**INSTITUTO DE PESQUISA ECONÔMICA APLICADA – IPEA**

# CLASSIFICAÇÃO AUTOMATIZADA DE RECLAMAÇÕES DE USUÁRIOS DE SERVIÇOS PÚBLICOS: UM ESTUDO DO CASO AÉREO

DISSERTAÇÃO DE MESTRADO

**ESA PEKKA TAPANI HORTTANAINEN**

**BRASÍLIA-DF**

**2019**

**ESA PEKKA TAPANI HORTTANAINEN**

# CLASSIFICAÇÃO AUTOMATIZADA DE RECLAMAÇÕES DE USUÁRIOS DE SERVIÇOS PÚBLICOS: UM ESTUDO DO CASO AÉREO

Dissertação apresentada ao Instituto de Pesquisa Econômica Aplicada (IPEA), como parte das exigências do Programa de Pós-Graduação em Políticas Públicas e Desenvolvimento, área de concentração em Economia, para a obtenção do título de Mestre.

Orientador: Prof. Dr. Bernardo Alves Furtado

Coorientador: Prof. Dr. Erivelton Pires Guedes

**BRASÍLIA-DF**

**2019**

Horttanainen, Esa Pekka Tapani
H823c Classificação automatizada de reclamações de usuários de serviços públicos: um estudo do caso aéreo / Esa Pekka Tapani Horttanainen – Brasília : IPEA, 2019.
51 f. : il.

Dissertação (mestrado) – Instituto de Pesquisa Econômica Aplicada, Programa de Pós-Graduação em Políticas Públicas e Desenvolvimento, área de concentração em Economia, 2019
Orientação: Bernardo Alves Furtado
Coorientação: Erivelton Pires Guedes
Inclui Bibliografia.

1. Automação. 2. Classificação. 3. Documentos. 4. Administração Pública. 5. Processamento da Informação. 6. Inteligência Artificial. I. Furtado, Bernardo Alves. II. Guedes, Erivelton Pires. III. Instituto de Pesquisa Econômica Aplicada. IV. Título.

CDD 006.35

Ficha catalográfica elaborada por Andréa de Mello Sampaio CRB-1/1650

**ESA PEKKA TAPANI HORTTANAINEN**

# CLASSIFICAÇÃO AUTOMATIZADA DE RECLAMAÇÕES DE USUÁRIOS DE SERVIÇOS PÚBLICOS: UM ESTUDO DO CASO AÉREO

Dissertação apresentada ao Instituto de Pesquisa Econômica Aplicada (IPEA), como parte das exigências do Programa de Pós-Graduação em Políticas Públicas e Desenvolvimento, área de concentração em Economia, para a obtenção do título de Mestre.

Defendida em 27 de setembro de 2019

## COMISSÃO JULGADORA

_____

Prof. Dr. José Eduardo Malta de Sá Brandão – Ipea

_____

Prof. Dr. Fabiano Mezadre Pompermayer – Ipea

_____

Prof. Dr. Bernardo Alves Furtado

_____

Dr. Erivelton Pires Guedes

**BRASÍLIA-DF**

**2019**

**Resumo**

Este trabalho analisou a possibilidade de automatizar a classificação de diversos documentos recebidos constantemente pela Administração Pública brasileira. Normalmente estes documentos são lidos e classificados por pessoas diferentes dos destinatários finais. Este processo gera custos adicionais aumentando a utilização do tempo laboral. O objetivo do trabalho foi sugerir meios de atuar de forma mais eficiente na classificação de documentos. O Processamento de Linguagem Natural (PLN), que possui como uma de suas principais aplicações a classificação de documentos, tem se beneficiado muito com o progresso da Inteligência Artificial (IA). Todavia, o PLN tradicional tem seus limites, e exige um pré-processamento extensivo dos documentos. Além disto, os avanços recentes da IA, especificamente na Aprendizagem Profunda (*Deep Learning*; DL), criaram novas possibilidades que podem ultrapassar os limites do estado atual do PLN tradicional. Um destes avanços é na classificação de documentos. Este trabalho utilizou o estado da arte do PLN em um problema real de classificação de documentos da Agência Nacional de Aviação Civil (Anac). Passageiros aéreos registram anualmente dezenas de milhares de reclamações (30 mil em 2018), e a Anac precisa classificá-las manualmente, o que tem consumido muitos recursos da Agência. Se a classificação automatizada fosse bem-sucedida, isso liberaria recursos da Anac para outras tarefas, beneficiando a sociedade. Ademais, a metodologia aplicada poderia ser utilizada por outros órgãos, e consequentemente, multiplicando a economia de recursos. A meta original deste trabalho era alcançar a acurácia de 95% na classificação automatizada das reclamações em onze temas e 66 subtemas. Esta meta ambiciosa não foi alcançada. Entretanto, os resultados são encorajadores para a continuação da pesquisa. Além disso, mesmo no estado atual do classificador automático, ele pode ser aplicado na auditoria das classificações, podendo gerar economia de recursos. Como o estado da arte do PLN está evoluindo constantemente, novas metodologias surgiram durante a elaboração desta dissertação. Essas metodologias poderão ser aplicadas ao presente caso.

**Palavras-chave**: classificação de documentos, automação, deep learning, processamento de linguagem natural.

**Abstract**

This work analyzed a possibility to automate the classification of different documents that Brazil's Public Administration receives all the time. Usually these documents have to be read and classified by a different person than the final recipient. This process creates additional costs in the form of spent work hours. The aim of this work was to present more efficient means to document classification. Natural Language Processing (NLP), whose one main application is document classification, has benefited greatly from the progress of Artificial Intelligence (AI). Traditional NLP has its limits, besides the need for extensive preprocessing of the documents. On the other hand, recent advances of the AI, especially in *Deep Learning* (DL), have created new opportunities that can go beyond the limits of the current state of the traditional NLP. One of these advances has been in document classification. This work applied a current state of the art to a real document classification problem of Brazil's Civil Aviation Authority (Anac). Air transport passengers submit annually tens of thousands of complaints (30 thousand in 2018), and Anac has had to classify them manually, which has tied up a lot of its resources. If the automatic classification succeeded, it would free Anac's resources to other tasks, which would directly benefit the society. Moreover, the successful method could be copied to other organizations, multiplying the savings in resources. The original target was to reach 95% accuracy in automatic classification of the complaints into eleven topics and 66 subtopics. This ambitious target was not reached, but the results are encouraging to continue the research. Even at the current state of the automatic classifier, it can be used to audit the classifications, creating some savings in resources. Since the current state of the art in NLP is a moving target, new methods were discovered during the writing of this thesis, and a logical continuation to this project is to apply these methods to the current case.

**Keywords**: document classification, automation, deep learning, natural language processing.

## Lista de ilustrações


Figura 1 – Modelo de neurônio artificial. . . . . 7
Figura 2 – Exemplos de funções de ativação com seus derivativos. . . . . 8
Figura 3 – Exemplo de uma rede neural artificial (ANN) com uma camada oculta. . 8
Figura 4 – Exemplo de uma rede neural artificial profunda com três camadas ocultas. 10
Figura 5 – Modelo de rede neural recursiva (RNN). . . . . 11
Figura 6 – O módulo da RNN que contém uma camada única. . . . . 12
Figura 7 – O módulo da rede LSTM contém quatro camadas interagindo entre si. . 12
Figura 8 – O detalhamento do módulo LSTM com os fluxos de informação. . . . . 14
Figura 9 – Os três passos do *ULMFiT*. . . . . 21
Figura 10 – A estrutura da rede neural. . . . . 47
Figura 11 – A rede neural inicializada. . . . . 47
Figura 12 – O primeiro passo da retropropagação. . . . . 48
Figura 13 – 2º passo da retropropagação. . . . . 49

## Lista de tabelas

Tabela 1 – Distribuição das reclamações na base por temas (Anac). . . . . 19
Tabela 2 – Tabulação cruzada entre grupos de problema (Consumidor.gov) e temas (Anac). . . . . 19
Tabela 3 – Tabulação cruzada entre problemas (Consumidor.gov) e temas (Anac). . 20
Tabela 4 – Matriz de confusão das previsões no conjunto de validação (10% da base). 27
Tabela 5 – Distribuição das reclamações na base por temas e subtemas . . . . . 46

**Sumário**


**Lista de ilustrações** . . . . . **vii**
**Lista de tabelas** . . . . . **viii**
**1 Introdução** . . . . . **1**
1.1 Apresentação do problema . . . . . 1
1.2 Justificativa sobre a relevância do problema . . . . . 2
1.3 Apresentação da pergunta de pesquisa . . . . . 3
1.4 Apresentação da estrutura da dissertação . . . . . 3
**2 Referencial teórico** . . . . . **4**
2.1 Processamento de linguagem natural (PLN) . . . . . 4
2.1.1 Processamento estatístico da linguagem natural . . . . . 4
2.1.2 Classificação de textos . . . . . 5
2.2 Inteligência artificial no PLN . . . . . 6
2.2.1 Redes neurais . . . . . 6
2.2.2 Deep Learning . . . . . 9
2.2.3 LSTM - rede de memória longa de curto prazo . . . . . 12
2.3 Apresentação do estado da arte sobre o problema . . . . . 14
**3 Abordagem metodológica** . . . . . **16**
3.1 A base de dados utilizada . . . . . 16
3.1.1 Consumidor.gov.br . . . . . 16
3.1.2 Construção da base . . . . . 16
3.1.3 Descrição da base . . . . . 18
3.2 Apresentação da metodologia utilizada (ULMFiT) . . . . . 19
3.3 Aplicação da metodologia no caso da Anac . . . . . 22
**4 Resultados** . . . . . **23**
4.1 Descrição do processo . . . . . 23
4.1.1 Modelo de linguagem . . . . . 23
4.1.2 Ajuste fino do modelo de linguagem . . . . . 24
4.1.3 Ajuste fino do classificador . . . . . 25
4.1.4 Validação dos resultados . . . . . 26
4.2 Análise dos resultados obtidos . . . . . 26
4.2.1 Acurácia da previsão dos temas e subtemas . . . . . 26
**5 Conclusões** . . . . . **29**
5.1 Apresentação da resposta à pergunta de pesquisa . . . . . 29

5.2 Enquadramento da resposta obtida à pergunta de pesquisa no estado da arte 30
5.3 Síntese dos avanços metodológicos e empíricos . . . . . 30
5.4 Apresentação dos limites da pesquisa realizada . . . . . 30
5.5 Trabalhos futuros . . . . . 31

**Referências** . . . . . **32**

**Anexos** **36**
**ANEXO A Lista de variáveis da base do Consumidor.gov.br** . . . . . **37**
**ANEXO B Classificação do Consumidor.gov.br - Grupos de problema e Problemas** . . . . . **39**
**ANEXO C Classificação da Anac - Temas e subtemas** . . . . . **42**
**ANEXO D Distribuição das reclamações por temas e subtemas** . . . . . **45**
**ANEXO E Exemplo numérico de rede com retropropagação** . . . . . **47**
**ANEXO F Python Jupyter Notebook do processo de modelagem** . . . . . **51**

# Capítulo 1 – Introdução

A Administração Pública brasileira recebe constantemente demandas dos cidadãos por meio eletrônico. Estas demandas são encaminhadas para equipes especializadas de acordo com a sua natureza. Normalmente não é a equipe competente que lê a demanda pela primeira vez, de maneira que ocorre um trabalho prévio de classificação dessas, que em seguida são direcionadas às equipes que irão lidar com as demandas.

A classificação automática de textos é um novo avanço da inteligência artificial (IA) que pode reduzir bastante o trabalho humano manual, e, portanto, gerar economia e liberar recursos para outras tarefas mais produtivas. Nesta dissertação procuramos realizar uma classificação automática das reclamações de passageiros aéreos registradas na plataforma Consumidor.gov.br.

Com este estudo, espera-se que os resultados positivos possam ser replicados em outros setores da Administração Pública, consequentemente, criando um efeito cascata de economia de recursos.

## 1.1 Apresentação do problema

O trabalho de classificação de textos gera desperdício de recursos para a Administração Pública brasileira. A leitura de demandas por um servidor que não irá analisá-las é uma etapa que consome recursos de outros serviços que poderiam ser mais produtivos para a sociedade.

Na Agência Nacional de Aviação Civil (Anac), a Gerência de Regulação das Relações de Consumo (GCON) da Superintendência de Acompanhamento de Serviços Aéreos (SAS) atualmente classifica as reclamações que os usuários do serviço de transporte aéreo registram na plataforma Consumidor.gov.br. Este trabalho é feito por meio da leitura de reclamações individuais e a sua classificação dentre onze temas, subdivididos em um universo de 66 subtemas (veja Anexo C). A classificação elaborada pela Anac tem por lastro a Resolução ANAC nº 400/2016 (ANAC, 2016), a qual dispõe sobre as condições gerais de transporte aéreo, e visa analisar em quais fases da comercialização e execução do serviço aéreo o passageiro enfrenta mais problemas e contratempos.

Em 2018, a quantidade das reclamações alcançou a marca de 30 mil. Isto significa um trabalho estimado de aproximadamente 1.500 homens-horas distribuído entre os servidores da GCON, o que equivale aproximadamente à atividade laboral contínua de um servidor em um ano, ou seja, um custo alto.

Neste sentido, a GCON buscou uma solução para reduzir esta carga de trabalho dos seus servidores. Dentre as alternativas existentes, poder-se-ia cogitar a utilização da leitura amostral. Este método, porém, apresenta dois problemas. O primeiro é que a divisão atual em 66 subtemas implica que o tamanho de alguns subtemas fique abaixo da margem de erro da amostra, ou seja, esses subtemas podem não aparecer na amostra. O segundo consiste no fato de que mesmo que se fosse resolvida a questão do tamanho de subtemas (por exemplo, por meio da diminuição de sua quantidade), numa eventual análise subsequente das reclamações poderia não haver representatividade das reclamações dentro de algum subtema.

Atualmente, a classificação dentro da plataforma Consumidor.gov.br segue o Código de Defesa do Consumidor (CDC) (BRASIL, 1990). Uma outra solução, que seria a modificação da plataforma para se adequar à arvore de classificação da Anac enfrenta dificuldades mais políticas do que técnicas. Além disso, o consumidor nem sempre classifica corretamente a sua reclamação. Esse fato foi confirmado pela análise do conteúdo das reclamações e a comparação com a classificação feita pelo consumidor (baseada no CDC). Vale ressaltar, porém, que nem a equipe da GCON é totalmente homogênea quanto às suas classificações (baseada na Resolução nº 400/2016) e, portanto, esses eventuais erros na autoclassificação pelos consumidores não poderiam ser usados como contra-argumento.

Obviamente, a implementação da solução a ser proposta tem custos que têm de ser considerados. Esses custos não podem exceder o custo atual, principalmente se a efetividade da classificação automática não alcançar o nível de classificação manual. Por outro lado, a realização desta pesquisa já permitiu a criação da maior parte das ferramentas para a implementação da solução final e, assim, evitou custos elevados.

Por fim, cabe registrar que os recentes avanços no campo de processamento de linguagem natural – PLN ou NLP, sigla das palavras inglesas *Natural Language Processing* como de Howard e Ruder (2018) e de Devlin et al. (2018) – incentivaram a realização deste estudo. Além do interesse acadêmico, um resultado bem sucedido traria economia imediata, bem como a possibilidade de uma análise mais profunda e focada das reclamações por temas ou subtemas, o que atualmente é inviável devido aos elevados custos e trabalho demandado.

## 1.2 Justificativa sobre a relevância do problema

As agências reguladoras, entre outras competências, são responsáveis pela implementação de políticas públicas (MARTINS, 2011; BRANDÃO JÚNIOR, 2017). A responsabilidade da Anac é “garantir segurança no transporte aéreo, a qualidade dos serviços e respeito aos direitos do consumidor” (ANAC, 2015).

Um dos vinte objetivos estratégicos do Plano Estratégico da Anac vigente para os

anos 2015-2019 é: "Criar mecanismos efetivos de correção da conduta dos entes que colocarem em risco a qualidade e a segurança da aviação civil e a qualidade do transporte aéreo." (ANAC, 2015).

Nesse Plano Estratégico, também temos a Estratégia 2.11.2: "Otimizar os processos de fiscalização." Ao todo, cada uma das 49 estratégias deste plano se subdivide em iniciativas estratégicas, as quais totalizam 102 iniciativas. No problema apresentado, pode-se enquadrar duas iniciativas estratégicas:

2.11.2.2 Estruturar e padronizar o processo de ação fiscal

2.11.2.3 Estruturar e padronizar os processos de vigilância continuada.

Como uma das competências da SAS é "fiscalizar a prestação dos serviços aéreos públicos" e, por outro lado, a competência específica da GCON é "promover a proteção e defesa coletiva dos direitos dos usuários dos serviços de transporte aéreo público", uma análise mais profunda das reclamações contribui com ambas competências. Portanto, é interessante ter mais recursos para essa análise, o que seria obtido com a classificação automática.

Além do possível benefício imediato à Anac, a automatização da classificação de textos poderá ser útil para outros setores da Administração Pública, liberando recursos para um melhor atendimento ao cidadão.

## 1.3 Apresentação da pergunta de pesquisa

A pergunta de pesquisa é: "Até qual nível de sucesso é possível chegar na classificação automática das reclamações dos usuários dos serviços de transporte aéreo?" A resposta define o sucesso da pesquisa, sendo a meta pretendida de 95% de acertos para os 66 subtemas atuais, o que permitiria automatizar o processo inteiro de classificação. De toda forma, a resposta será uma contribuição valiosa para ajudar a definir o estado atual da classificação automática para a língua portuguesa.

## 1.4 Apresentação da estrutura da dissertação

Nos próximos capítulos é desenvolvido o estudo cujo resultado é a classificação automática das reclamações. Primeiramente, é apresentado o referencial teórico a partir do problema da classificação de textos, o qual é um subtema dentro do processamento da linguagem natural (PLN). Depois, é apresentada a base de dados, com a descrição da coleta e a formação dela. A metodologia escolhida, que é o estado da arte atual nesse ramo de pesquisa que evolui em alta velocidade, é apresentada na sequência. No final, são apresentados os resultados, as análises e as conclusões extraídas deles.

# Capítulo 2 – Referencial teórico

Neste capítulo são apresentados os conceitos de Processamento de Linguagem Natural (PLN), Inteligência Artificial (IA) e sua convergência em *Deep Learning* (DL) no âmbito da classificação de textos.

## 2.1 Processamento de linguagem natural (PLN)

O Processamento de Linguagem natural (PLN) é um campo de ciência que desenha métodos e algoritmos para processar ou produzir linguagem natural (GOLDBERG, 2017). Ele nasceu nos anos 1950 como interseção da Inteligência Artificial e da Linguística (NADKARNI; OHNO-MACHADO; CHAPMAN, 2011).

As principais aplicações do PLN são a categorização ou *clusterização* de documentos, tradução automática, recuperação de documentos, resposta automática a perguntas, sumarização, extração de informação, resolução de entidades, reconhecimento de fala e síntese de voz (JACKSON; SCHILDER, 2006). Classificação de textos encaixa-se na primeira dessas aplicações.

