Utilização de Redes Neuronais em contexto Biomédico: Previsão de Texto na elaboração de Relatórios Clínicos

Alessandro Santos [1120198@isep.ipp.pt]

Neurociência Computacional

Mestrado em Engenharia Biomédica

Instituto Superior de Engenharia do Porto

Abstract. O presente projeto procura expor ao leitor o desenvolvimento de um sistema inteligente baseado em Redes Neuronais para a resolução de um problema em contexto biomédico. O problema proposto foi apresentado com maior detalhe no primeiro trabalho, consistindo no processo de redação de um relatório médico durante a descrição das evidências clínicas presentes num estudo imagiológico. Desta forma, é apresentado ao leitor o desenvolvimento de um sistema de previsão de texto, que visa auxiliar o profissional de saúde a elaborar o relatório clínico. Para tal, foi implementada uma Rede Neuronal Recorrente, especializada no processamento de dados sequenciais, utilizando uma variante LSTM. No final, os resultados obtidos são comparados com outra abordagem estatística baseada na utilização de *N-Grams*.

Palavras-chave: *language models*, previsão, *tokens*, LSTM.

1 Introdução

1.1 Âmbito e Objetivos

O presente trabalho e relatório foram desenvolvidos no âmbito da unidade curricular de Neurociência Computacional (NRC), lecionada no primeiro ano do Mestrado de Engenharia Biomédica (MEB), no Instituto Superior de Engenharia do Porto (ISEP).

O objetivo deste trabalho passa pelo desenvolvimento de um sistema computacional, recorrendo à implementação de uma Rede Neuronal (RN), para a resolução de um problema no contexto médico apresentado no primeiro trabalho prático.

1.2 Contexto e Problema

No primeiro trabalho prático foi apresentado ao leitor o fluxo de trabalho do serviço de imagiologia no processo de diagnóstico do cancro da mama, assim como, de que forma a utilização de sistemas baseados em RN podem beneficiar este processo. De seguida, é apresentado o fluxo de trabalho do serviço de imagiologia apresentado no primeiro trabalho:

1. Um paciente, através da auto palpação, identifica um nódulo na mama;

2. Visita o seu médico de família, que confirma a existência desse nódulo;
3. Considerando o nódulo encontrado e outros fatores, como a idade e o histórico de família, existe uma suspeita de cancro da mama;
4. O médico de família (referenciador) procede à requisição de um exame de mamografia;
5. O exame é agendado no serviço de imagiologia consoante a disponibilidade do equipamento;
6. Nesse departamento, o exame é executado por um técnico de radiologia;
7. As imagens são depois enviadas para um médico radiologista;
8. O médico radiologista analisa as imagens e elabora um relatório clínico a descrever as evidências encontradas;
9. Finalmente, o relatório é enviado de volta para o médico referenciador que, no caso de a suspeita se confirmar, decide qual a melhor terapia a seguir.

Para a realização do segundo trabalho prático, é dado particular enfase ao oitavo passo do fluxo apresentado, a elaboração do relatório médico. Considerando que um diagnóstico e tratamento precoce é o principal fator para o sucesso do tratamento do cancro da mama, é fácil compreender a importância da celeridade de todo o processo apresentado e, consequentemente, da fase de elaboração do relatório clínico. Por este motivo é proposto o desenvolvimento de um sistema de previsão de texto que procura acelerar este processo.

1.3 Estrutura do Relatório

No presente capítulo é feita uma breve apresentação do trabalho desenvolvido, sendo explicado o seu âmbito e objetivos; contextualização e apresentação do problema; e ainda a estrutura deste documento. No segundo capítulo é apresentado ao leitor a implementação do projeto, abordando-se alguns dos conceitos teóricos necessários para a compreensão do mesmo; a *stack* tecnológica escolhida; o pré processamento dos dados; e a implementação e configuração da rede neuronal. No terceiro capítulo é apresentada a avaliação da solução implementada, apresentando-se as métricas e hipóteses utilizadas; a metodologia de avaliação; e no final os resultados obtidos, assim como a respetiva análise. Por último, no quarto capítulo, são apresentadas as conclusões do trabalho desenvolvido, apresentando-se o trabalho futuro e limitações; e ainda uma apreciação final.

