Sociedade: Estudos, v. 18, n. 2.

Maeda, A. C.; Moraes, S. M. W. (2017) “Chatbot baseado em Deep Learning: um Estudo para Língua Portuguesa”.

Sutskever, I., Vynyaks, O., and Le, Q. V. (2014) “Sequence to sequence learning with neural networks”. In Advances in Neural Information Processing Systems 27, Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, and K. Q. Weinberger (Eds.). Curran Associates, Inc., pp. 3104–3112.

Barreto, Jorge M. (2002) “Introduçãos redes neurais artificiais”. V Escola Regional de Informática. Sociedade Brasileira de Computaçao, Regional Sul, Santa Maria, Florianópolis, Maringá, p. 5-10.

Penha, Deyvison de Paiva et al. (2018) “Rede neural convolucional aplicada à identificação de equipamentos residenciais para sistemas de monitoramento não-intrusivo de carga”.

Nelson, David Michel Quirino. (2017) “Uso de redes neurais recorrentes para previsão de séries temporais financeiras”.

Greff, K.; Srivastava, R. K.; Koutník, J.; Steunebrink, B. R. & Schmidhuber, J. (2015). “LSTM: A search space odyssey”. arXiv preprint arXiv:1503.04069.

Raschka, Sebastian and Mirjalili, Vahid. (2017) “Python Machine Learning”. Packt Publishing Ltd. 2nd Edition.