Jones (2001) fez uma revisão do PLN desde o fim da década de 1940 até a data do seu artigo (2001). Neste artigo o autor identifica quatro fases principais na história do PLN, caracterizadas, respectivamente, pela ênfase em tradução automática, pela influência da inteligência artificial, pela adoção do estilo lógico-gramatical e pelo uso de dados linguísticos em massa.

Cambria e White (2014) mencionam três paradigmas da evolução do PLN: *bag-of-words* (saco de palavras), *bag-of-concepts* (saco de conceitos) e *bag-of-narratives* (saco de narrativas) que estão relacionados a curvas de evolução de sintática, semântica e pragmática, cada curva se sobrepondo e dando um salto de desempenho. Atualmente, estaríamos na transição da curva sintática para a curva semântica, a curva pragmática prevista ainda para o futuro.

### 2.1.1 Processamento estatístico da linguagem natural

Com a quarta fase histórica do PLN e a disponibilização de dados linguísticos em massa, técnicas baseadas em conceitos estatísticos estão sendo difundidas para análises desses dados (JONES, 2001). Portanto, o PLN se baseou nas teorias da probabilidade e da informação (VARASCHIN; LIMA, 2001).

Uma grande base de textos é denominada corpus (plural: corpora) (VARASCHIN; LIMA, 2001). O treinamento de modelos estatísticos requer que o corpus seja dividido em partes de treinamento e de teste, para um melhor resultado. A conclusão principal do trabalho de Varaschin e Lima é que até a data de sua publicação (2001), não foram encontrados trabalhos que utilizassem o processamento estatístico da linguagem portuguesa. Posteriormente já foram publicados alguns trabalhos em língua portuguesa. Orair (2009) pesquisa a classificação multi-rótulo hierárquica no contexto de métodos do PLN tradicionais. Peclat (2015) faz uma avaliação semântica dos editais de licitação de fábricas de software na Administração Pública Federal. Marzagão (2015) avalia a democracia mundial utilizando métodos tradicionais do PLN e, em outra pesquisa (MARZAGÃO, 2016), o mesmo autor classifica compras governamentais, também com métodos tradicionais. Monteiro (2018) está experimentando a utilização da mesma metodologia deste trabalho (*Universal Language Model Fine-Tuning, ULMFiT*) com documentos públicos de controle externo do Tribunal de Contas da União (TCU). Valle (2019) faz análise de sentimentos de avaliações de passageiros aéreos com métodos tradicionais e Fernandes et al. (2019) fazem uso das novas metodologias de *Deep Learning* para avaliarem modificações de sentenças da justiça brasileira.

### 2.1.2 Classificação de textos

Varaschin e Lima (2001, p. 52) definem classificação como "associar objetos de um universo a duas ou mais classes." Os problemas de classificação da linguagem natural podem ser categorizados de acordo com o item a ser classificado, como palavra isolada, texto, texto emparelhado (um par de palavras ou textos que têm alguma relação entre si como similaridade, tradução, autoria, inferência etc.), palavra em contexto ou relação entre duas palavras no mesmo documento (GOLDBERG, 2017). No caso de textos, há um conjunto de documentos para classificar, tais como avaliações de filmes. Uma tarefa de classificação poderia ser dividir as avaliações em positivas e negativas. Nesse caso, a classificação é binária, ou seja, em duas classes. No caso da Anac, as reclamações têm que ser classificadas em 11 classes (temas) e, ainda mais precisamente, em 66 classes (subtemas) (Veja Anexo C).

Sachan, Zaheer e Salakhutdinov (2018) conduzem uma análise comparativa de várias metodologias de classificação de texto, incluindo modelos tradicionais do PLN, como também modelos mais recentes (Deep Learning). A conclusão deles foi que os modelos mais recentes têm quase sempre desempenho melhor, mas que é importante se atentar para a separação do corpus nas partes de treinamento e teste.

## 2.2 Inteligência artificial no PLN

Nos últimos anos, a classificação de textos tem começado a usar predominantemente métodos não lineares. Isto foi possível graças a avanços rápidos na Inteligência Artificial (IA), mais especificamente na Aprendizagem Profunda (em inglês *Deep Learning* ou DL), a qual é um subconjunto do Aprendizado de Máquina (*Machine Learning* ou ML), um ramo da IA. O DL está baseado em redes neurais. Outro avanço tecnológico que possibilitou a utilização de métodos não lineares foi o avanço do *hardware* específico, GPU (*Graphics Processing Unit* ou unidade de processamento gráfico). O desempenho de GPU para calcular os modelos de DL é muito maior do que o de CPU (*Central Processing Unit*) e o preço deles tem baixado concomitantemente aos seus avanços de velocidade. (EISENSTEIN, 2018).

### 2.2.1 Redes neurais

O uso de redes neurais no ML se inspira na estrutura biológica do cérebro humano. O objetivo é simular a capacidade de aprendizagem do cérebro humano, desenvolvendo o algoritmo levando em consideração a estrutura e funcionamento do sistema nervoso. O inventor de um dos primeiros neurocomputadores, Dr. Robert Hecht-Nielsen, define rede neural como "...sistema de computação feito de elementos de processamento simples e altamente interconectados que processam informação pela sua resposta dinâmica de estado às entradas externas." (SHETTY et al., 2015, p. 443). A utilidade principal das redes neurais é a previsão de relações lineares e não lineares entre estruturas complexas de dados.

Uma rede neural é composta por uma ou várias unidades de processamento simples interconectadas. Estas unidades chamam-se neurônios artificiais. Eles estão dispostos por uma ou mais camadas e interligados por um alto número de conexões.

A rede neural artificial mais simples é chamada perceptron (NIELSEN, 2015). Ela é uma função linear de suas entradas (soma ponderada por pesos diferentes associados para cada entrada, sendo que estes pesos refletem a importância da entrada associada) e tem somente a camada de entrada e a de saída, conforme a Figura 1. Perceptron é normalmente usado para classificar dados em duas partes. Portanto, ele também é chamado de classificador linear binário. A forma equacional do perceptron é apresentada pela seguinte equação:

$$ANN_{Perceptron}(\mathbf{x}) = \mathbf{W}\mathbf{x} + \mathbf{b}, \tag{2.1}$$

onde $\mathbf{x}$ é o vetor de entradas, $\mathbf{W}$ é a matriz de pesos com suas dimensões dadas pelo comprimento do $\mathbf{x}$ e o número de saídas (no caso de uma saída, a $\mathbf{W}$ é também um vetor) e o $\mathbf{b}$ vetor de *bias* ou tendência, com número de elementos igual ao de saídas. Para uma revisão de

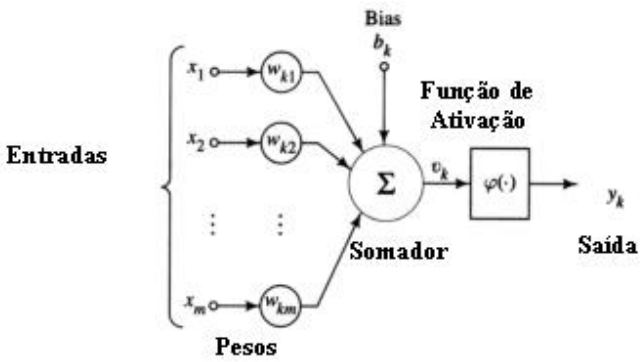


Figura 1: Modelo de neurônio artificial.
Fonte: https://www.gsigma.ufsc.br/~popov/aulas/rna/neuronio_artificial/index.html

álgebra linear para fins de IA e especificamente para *Deep Learning*, veja Goodfellow, Bengio e Courville (2016).

Segundo a definição original do perceptron, a saída binária seria 1, quando $\mathbf{Wx} + \mathbf{b} \geq 0$ e 0 quando $\mathbf{Wx} + \mathbf{b} < 0$. Isso se reflete à função de ativação (uma função que muda o valor da saída) *Heaviside*, que faz justamente isso, muda o valor de 0 a 1 instantaneamente (TONIDANDEL; ARAUJO, 2015). Aqui o significado da tendência (*bias*) é evidente, ela define o valor necessário da soma ponderada para iniciar a ativação.

Atualmente, para facilitar as aplicações modernas de perceptron, outras funções de ativação são aplicadas. Essas funções têm tido normalmente a forma de uma sigmoide, como a própria sigmoide $\sigma(x) = 1/(1 + e^{-x})$ ou tanh $= \frac{e^x - e^{-x}}{e^x + e^{-x}}$, mas, ultimamente, nas saídas das camadas ocultas, uma função *retificadora* ou $ReLU = \max(0, x)$ (do inglês, Rectifier Linear Unit) tem ocupado espaço nas aplicações, graças a sua facilidade de evitar problemas numéricos relacionados a sigmoides. Na Figura 2 há quatro exemplos de funções de ativação, junto com seus derivativos. O derivativo da $ReLU$ é justamente a função *Heaviside*.

Para estender a aplicabilidade de rede neural além da linearidade, é introduzida uma camada oculta, conforme a Figura 3. Esta rede é chamada perceptron de camadas múltiplas (MLP de inglês, *Multi-Layer Perceptron*). No caso de uma camada oculta, a sigla é MLP1 e a sua equação no caso de rede *feed-forward* (onde as saídas de uma camada são usadas como entradas para a camada seguinte e não para camadas anteriores) é

$$ANN_{MLP1}(\mathbf{x}) = \mathbf{W_2} g(\mathbf{W_1 x} + \mathbf{b_1}) + \mathbf{b_2}, \tag{2.2}$$

onde $\mathbf{W_1}$ é a matriz de pesos e $\mathbf{b_1}$ o vetor de bias para a primeira transformação linear, $g(\cdot)$ função de ativação não linear e $\mathbf{W_2}$ a matriz de pesos e $\mathbf{b_2}$ o vetor de tendência (*bias*) para a segunda transformação linear. A não linearidade da função $g$ é necessária para que a rede

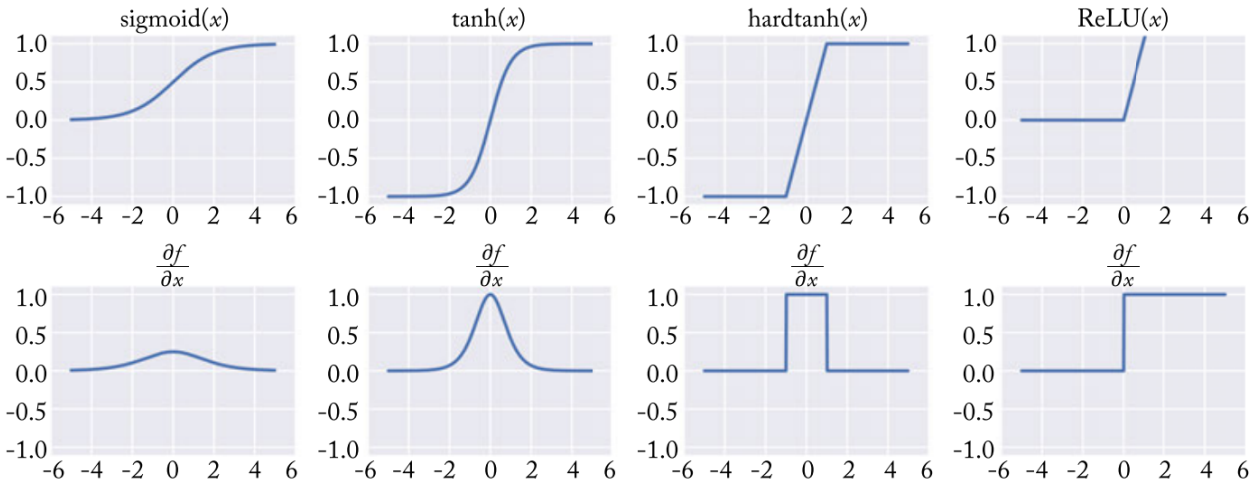


Figura 2: Exemplos de funções de ativação com seus derivativos.
Fonte: Goldberg (2017)

neural possa aproximar funções não lineares.

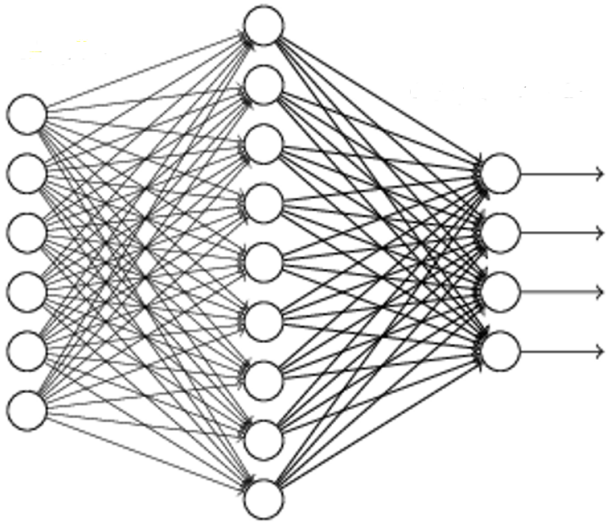

Figura 3: Exemplo de uma rede neural artificial (ANN) com uma camada oculta.
Fonte: Nielsen (2015)

Para calibrar ou ensinar a rede neural, as saídas finais da última camada da rede são comparadas com as saídas desejadas (os resultados reais do fenômeno modelado), tal como a classificação do documento. Por exemplo, se o documento pertence somente a uma classe, o valor da saída correspondente seria um e das demais zero. Porém, é quase impossível que a saída da rede neural seja tão discreta. O mais provável é que mesmo quando a rede neural esteja totalmente calibrada, o neurônio da saída mais provável receba um valor alto e alguns

outros neurônios recebam valores menores. Geralmente na classificação, a soma das saídas é normalizada a 1 e os valores correspondem às probabilidades de a entrada pertencer às classes. Isso é realizado com ajuda da função *softmax*, cujo denomidador é a soma dos exponenciais dos valores das saídas, sendo o numerador o exponencial de cada saída individualmente considerada, realizando, portanto, justamente esta normalização. A forma da *softmax* para um vetor $\mathbf{x} = x_1, \ldots, x_k$ é

$$softmax(x_i) = \frac{e^{x_i}}{\sum_{j=1}^{k} e^{x_j}} \tag{2.3}$$

$$\sum_{i=1}^{k} softmax(x_i) = 1 \tag{2.4}$$

A diferença entre as saídas da rede e as saídas desejadas é o erro da rede. Para quantificar este erro, define-se uma função de custo ou perda, cuja forma pode ser, a título de exemplo, a seguinte:

$$C(\mathbf{W}, \mathbf{b}) = \frac{1}{2n} \sum_{i=1}^{n} (\mathbf{y}(\mathbf{x_i}) - \hat{\mathbf{y}}(\mathbf{x_i}, \mathbf{W}, \mathbf{b}))^2, \tag{2.5}$$

onde $\mathbf{W}$ e $\mathbf{b}$ são todos os pesos e tendências (bias) da rede neural, $n$ é o número de vetores de treinamento, $\mathbf{x_i}$ é um vetor de entrada de treinamento, $\mathbf{y}(\mathbf{x_i})$ é a saída desejada (o fenômeno real), $\hat{\mathbf{y}}(\mathbf{x_i}, \mathbf{W}, \mathbf{b})$ é a saída da rede neural com a entrada $\mathbf{x_i}$ e parâmetros $\mathbf{W}$ e $\mathbf{b}$ e a soma é feita sobre todas as entradas de treinamento.

Uma técnica consolidada para minimizar este tipo de função de várias dimensões é o Método Estocástico do Gradiente (*Stochastic Gradient Descent* ou SGD em inglês). A fórmula básica para atualização por SGD é

$$\boldsymbol{\theta}_t = \boldsymbol{\theta}_{t-1} - \eta \cdot \nabla_{\boldsymbol{\theta}} C(\boldsymbol{\theta}), \tag{2.6}$$

onde $\boldsymbol{\theta}$ é o vetor de todos os parâmetros da rede neural ($\mathbf{W}$ e $\mathbf{b}$), $\eta$ é a taxa de aprendizado (*learning rate* em inglês), $\nabla_{\boldsymbol{\theta}}$ é o operador nabla de diferenciais parciais (gradiente) com respeito a $\boldsymbol{\theta}$ e $C(\cdot)$ é a função de custo da equação 2.5. O SGD é aplicada na aprendizagem da rede neural por um processo de *backpropagation* ou retropropagação. (SANTOS, 2015).

No Anexo E encontra-se exemplo do fluxo de rede neural com retropropagação.

### 2.2.2 Deep Learning

O *Deep Learning* (DL) nada mais é que um novo nome para a utilização de redes neurais de várias camadas ocultas (GOLDBERG, 2017). A rede da Figura 4 é um exemplo de

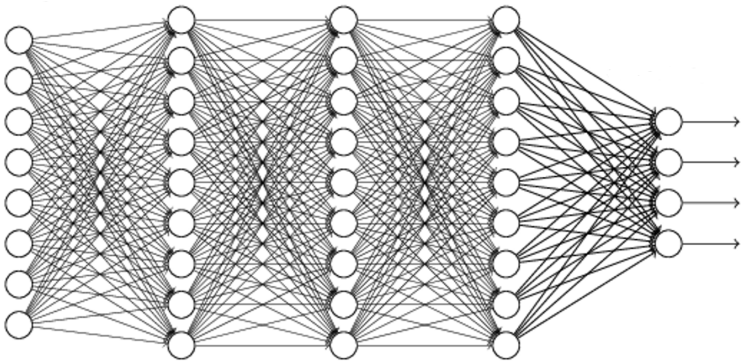

Figura 4: Exemplo de uma rede neural artificial profunda com três camadas ocultas.
Fonte: Nielsen (2015)

rede profunda, tendo três camadas ocultas. Goodfellow, Bengio e Courville (2016) resumem a abordagem do DL como ramo do ML que concentra nosso conhecimento acumulado durante as últimas décadas sobre o cérebro humano, a estatística e a matemática aplicada. O avanço rápido do DL nos últimos anos tem acontecido principalmente nos campos de PLN, visão computacional, reconhecimento automático de fala, aprendizagem reforçada e modelagem estatística (ZHANG et al., 2019). A metodologia já foi usada no Brasil para detectar anomalias e possíveis fraudes entre exportadores brasileiros (PAULA et al., 2016).