2 Implementação

2.1 Conceitos

O problema de previsão de texto é conhecido na área das Ciências da Computação por *Language Modeling* (LM), fazendo parte de um conjunto alargado de problemas relacionados com processamento de texto que pertencem à área de *Natural Language Processing* (NLP). O NLP é responsável pela investigação e desenvolvimento de métodos e processos que permitam às máquinas compreender e manipular a linguagem natural das pessoas. Alguns exemplos dos problemas mais comuns que o NLP

visa resolver são: tradução de línguas; extração de informação em texto; reconhecimento da fala; identificação de erros gramaticais e de sintaxe; entre outros (Nadkarni, Machado, & Chapman, 2011).

Na área de LM, a um documento de texto, que no contexto do presente trabalho corresponde a um relatório clínico, é dado o nome de *corpus*, ou *corpora* no plural (Copestake, 2004). Quanto ao elemento atômico dum *corpus*, que pode corresponder a uma palavra ou símbolo, é dado o nome de *token*. Dependendo do problema a resolver, o conceito de *token* pode ser mais abrangente ou restrito. Por exemplo, num problema em que sinais de pontuação não sejam importantes, os mesmos podem ser retirados da definição de *token*, tornando assim o conceito mais restrito (Manning, Raghavan , & Schütze, 2009).

Na previsão de texto deseja-se prever o próximo *token* considerando o seu contexto. Tal pode ser feito com recurso a *language models*, modelos que atribuem uma probabilidade a uma sequência de *tokens*, como é o caso das Redes Neuronais.

As RN procuram imitar o funcionamento dos neurónios dos seres vivos, principalmente no que toca à capacidade de processamento paralelo da informação. No âmbito do presente projeto as RN necessitam de ter em conta alguns dos conceitos anteriormente apresentados, como é o caso dos *tokens* e contexto. Na figura 1 é apresentada uma esquematização simples de uma rede neuronal para a previsão de texto, onde T_1, T_2, T_3, …, T_N são os *tokens* de contexto, e T_{N+1} o *token* previsto.

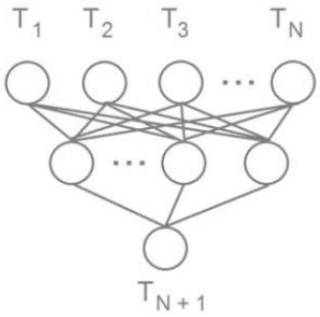

Figura 1- Arquitetura de uma RN no problema de previsão de texto

A camada superior, conhecida por *input layer*, deverá receber os *tokens* a considerar para a previsão, *tokens* de contexto. A camada central, *hidden layer*, que é constituída por uma ou mais camadas, procura transformar o *input* em *output* através da aplicação de uma série de funções. Por último, a última camada, *output layer*, corresponde ao *token* previsto.

2.2 Tecnologias

Um dos primeiros tópicos que foi necessário definir aquando da implementação do presente projeto foi *stack* tecnológica a utilizar. A nível de linguagem de programação foi utilizado o python (versão 3.9), conhecido por ter uma panóplia de *frameworks* e bibliotecas já implementadas para a área de Inteligência Artificial (IA). Para o desen-

volvimento deste sistema foi utilizado o TensorFlow, uma *framework open source* para o desenvolvimento de sistemas de *Machine Learning* (ML) e IA. Em particular foi utilizada a biblioteca Keras, útil para a implementação de RN.

2.3 *Dataset*

Para o desenvolvimento da rede neuronal, em particular no que toca ao processo de treino, foi utilizado um *dataset* privado. Este *dataset* é composto por um total de 53099 relatórios médicos, armazenados numa base de dados em formato HTML.

De forma a reduzir o tempo despendido no processo de treino, foi selecionada uma amostra aleatória de 1000 relatórios clínicos, todos redigidos pelo mesmo profissional de saúde.

2.4 Pré processamento dos Dados

Para um melhor tratamento e processamento dos dados, é ideal realizar uma série de operações que visam a limpeza e uniformização dos mesmos. Desta forma, os relatórios utilizados foram submetidos às seguintes operações de pré processamento:

1. Descodificação do HTML;
2. Conversão para minúsculas;
3. Remoção de espaços brancos extra;
4. Classificação de *tokens*.

Como referido anteriormente, os relatórios utilizados no processo de treino encontram-se armazenados no formato HTML. Há assim a necessidade de, ao processar os relatórios, realizar a descodificação dos mesmos, de forma a remover as *tags* de HTML e extrair o texto dos relatórios. De seguida é apresentado um excerto de um relatório clínico:

```
<p style="text-align: justify">
  Rins em topografia habitual, com    dimens&otilde;es preservadas
(eixo bipolar de 11cm &agrave; esquerda e 11,5cm &agrave; direita),
com contornos regulares.
</p>
```

Após o processo de descodificação e extração do texto o resultado final será:

```
Rins em topografia habitual, com    dimensões preservadas (eixo bi-
polar de 11cm à esquerda e 11,5cm à direita), com contornos regula-
res.
```

De seguida, é realizada a conversão de todo o texto para minúsculas. Este passo é realizado de forma a reduzir o número total de possibilidades existentes na comparação das *strings*. Esta conversão apenas é possível pois a distinção entre maiúsculas e minúsculas não é importante no âmbito da previsão de texto.

```
rins em topografia habitual, com    dimensões preservadas (eixo bi-
polar de 11cm à esquerda e 11,5cm à direita), com contornos regula-
res.
```

Seguidamente é feita uma remoção de todos os espaços brancos extra, ou seja, de todos os espaços brancos repetidos. Esta remoção é bastante simples utilizando expressões regulares. Após a remoção dos espaços brancos o resultado será:

```
rins em topografia habitual, com dimensões preservadas (eixo bipo-
lar de 11cm à esquerda e 11,5cm à direita), com contornos regula-
res.
```

Finalmente, a última operação a ser aplicada é a classificação de *tokens*, onde se atribui uma determinada classe a certos *tokens*. No texto extraído existem alguns *tokens* cujo valor em si não apresenta qualquer benefício para a previsão de texto, mas o tipo (classe) do *token* sim. Por exemplo, o valor de um comprimento não tem qualquer utilidade para a previsão de texto, no entanto, saber que um comprimento surge apôs aquele termo já pode ter alguma utilidade. Assim, sempre que um comprimento é identificado, o mesmo é substituído por um *placeholder*. A identificação destas classes é feita através da utilização de expressões regulares, que identificam datas, volumes, áreas, comprimentos e números. Apôs o processo de classificação de *tokens* o resultado final será:

```
rins em topografia habitual, com dimensões preservadas (eixo bipo-
lar de _comprimento_ à esquerda e _comprimento_ à direita), com
contornos regulares.
```

O texto resultante da aplicação das operações acima descritas, será depois disponibilizado à RN para processamento, sendo gerado o modelo a partir deste.

É importante referir outra operação que, embora não tenha sido utilizada, tem alguma relevância neste tópico, a anonimização dos dados, que consiste na remoção de qualquer elemento do texto que possa permitir a identificação do paciente. Devido à estrutura da base de dados do sistema de onde o *dataset* foi retirado, as informações dos pacientes não se encontram presentes no texto do relatório, nem é habitual os médicos adicionarem essa informação no texto. Por este motivo, este passo foi desprezado no presente trabalho.