Expandindo a equação 2.2, se for adicionada outra camada oculta, já haverá, pela definição, uma rede profunda:

$$ANN_{MLP2}(\mathbf{x}) = \mathbf{W}_3(g_2(\mathbf{W}_2 g_1(\mathbf{W}_1\mathbf{x} + \mathbf{b}_1) + \mathbf{b}_2)), \tag{2.7}$$

onde $\mathbf{W_1}$ é a matriz de pesos e $\mathbf{b_1}$ o vetor de bias para a primeira transformação linear, $\mathbf{W_2}$ a matriz de pesos e $\mathbf{b_2}$ o vetor de bias para a segunda transformação linear, $\mathbf{W_3}$ a matriz de pesos para a terceira transformação linear e $g_1(\cdot)$ e $g_2(\cdot)$ funções de ativação não lineares. Como Goldberg (2017) mostra, fica mais claro apresentar redes profundas usando variáveis intermediárias:

$$ANN_{MLP2}(\mathbf{x}) = \mathbf{y} \tag{2.8}$$

$$\mathbf{h}_1 = g_1(\mathbf{W}_1\mathbf{x} + \mathbf{b}_1) \tag{2.9}$$

$$\mathbf{h}_2 = g_2(\mathbf{W}_2\mathbf{h}_1 + \mathbf{b}_2) \tag{2.10}$$

$$\mathbf{y} = \mathbf{W}_3\mathbf{h}_2, \tag{2.11}$$

onde $\mathbf{h}_1$ e $\mathbf{h}_2$ representam as camadas ocultas ($\mathbf{h}$ de *hidden* em inglês).

Existem dois tipos básicos de arquiteturas de redes neurais apropriadas para o

PLN: redes *feed-forward* e redes recursivas (*Recurrent Neural Networks* ou RNN) (GOLDBERG, 2017). Nas redes *feed-forward* a informação das saídas flui somente em uma direção, não tendo retroalimentação pelas saídas (mas pela retropropagação sim para que a rede possa ser calibrada). As equações até este ponto representam redes *feed-forward*.

A ampliação de retroalimentação para uma rede *feed-forward* produz uma rede recursiva (RNN). A formulação de uma RNN é a seguinte:

$$RNN(\mathbf{x}_t) = \mathbf{y}_t \tag{2.12}$$

$$\mathbf{h}_t = g_h(\mathbf{W}_h\mathbf{x}_t + \mathbf{U}_h\mathbf{h}_{t-1} + \mathbf{b}_h) \tag{2.13}$$

$$\mathbf{y}_t = g_y(\mathbf{W}_y\mathbf{h}_t + \mathbf{b}_y), \tag{2.14}$$

onde $g_h(\cdot)$ e $g_y(\cdot)$ são funções de ativação resultando em estados $\mathbf{h}_t$ (camada oculta) e $\mathbf{y}_t$ (saída), $\mathbf{W}_h$ a matriz de pesos para as entradas atuais $\mathbf{x}_t$, $\mathbf{U}_h$ a matriz de pesos para o estado oculto (camada oculta) anterior, $\mathbf{W}_y$ a matriz de pesos para o estado oculto atual e $\mathbf{b}_h$ e $\mathbf{b}_y$ respectivos vetores de bias.

A Figura 5 mostra um diagrama da RNN. Um problema com a RNN é que na aprendizagem da rede usando a SGD por vários passos de tempo, os gradientes tendem a diminuir ou crescer demasiadamente durante a *backpropagation*. Isso dificulta ou até impossibilita a aprendizagem. Para mitigar esse efeito, Hochreiter e Schmidhuber desenvolveram em 1997 uma solução chamada redes LSTM (*Long Short-Term Memory* ou Memória Longa de Curto Prazo).

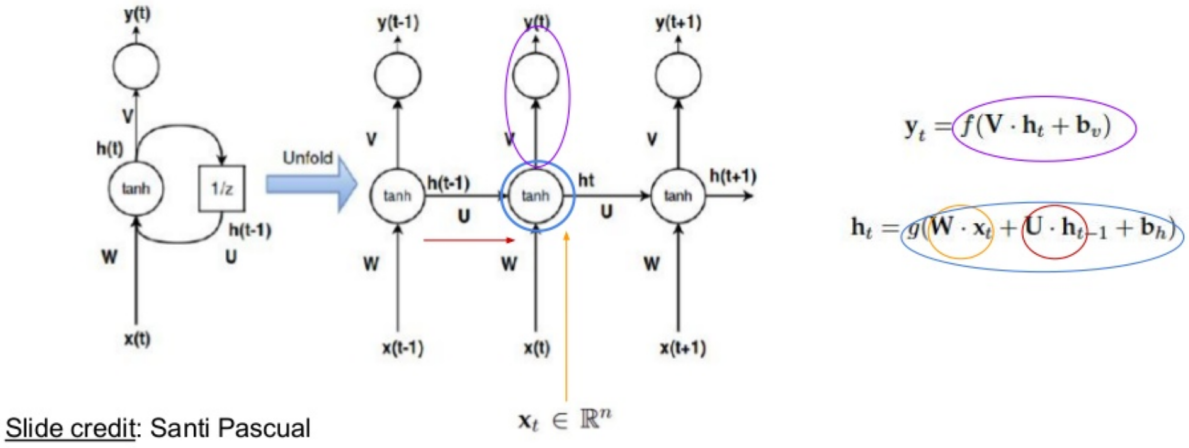


Figura 5: Modelo de rede neural recursiva (RNN).
Fonte: https://www.slideshare.net/xavigiro/video-analysis-with-recurrent-neural-networks-master-computer-vision-barcelona-2017

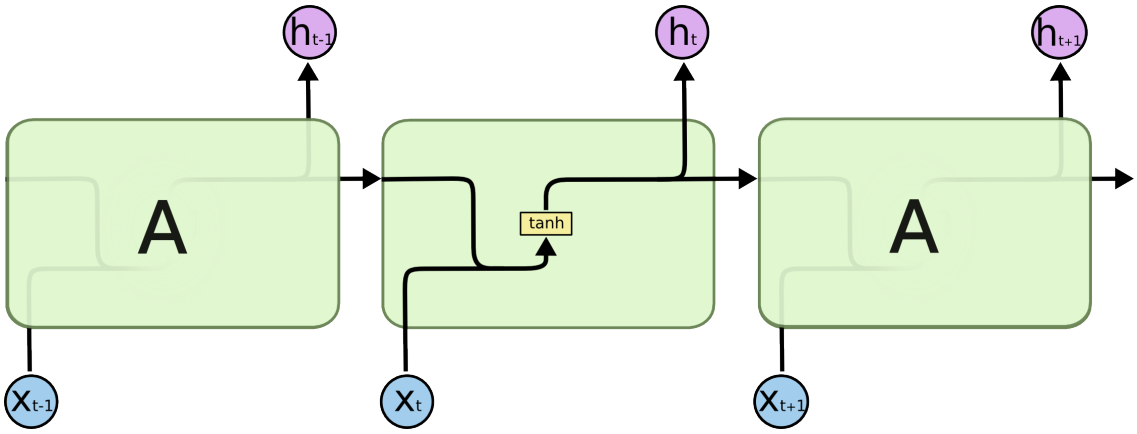


Figura 6: O módulo da RNN que contém uma camada única.
Fonte: Olah (2015)

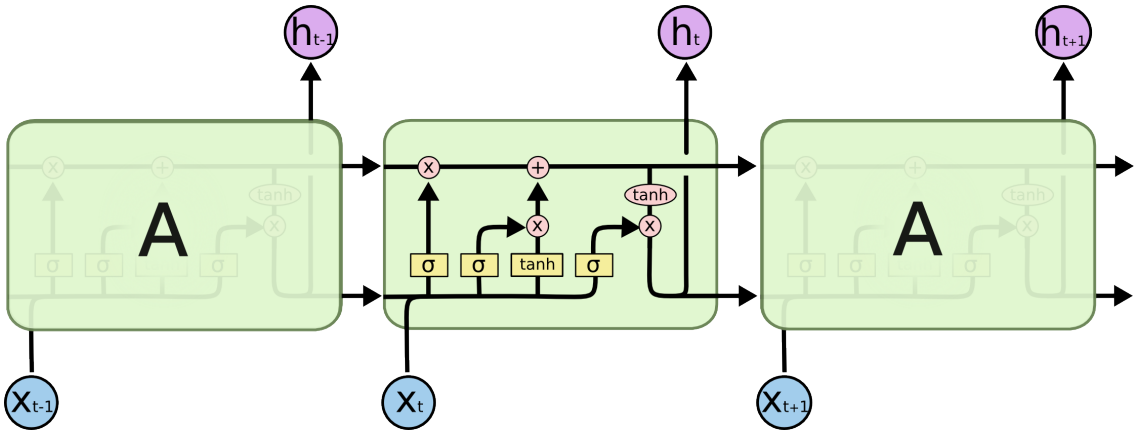


Figura 7: O módulo da rede LSTM contém quatro camadas interagindo entre si.
Fonte: Olah (2015)

### 2.2.3 LSTM - rede de memória longa de curto prazo

A rede LSTM foi desenhada para aprender dependências de longo prazo. Para alcançar tal objetivo, a estrutura de um módulo da rede LSTM é mais complexo do que o da rede RNN (compara as Figuras 6 e 7).

A formulação da rede LSTM é a seguinte:

$$LSTM(\mathbf{s}_{t-1}, \mathbf{x}_t) = \mathbf{s}_t = [\mathbf{C}_t; \mathbf{h}_t] \tag{2.15}$$

$$\mathbf{f}_t = \sigma(\mathbf{W}_f \mathbf{x}_t + \mathbf{U}_f \mathbf{h}_{t-1} + \mathbf{b}_f) \tag{2.16}$$

$$\mathbf{i}_t = \sigma(\mathbf{W}_i \mathbf{x}_t + \mathbf{U}_i \mathbf{h}_{t-1} + \mathbf{b}_i) \tag{2.17}$$

$$\tilde{\mathbf{C}}_t = \tanh(\mathbf{W}_C \mathbf{x}_t + \mathbf{U}_C \mathbf{h}_{t-1} + \mathbf{b}_C) \tag{2.18}$$

$$\mathbf{C}_t = \mathbf{f}_t \odot \mathbf{C}_{t-1} + \mathbf{i}_t \odot \tilde{\mathbf{C}}_t \tag{2.19}$$

$$\mathbf{o}_t = \sigma(\mathbf{W}_o \mathbf{x}_t + \mathbf{U}_o \mathbf{h}_{t-1} + \mathbf{b}_o) \tag{2.20}$$

$$\mathbf{y}_t = \mathbf{h}_t = \mathbf{o}_t \odot \tanh(\mathbf{C}_t), \tag{2.21}$$

onde $\mathbf{s}$ é vetor do estado do módulo; $\mathbf{x}$ é vetor da entrada; $\mathbf{C}$ é vetor do estado da célula ou a "memória" da rede; $\mathbf{h}$ é vetor do estado oculto; $\mathbf{f}$, $\mathbf{i}$ e $\mathbf{o}$ são funções de ativação sigmoide chamadas portões; $\mathbf{W}$'s e $\mathbf{U}$'s são matrizes de pesos das entradas e dos estados ocultos, respectivamente; e o símbolo $\odot$ significa multiplicação por componentes, ao contrário da multiplicação vetorial.

O módulo da rede LSTM recebe informação do módulo anterior (à esquerda, veja Figuras 7 e 8) no formato do estado da célula $\mathbf{C}_{t-1}$ (a "memória" da rede) e do estado oculto $\mathbf{h}_{t-1}$. O módulo LSTM também recebe a entrada $\mathbf{x}_t$. As saídas do módulo são o estado da célula atualizado $\mathbf{C}_t$ e o estado oculto $\mathbf{h}_t$.

Em seguida é apresentado o que acontece em cada camada do módulo LSTM, seguindo um detalhamento elaborado por Olah (2015). Essas camadas são chamadas de portões quando são funções de ativação sigmoide porque seu limite inferior tende a zero, o que significa que elas ou preservam ou descartam a informação que passa por elas. Estes portões são chamados de esquecimento ($\mathbf{f}$ de *forget*), entrada ($\mathbf{i}$ de *input*) e saída ($\mathbf{o}$ de *output*). Há também duas funções de ativação *tanh* que deixam as suas saídas entre os valores de $-1$ e 1.

O primeiro passo é a camada de portão de esquecimento. Seguindo o fluxo da Figura 8 e a equação 2.16, ela recebe as entradas $\mathbf{h}_{t-1}$ e $\mathbf{x}_t$ e emite um número entre 0 e 1 para cada componente do vetor $\mathbf{C}_{t-1}$, o número 1 significando manter o componente totalmente e o 0 significando esquecer o componente totalmente.

O segundo passo é a escolha da informação nova para o estado da célula $\mathbf{C}_t$. Aqui, a camada de portão de entrada decide quais valores do $\mathbf{C}_{t-1}$ são atualizados segundo a equação 2.17. As entradas passam também pela camada *tanh* (equação 2.18) que cria o vetor $\tilde{\mathbf{C}}_t$, o candidato de novos valores para $\mathbf{C}_t$.

O próximo passo é atualizar o estado da célula de $\mathbf{C}_{t-1}$ a $\mathbf{C}_t$, segundo a equa-

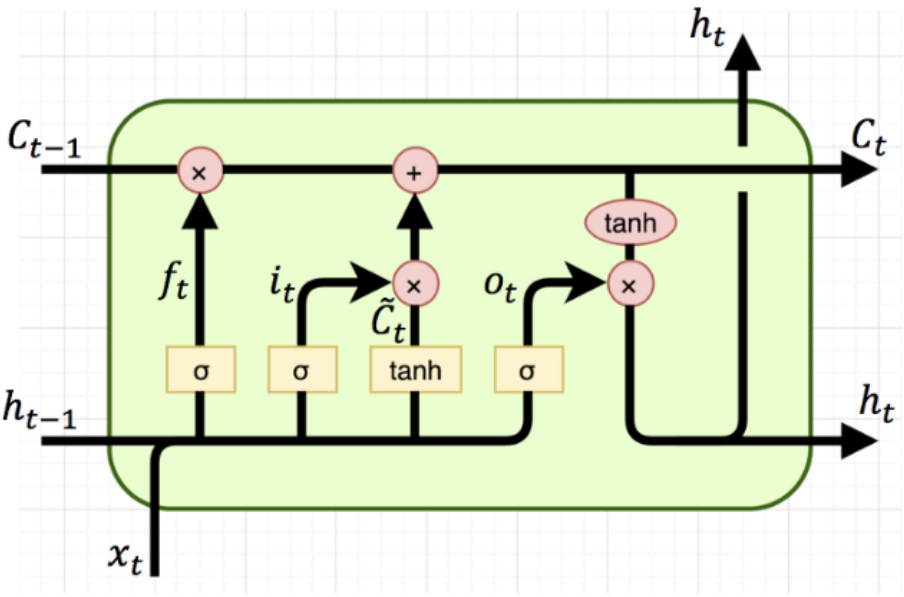


Figura 8: O detalhamento do módulo LSTM com os fluxos de informação.
Fonte: https://towardsdatascience.com/grus-and-lstm-s-741709a9b9b1

ção 2.19. Os componentes do $\mathbf{C}_{t-1}$ são multiplicados pelos componentes do vetor $\mathbf{f}_t$ o que significa esquecer ou manter os componentes como havia sido decidido no primeiro passo. Similarmente, os componentes do $\tilde{\mathbf{C}}_t$ são multiplicados pelos componentes do vetor $\mathbf{i}_t$ o que significa a atualização dos valores do $\mathbf{C}_{t-1}$. A soma destes dois vetores é o novo estado da célula, $\mathbf{C}_t$.

O último passo é a definição da saída (estado oculto $\mathbf{h}_t$). O portão de saída decide quais componentes do estado da célula são considerados (equação 2.20). O vetor do estado da célula passa por função *tanh* que deixa seus valores entre $-1$ e 1 e depois seus componentes são multiplicados pelos componentes do vetor de saída.

## 2.3 Apresentação do estado da arte sobre o problema

Segundo Goldberg (2017), o progresso da pesquisa de redes neurais e *Deep Learning* é muito rápido e o estado da arte deste campo é um alvo em movimento. O modelo seguinte aqui relatado era o estado da arte no início de 2019.

Em 2018, Howard e Ruder publicaram um artigo que apresenta sua metodologia de *Universal Language Model Fine-tuning* (*ULMFiT*) ou Ajuste Fino de Modelo Universal de Linguagem. Esse modelo superou o estado da arte atual em seis tarefas de classificação de texto e está muito bem posicionado nos rankings no site paperswithcode.com, o qual acompa-

nha a pesquisa de ML, especialmente os artigos que publicam seu código para a comunidade poder reproduzir as pesquisas. Em fevereiro de 2019, dos três rankings de classificação de textos no site, o *ULMFiT* ocupava dois primeiros lugares e um segundo lugar, cujo critério era a acurácia dos modelos.

O grande diferencial do *ULMFiT* comparado com os modelos mais tradicionais de PLN é que ele procura "entender" a linguagem e depois usar esse entendimento na aplicação final do modelo. Esse "entendimento" vem do modelo de linguagem que é a base da metodologia. O modelo de linguagem é treinado com um *corpus* grande da língua em questão e ele tenta prever a palavra seguinte na frase, apoiando-se nas palavras anteriores introduzidas ao modelo e depois emitidas pelo próprio modelo. Isto permite que o *corpus* a ser classificado não necessite ser tão grande quanto para os modelos tradicionais. (HOWARD; RUDER, 2018).

O modelo-base para o *ULMFiT* é o modelo de linguagem do estado da arte atual treinado com a rede AWD-LSTM (da sigla *AvSGD Weight-Dropped Long Short Term Memory*, da qual a AvSGD vem de *Averaged Stochastic Gradient Descent*) (MERITY; KESKAR; SOCHER, 2017). AWD-LSTM é uma metodologia aplicada à rede LSTM, conforme foi detalhada na seção anterior. Este modelo de rede profunda tenta captar características básicas de linguagem e ele é treinado para prever a próxima palavra na sequência. O modelo é alimentado por uma ou mais palavras e ele gera a palavra seguinte da sequência ou da frase. Se este processo é repetido, o modelo, dependendo de sua qualidade, gera um texto que pode até parecer natural. Este modelo-base pode ser ajustado para a aplicação de classificação com uma técnica chamada Aprendizagem de Transferência (*Transfer Learning*), que foi responsável pelos grandes avanços na visão computacional antes de ser aplicada na PLN (HOWARD; RUDER, 2018).

A metodologia de *ULMFiT* foi escolhida para este trabalho porque ela se baseia na teoria robusta das redes LSTM, tendo sido aperfeiçoada desde sua concepção em 1997 (HOCHREITER; SCHMIDHUBER, 1997) até hoje, e no estado da arte do modelo de linguagem. Ademais, no trabalho de Howard e Ruder (2018) ela superou o então estado de arte de classificação de texto em seis tarefas e ocupa posições altas no ranking em paperswithcode.com. Além disso, ela não exige altíssima capacidade de computação nem quantidade excessiva de dados, sendo possível treinar o modelo inteiro numa máquina virtual de serviço de computação em nuvem em menos de um dia e em um computador portátil em um período de poucos dias.