2.5 Rede Neuronal

As Redes Neuronais Artificiais (RNA) e Redes Neuronais Convulsionais (RNC) são tipos de Redes Neuronais indicadas para a resolução de problemas de classificação. No entanto, estas redes apresentam uma limitação enorme no que toca à capacidade de resolver problemas com séries temporais, isto por não possuírem capacidades de lidar com dados sequenciais. Para colmatar esta falha, surgiram as chamadas Redes Neuronais Recorrentes (RNR), uma abordagem essencial para a resolução de problemas no âmbito da área de processamento de linguagem natural (Gupta, 2021).

Ainda assim, as RNR têm alguma dificuldade em processar dados temporais demasiado extensos, ou seja, por outras palavras, podem ser vistas como tendo apenas uma *short memory*. Para resolver este problema, foi desenvolvida uma versão mais avançada das RNR conhecida por *Long Short-Term Memory* (LSTM). As LSTM são capazes de modelar sequências e respetivas dependências de forma mais precisa que as RNR convencionais (Mittal, 2019).

Relativamente à arquitetura da RN implementada, é importante definir vários parâmetros como: o número de células na *input layer*; o número de *layers* da *hidden layer*; o número de células em cada uma das *hidden layers*; e ainda, o número de células na *output layer*. O seguinte quadro apresenta esta informação de forma resumida:

Tabela 1 – Características da arquitetura da Rede Neuronal

	Nº de *Layers*	**Nº de Nós**
Input Layer	1	Nº de *tokens* únicos existentes
Hidden Layer	A definir após a avaliação dos modelos	A definir após a avaliação dos modelos
Output Layer	1	Nº de *tokens* únicos existentes

Da tabela apresentada, as únicas incógnitas estão relacionadas com a *hidden layer*, tanto a nível de *layers* a utilizar, como no número de células por *layer*. A definição destes valores apenas é feita depois da escolha do melhor modelo.

Para definir o melhor modelo possível, foram realizados uma série de treinos, com variação em alguns parâmetros, de forma a encontrar a combinação que proporcione o melhor modelo. Os parâmetros considerados foram:

- ***Epochs*:** Corresponde ao número de vezes que o *dataset* completo é revisto pela rede neuronal. Para a presente implementação foi utilizado um valo fixo de 10 *epochs*;
- ***Batch Size*:** É o número de amostras processadas pela rede neuronal antes dos pesos serem atualizados;
- ***Hidden Layers*:** Número de *hidden layers* utilizadas na rede neuronal. Para determinar o número de camadas foram avaliados modelos com uma e duas *layers*;
- ***Hidden Cells*:** Número de células utilizadas em cada *hidden layer* da rede neuronal. Foram avaliados modelos com 128 e 256 células;
- ***Learning Rate*:** É um parâmetro que indica quanto um modelo pode ser alterado após o processamento de um *batch*, ou seja, pode ser visto como a velocidade com que o modelo pode aprender. Um valor demasiado pequeno traduz-se num processo de treino muito longo, enquanto um valor demasiado grande, pode significar um modelo impreciso (Brownlee, 2019).

De seguida, são apresentados os resultados obtidos para cada combinação dos parâmetros de treino. As métricas utilizadas para esta avaliação foram a *accuracy* e o *loss*. A *accuracy* é a percentagem de previsões corretas, sendo particularmente útil para medir a *performance* de um modelo. O *loss* é a diferença entre o valor correto e a previsão, podendo ser visto como um acumular de erros, ou seja, quanto menor for o seu valor, melhor o modelo (Kumar, 2018).

Tabela 2 - *Accuracy* e *Loss* nos vários modelos testados.