# Capítulo 3 – Abordagem metodológica

## 3.1 A base de dados utilizada

### 3.1.1 Consumidor.gov.br

Os passageiros registram as reclamações na plataforma Consumidor.gov.br, que é um "serviço público que permite a interlocução direta entre consumidores e empresas para solução de conflitos de consumo pela internet". A plataforma é monitorada pela Secretaria Nacional do Consumidor – Senacon – do Ministério da Justiça, Procons, Defensorias Públicas, Ministérios Públicos e também por toda a sociedade. Ela "possibilita uma resolução de conflitos de consumo de forma rápida e desburocratizada." Aproximadamente 80% das reclamações registradas no Consumidor.gov.br são solucionadas pelas empresas participantes, com um prazo médio de resposta às demandas de 7 dias. A Anac firmou um termo de cooperação com a Senacon e é gestora da plataforma quanto a reclamações de passageiros aéreos. Até janeiro de 2019, a plataforma tinha, no total, 481 empresas cadastradas; 1,2 milhão de usuários cadastrados e 1,6 milhão reclamações finalizadas. (SENACON, 2019).

### 3.1.2 Construção da base

Para registrar uma reclamação, o usuário deve se cadastrar na plataforma. Assim, alguns de seus dados básicos já são coletados. No momento do registro da reclamação, o usuário também deve classificá-la. A lista das variáveis coletadas e criadas no momento do registro se encontra no Anexo A. Porém, esta classificação (veja Anexo B) se baseia na estrutura do Código de Defesa do Consumidor (CDC) (BRASIL, 1990), a qual não está totalmente compatível com os fins da análise da Anac.

A Anac como entidade gestora do Consumidor.gov.br, tem acesso a todos esses dados, bem como ao conteúdo da reclamação, que está identificada pelo número de protocolo, e à tramitação consecutiva, até a finalização da reclamação. Porém, a interface de programação de aplicação (API de termo inglês *Application Programming Interface*) da plataforma não está ainda disponível, o que dificulta o acesso automático aos conteúdos das reclamações. Para contornar isso, há um robô programado para extrair essas reclamações protocolo por protocolo.

As reclamações são posteriormente lidas e classificadas uma por uma pela equipe da GCON nos termos das definições estabelecidas pela Anac. Esta classificação visa a ajudar na fiscalização de serviços aéreos, conforme a Resolução ANAC nº 400/2016 (ANAC, 2016).

Essa Resolução dispõe sobre as condições gerais de transporte aéreo, que trata, por meio de seus artigos, de todo o processo de oferta do serviço aéreo contemplando, inclusive, as eventualidades de atraso, cancelamento e interrupção do serviço, extravio, avaria e violação da bagagem, e atendimento ao passageiro. O descumprimento de qualquer artigo caracteriza uma infração sujeita a multa. Observa-se, assim, que a fiscalização de cumprimento da Resolução é trabalho essencial da Anac. A divisão em temas e subtemas segue a estrutura dos artigos da Resolução nº 400. A discriminação da classificação se encontra no Anexo C.

Segue um exemplo de reclamação:

> Descrição da Reclamação
>
> Olá em 26/06 comprei o localizador [XXXXXX] para 04 passageiros Sinop x Brasilia, indo 25/02 e retornando em 28/02.
>
> Comprei bem antecipado para poder me programar na viagem (principalmente no retorno); no entanto houve uma alteração no horário o que descontrolou todos meus planos (principalmente no retorno).
>
> Entrei em contato com a central via fone e facebook; só me foi ofertada uma opção de voo, no entanto este não me atende; solicitado a alteração do destino da viagem sem custo.
>
> Pedido à Empresa
>
> Alteração no destino da viagem sem custo.

Essa reclamação foi classificada pelo passageiro no Grupo Problema "Contrato/Oferta" e no Problema "Dificuldade / demora para alterar o contrato de serviço (alteração de voo)". A classificação da Anac foi "Alteração pela empresa aérea" (tema) e "Alternativas (reacomodação, reembolso integral ou outra modalidade de transporte)" (subtema). O processo de classificação na Anac consta de: leitura do texto; escolha do tema entre os onze temas; e escolha seguinte do subtema relacionado ao tema. Essas escolhas são feitas por menus suspensos e a escolha do tema limita as opções de subtemas somente aos que estejam relacionados, de modo a dar um pouco mais de agilidade ao processo.

Por exemplo, se numa certa semana há mil reclamações para classificar e a equipe tem dez pessoas em exercício, cada uma classifica cem reclamações. Há uma simples auditoria amostral não documentada sobre as classificações. Foi prevista uma auditoria mais organizada para estimar a acurácia da classificação manual, mas, por falta de recursos, ela não foi feita.

No momento, o produto relacionado à classificação é o *Boletim Trimestral de Monitoramento do Consumidor.gov.br - Transporte Aéreo* (ANAC, 2018), o qual mostra a distribuição das reclamações por empresas aéreas e por temas e subtemas. A intenção da Anac é conduzir uma análise mais profunda dentro dos subtemas de destaque, mas atualmente a Agência não possui recursos para realizá-la, devido ao tempo gasto na classificação.

Além da classificação em temas e subtemas, é avaliado se a reclamação decorre de uma eventual infração à Resolução nº 400 por parte da empresa aérea e, em caso afirmativo, quais artigos da Resolução teriam sido violados. Caso seja possível extrair essas informações do conteúdo da reclamação, anota-se se a operação é nacional ou internacional, bem como, se aplicável, o nome do aeroporto onde a reclamação aconteceu.

Essas informações são anexadas à base original extraída do Consumidor.gov.br, o que resulta numa base completa. A lista das variáveis desta base está no Anexo A.

### 3.1.3 Descrição da base

A base utilizada para este trabalho contempla 39.076 reclamações dos anos 2017 e 2018. A distribuição das reclamações está apresentada na Tabela 1 por temas e na Tabela 5, no Anexo D, também por subtemas. Na Tabela 1, destaca-se o tema "Assistência ao PNAE", que tem muito menos reclamações que os outros temas. E este problema agrava-se ao nível dos subtemas, entre os quais há alguns com poucas reclamações, como o subtema "Cão-guia", do tema "Assistência ao PNAE" (1 reclamação durante 2017-2018) ou o subtema "Declaração especial do valor de bagagem" do tema "Check-in e embarque" (2 reclamações durante 2017-2018; veja Tabela 5 do Anexo D). Portanto, não seria possível, nem teoricamente, uma classificação perfeita dos subtemas.

Ignorando o tema "Assistência ao PNAE", a distribuição das reclamações entre temas é bastante equilibrada. Foi feita uma simples análise de aglomeração, usando a metodologia *K-Means Clustering* para 2 a 20 clusters. Segundo a análise, o melhor número de classes para esta base seria aproximadamente entre 10 e 15, mas para a mudança de número de temas ou subtemas, seria necessária uma análise mais profunda que foge do escopo deste trabalho.

Para remover o desequilíbrio entre os temas, e eventualmente melhorar a acurácia da classificação automatizada, foi ponderada a exclusão do tema "Assistência ao PNAE" da base. Porém, isso se mostrou inviável porquanto não seria possível detectar as reclamações somente a este tema sem incorrer no risco de excluir as reclamações relacionadas a outros temas. Por exemplo, a autoclassificação pelo consumidor no momento da reclamação (um metadado da base) não tem classes que correspondam somente com esse tema. Isso é evidente quando se observa as Tabelas 2 e 3, em que, respectivamente, os grupos de problemas e problemas escolhidos pelos passageiros são tabulados com os temas da classificação da Anac. Não existe correspondência clara entre nenhum problema ou grupo de problemas e os temas. Porém, é possível ainda tentar usar estes metadados de autoclassificação na tentativa de melhorar a acurácia do modelo.

O modelo de linguagem é alimentado pelo conteúdo das reclamações (e dos pedidos

Tabela 1: Distribuição das reclamações na base por temas (Anac).

| Tema | Quantidade de reclamações |
|---|---|
| Reclamações contra valores e regras do contrato | 5.165 |
| Execução do voo | 5.097 |
| Oferta e compra | 4.914 |
| Transporte de bagagem | 4.452 |
| Alteração pela empresa aérea | 4.340 |
| Alteração pelo passageiro | 3.992 |
| Programas de Fidelidade | 3.521 |
| Outros | 2.484 |
| Reembolso | 2.414 |
| Check-in e embarque | 2.311 |
| Assistência ao PNAE | 386 |
| TOTAL | 39.076 |

Tabela 2: Tabulação cruzada entre grupos de problema (Consumidor.gov) e temas (Anac).

| Grupo Problema \ Tema | Alteração pela empresa aérea | Alteração pelo passageiro | Assistência ao PNAE | Check-in e embarque | Execução do voo | Oferta e compra | Outros | Programas de Fidelidade | Reclamações contra valores e regras do contrato | Reembolso | Transporte de bagagem | Total |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Atendimento / SAC | 522 | 850 | 154 | 495 | 559 | 774 | 738 | 463 | 514 | 342 | 525 | 5936 |
| Cobrança / Contestação | 334 | 1822 | 28 | 429 | 394 | 1816 | 528 | 431 | 3742 | 1657 | 474 | 11655 |
| Contrato / Oferta | 1308 | 1040 | 35 | 564 | 401 | 1835 | 533 | 1994 | 666 | 232 | 118 | 8726 |
| Entrega do Produto | 8 | 11 | 0 | 5 | 9 | 24 | 28 | 371 | 1 | 12 | 165 | 634 |
| Informação | 333 | 73 | 28 | 226 | 387 | 227 | 108 | 96 | 49 | 26 | 69 | 1622 |
| Saúde e Segurança | 18 | 5 | 21 | 21 | 78 | 11 | 144 | 5 | 5 | 2 | 54 | 364 |
| Vício de Qualidade | 1817 | 191 | 120 | 571 | 3269 | 227 | 405 | 161 | 188 | 143 | 3047 | 10139 |
| Total | 4340 | 3992 | 386 | 2311 | 5097 | 4914 | 2484 | 3521 | 5165 | 2414 | 4452 | 39076 |

à empresas aéreas que as acompanham). Na tentativa de melhorar a acurácia da classificação, outras variáveis da base podem também ser usadas, porém, a maioria serve somente como metadados.

## 3.2 Apresentação da metodologia utilizada (ULMFiT)

A classificação é conduzida pela metodologia *ULMFiT* de Howard e Ruder (2018), o atual estado da arte como foi apresentado no capítulo anterior. Na sequência é apresentada a implementação da metodologia conforme artigo dos autores citados.

Tabela 3: Tabulação cruzada entre problemas (Consumidor.gov) e temas (Anac).

| Problema \Tema | Alteração pela empresa aérea | Alteração pelo passageiro | Assistência ao PNAE | Check-in e embarque | Execução do voo | Oferta e compra | Outros | Programas de Fidelidade | Valores e regras do contrato | Reembolso | Transporte de bagagem | Total |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Alteração / rescisão de contrato sem solicitação / aviso prévio | 29 | 10 | 0 | 4 | 6 | 19 | 8 | 30 | 7 | 5 | 0 | 118 |
| Alteração da apólice sem solicitação / aviso prévio | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 2 |
| Atraso de voo | 124 | 13 | 0 | 29 | 1204 | 3 | 49 | 6 | 2 | 9 | 17 | 1456 |
| Cancelamento de voo | 1525 | 149 | 0 | 27 | 1192 | 96 | 78 | 14 | 168 | 79 | 19 | 3347 |
| Cancelamento indevido da passagem de volta | 268 | 212 | 3 | 60 | 131 | 146 | 27 | 5 | 24 | 13 | 5 | 894 |
| Clonagem, fraude, furto e roubo | 0 | 0 | 0 | 1 | 1 | 1 | 7 | 30 | 0 | 0 | 10 | 50 |
| Cláusulas irregulares / abusivas / de interpretação dúbia | 0 | 0 | 1 | 0 | 1 | 1 | 0 | 0 | 0 | 1 | 0 | 4 |
| Cobrança após cancelamento do serviço | 0 | 0 | 0 | 0 | 0 | 2 | 0 | 0 | 1 | 0 | 1 | 4 |
| Cobrança de tarifas, taxas, valores não previstos / não informados | 33 | 200 | 16 | 79 | 12 | 510 | 107 | 105 | 454 | 44 | 93 | 1653 |
| Cobrança em duplicidade / Cobrança referente a pagamento já efetuado | 4 | 33 | 0 | 19 | 11 | 358 | 45 | 8 | 13 | 56 | 54 | 601 |
| Cobrança indevida / abusiva para alterar ou cancelar o contrato | 80 | 821 | 3 | 216 | 24 | 265 | 68 | 80 | 2367 | 85 | 30 | 4039 |
| Cobrança indevida por excesso de bagagem | 5 | 6 | 3 | 5 | 3 | 102 | 15 | 8 | 32 | 6 | 114 | 299 |
| Cobrança por serviço não fornecido/ em desacordo com a utilização / fora do prazo | 1 | 5 | 0 | 4 | 4 | 34 | 8 | 34 | 4 | 7 | 2 | 103 |
| Cobrança por serviço/produto não contratado / não reconhecido / não solicitado | 7 | 25 | 2 | 10 | 3 | 195 | 53 | 19 | 14 | 24 | 13 | 365 |
| Cobrança submetendo a ofensa, constrangimento ou ameaça | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 3 | 1 | 3 | 8 |
| Dados pessoais incorretos – dificuldade de retificação | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 2 |
| Dados pessoais ou financeiros consultados, coletados, publicados ou repassados sem autorização | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 1 |
| Dificuldade / atraso na devolução de valores pagos / reembolso / retenção de valores | 201 | 722 | 4 | 94 | 337 | 342 | 226 | 176 | 828 | 1433 | 163 | 4526 |
| Dificuldade / demora para alterar o contrato de serviço (alteração de voo) | 346 | 385 | 1 | 25 | 33 | 38 | 36 | 23 | 164 | 15 | 4 | 1070 |
| Dificuldade / demora para alterar o nome do passageiro | 3 | 13 | 0 | 328 | 0 | 19 | 3 | 2 | 14 | 3 | 0 | 385 |
| Dificuldade / demora para cancelar o serviço | 27 | 191 | 0 | 3 | 4 | 18 | 10 | 11 | 182 | 58 | 1 | 505 |
| Dificuldade de contato / acesso a outros canais (exceto SAC) | 32 | 97 | 8 | 27 | 17 | 82 | 61 | 31 | 20 | 15 | 33 | 423 |
| Dificuldade de contato / demora no atendimento | 7 | 28 | 5 | 6 | 8 | 31 | 47 | 100 | 7 | 4 | 27 | 270 |
| Dificuldade para alterar ou cancelar o contrato /serviço | 14 | 62 | 0 | 21 | 0 | 15 | 17 | 45 | 41 | 18 | 2 | 235 |
| Dificuldade para contratar / recusa injustificada | 2 | 3 | 6 | 2 | 0 | 35 | 8 | 72 | 7 | 2 | 1 | 138 |
| Dificuldade para contratar o serviço (problema operacionais, acesso aos canais de venda etc) | 7 | 14 | 9 | 12 | 3 | 371 | 28 | 49 | 37 | 9 | 8 | 547 |
| Dificuldade para embarque com bagagem de mão | 1 | 2 | 3 | 39 | 5 | 11 | 15 | 0 | 0 | 0 | 142 | 218 |
| Dificuldades com Programas de Fidelidade / Pontuação (resgate, alteração, transferência etc) | 14 | 52 | 0 | 6 | 4 | 95 | 22 | 461 | 41 | 38 | 1 | 734 |
| Extravio/ avaria de bagagem, carga, correspondência (ressarcimento, indenizações) | 5 | 3 | 11 | 6 | 30 | 11 | 53 | 0 | 2 | 27 | 2638 | 2786 |
| Falha de segurança (furto, roubo, violação) | 1 | 1 | 0 | 3 | 1 | 5 | 23 | 3 | 0 | 0 | 180 | 217 |
| Falta de acessibilidade ao idoso ou à pessoa com deficiência | 1 | 1 | 34 | 3 | 2 | 2 | 3 | 0 | 1 | 0 | 1 | 48 |
| Falta de acesso a comunicação, alimentação, acomodação e/ou traslado em caso de atraso, cancelamento ou interrupção de voo | 75 | 2 | 3 | 23 | 329 | 8 | 25 | 1 | 2 | 3 | 5 | 476 |
| Falta de atendimento prioritário | 6 | 5 | 64 | 14 | 15 | 12 | 38 | 3 | 4 | 1 | 7 | 169 |
| Falta de informação sobre atraso, cancelamento ou interrupção de voo | 266 | 15 | 3 | 45 | 328 | 34 | 20 | 2 | 13 | 3 | 12 | 741 |
| Imposição de multa indevida / abusiva (não prevista em contrato ou em valor diverso do contratado) | 1 | 8 | 0 | 1 | 0 | 3 | 0 | 0 | 23 | 0 | 0 | 36 |
| Inadequação das instalações/ acomodações | 5 | 0 | 6 | 7 | 10 | 14 | 88 | 2 | 3 | 0 | 10 | 145 |
| Inadequação dos serviços a pessoas com necessidade de assistência especial (embarque, assento, equipamentos etc) | 0 | 0 | 44 | 4 | 7 | 11 | 21 | 1 | 1 | 0 | 2 | 91 |
| Informações incompletas / inadequadas sobre o serviço | 67 | 58 | 25 | 180 | 59 | 193 | 87 | 93 | 36 | 23 | 57 | 878 |
| Interrupção/instabilidade do fornecimento | 3 | 0 | 0 | 1 | 0 | 8 | 3 | 33 | 0 | 0 | 0 | 48 |
| Má qualidade do serviço | 0 | 1 | 0 | 0 | 3 | 0 | 0 | 0 | 0 | 0 | 0 | 4 |
| Má qualidade no atendimento (descortesia / despreparo/ constrangimento) | 12 | 17 | 9 | 26 | 50 | 28 | 39 | 49 | 5 | 6 | 23 | 264 |
| Má qualidade no atendimento presencial ou outros canais (exceto SAC) | 37 | 39 | 12 | 182 | 158 | 63 | 116 | 23 | 20 | 9 | 80 | 739 |
| Negativa de cobertura / Demora injustificada ou Divergência na indenização | 0 | 0 | 0 | 1 | 3 | 0 | 0 | 0 | 0 | 1 | 3 | 8 |
| Negativa de embarque de passageiro (preterição / overbooking) | 70 | 17 | 17 | 423 | 467 | 45 | 31 | 3 | 8 | 17 | 15 | 1113 |
| Negativação indevida (SPC, Serasa, SCPC etc) | 2 | 2 | 0 | 1 | 0 | 5 | 5 | 1 | 3 | 1 | 1 | 21 |
| Não entrega / demora na entrega do produto | 2 | 5 | 0 | 1 | 6 | 16 | 17 | 306 | 0 | 10 | 52 | 415 |
| Não entrega do contrato ou documentação relacionada ao serviço | 13 | 4 | 1 | 11 | 7 | 72 | 40 | 7 | 2 | 1 | 4 | 162 |
| Não envio / atraso do boleto ou fatura / dificuldade com o pagamento | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 2 |
| Oferta não cumprida / serviço não fornecido/ venda enganosa / publicidade enganosa | 578 | 79 | 14 | 78 | 209 | 960 | 303 | 1265 | 123 | 67 | 85 | 3761 |
| Produto entregue incompleto / diferente do pedido / danificado | 6 | 6 | 0 | 4 | 3 | 8 | 11 | 65 | 1 | 2 | 113 | 219 |
| Publicidade abusiva / ofensiva / discriminatória | 3 | 2 | 0 | 2 | 0 | 11 | 19 | 17 | 7 | 0 | 0 | 61 |
| Risco, dano físico ou mal estar decorrente da prestação do serviço/utilização do produto | 18 | 5 | 21 | 21 | 78 | 11 | 144 | 5 | 5 | 2 | 54 | 364 |
| SAC - Demanda não resolvida / não respondida / respondida após o prazo | 350 | 291 | 45 | 158 | 196 | 419 | 289 | 192 | 167 | 227 | 278 | 2612 |
| SAC - Descortesia / despreparo / demora | 29 | 28 | 8 | 43 | 95 | 41 | 70 | 16 | 10 | 13 | 47 | 400 |
| SAC - Dificuldade em obter protocolo, gravação ou histórico | 3 | 14 | 0 | 4 | 5 | 11 | 19 | 5 | 4 | 1 | 7 | 73 |
| SAC - Dificuldade para cancelar o serviço | 16 | 267 | 0 | 9 | 4 | 28 | 13 | 11 | 254 | 50 | 1 | 653 |
| SAC - Dificuldadede de contato / acesso | 30 | 64 | 3 | 26 | 11 | 59 | 46 | 33 | 23 | 16 | 22 | 333 |
| Serviço não realizado no prazo ou incompleto | 7 | 2 | 2 | 5 | 18 | 11 | 8 | 68 | 1 | 8 | 8 | 138 |
| Venda casada | 3 | 13 | 0 | 11 | 0 | 33 | 11 | 7 | 17 | 0 | 4 | 99 |
| Venda casada (imposição de seguro, empréstimo, título de capitalização etc) | 1 | 0 | 0 | 0 | 0 | 2 | 0 | 0 | 0 | 0 | 0 | 3 |
| Total | 4340 | 3992 | 386 | 2311 | 5097 | 4914 | 2484 | 3521 | 5165 | 2414 | 4452 | 39076 |

Primeiramente, um modelo de várias camadas é treinado com um corpus geral da linguagem, como o conjunto dos artigos do sítio eletrônico Wikipédia. Esse modelo chama-se Modelo de Linguagem (*Language Model*, LM) e quando alimentado com uma ou mais palavras, a palavra seguinte será prevista. Esse é um modelo universal que captura características gerais da linguagem e que não necessariamente tem um poder de previsão muito alto, justamente por ser um modelo genérico (MERITY; KESKAR; SOCHER, 2017). O modelo também precisa de um pouco mais de tempo para treinar e geralmente precisa ser treinado apenas uma vez. O processo do primeiro passo é apresentado na Figura 9a.