ID	*Epochs*	*Layers*	*Batch*	*Learning Rate*	*Hidden Cells*	*Accuracy*	*Loss*
1	10	1	128	0.500	128	0.2306	11.3580
2			128	0.250	128	0.3110	9.0739
3			128	0.150	128	0.4614	5.7375
4			64	0.050	128	0.5706	3.8353
5					256	0.5569	4.3603
6			128		128	0.6474	2.1700
7					256	0.6936	1.9591
8			64	0.010	128	0.6062	2.6951
9					256	0.7397	1.2842
10			128		128	0.5493	2.5362
11					256	0.7419	1.2004
12		2	64	0.050	128	0.5072	3.7216
13					256	0.5248	3.5453
14			128		128	0.5968	2.3053
15					256	0.2758	5.3616
16			64	0.010	128	0.6778	1.6896
17					256	0.7308	1.3230
18			128		128	0.6990	1.4598
19					256	0.7657	1.0567

É importante referir que, devido a limitações de tempo, apenas foi testado uma variação muito finita dos parâmetros de treino. Numa situação em que esta limitação não existisse, seria interessante testar mais modelos com diferentes arquiteturas (nº de *layers*, *hidden cells* e *learning rate*).

Face aos resultados apresentados na tabela 2, é possível verificar que a combinação de parâmetros de treino com maior *accuracy* e menor *loss* é a combinação com o ID 19. Com isto, está definida a arquitetura final da RN a implementar, sendo constituída por 2 *hidden layers*, com 256 nós cada, um *batch* de 128 e um *learning rate* de 0.010.

Tabela 3 – Características definitivas da arquitetura da Rede Neuronal.

	Nº de *Layers*	**Nº de Nós**
Input Layer	1	Nº de *tokens* únicos existentes
Hidden Layer	2	256
Output Layer	1	Nº de *tokens* únicos existentes

Graficamente, a arquitetura da RN pode ser representada de seguinte forma:

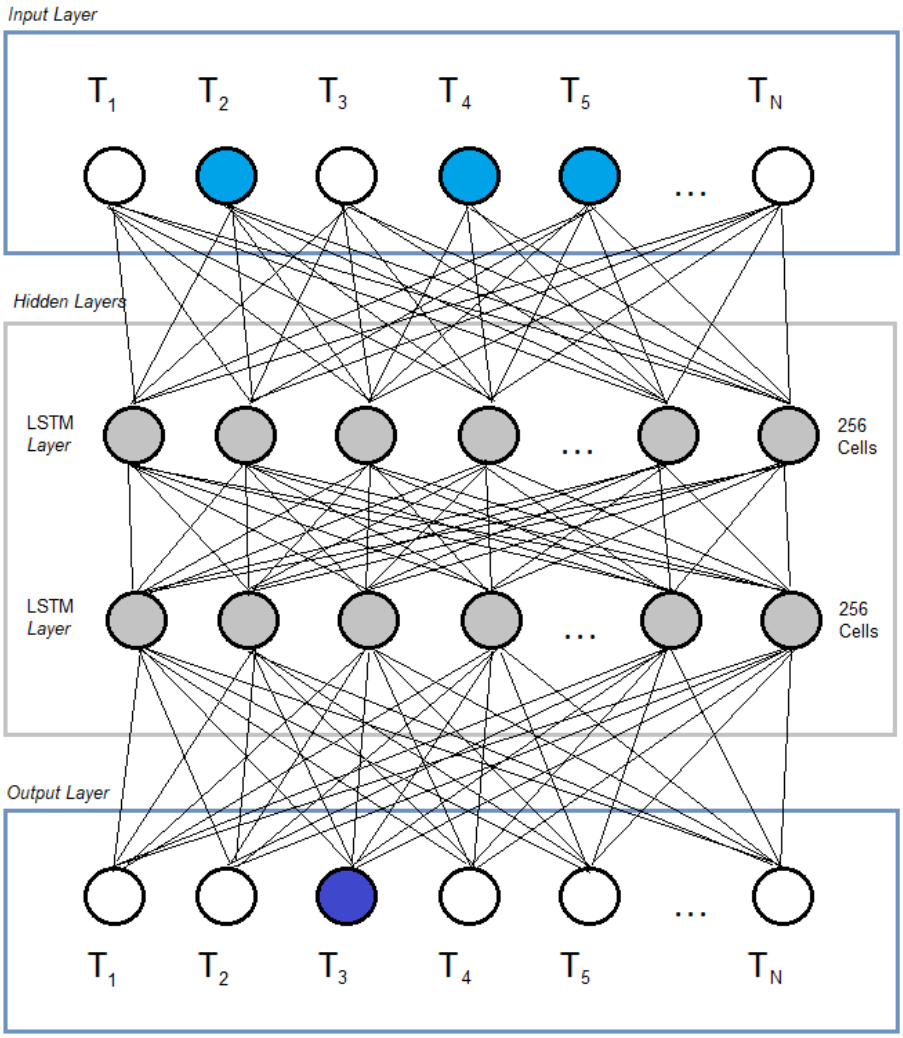

Figura 2 - Diagrama da arquitetura da Rede Neuronal.

3 Avaliação

3.1 Métricas e Hipóteses

Uma das principais questões que deve ser analisada aquando da avaliação de um qualquer sistema é de que forma o mesmo poderá ser avaliado. Para tal, foram identificados os indicadores que se consideram interessantes e relevantes para a avaliação de sistemas do ramo de LM:

- Tempo de Previsão (TP), medido em milissegundos;
- *Keystroke Savings* (KS), medido em percentagem.

Considerando que o sistema de previsão de texto desenvolvido visa ser utilizado durante o processo de redação de um relatório clínico, é importante que as sugestões das próximas palavras sejam apresentadas rapidamente, de forma a não quebrar o raciocínio do utilizador. Assim, uma das métricas a utilizar para a avaliação do sistema será o Tempo de Previsão, medido em milissegundos, desde o momento em que o último carácter é introduzido até ao momento da apresentação das sugestões.

Outro fator que se considera relevante para a avaliação da solução implementada é a eficácia das previsões realizadas pelo sistema. Para determinar esta eficácia, foi utilizado o conceito de *Keystroke Savings*, que indica a quantidade de *keystrokes* evitados devido à previsão de texto (Trnka & McCoy, 2008). Este valor pode ser calculado através da razão da diferença entre o número de *keystrokes* sem previsão com o número de *keystrokes* com previsão, sobre o número de *keystrokes* sem previsão:

$$KS(\%) = \frac{Nº\ de\ keystrokes\ sem\ previsão - Nº\ de\ keystrokes\ com\ previsão}{Nº\ keystrokes\ sem\ previsão} \times 100$$

Após a identificação das métricas a utilizar, é importante definir que valores devem ser considerados bons resultados. Para tal, foram definidas as seguintes hipóteses:

- O sistema deve apresentar um TP inferior a 500 milissegundos;
- O sistema deve apresentar uma taxa de KS superior a 75%.

3.2 Metodologia de Avaliação

Para a avaliação da solução implementada, procedeu-se numa primeira instância à criação do modelo a utilizar posteriormente pela RN. Como é normal na avaliação de sistemas de ML, os dados utilizados para a geração do modelo devem ser diferentes dos que são utilizados para a sua avaliação. Desta forma, e considerando a quantidade elevada de relatórios disponíveis, optou-se pela utilização de uma técnica de validação cruzada, *holdout*, que divide os dados em dois grupos distintos, um para o processo de treino, e outro para o processo de avaliação (Allibhai, 2018).

Assim, utilizou-se a amostra aleatória de 1000 documentos do mesmo autor, tendo-se procedido à divisão destes numa proporção de 75% (750 relatórios) para o processo de treino e 25% (250 relatórios) para a avaliação.

Foi desenvolvido um sistema que vai simulando a redação de um relatório e que, utilizando a solução de previsão de texto implementada, vai construindo o relatório tendo em conta as previsões realizadas. O sistema desenvolvido para a avaliação vai registando o número total de *keystrokes* sem previsão, o número total de *keystrokes* com previsão, e ainda o tempo de cada previsão, sendo no final possível calcular a percentagem de *keystroke savings* e o tempo médio das previsões.

3.3 Resultados

Os resultados obtidos foram comparados com outro trabalho similar (Santos, 2017), mas que utiliza uma abordagem diferente, onde a previsão da próxima palavra é realizada com base em informação estatística, em particular a frequência. Seguidamente são apresentados os resultados dessa e da presente abordagem.

Tabela 4 - Resultados obtidos com as abordagens baseadas em frequência e Redes Neuronais.

Abordagem	Nº *keystrokes* sem previsão	Nº *keystrokes* com previsão	KS (%)	Tempo de previsão (ms)
Frequência	731 476	77 656	~89	983
Redes Neuronais	345 725	101 112	~71	87

O trabalho baseado na frequência dos *tokens* utilizou um *corpus* com um total de 731 475 caracteres, que equivale ao número de *keystrokes* sem previsão; o número de *keystrokes* com previsão foi de 77 656, o que perfaz uma taxa de *keystroke savings* de aproximadamente 89%; quanto ao tempo médio de previsão foi obtido um valor de 983 milissegundos. Relativamente ao presente trabalho, que utiliza uma abordagem baseada em RN, o *corpus* tem um total de 345 725 caracteres; com previsão foram necessários 101 112 *keystrokes*, o que se traduz numa taxa de KS de aproximadamente 71%; o tempo médio por previsão é de 87 milissegundos.

3.4 Discussão dos Resultados

Um dos primeiros pontos que se verifica na análise dos resultados é o tamanho dos *corpora*, que corresponde ao número de *keystrokes* sem previsão, utilizado em ambos os trabalhos. Na abordagem baseada em frequência foi utilizado uma base de relatórios com mais do dobro dos documentos que na abordagem baseada em RN.

Relativamente aos resultados em si, verificou-se que a taxa de KS na abordagem baseada em RN foi aproximadamente 18% inferir à abordagem baseada na frequência (redução de ~89% para ~71%). Acredita-se que esta redução se deva principalmente a duas razões: a utilização de um modelo medíocre (com uma *accuracy* de 76%); e à utilização de uma menor quantidade de dados para treino. Quanto aos tempos médios de previsão, verificou-se que a abordagem baseada em RN apresenta uma melhoria significativa quando comparada à abordagem baseada em frequência, aproximadamente 91% (redução de 983 ms para 87 ms).

Resumidamente, a abordagem implementada no presente trabalho, baseada em RN, apresenta uma taxa de KS inferior em cerca de 18% (~71%) quando comparada com a abordagem baseada na frequência. Por outro lado, ao nível do tempo médio de previsão, apresenta uma melhoria considerável de 91% (87 ms).

4 Conclusão

4.1 Limitações e Trabalho Futuro

Durante o desenvolvimento da solução foram identificadas algumas limitações que se consideram importantes de referir. Uma dessas limitações é a incapacidade de o sistema conseguir lidar com erros lexicais, sintáticos e semânticos. Este ponto é particularmente critico dado que a RN vai aprender com os textos disponibilizados e, na eventualidade destes possuírem erros, o modelo criado terá em consideração esses erros. Uma das formas de tentar resolver este problema poderia ser, por exemplo, através da integração de um corretor automático no editor de texto. Outra limitação identificada, foi a duração do processo de treino, que se verificou muito demorada, especialmente com o aumento do número de *epochs* e diminuição do *learning rate*. Esta limitação foi particularmente critica porque limitou a quantidade de arquiteturas testadas. Uma das formas de colmatar este problema passaria pela utilização de *hardware* com melhores especificações, especialmente no que toca à capacidade de processamento. De seguida é apresentado uma lista com as limitações identificadas.

- Incapacidade de lidar com erros lexicais, sintáticos e semânticos;
- Processo de treino muito demorado.