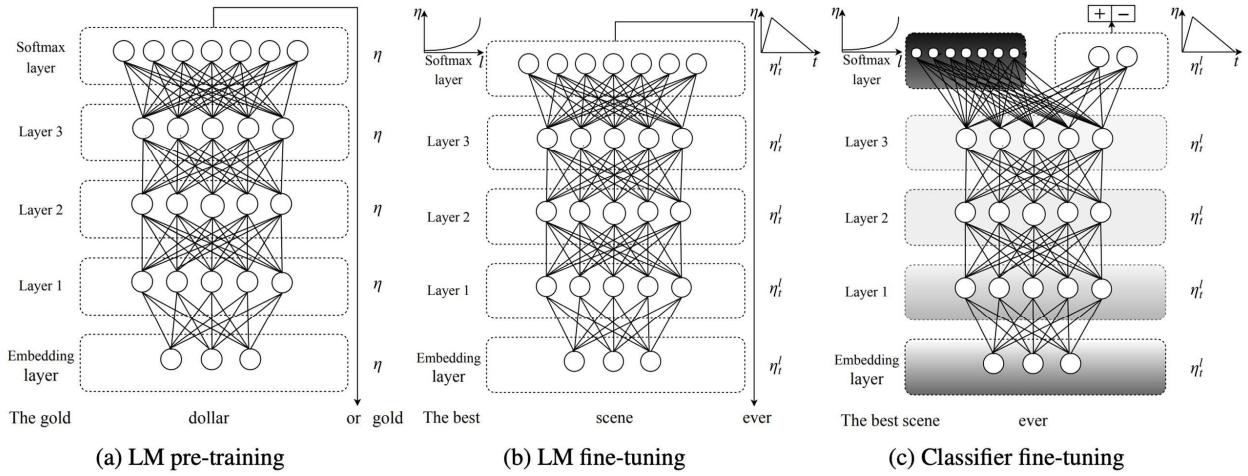


Figura 9: Os três passos do *ULMFiT*. (a) O pré-treinamento do modelo de linguagem universal com um corpus genérico ("The gold dollar or"). A última camada (a superior) da rede neural tem número de neurônios igual ao tamanho do vocabulário do modelo, o que pode ser dezenas de milhares de palavras. A palavra correspondente ao neurônio com valor mais alto é escolhida (aqui "gold"). (b) O modelo de linguagem é ajustado com o corpus de classificação ("The best scene"). Aqui, as taxas de aprendizagem são diferentes para cada camada e elas mudam conforme o progresso de treinamento, de forma triangular assimétrica, subindo primeiro rapidamente até seu valor máximo e depois decrescendo gradualmente. O modelo prevê a próxima palavra ("ever") ajustando-se ao corpo de classificação. (c) O modelo é modificado na última camada. Agora ele prevê a classificação definida (no exemplo da figura classificação binária de sentimento). As camadas são congeladas para um descongelamento gradual, mantendo o uso das técnicas para a taxa de aprendizagem.
Fonte: http://ruder.io

Depois, o corpus da classificação é usado para dar um ajuste fino ao modelo geral, justamente para prepará-lo para o último passo de classificação. Howard e Ruder (2018) propõem duas novas técnicas para este passo: Ajuste fino discriminativo que utiliza parâmetros

(taxas de aprendizado do modelo) diferentes para cada camada ou grupos de camada do modelo; e taxas de aprendizado triangulados assimétricas (*Slanted Triangular Learning Rates*, STLR) em que esses parâmetros se elevam, primeiramente, de um valor mínimo até um valor máximo e depois diminuem até alcançarem novamente o valor mínimo. Estas técnicas são apresentadas na Figura 9b.

Essas metodologias têm sido aperfeiçoadas empiricamente e cada aplicação pode requerer os próprios ajustes para que se achem os valores ótimos. Um trabalho recente aborda a definição de alguns dos hiperparâmetros para redes neurais de *Deep Learning* (SMITH, 2018).

O último passo é o treinamento do modelo para a classificação. Este passo é também um ajuste fino no sentido que o modelo é aumentado por duas camadas, que são suas últimas camadas. Os parâmetros dessas camadas são os únicos que são iniciados do zero para o modelo em questão. Para este passo de ajuste fino, o modelo antigo é "congelado" e gradualmente "descongelado" a partir das camadas finais as quais contêm informação mais específica e, portanto, os parâmetros dessas camadas são mais sujeitos a mudanças durante o treinamento. Estes processos são demonstrados na Figura 9c. No fim deste passo, o modelo inteiro é ajustado até a convergência e, assim, encontra-se pronto para ser validado e posteriormente usado para a aplicação final.

## 3.3 Aplicação da metodologia no caso da Anac

Na tarefa de classificação da Anac, o corpus da classificação é o conjunto de todas as reclamações já classificadas manualmente. Esse corpus é usado para o ajuste fino do modelo de linguagem, previamente treinado com o corpus de Wikipédia em português, sem a informação de classificação. No último passo, a informação de classificação é usada. Uma entrada é uma reclamação e a saída esperada é a classificação dela. É possível utilizar outras variáveis da base de classificações, se o resultado do treinamento da rede neural não for satisfatório usando apenas os conteúdos das reclamações. Uma parte das classificações é usada no treinamento em si e outra parte para validar o modelo, quando ele já tenta classificar a reclamação e compara o resultado para a classificação real.

# Capítulo 4 – Resultados

## 4.1 Descrição do processo

Neste capítulo descreve-se o processo de modelagem detalhadamente. A execução é feita no *Jupyter Notebook* na linguagem Python. Um *Notebook* é um ambiente de programação, onde é possível combinar o código de programação com texto formatável e gráficos, por exemplo. O *Notebook* da primeira execução do modelo deste trabalho pode ser encontrado no Anexo F.

Jeremy Howard, coautor do artigo e da metodologia *ULMFiT* (HOWARD; RUDER, 2018), criou um projeto chamado fast.ai, cujo objetivo é democratizar o uso das metodologias de *Deep Learning*. Para tal efeito, o projeto oferece cursos gratuitos e acesso às bibliotecas de programação desenvolvidas pelos colaboradores do projeto, incluindo todas as bibliotecas usadas na metodologia do *ULMFiT*. Portanto, códigos e notebooks prontos do fast.ai foram utilizados e modificados para língua portuguesa.

### 4.1.1 Modelo de linguagem

Para treinar o modelo de linguagem, um corpus de Wikipédia em português é carregado e alimentado ao treinador. O tamanho compactado do arquivo que contém milhares de artigos da Wikipédia portuguesa foi de 1,2 GB. Os valores dos hiperparâmetros são os mesmos usados nos trabalhos de Merity, Keskar e Socher (2017) e Howard e Ruder (2018). Em cada passo, a função de aprendizagem já divide o corpus em partes de treinamento (90%) e de validação (10%).

O modelo de linguagem tem três camadas ocultas e 1152 células LSTM em cada camada oculta. O treinamento do modelo durou 60 horas com um computador portátil com um processador Intel de sétima geração (i7-7700HQ) com 2,8 GHz de velocidade e com 16 GB de memória RAM; e com uma GPU NVidia GTX1060 com 6 GB de memória RAM dedicada. Para comparação, quando foi treinado o modelo de linguagem à direção contrária, para aproveitar uma eventual melhoria na acurácia da classificação do modelo bidirecional, foi utilizado o serviço de nuvem da Google. O modelo de linguagem à direção contrária tenta prever a palavra anterior no texto e, unindo os resultados dos dois modelos, é possível ter uma pequena melhoria na classificação (HOWARD; RUDER, 2018).

A Google oferece crédito de USD 300 para novos usuários para que testem os seus serviços de computação virtual. Nessa instância de máquina virtual, uma GPU potente, modelo NVidia Tesla V100 com 16 GB de memória RAM dedicada, além de 52 GB de

memória RAM, o mesmo modelo (porém à direção contrária) foi treinado em apenas 8,5 horas, equivalendo um custo monetário de aproximadamente 150 reais. Essa parte foi o passo mais demorado, mas, para futuros trabalhos, o modelo de linguagem está pronto e serão necessários somente os ajustes finos.

Para ilustrar o funcionamento do modelo de linguagem, gera-se com ele trinta palavras para uma sequência inicial de palavras. Quando as palavras são "`perdi meu voo`", a sequência se torna:

```
perdi meu voo em baixo alemão . Em meados de torno da sua produção , o
" povo " possui abaixo dos sucesso : Jovens ativos na depressão , alguns
```

Ou seja, o modelo não sabe produzir muito bem um texto relacionado à aviação. Mas, se alimentado com uma sequência, tal como "`estudar para a vida`", o resultado é mais coerente:

```
estudar para a vida dela . Além disso , já trabalhou com outro problema
envolvendo as reuniões de diversas regiões do país , como a
desenvolvimento , buscou e observar as condições áreas
```

### 4.1.2 Ajuste fino do modelo de linguagem

Para o passo tratado neste item, os conteúdos das 39.076 reclamações dos passageiros foram carregados e alimentados ao treinador do modelo. Como foi descrito no capítulo anterior, o objetivo é ajustar o modelo de linguagem para as peculiaridades da língua usada nas reclamações de transporte aéreo. De novo, os hiperparâmetros de Howard e Ruder (2018) são usados, porque, além de inglês, os mesmos parâmetros já geraram bons resultados em outras línguas. Esse passo durou aproximadamente cinco horas com o computador portátil mencionado e menos que uma hora com a máquina virtual da Google.

Agora vê-se como o modelo ajustado gera um texto diferente, alimentando com as mesmas sequências iniciais:

```
perdi meu voo de volta de Recife para Salvador na companhia dificulta
( mais de 10 passageiros ) porque tenho um grupo de melo no aeroporto que
ficou sem bagagem ,
```

e

```
estudar para a vida de minha filha , liguei em uma agência e a mesma me
informou que não poderia fazer nada pela minha solicitação pois eu havia
aberto uma reclamação na Gol
```

Agora pode-se observar que o modelo de fato se ajustou à linguagem de aviação (compara com a Figura 9b. Na sequência, pode-se treinar o classificador de reclamações.

### 4.1.3 Ajuste fino do classificador

Esse passo, que envolve descongelamento parcial das camadas da rede neural, durou duas horas no computador portátil e alguns minutos na máquina virtual da Google. Agora, o modelo não gera mais palavras, senão prevê a classificação (tema ou subtema) para a sequência de palavras (uma reclamação) alimentada a ele (veja ainda Figura 9c.

Da primeira vez, o classificador foi treinado somente com os teores de reclamações para os temas, tendo alcançado acurácia de 74,4%. Acurácia significa que a indicação do tema pelo classificador tem probabilidade de 74,4% de estar correta, ou que a indicação é correta em 74,4% dos casos. Com a utilização do modelo de linguagem inverso (levando ao modelo bidirecional), a acurácia subiu a 75,0%. Ou seja, o modelo prevê o tema correto, na média, em três a cada quatro vezes.

Para melhorar a acurácia, foi feita uma tentativa de incluir metadados, conforme Teschke e Sacash (2019) e Teschke, Sacash e Gumabay (2019), em que a acurácia melhorou em até 20%. Foram utilizados como metadados as autoclassificações dos passageiros em dois níveis, grupos de problema e problemas. Porém, no nosso modelo, a acurácia não melhorou, pelo contrário, piorou alguns decimais de por cento. Uma possível explicação para esse fato é que mesmo o modelo complexo de *Deep Learning* não consegue achar as dependências complexas entre os metadados e as classificações, como pode se observar nas Tabelas 2 e 3.

Como este resultado não parecia inicialmente muito satisfatório, foram incluídos o segundo e o terceiro temas mais prováveis, segundo o classificador. Ou seja, somam-se as probabilidades dos dois ou três temas mais prováveis. Com os dois temas mais prováveis (Top-2), a acurácia subiu a 90,2% (com o modelo bidirecional a 91,3%) e, com os três temas mais prováveis (Top-3), até 95,6% (modelo bidirecional até 96,0%). Esse ajuste aos resultados foi feito por Teschke (2018), cujo trabalho tem número parecido de classes e, portanto, os resultados também parecidos. Também, Marzagão (2016) considerou três classes mais prováveis em seu trabalho.

Na sequência, o classificador foi treinado para os subtemas. Esse passo tem que ser refeito desde o fim do passo anterior, mas, de qualquer forma, é o passo mais rápido, principalmente quando se usa a máquina virtual na nuvem. A acurácia para um subtema foi

de 59,2%, a acurácia "Top-2" de 74,9% e a acurácia "Top-3" de 82,2%.

### 4.1.4 Validação dos resultados

Para verificar que os resultados fossem estatisticamente estáveis, foi executada uma validação sistemática para o classificador de temas. Nessa validação, a base foi dividida em 10 partes e cada parte foi usada uma vez como base de validação; as demais sendo usadas como base de treinamento. Esse procedimento foi realizado 10 vezes, ou seja, tantas vezes a quantas partes a base foi dividida. Essa validação chama-se validação cruzada de dobras $k$ (*k-fold cross validation*). Como os temas não eram distribuídos equilibradamente, foi utilizada uma versão chamada validação cruzada estratificada de dobras $k$ (*stratified k-fold cross validation*), em que cada parte (dobra) tem aproximadamente a mesma porcentagem da representatividade de cada tema. O valor de $k$ foi escolhido 10, segundo a recomendação de Kohavi (1995).

A validação foi feita duas vezes, sem "embaralhar" a base e a embaralhando. A média das acurácias da primeira validação foi de 72,6%, um pouco inferior da acurácia do primeiro treinamento, de 74,4%. Como a base não tinha sido embaralhada, as dobras de cada tema podem ter tido algum viés e um dos passos resultou em acurácia muito abaixo da média, sendo 63,1%. Este valor afetou também o valor do desvio-padrão, o qual foi para as 10 acurácias de 3,6%. Por esse motivo, a mediana das acurácias ficou maior que a média, sendo 73,7%, já mais próxima da acurácia original. Já a validação com a base embaralhada gerou uma média de acurácias de 74,5%, com desvio-padrão de 0,7%. De qualquer forma, pode-se perceber que o valor original da acurácia não está muito longe de nenhuma média das validações.

## 4.2 Análise dos resultados obtidos

### 4.2.1 Acurácia da previsão dos temas e subtemas

A acurácia de classificar o tema corretamente ficou em 75%, aquém da meta de 95%. Porém, a meta foi definida muito alta. Como a acurácia de estado da arte se encontra em torno de 95% na classificação binária, já era previsível que seria difícil alcançar a mesma acurácia para as onze classes, ou ainda para as 66 classes.

Para entender a dificuldade de distinguir as classes entre si, observa-se a Tabela 4 em que há a matriz de confusão da classificação em temas, aqui elaborada pelas previsões do modelo na base de validação, a qual teve 10% do tamanho da base total. Em geral, a maioria (74,4%) tem sua classificação correta (na diagonal da matriz). Porém, uma grande

Tabela 4: Matriz de confusão das previsões no conjunto de validação (10% da base).

| Atuais \Previstos | Alteração pela empresa aérea | Alteração pelo passageiro | Assistência ao PNAE | Check-in e embarque | Execução do voo | Oferta e compra | Outros | Programas de Fidelidade | Valores e regras do contrato | Reembolso | Transporte de bagagem | Total |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Alteração pela empresa aérea | 354 | 20 | 0 | 3 | 30 | 5 | 9 | 2 | 4 | 10 | 0 | 437 |
| Alteração pelo passageiro | 7 | 220 | 0 | 4 | 6 | 29 | 9 | 11 | 51 | 24 | 0 | 361 |
| Assistência ao PNAE | 0 | 1 | 24 | 2 | 1 | 1 | 3 | 0 | 0 | 0 | 0 | 32 |
| Check-in e embarque | 2 | 6 | 5 | 201 | 15 | 7 | 5 | 1 | 8 | 6 | 3 | 259 |
| Execução do voo | 44 | 3 | 1 | 20 | 416 | 2 | 15 | 1 | 2 | 17 | 7 | 528 |
| Oferta e compra | 4 | 19 | 3 | 11 | 1 | 373 | 46 | 16 | 35 | 17 | 17 | 542 |
| Outros | 3 | 3 | 3 | 4 | 7 | 29 | 114 | 5 | 3 | 11 | 7 | 189 |
| Programas de Fidelidade | 4 | 4 | 0 | 0 | 0 | 25 | 9 | 272 | 6 | 4 | 1 | 325 |
| Valores e regras do contrato | 5 | 123 | 2 | 3 | 0 | 14 | 1 | 11 | 390 | 10 | 2 | 561 |
| Reembolso | 3 | 30 | 0 | 0 | 2 | 8 | 11 | 7 | 14 | 154 | 3 | 232 |
| Transporte de bagagem | 2 | 0 | 2 | 5 | 5 | 17 | 16 | 1 | 1 | 2 | 390 | 441 |
| Total | 428 | 429 | 40 | 253 | 483 | 510 | 238 | 327 | 514 | 255 | 430 | 3907 |

parte do tema “Reclamações contra valores e regras do contrato” (561 reclamações na base) foi classificada como tema “Alteração pelo passageiro” (123 vezes), o que é compreensível, uma vez que a diferença entre os dois temas é difícil de percepção até para um classificador humano. Outro destaque é a previsão do tema “Outros” mais vezes em outros temas do que no próprio tema (124 previsões fora do tema, 114 dentro do tema). Isso é também compreensível pela natureza do tema; no tema “Outros” consta qualquer reclamação que não se encaixa em outros temas.

Uma das explicações é que o tema “correto” nem sempre é claro para o classificador humano, outra é que uma reclamação pode ter elementos de mais de um tema. E no momento da classificação, o classificador escolhe o tema mais importante, segundo a sua opinião, mesmo se este tema tivesse uma pequena menção na reclamação e a maior parte da reclamação falaria sobre o tema que não foi considerado o mais importante. O ser humano pode julgar as importâncias dos temas assim, mas isto é algo que a máquina ainda não consegue fazer, mesmo que tivesse uma acurácia próxima a 100%.

Por outro lado, não há uma estimativa da acurácia das classificações manuais. Como este trabalho de classificação é conduzido por várias pessoas e não harmonizado posteriormente, tendo apenas uma auditoria não documentada por amostra não definida cientificamente, é certo que a acurácia não é de 100%. Considerando, inclusive, que há opiniões diferentes entre pessoas a respeito da classificação e, como foi comentado no parágrafo anterior, uma reclamação pode ter mais que um problema (tema ou subtema). Nessa classificação apenas um tema tinha que ser escolhido e pode haver ainda divergências entre as pessoas sobre o tema mais importante.

Se o classificador humano pode ficar na dúvida entre os temas, o classificador automático também tem confianças diferentes entre eles. Como a função *softmax* produz probabilidade para cada tema, o classificador automático poderia dar a sua confiança da classificação. Quando a probabilidade do tema fosse, por exemplo, 90% ou acima, não seria necessária intervenção humana. Porém, se o tema mais provável ficasse por volta de 50% e o segundo tema mais provável tivesse quase mesma probabilidade, o classificador automático já pediria a intervenção. Desta forma, o classificador automático já poderia economizar bastante tempo do classificador humano.

O plano original era que fosse realizada uma auditoria mais estruturada pelos servidores mais experientes da equipe de classificação, para que se obtivesse uma estimativa, mesmo que aproximada, sobre a acurácia humana. Mas, por falta de te tempo desses servidores, a auditoria não foi realizada e, portanto, não há uma estimativa da acurácia humana. Como uma tentativa de estimar a variação de acurácia entre classificadores humanos, a base inteira é reclassificada pelo classificador automático e observa-se a acurácia das classificações automáticas em comparação com as classificações realizadas pelos 16 servidores que participaram da classificação durante 2017–2018. Como o modelo foi treinado com essa base, não se pode considerar as acurácias como números absolutos, tendo elas alcançado percentuais acima de 75% na maioria dos casos. Porém, a amplitude entre a maior e a menor acurácia ficou em 13,6%, ou seja, uma variação considerável. Isso nos indica que a acurácia individual pode chegar no mínimo bem abaixo de 90% e, na prática, até a acurácia média pode estar, na melhor das hipóteses, por volta de 90%.

Para comparação, um grupo de estudos de PLN e DL em Brasília que teve acesso à base, criou uma referência da acurácia usando PLN tradicional, além de incentivar e colaborar com o projeto. O melhor resultado foi alcançado pela metodologia *TF-IDF* (*Term Frequency - Inverse Document Frequency* ou frequência de termo - frequência inverso de documento) e regressão logística, sendo a acurácia 73,2% para temas e 53,0% para subtemas. Mesmo que a acurácia para temas seja somente 1,8 pontos percentuais melhor com *ULMFiT* (mas já 6,2 pontos percentuais melhor para subtemas), vale observar que isso significa uma diminuição relativa de 7% do erro da classificação. Quando as acurácias ficam acima de 50% e, principalmente, quando elas ficam acima de 90%, é interessante mudar a ótica ao erro de classificação (o que é 1 - acurácia) para ter melhor entendimento sobre os efeitos de até uma pequena melhora na acurácia. Convém ainda ressaltar que os métodos tradicionais do PLN não houveram avanços significativos durante os últimos anos, enquanto os métodos baseados em DL têm avançado praticamente mensalmente desde 2018 e ainda não chegaram ao seu ponto de saturação.

# Capítulo 5 – Conclusões

Este trabalho apresentou a evolução do processamento de linguagem natural (PLN) no âmbito de classificação de textos, desde seu início até seu estado da arte atual, que é a utilização de metodologias baseadas em *Deep Learning* (DL), ramo da inteligência artificial que está se desenvolvendo em uma velocidade muito rápida. O estudo de caso foi a classificação de reclamações de usuários de transporte aéreo. A metodologia aplicada foi o *ULMFiT* (HOWARD; RUDER, 2018) e a base de dados continha 39.076 reclamações manualmente classificadas, contemplando os anos de 2017 e 2018.

## 5.1 Apresentação da resposta à pergunta de pesquisa

A pergunta de pesquisa era "Até qual nível de sucesso é possível chegar na classificação automática das reclamações dos usuários dos serviços de transporte aéreo?". No nível de tema, a classificação automática chegou à acurácia de 75% e, no nível de subtema, a de 59%. Ou seja, a meta desafiadora não foi alcançada. Isso significa que não se poderá ainda automatizar toda a classificação na Anac uma vez que a probabilidade de releitura das já classificadas seria baixa, principalmente daquelas que forem classificadas em assuntos não relacionados a infrações. E se for considerado que essas reclamações tenham informação de infração da empresa aérea, essa informação poderia ser ignorada.

Por outro lado, para uma classificação de demandas a serem tratadas por destinatários diferentes, uma acurácia de 75% pode ser adequada porque em vez de leitura prévia de todas as demandas, somente aquelas distribuídas equivocadamente serão lidas duas vezes. Apenas isso já criaria uma economia de recursos. Obviamente não se sabe ainda a acurácia para outra finalidade, porque precisa-se de uma base classificada dessas demandas para retreinar o classificador. E, também, o número de classes sempre afetará a acurácia. Contudo, o resultado atual já é promissor e justifica pesquisas com outras bases de textos.

Se for considerada a acurácia da classificação entre os três temas mais prováveis (Top-3), ela chega a 96%, o que já é bem confiável (no caso de subtemas, a acurácia "Top-3" chegou a 82%). No mínimo, isso pode agilizar uma classificação manual, mas uma aplicação mais interessante poderia ser na auditoria da classificação. Lá a primeira filtragem de verificação poderia ser realizada com o classificador automático, auditando primeiramente aquelas classificações que não estivessem nem entre as três classes mais prováveis.

## 5.2 Enquadramento da resposta obtida à pergunta de pesquisa no estado da arte

A meta original de acurácia da classificação automatizada foi definida muito alta. Esse tipo de acurácia tem sido alcançado normalmente quando o número de classes é baixo, como na classificação binária. Um outro trabalho com número de classes similar chegou a ter acurácias parecidas (TESCHKE, 2018). Ou seja, logicamente a acurácia tende a baixar quando o número de classes aumenta. Considerando isso, o resultado do estudo foi satisfatório e incentiva a continuação das pesquisas com novas metodologias e técnicas que surgiram durante a realização deste estudo.

## 5.3 Síntese dos avanços metodológicos e empíricos

Esta dissertação avançou na aplicação de metodologias de DL em língua portuguesa, a qual já vinha sendo conduzida, mas ainda carece de trabalhos acadêmicos publicados. Existe um grupo de estudos de DL e PLN em Brasília. Esse grupo, que já capacitou dezenas de pessoas interessadas, incentivou e apoiou também a realização deste trabalho. Os participantes do grupo já realizaram vários projetos, mas, por enquanto, a divulgação deles tem acontecido mais em eventos e *blogs* na internet. Num prazo muito curto espera-se ter mais trabalhos acadêmicos do DL aplicado a PLN em português.

## 5.4 Apresentação dos limites da pesquisa realizada

Neste trabalho não foi tratada a escolha da dimensão da rede ou o ajuste dos hiperparâmetros. Ambos foram escolhidos segundo o trabalho de Howard e Ruder (2018), que já obteve bons resultados em várias línguas, usando os mesmos parâmetros do trabalho original. Porém, além do ajuste dos hiperparâmetros, o número de camadas e de neurônios em cada camada pode ser escolhido. Como essas escolhas se baseiam ainda em experimentos empíricos, os ajustes fogem do escopo deste trabalho e poderão ser tratados em pesquisas futuras.

Uma deficiência do estudo era a falta da estimativa da acurácia da classificação manual. Como mais de dez pessoas classificaram as reclamações, é certo que a acurácia não deve ser 100%, provavelmente, nem perto. Assim, foi apresentada uma estimativa da variação da acurácia de aproximadamente 14%, o que indica que, na melhor das hipóteses, a acurácia média da classificação manual estaria por volta de 90%. Contudo, o fato de o *ULMFiT* superar a acurácia dos métodos tradicionais do PLN já convalida sua importância.

A metodologia *ULMFiT* é também muito menos exigente de recursos do que outras metodologias do estado da arte. Convém ressaltar que as metodologias do PLN tradicional ainda são menos exigentes, porém elas parecem ter chegado no limite de seu desempenho.

Quando a metodologia *ULMFiT* foi escolhido, no começo de 2019, ele era o estado da arte na classificação de textos no sítio eletrônico paperswithcode.com. No decorrer deste projeto de dissertação, o *ULMFiT* perdeu três posições na tarefa de classificação mais parecida com a da Anac, ocupando o quarto lugar em agosto de 2019. Dos três métodos que estão à frente do *ULMFiT*, os dois primeiros são de uma escala totalmente diferente e o custo de treinamento deles é estimado em milhares ou dezenas de milhares de dólares.

Porém, se esses métodos oferecerem modelos de linguagem pré-treinados – e, portanto, prontos para usar – em língua portuguesa, eles serão uma opção óbvia para a continuação desta pesquisa. De qualquer forma, o estado da arte do PLN é um alvo em movimento e novas metodologias que não exijam muitos recursos devem surgir em breve possibilitando que essas sejam prontamente testadas.

## 5.5 Trabalhos futuros

Ainda que este trabalho consiga responder à pergunta de pesquisa, ele criou outras perguntas. Estas perguntas podem criar novas pesquisas imediatas. Por exemplo, é possível, ainda posteriormente, realizar a auditoria sistemática da base. Dependendo dos resultados dessa auditoria, a aplicabilidade imediata do classificador resultante deste trabalho pode ser reavaliada.

Outro trabalho imediato seria uma remodelagem com um subconjunto da base. Durante a realização deste trabalho, houve uma redefinição da árvore de classificação. Além disso, a responsabilidade da classificação foi transferida para as empresas aéreas. Há aproximadamente 8.500 classificações com a nova árvore de classificação. Assim, com a base original, é possível testar se e como a acurácia diminui, reduzindo o tamanho dela. Se a acurácia não diminuir muito, será possível criar o novo classificador com a base menor, a ser utilizado na auditoria das classificações.

Um projeto acadêmico adicional seria um estudo empírico sobre a variação dos hiperparâmetros e das dimensões da rede LSTM. Como esse ramo de pesquisa ainda carece de uma teoria robusta, é possível que outros parâmetros produzam melhores resultados. Há espaço tanto para trabalhos empíricos como teóricos, ainda mais com o rápido desenvolvimento dos resultados cada vez melhores, mesmo sem uma robusta base teórica.

A base de 39 mil reclamações classificadas é uma boa base para várias pesquisas, com potencial para avançar a pesquisa do PLN na língua portuguesa.

## Referências

ANAC. *Planejamento Estratégico: Plano Estratégico – 2015/2019*. Brasília: Agência Nacional de Aviação Civil, 2015. 62 p. Disponível em: <https://www.anac.gov.br/A_Anac/institucional/ANAC_Planj_estrategico_2015_2016.pdf>. Citado 2 vezes nas páginas 2 e 3.

ANAC. *Resolução nº 400, de 13 de dezembro de 2016*. Brasília: Agência Nacional de Aviação Civil, 2016. Disponível em: <https://www.anac.gov.br/assuntos/legislacao/legislacao-1/resolucoes/resolucoes-2016/resolucao-no-400-13-12-2016/@@display-file/arquivo_norma/RA2016-0400%20-%20Compilado%20at%C3%A9%20RA2017-0434.pdf>. Citado 2 vezes nas páginas 1 e 16.

ANAC. *Boletim Trimestral de Monitoramento do Consumidor.gov.br - Transporte Aéreo*. Brasília, 2018. Disponível em: <https://www.anac.gov.br/noticias/2019/anac-divulga-dados-de-manifestacoes-de-passageiros-do-transporte-aereo/BoletimTrimestraldeMonitoramentodoConsumidorgovbr_3trimestre_2018.pdf>. Citado na página 17.

BRANDÃO JÚNIOR, F. R. *O Papel das Agências Reguladoras na Formação da Agenda e na Formulação de Políticas Públicas*. 41 p. Dissertação (Artigo Científico para Conclusão de Curso), Brasília, 2017. Citado na página 2.

BRASIL. *Lei nº 8.078, de 11 de setembro de 1990. Código de Defesa do Consumidor*. 1990. Disponível em: <http://www.planalto.gov.br/ccivil_03/leis/l8078.htm>. Citado 2 vezes nas páginas 2 e 16.

CAMBRIA, E.; WHITE, B. Jumping nlp curves: A review of natural language processing research [review article]. *Comp. Intell. Mag.*, IEEE Press, Piscataway, NJ, USA, v. 9, n. 2, p. 48–57, maio 2014. ISSN 1556-603X. Disponível em: <https://doi.org/10.1109/MCI.2014.2307227>. Citado na página 4.

DEVLIN, J. et al. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. *CoRR*, abs/1810.04805, 2018. Disponível em: <http://arxiv.org/abs/1810.04805>. Citado na página 2.

EISENSTEIN, J. *Natural Language Processing*. Cambridge, MA: The MIT Press, 2018. 587 p. Disponível em: <https://github.com/jacobeisenstein/gt-nlp-class/blob/master/notes/eisenstein-nlp-notes.pdf>. Citado na página 6.

FERNANDES, W. P. D. et al. Appellate Court Modifications Extraction for Portuguese. *Artificial Intelligence and Law*, jul. 2019. ISSN 0924-8463, 1572-8382. Disponível em: <http://link.springer.com/10.1007/s10506-019-09256-x>. Citado na página 5.

GOLDBERG, Y. Neural Network Methods for Natural Language Processing (Synthesis Lectures on Human Language Technologies). *Synthesis Lectures on Human Language Technologies*, v. 10, n. 1, p. 1–309, 2017. ISSN 1947-4040. Disponível em: <http:

//www.morganclaypool.com/doi/10.2200/S00762ED1V01Y201703HLT037>. Citado 7 vezes nas páginas 4, 5, 8, 9, 10, 11 e 14.

GOODFELLOW, I.; BENGIO, Y.; COURVILLE, A. *Deep Learning*. Cambridge, MA: The MIT Press, 2016. <http://www.deeplearningbook.org>. Citado 2 vezes nas páginas 7 e 10.

HOCHREITER, S.; SCHMIDHUBER, J. Long short-term memory. *Neural computation*, MIT Press, v. 9, n. 8, p. 1735–1780, 1997. Disponível em: <http://www.bioinf.jku.at/publications/older/2604.pdf>. Citado 2 vezes nas páginas 11 e 15.

HOWARD, J.; RUDER, S. Universal Language Model Fine-tuning for Text Classification. *CoRR*, abs/1801.06146, 2018. ISSN 23330384. Disponível em: <http://arxiv.org/abs/1801.06146>. Citado 9 vezes nas páginas 2, 14, 15, 19, 21, 23, 24, 29 e 30.

JACKSON, P.; SCHILDER, F. Natural Language Processing: Overview. In: *Encyclopedia of Language & Linguistics*. Amsterdam: Elsevier Press, 2006. p. 503–518. ISBN 978-0-08-044854-1. Disponível em: <http://linkinghub.elsevier.com/retrieve/pii/B0080448542009275>. Citado na página 4.

JONES, K. S. Natural Language Processing : a Historical Review. *Natural Language Processing : a Historical Review*, v. 9, n. 10, p. 53–59, 2001. Disponível em: <https://www.cl.cam.ac.uk/archive/ksj21/histdw4.pdf>. Citado na página 4.

KOHAVI, R. A study of cross-validation and bootstrap for accuracy estimation and model selection. In: *Proceedings of the 14th International Joint Conference on Artificial Intelligence - Volume 2*. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 1995. (IJCAI'95), p. 1137–1143. ISBN 1-55860-363-8. Disponível em: <http://dl.acm.org/citation.cfm?id=1643031.1643047>. Citado na página 26.

MARTINS, M. S. M. A implementação de políticas públicas por meio das agências reguladoras. *Revista Jus Navigandi*, Teresina, ano 16, n. 2761, janeiro 2011. Disponível em: <https://jus.com.br/artigos/18332>. Citado na página 2.

MARZAGÃO, T. Using NLP to measure democracy. *CoRR*, abs/1502.06161, 2015. Disponível em: <http://arxiv.org/abs/1502.06161>. Citado na página 5.

MARZAGÃO, T. Using SVM to pre-classify government purchases. *CoRR*, abs/1601.02680, 2016. Disponível em: <http://arxiv.org/abs/1601.02680>. Citado 2 vezes nas páginas 5 e 25.

MERITY, S.; KESKAR, N. S.; SOCHER, R. Regularizing and Optimizing LSTM Language Models. *CoRR*, abs/1708.02182, ago. 2017. Disponível em: <http://arxiv.org/abs/1708.02182>. Citado 3 vezes nas páginas 15, 21 e 23.

MONTEIRO, M. *Modelo de Linguagem para Português com ULMFit*. 2018. Disponível em: <https://github.com/monilouise/deep-learning/blob/master/nlp/language-models/pt/Modelo%20de%20Linguagem%20para%20Portugue%CC%82s%20com%20ULMFit.pdf>. Citado na página 5.

NADKARNI, P. M.; OHNO-MACHADO, L.; CHAPMAN, W. W. Natural language processing: An introduction. *Journal of the American Medical Informatics Association*, v. 18, n. 5, p. 544–551, 2011. ISSN 10675027. Disponível em: <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC3168328/>. Citado na página 4.