Relativamente ao trabalho futuro, para além das melhorias que visam a resolução das limitações identificadas, identificaram-se outras tarefas que poderiam ser uma mais-valia para a solução desenvolvida. Uma dessas tarefas passa pela realização de mais testes de outros modelos, com mais variações de parâmetros, de forma a encontrar melhores modelos. Infelizmente o número de modelos testados foi limitado devido ao tempo e *hardware* disponíveis. Outro ponto que é certamente uma mais-valia para a obtenção de melhores resultados é a divisão dos relatórios clínicos nos respetivos exames realizados, ou seja, agrupar os relatórios pelo procedimento médico e respetiva zona do corpo como, por exemplo, raio-X à mão, ao tórax; Tomografia Computorizada (CT) ao crânio; ecografia ao abdómen; entre outros. Outro ponto que seria positivo para a solução desenvolvida, seria a atualização regular do modelo utilizado, de forma a melhorar o modelo ao longo do tempo, adicionando algum tipo de aprendizagem na solução implementada. Por último, outro ponto que poderia ser interessante de analisar, seria encontrar uma forma de integrar o sistema desenvolvida com reconhecimento de fala e/ou comandos de voz. De seguida é apresentado uma lista com o trabalho futuro identificado.

- Resolução das limitações identificadas;
- Avaliação de mais modelos;
- Divisão dos relatórios por procedimento médico e zona do corpo;

- Atualização regular do modelo, de forma a simular aprendizagem;
- Integração com reconhecimento de fala e/ou comandos de voz.

4.2 Apreciação Final

No geral, o desenvolvimento deste projeto considera-se positivo, apresentando resultados razoáveis e promissores. Ainda que a taxa de KS obtida (71%) tenha sido inferior ao mínimo estabelecido (75%), acredita-se que seria possível ultrapassar este valor com a utilização de um melhor modelo e ainda com o agrupamento dos relatórios nos respetivos procedimentos médicos. A nível de *performance*, a presente abordagem, baseada em RN, apresentou melhorias consideráveis (91%) no tempo médio de previsão, com cada previsão a ser feita em 87 ms.

Referências

Allibhai, E. (3 de Outubro de 2018). *Hold-out vs. Cross-validation in Machine Learning.* Obtido de https://medium.com/@eijaz/holdout-vs-cross-validation-in-machine-learning-7637112d3f8f

Brownlee, J. (25 de Janeiro de 2019). *Understand the Impact of Learning Rate on Neural Network Performance.* Obtido de Machine Learning Mastery: https://machinelearningmastery.com/understand-the-dynamics-of-learning-rate-on-deep-learning-neural-networks/ tendo

Copestake, A. (2004). Natural Language Processing. p. 19. Obtido de https://www.cl.cam.ac.uk/teaching/2002/NatLangProc/revised.pdf

Gupta, S. (27 de Outubro de 2021). *RNN vs. CNN: Which Neural Network Is Right for Your Project?* Obtido de https://www.springboard.com/blog/ai-machine-learning/rnn-vs-cnn/

Kumar, H. (7 de Dezembro de 2018). *Loss vs Accuracy.* Obtido de https://kharshit.github.io/about/

Manning, C., Raghavan , P., & Schütze, H. (7 de Abril de 2009). *Tokenization.* Cambridge University Press. Obtido de Determining the vocabulary of terms: https://nlp.stanford.edu/IR-book/html/htmledition/tokenization-1.html

Mittal, A. (12 de Outubro de 2019). *Understanding RNN and LSTM.* Obtido de https://aditi-mittal.medium.com/understanding-rnn-and-lstm-f7cdf6dfc14e

Nadkarni, P., Machado, L., & Chapman, W. (15 de Setembro de 2011). Natural language processing: an introduction. *Journal of the American Medical Informatics Association*, pp. 544-551.

Santos, A. (Outubro de 2017). Elaboração Automática de Relatórios Médicos. Obtido https://recipp.ipp.pt/bitstream/10400.22/11318/1/DM_AlessandroSantos_2017_MEI.pdf

Trnka, K., & McCoy, K. (Junho de 2008). Evaluating Word Prediction: Framing Keystroke Savings. *Proceedings of ACL-08: HLT* (pp. 261-264). Ohio: Association for Computational Linguistics. Obtido de https://aclanthology.org/P08-2066.pdf