NIELSEN, M. A. *Neural Networks and Deep Learning*. Determination Press, 2015. Disponível em: <http://neuralnetworksanddeeplearning.com/>. Citado 3 vezes nas páginas 6, 8 e 10.

OLAH, C. *Understanding LSTM Networks*. 2015. Disponível em: <http://colah.github.io/posts/2015-08-Understanding-LSTMs/>. Citado 2 vezes nas páginas 12 e 13.

ORAIR, G. H. *Classificação multi-rótulo hierárquica automática de documentos textuais*. Dissertação (mestrado), 2009. Disponível em: <https://repositorio.ufmg.br/bitstream/1843/SLSS-7WMHNG/1/gustavohenriqueorair.pdf>. Citado na página 5.

PAULA, E. L. et al. Deep learning anomaly detection as support fraud investigation in brazilian exports and anti-money laundering. In: *2016 15th IEEE International Conference on Machine Learning and Applications (ICMLA)*. Los Alamitos, CA, USA: IEEE Computer Society, 2016. p. 954–960. Disponível em: <https://doi.ieeecomputersociety.org/10.1109/ICMLA.2016.0172>. Citado na página 10.

PECLAT, R. N. *Avaliação Semântica da Integração da Gestão de Riscos de Segurança em Documentos de Software da Administração Pública*. 125 p. Dissertação (Mestrado Profissional em Computação Aplicada) — Universidade de Brasília, 2015. Disponível em: <http://repositorio.unb.br/handle/10482/18827>. Citado na página 5.

SACHAN, D. S.; ZAHEER, M.; SALAKHUTDINOV, R. Investigating the working of text classifiers. *CoRR*, abs/1801.06261, 2018. Disponível em: <http://arxiv.org/abs/1801.06261>. Citado na página 5.

SANTOS, C. M. dos. *Classificação de Documentos com Processamento de Linguagem Natural*. 217 p. Dissertação (Mestrado) — Instituto Superior de Engenharia de Coimbra, 2015. Disponível em: <http://files.isec.pt/DOCUMENTOS/SERVICOS/BIBLIO/Teses/Tese_Mest_Cedric-Michael-Santos.pdf>. Citado na página 9.

SENACON. *Consumidor.gov.br*. 2019. Secretaria Nacional do Consumidor. Disponível em: <https://www.consumidor.gov.br/pages/conteudo/sobre-servico>. Citado na página 16.

SHETTY, R. et al. Study of emotion detection in tunes using machine learning. *International Journal of Scientific and Research Publications*, v. 5, p. 5, 11 2015. Disponível em: <http://www.ijsrp.org/research-paper-1115.php?rp=P474811>. Citado na página 6.

SMITH, L. N. A disciplined approach to neural network hyper-parameters: Part 1 – learning rate, batch size, momentum, and weight decay. *CoRR*, abs/1803.09820, 2018. Disponível em: <http://arxiv.org/abs/1803.09820>. Citado na página 22.

TESCHKE, M. *NLP Transfer Learning on SageMaker | Novetta Nexus*. 2018. Disponível em: <https://www.novetta.com/2018/10/nlp-transfer-learning-sagemaker/>. Citado 2 vezes nas páginas 25 e 30.

TESCHKE, M.; SACASH, B. *Introducing Metadata Enhanced ULMFiT*. 2019. Disponível em: <https://www.novetta.com/2019/03/introducing_me_ulmfit/>. Citado na página 25.

TESCHKE, M.; SACASH, B.; GUMABAY, E. *20% Accuracy Bump in Text Classification with ME-ULMFiT*. 2019. Disponível em: <https://www.novetta.com/2019/06/accuracy_bump_meulmfit/>. Citado na página 25.

TONIDANDEL, D.; ARAUJO, A. A função delta revisitada: De Heaviside a Dirac. *Revista Brasileira de Ensino de Física*, scielo, v. 37, p. 3306–1 – 3306–9, 09 2015. ISSN 1806-1117. Disponível em: <http://www.scielo.br/scielo.php?script=sci_arttext&pid=S1806-11172015000300306&nrm=iso>. Citado na página 7.

VALLE, L. M. *Airlines tripadvisor's comments analysis using NLP*. 2019. Disponível em: <https://github.com/LeticiaMValle/TripAdvisor-s-NLP-portuguese/blob/master/Estudo_orientado_2.pdf>. Citado na página 5.

VARASCHIN, C.; LIMA, V. L. S. D. *Fundamentos do processamento estatístico da linguagem natural*. Porto Alegre, 2001. v. 7, n. 021, 59 p. Relatório técnico. Disponível em: <http://www.pucrs.br/facin-prov/wp-content/uploads/sites/19/2016/03/tr021.pdf>. Citado 2 vezes nas páginas 4 e 5.

ZHANG, A. et al. Dive into Deep Learning. 2019. <http://www.d2l.ai>. Citado na página 10.

Anexos

# ANEXO A – Lista de variáveis da base do Consumidor.gov.br

- Gestor
- Protocolo
- Canal de Origem
- Consumidor
- CPF
- Região
- UF
- Cidade
- Sexo
- Faixa Etária
- Ano Abertura
- Mês Abertura
- Data Abertura
- Hora Abertura
- Data Resposta
- Hora Resposta
- Data Análise
- Hora Análise
- Data Recusa
- Hora Recusa
- Data Finalização
- Hora Finalização
- Prazo Resposta
- Prazo Analise Gestor
- Tempo Resposta
- Grupo Econômico
- Nome Fantasia
- Segmento de Mercado
- Área
- Assunto
- Grupo Problema
- Problema
- Como Comprou Contratou

- Procurou Empresa
- Técnico do Último Trâmite
- Respondida
- Situação
- Avaliação Reclamação
- Nota do Consumidor
- Análise da Recusa
- Edição de Conteúdo
- Interação do Gestor
- RESPONSÁVEL
- OPERAÇÃO
- TEMA
- SUBTEMA
- AEROPORTO 20+
- AEROPORTO Outros
- INDÍCIOS
- ART.1
- ART.2
- ART.3

# ANEXO B – Classificação do Consumidor.gov.br - Grupos de problema e Problemas

Estes grupos de problema e problemas são disponíveis quando a área de reclamação é selecionada como “Transportes” e o assunto como “Aéreo”.

1. Atendimento / SAC
    - 1.1. Dificuldade de contato / acesso a outros canais (exceto SAC)
    - 1.2. Dificuldade de contato / demora no atendimento
    - 1.3. Falta de atendimento prioritário
    - 1.4. Má qualidade no atendimento (descortesia / despreparo/ constrangimento)
    - 1.5. Má qualidade no atendimento presencial ou outros canais (exceto SAC)
    - 1.6. SAC - Demanda não resolvida / não respondida / respondida após o prazo
    - 1.7. SAC - Descortesia / despreparo / demora
    - 1.8. SAC - Dificuldade em obter protocolo, gravação ou histórico
    - 1.9. SAC - Dificuldade para cancelar o serviço
    - 1.10. SAC - Dificuldadede de contato / acesso
2. Cobrança / Contestação
    - 2.1. Cobrança após cancelamento do serviço
    - 2.2. Cobrança de tarifas, taxas, valores não previstos / não informados
    - 2.3. Cobrança em duplicidade / Cobrança referente a pagamento já efetuado
    - 2.4. Cobrança indevida / abusiva para alterar ou cancelar o contrato
    - 2.5. Cobrança indevida por excesso de bagagem
    - 2.6. Cobrança por serviço não fornecido/ em desacordo com a utilização / fora do prazo
    - 2.7. Cobrança por serviço/produto não contratado / não reconhecido / não solicitado
    - 2.8. Cobrança submetendo a ofensa, constrangimento ou ameaça
    - 2.9. Dificuldade / atraso na devolução de valores pagos / reembolso / retenção de valores
    - 2.10. Imposição de multa indevida / abusiva (não prevista em contrato ou em valor diverso do contratado)
    - 2.11. Negativação indevida (SPC, Serasa, SCPC etc)
3. Contrato / Oferta
    - 3.1. Alteração / rescisão de contrato sem solicitação / aviso prévio

3.2. Alteração da apólice sem solicitação / aviso prévio
3.3. Cancelamento indevido da passagem de volta
3.4. Cláusulas irregulares / abusivas / de interpretação dúbia
3.5. Dificuldade / demora para alterar o contrato de serviço (alteração de voo)
3.6. Dificuldade / demora para alterar o nome do passageiro
3.7. Dificuldade / demora para cancelar o serviço
3.8. Dificuldade para alterar ou cancelar o contrato /serviço
3.9. Dificuldade para contratar / recusa injustificada
3.10. Dificuldade para contratar o serviço (problema operacionais, acesso aos canais de venda etc)
3.11. Dificuldades com Programas de Fidelidade / Pontuação (resgate, alteração, transferência etc)
3.12. Não entrega do contrato ou documentação relacionada ao serviço
3.13. Negativa de cobertura / Demora injustificada ou Divergência na indenização
3.14. Oferta não cumprida / serviço não fornecido/ venda enganosa / publicidade enganosa
3.15. Publicidade abusiva / ofensiva / discriminatória
3.16. Venda casada
3.17. Venda casada (imposição de seguro, empréstimo, título de capitalização etc)

4. Entrega do Produto

4.1. Não entrega / demora na entrega do produto
4.2. Produto entregue incompleto / diferente do pedido / danificado

5. Informação

5.1. Dados pessoais incorretos – dificuldade de retificação
5.2. Dados pessoais ou financeiros consultados, coletados, publicados ou repassados sem autorização
5.3. Falta de informação sobre atraso, cancelamento ou interrupção de voo
5.4. Informações incompletas / inadequadas sobre o serviço

6. Saúde e Segurança

6.1. Risco, dano físico ou mal estar decorrente da prestação do serviço/utilização do produto

7. Vício de Qualidade

7.1. Atraso de voo

7.2. Cancelamento de voo
7.3. Clonagem, fraude, furto e roubo
7.4. Dificuldade para embarque com bagagem de mão
7.5. Extravio/ avaria de bagagem, carga, correspondência (ressarcimento, indenizações)
7.6. Falha de segurança (furto, roubo, violação)
7.7. Falta de acessibilidade ao idoso ou à pessoa com deficiência
7.8. Falta de acesso a comunicação, alimentação, acomodação e/ou traslado em caso de atraso, cancelamento ou interrupção de voo
7.9. Inadequação das instalações/ acomodações
7.10. Inadequação dos serviços a pessoas com necessidade de assistência especial (embarque, assento, equipamentos etc)
7.11. Interrupção/instabilidade do fornecimento
7.12. Má qualidade do serviço
7.13. Não envio / atraso do boleto ou fatura / dificuldade com o pagamento
7.14. Negativa de embarque de passageiro (preterição / overbooking)
7.15. Serviço não realizado no prazo ou incompleto

# ANEXO C – Classificação da Anac - Temas e subtemas

O 11º tema seria Programas de Fidelidade, cujas reclamações não entram na análise pelo eventual viés da análise porque algumas empresas aéreas têm uma pessoa jurídica separada a qual recebe as reclamações, e em outros casos a própria empresa aérea recebe essas reclamações. A Anac tampouco regula os programas de fidelização de cliente.

Porém, como não é possível separar as reclamações sobre programas de fidelidade antes de qualquer análise, elas são classificadas e incluídas na base de classificação.

1. Oferta e compra
    - 1.1. Cancelamento da passagem pela empresa aérea (inclusive por suspeita de fraude)
    - 1.2. Comprovante de passagem aérea
    - 1.3. Informação/Regras e valores de itens opcionais
    - 1.4. Informação/Regras e valores do transporte de bagagem
    - 1.5. Informação/Regras e valores do transporte de bagagem especial, animais, carga e itens proibidos
    - 1.6. Informação/Regras e valores do transporte de passageiros
    - 1.7. Outros
2. Alteração pelo passageiro
    - 2.1. Cancelamento automático do retorno (no-show na ida)
    - 2.2. Desistência 24/7
    - 2.3. Multas indevidas e devolução da tarifa aeroportuária
    - 2.4. Remarcação e resilição
    - 2.5. Outros
3. Alteração pela empresa aérea
    - 3.1. Alternativas (reacomodação, reembolso integral ou outra modalidade de transporte)
    - 3.2. Assistência material (comunicação, alimentação, hospedagem e traslado)
    - 3.3. Aviso
    - 3.4. Outros
4. Check-in e embarque
    - 4.1. Atraso do passageiro
    - 4.2. Correção de nome (inclusive negativa de embarque)
    - 4.3. Dificuldade em realizar o check-in (problemas com site, app, toten, filas etc.)

4.4. Documentação (inclusive negativa de embarque)
4.5. Outros

5. Execução do voo

5.1. Atraso/Alternativas (reacomodação, reembolso integral ou outra modalidade de transporte)
5.2. Atraso/Assistência material (comunicação, alimentação, hospedagem e traslado)
5.3. Atraso/Informação
5.4. Atraso/Outros
5.5. Cancelamento/Alternativas (reacomodação, reembolso integral ou outra modalidade de transporte)
5.6. Cancelamento/Assistência material (comunicação, alimentação, hospedagem e traslado)
5.7. Cancelamento/Informação
5.8. Cancelamento/Outros
5.9. Interrupção do serviço/Alternativas (reacomodação, reembolso integral ou outra modalidade de transporte)
5.10. Interrupção do serviço/Assistência material (comunicação, alimentação, hospedagem e traslado)
5.11. Interrupção do serviço/Informação
5.12. Interrupção do serviço/Outros
5.13. Perda de conexão/Alternativas (reacomodação, reembolso integral ou outra modalidade de transporte)
5.14. Perda de conexão/Assistência material (comunicação, alimentação, hospedagem e traslado)
5.15. Perda de conexão/Informação
5.16. Perda de conexão/Outros
5.17. Preterição/Alternativas (reacomodação, reembolso integral ou outra modalidade de transporte)
5.18. Preterição/Assistência material (comunicação, alimentação, hospedagem e traslado)
5.19. Preterição/Compensação
5.20. Preterição/Outros

6. Transporte de bagagem

6.1. Avaria
6.2. Extravio

6.3. Violação
6.4. Outros/Bagagem de mão
6.5. Outros/Bagagem despachada
6.6. Outros/Bagagem especial, animais, carga e itens proibidos

7. Reembolso
   7.1. Prazo
   7.2. Créditos
   7.3. Outros

8. Reclamações contra valores e regras do contrato
   8.1. Arrependimento (art. 49, CDC)
   8.2. Multas previstas no contrato
   8.3. Preços praticados
   8.4. Outros

9. Assistência ao PNAE
   9.1. Acompanhante
   9.2. Ajudas técnicas e equipamentos médicos
   9.3. Assistência durante a viagem
   9.4. Cão-guia
   9.5. Designação de assentos e mecanismos de contenção
   9.6. Procedimentos prévios à viagem
   9.7. Outros

10. Outros
    10.1. Assento conforto e outros itens opcionais
    10.2. Passageiro indisciplinado
    10.3. Outros

# ANEXO D – Distribuição das reclamações por temas e subtemas

Tabela 5: Distribuição das reclamações na base por temas e subtemas

| Tema | Subtema | #recl. |
|---|---|---|
| Alteração pela empresa aérea | | 4340 |
| | Alternativas (reacomodação, reembolso integral ou outra modalidade de transporte) | 3241 |
| | Assistência material (comunicação, alimentação, hospedagem e traslado) | 160 |
| | Aviso | 517 |
| | Outros3 | 422 |
| Alteração pelo passageiro | | 3992 |
| | Cancelamento automático do retorno (no-show na ida) | 314 |
| | Desistência 24/7 | 436 |
| | Multas indevidas e devolução da tarifa aeroportuária | 475 |
| | Outros2 | 153 |
| | Remarcação e resilição | 2614 |
| Assistência ao PNAE | | 386 |
| | Acompanhante | 82 |
| | Ajudas técnicas e equipamentos médicos | 44 |
| | Assistência durante a viagem | 67 |
| | Cão-guia | 1 |
| | Designação de assentos e mecanismos de contenção | 34 |
| | Outros7 | 32 |
| | Procedimentos prévios à viagem | 126 |
| Check-in e embarque | | 2311 |
| | Atraso do passageiro | 632 |
| | Correção de nome (inclusive negativa de embarque) | 622 |
| | Declaração especial de valor de bagagem | 2 |
| | Dificuldade em realizar o check-in (problemas com site, app, toten, filas etc.) | 291 |
| | Documentação (inclusive negativa de embarque) | 208 |
| | Outros4 | 556 |
| Execução do voo | | 5097 |
| | Atraso/Alternativas (reacomodação, reembolso integral ou outra modalidade de transporte) | 259 |
| | Atraso/Assistência material (comunicação, alimentação, hospedagem e traslado) | 623 |
| | Atraso/Informação | 196 |
| | Atraso/Outros | 551 |
| | Cancelamento/Alternativas (reacomodação, reembolso integral ou outra modalidade de transporte) | 623 |
| | Cancelamento/Assistência material (comunicação, alimentação, hospedagem e traslado) | 524 |
| | Cancelamento/Informação | 120 |
| | Cancelamento/Outros | 407 |
| | Interrupção do serviço/Alternativas (reacomodação, reembolso integral ou outra modalidade de transporte) | 120 |
| | Interrupção do serviço/Assistência material (comunicação, alimentação, hospedagem e traslado) | 89 |
| | Interrupção do serviço/Informação | 17 |
| | Interrupção do serviço/Outros | 86 |
| | Perda de conexão/Alternativas (reacomodação, reembolso integral ou outra modalidade de transporte) | 235 |
| | Perda de conexão/Assistência material (comunicação, alimentação, hospedagem e traslado) | 183 |
| | Perda de conexão/Informação | 32 |
| | Perda de conexão/Outros | 307 |
| | Preterição/Alternativas (reacomodação, reembolso integral ou outra modalidade de transporte) | 112 |
| | Preterição/Assistência material (comunicação, alimentação, hospedagem e traslado) | 43 |
| | Preterição/Compensação | 439 |
| | Preterição/Outros | 131 |
| Oferta e compra | | 4914 |
| | Cancelamento da passagem pela empresa aérea (inclusive por suspeita de fraude) | 621 |
| | Comprovante de passagem aérea | 404 |
| | Informação/Regras e valores de itens opcionais | 356 |
| | Informação/Regras e valores do transporte de bagagem | 548 |
| | Informação/Regras e valores do transporte de bagagem especial, animais, carga e itens proibidos | 151 |
| | Informação/Regras e valores do transporte de passageiros | 648 |
| | Outros1 | 2186 |
| Outros | | 2484 |
| | Assento conforto e outros itens opcionais | 595 |
| | Outros8 | 1877 |
| | Passageiro indisciplinado | 12 |
| Programas de Fidelidade | | 3521 |
| | Programas de Fidelidade | 3521 |
| Reclamações contra valores e regras do contrato | | 5165 |
| | Arrependimento (art. 49, CDC) | 630 |
| | Multas previstas no contrato | 3980 |
| | Outros6 | 151 |
| | Preços praticados | 404 |
| Reembolso | | 2414 |
| | Créditos | 307 |
| | Outros5 | 359 |
| | Prazo | 1748 |
| Transporte de bagagem | | 4452 |
| | Avaria | 1568 |
| | Extravio | 1570 |
| | Outros/Bagagem de mão | 311 |
| | Outros/Bagagem despachada | 459 |
| | Outros/Bagagem especial, animais, carga e itens proibidos | 167 |
| | Violação | 377 |
| TOTAL | | 39076 |

# ANEXO E – Exemplo numérico de rede com retropropagação

Extraído de https://mattmazur.com/2015/03/17/a-step-by-step-backpropagation-example/.

A estrutura da rede é a camada de entradas ($i_1$ e $i_2$), uma camada oculta de dois neurônios ($h_1$ e $h_2$), a camada de saídas ($o_1$ e $o_2$); oito pesos ($w_1$-$w_8$) e dois biases ($b_1$ e $b_2$):

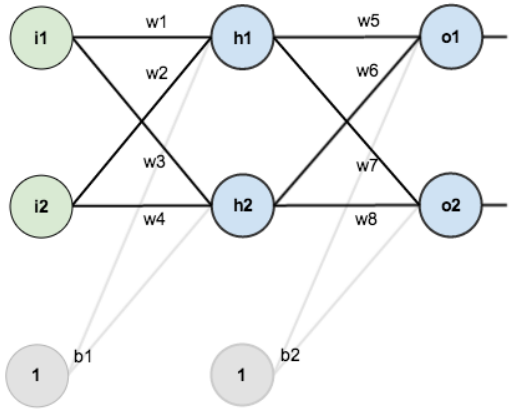


Figura 10: A estrutura da rede neural.

A rede é inicializada, sendo definidas as entradas, as saídas desejadas e os biases. Os pesos recebem valores aleatórios, com o intuito de ajustá-los para que as saídas estejam mais próximas o possível às desejadas:

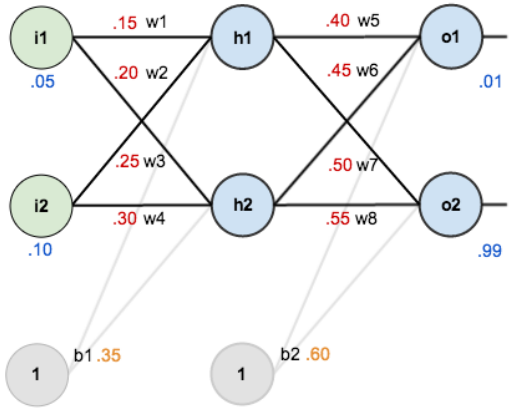


Figura 11: A rede neural inicializada.

A entrada atual ao neurônio $h_1$ é:

$$net_{h_1} = w_1 * i_1 + w_2 * i_2 + b_1 * 1 \tag{E.1}$$
$$net_{h_1} = 0,15 * 0,05 + 0,2 * 0,1 + 0,35 * 1$$
$$= 0,3775$$

Usando função logística para calcular a saída do $h_1$, obtemos:

$$out_{h_1} = \frac{1}{1 + e^{-net_{h_1}}} \tag{E.2}$$
$$= \frac{1}{1 + e^{-0,3775}} = 0,593269992$$

Da mesma maneira, a saída do $h_2$ é:

$$out_{h_2} = 0,596884378$$

Continuando o processo para os neurônios de saída, obtemos:

$$net_{o_1} = w_5 * out_{h_1} + w_6 * out_{h_2} + b_2 * 1 \tag{E.3}$$
$$net_{o_1} = 0.4 * 0,593269992$$
$$+ 0.45 * 0,596884378 + 0.6 * 1$$
$$= 1,105905967$$
$$out_{o_1} = \frac{1}{1 + e^{-net_{o_1}}} = \frac{1}{1 + e^{-1,105905967}}$$
$$= 0,75136507$$

e da mesma maneira para $o_2$:

$$out_{o_2} = 0,772928465$$

### Cálculo do erro total

Usando a função de custo ou erro quadrado, conforme na equação 2.5, temos:

$$E_{total} = \sum \frac{1}{2}(target - output)^2 \tag{E.4}$$

Calculamos o erro para $o_1$:

$$\begin{aligned} E_{o_1} &= \frac{1}{2}(target_{o_1} - out_{o_1})^2 \\ &= \frac{1}{2}(0,01 - 0,75136507)^2 \\ &= 0,274811083 \end{aligned}$$

E da mesma maneira obtemos para $o_2$:

$$E_{o_2} = 0,023560026$$

O erro total é a soma desses dois erros:

$$\begin{aligned} E_{total} &= E_{o_1} + E_{o_2} \\ &= 0,274811083 + 0,023560026 \\ &= 0,298371109 \end{aligned}$$

### Passo da retropropagação

Agora queremos atualizar os pesos para que as saídas produzidas pela rede estejam mais próximas às desejadas, minimizando o erro total. Primeiramente, queremos saber quanto a mudança nos pesos da camada de saídas influencia o erro total. Para $w_5$:

$$\frac{\partial E_{total}}{\partial w_5} = \frac{\partial E_{total}}{\partial out_{o_1}} * \frac{\partial out_{o_1}}{\partial net_{o_1}} * \frac{\partial net_{o_1}}{\partial w_5} \tag{E.5}$$

Visualmente podemos observar a mudança na Figura 12. Observamos cada um dos três elementos da equação E.5. Primeiro, quanto o erro total muda com respeito à saída?

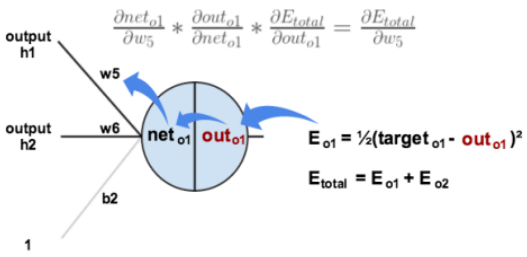


Figura 12: O primeiro passo da retropropagação.

$$\begin{aligned} E_{total} &= \frac{1}{2}(target_{o_1} - out_{o_1})^2 \\ &\quad + \frac{1}{2}(target_{o_2} - out_{o_2})^2 \end{aligned} \tag{E.6}$$

$$\frac{\partial E_{total}}{\partial out_{o_1}} = 2 * \frac{1}{2}(target_{o_1} - out_{o_1})^{2-1} * -1 + 0 \tag{E.7}$$

$$\begin{aligned} \frac{\partial E_{total}}{\partial out_{o_1}} &= -(target_{o_1} - out_{o_1}) \\ &= -(0,01 - 0,75136507) \\ &= 0,74136507 \end{aligned} \tag{E.8}$$

Depois, quanto a saída $o_1$ muda com respeito a sua entrada total?

$$\begin{aligned} out_{o_1} &= \frac{1}{1 + e^{-net_{o_1}}} \\ \frac{\partial out_{o_1}}{\partial net_{o_1}} &= out_{o_1}(1 - out_{o_1}) \\ &= 0,75136507(1 - 0,75136507) \\ &= 0,186815602 \end{aligned}$$

E, por fim, quanto a entrada total do $o_1$ muda com respeito a $w_5$?

$$\begin{aligned} net_{o_1} &= w_5 * out_{h_1} + w_6 * out_{h_2} + b_2 * 1 \\ \frac{\partial net_{o_1}}{\partial w_5} &= 1 * out_{h_1} * w_5^{(1-1)} + 0 + 0 \\ &= out_{h_1} = 0,593269992 \end{aligned}$$

Unindo tudo:

$$\frac{\partial E_{total}}{\partial w_5} = \frac{\partial E_{total}}{\partial out_{o_1}} * \frac{\partial out_{o_1}}{\partial net_{o_1}} * \frac{\partial net_{o_1}}{\partial w_5}$$
$$\frac{\partial E_{total}}{\partial w_5} = 0,74136507 * 0,186815602 * 0,593269992$$
$$= 0,082167041$$

Para diminuir o erro, deduzimos este valor do valor atual do $w_5$, utilizando uma taxa de aprendizagem ($\eta$, aqui de 0,5):

$$w_5^+ = w_5 - \eta * \frac{\partial E_{total}}{\partial w_5}$$
$$= 0,4 - 0,5 * 0,082167041$$
$$= 0,35891648$$

Repetimos esse processo para os demais pesos da camada de saídas, obtendo:

$$w_6^+ = 0,408666186$$
$$w_7^+ = 0,511301270$$
$$w_8^+ = 0,561370121$$

Estes valores atualizados dos pesos $w_5^+$-$w_8^+$ serão usados apenas na atualização das saídas da rede e ainda não no próximo passo da retropropagação que segu para a atualização dos pesos da camada anterior. Ou seja, ainda usamos valores originais $w_5$-$w_8$ continuando a retropropagação.

### Camada oculta

Agora calculamos novos valores para $w_1$-$w_4$. Precisamos saber a influência deles no erro total, observando aqui o $w_1$:

$$\frac{\partial E_{total}}{\partial w_1} = \frac{\partial E_{total}}{\partial out_{h_1}} * \frac{\partial out_{h_1}}{\partial net_{h_1}} * \frac{\partial net_{h_1}}{\partial w_1} \tag{E.9}$$

Visualmente:

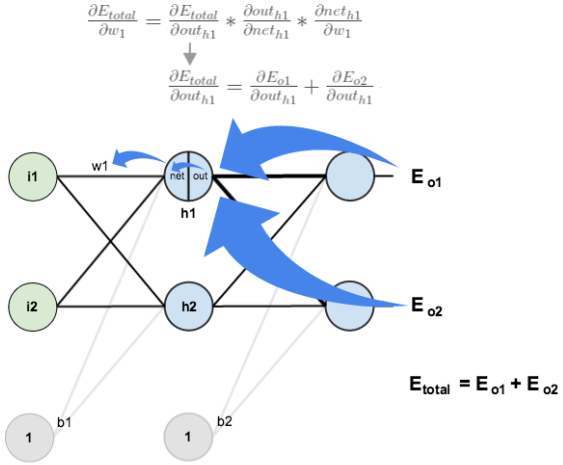


Figura 13: 2º passo da retropropagação.

O processo é similar ao anterior com a diferença que a saída de cada neurônio da camada oculta contribui com os erros de múltiplos neurônios de saída. Precisamos levar isso em consideração:

$$\frac{\partial E_{total}}{\partial out_{h_1}} = \frac{\partial E_{o_1}}{\partial out_{h_1}} + \frac{\partial E_{o_2}}{\partial out_{h_1}} \tag{E.10}$$

Começamos com $\frac{\partial E_{o_1}}{\partial out_{h_1}}$:

$$\frac{\partial E_{o1}}{\partial out_{h_1}} = \frac{\partial E_{o_1}}{\partial net_{o_1}} * \frac{\partial net_{o_1}}{\partial out_{h_1}} \tag{E.11}$$

Podemos calcular $\frac{\partial E_{o_1}}{\partial net_{o_1}}$ usando valores já calculados:

$$\frac{\partial E_{o_1}}{\partial net_{o_1}} = \frac{\partial E_{o_1}}{\partial out_{o_1}} * \frac{\partial out_{o_1}}{\partial net_{o_1}}$$
$$= 0,74136507 * 0,186815602$$
$$= 0,138498562$$

Por ourto lado, $\frac{\partial net_{o_1}}{\partial out_{h_1}}$ é igual a $w_5$:

$$net_{o_1} = w_5 * out_{h_1} + w_6 * out_{h_2} + b_2 * 1$$
$$\frac{\partial net_{o_1}}{\partial out_{h_1}} = w_5 = 0,40$$

Unindo:

$$\begin{aligned}\frac{\partial E_{o_1}}{\partial out_{h_1}} &= \frac{\partial E_{o_1}}{\partial net_{o_1}} * \frac{\partial net_{o_1}}{\partial out_{h_1}} \\ &= 0,138498562 * 0,40 \\ &= 0,055399425\end{aligned}$$

Seguindo o mesmo processo para $\frac{\partial E_{o_2}}{\partial out_{h_1}}$, obtemos:

$$\frac{\partial E_{o_2}}{\partial out_{h_1}} = -0,01949119$$

Portanto,

$$\begin{aligned}\frac{\partial E_{total}}{\partial out_{h_1}} &= \frac{\partial E_{o_1}}{\partial out_{h_1}} + \frac{\partial E_{o_2}}{\partial out_{h_1}} \\ &= 0,055399425 - 0,019049119 \\ &= 0,036350306\end{aligned}$$

Tendo $\frac{\partial E_{total}}{\partial out_{h_1}}$, precisamos calcular $\frac{\partial out_{h_1}}{\partial net_{h_1}}$ e $\frac{\partial net_{h_1}}{\partial w}$ para cada peso $w_1$ - $w_4$:

$$out_{h_1} = \frac{1}{1 + e^{-net_{h_1}}}$$
$$\begin{aligned}\frac{\partial out_{h_1}}{\partial net_{h_1}} &= out_{h_1}(1 - out_{h_1}) \\ &= 0,59326999(1 - 0,59326999) \\ &= 0,241300709\end{aligned}$$

Calculamos a diferencial parcial da entrada total ao $h_1$ com respeito a $w_1$ da mesma maneira que fizemos com $o_1$:

$$net_{h_1} = w_1 * i_1 + w_2 * i_2 + b_1 * 1$$
$$\frac{\partial net_{h_1}}{\partial w_1} = i_1 = 0,05$$

Unindo tudo:

$$\begin{aligned}\frac{\partial E_{total}}{\partial w_1} &= \frac{\partial E_{total}}{\partial out_{h_1}} * \frac{\partial out_{h_1}}{\partial net_{h_1}} * \frac{\partial net_{h_1}}{\partial w_1} \\ \frac{\partial E_{total}}{\partial w_1} &= 0.036350306 * 0.241300709 * 0.05 \\ &= 0.000438568\end{aligned}$$

Agora podemos atualizar $w_1$:

$$\begin{aligned}w_1^+ &= w_1 - \eta * \frac{\partial E_{total}}{\partial w_1} \\ &= 0,15 - 0,5 * 0,000438568 = 0,149780716\end{aligned}$$

E, da mesma maneira:

$$w_2^+ = 0,19956143$$
$$w_3^+ = 0,24975114$$
$$w_4^+ = 0,29950229$$

Com esse passo completo de retropropagação, o erro total diminuiu de 0,298371109 para 0,291027924. Se isso parece pouco, repentindo o mesmo processo por 10 mil vezes, o erro cai para 0,0000351085, e as saídas da rede ficam em 0,016 (a desejada era 0,01) e 0,984 (a desejada era 0,99).

# ANEXO F – Python Jupyter Notebook do processo de modelagem

# ULMFiT em português do zero

Este notebook é ajustado e traduzido do original no endereço https://github.com/fastai/course-nlp/blob/master/nn-vietnamese.ipynb (https://github.com/fastai/course-nlp/blob/master/nn-vietnamese.ipynb)

In [1]:
```
%reload_ext autoreload
%autoreload 2
%matplotlib inline

from fastai import *
from fastai.text import *
```

In [2]:
```
# bs=48
# bs=24
bs=128
```

In [3]:
```
torch.cuda.set_device(0)
```

In [4]:
```
data_path = Config.data_path()
```

Os próximos comandos criam pasta `ptwiki`, contendo um arquivo de texto `ptwiki` que contém os textos de Wikipédia. (Para outras linguagens, substitua `pt` com o código de list of wikipedias (https://meta.wikimedia.org/wiki/List_of_Wikipedias).)

In [5]:
```
lang = 'pt'
```

In [6]:
```
name = f'{lang}wiki'
path = data_path/name
path.mkdir(exist_ok=True, parents=True)
lm_fns = [f'{lang}_wt', f'{lang}_wt_vocab']
```

## Modelo de linguagem de português de wikipédia

Este passo é o mais demorado, mas precisa ser executado somente uma vez. Portanto, os comandos foram comentados para evitar execução acidental.

### Download de dados

In [7]:
```
#from nlputils import split_wiki,get_wiki
```

In [8]:
```
#get_wiki(path,lang)
```

In [10]:
```
#path.ls()
```

In [11]:
```
#!head -n4 {path}/{name}
```

A função seguinte divide o arquivo único de wikipédia em arquivos separados de artigos. O tratamento dos arquivos será mais fácil assim.

In [12]:
```
# dest = split_wiki(path,lang)
```

In [13]:
```
#dest.ls()[:5]
```

### Criar modelo pré-treinado

In [14]:
```
#data = (TextList.from_folder(dest)
#            .split_by_rand_pct(0.1, seed=42)
#            .label_for_lm()
#            .databunch(bs=bs, num_workers=1))

#data.save(f'{lang}_databunch')
#len(data.vocab.itos),len(data.train_ds)
```

In [15]:
```
#data.save(path, f'{lang}_databunch')
#len(data.vocab.itos),len(data.train_ds)
```

In [16]:
```
#print(data.path)
#print(path)
```

In [17]:
```
#data = load_data(path, f'{lang}_databunch', bs=bs)
```

In [18]:
```
# learn = language_model_learner(data, AWD_LSTM,
# drop_mult=0.5, pretrained=False).to_fp16()
```

In [19]:
```
#lr = 1e-2
#lr *= bs/48  # Scale learning rate by batch size
```

In [20]:
```
#learn.unfreeze()
#learn.fit_one_cycle(10, lr, moms=(0.8,0.7))
```

Salvar o modelo pré-treinado e o vocabulário:

In [21]:
```
#mdl_path = path/'models'
#mdl_path.mkdir(exist_ok=True)
#learn.to_fp32().save(mdl_path/lm_fns[0], with_opt=False)
#learn.data.vocab.save(mdl_path/(lm_fns[1] + '.pkl'))
```

# Classificação de documentos em português

## Modelo de linguagem

In [68]:
```
#path.ls()
```

In [23]:
```
train_df = pd.read_csv(path/'traintema.csv', sep=';')
train_df.loc[pd.isna(train_df.manifestacao_clean),'manifestacao_clean']='NA'
train_df.head()
```

Out[23]:

| | manifestacao_clean | TEMA |
|---|---|---|
| **0** | Olá em 26/06 comprei o localizador 8RK87X para... | Alteração pela empresa aérea |
| **1** | Prezados, Abri o processo junto a LATAM pra te... | Transporte de bagagem |
| **2** | No dia 17/12 comprei uma passagem, Vitória x C... | Alteração pelo passageiro |
| **3** | Estou tendo um problema! Recentemente fiz a co... | Programas de Fidelidade |
| **4** | Olá, Minha mala retornou da viagem de Recife c... | Transporte de bagagem |

In [24]:
```
df = train_df.copy()
```

In [25]:
```
data_lm = (TextList.from_df(df, path, cols='manifestacao_clean')
    .split_by_rand_pct(0.1, seed=42)
    .label_for_lm()
    .databunch(bs=bs, num_workers=1))
```

In [26]:
```
learn_lm = language_model_learner(data_lm, AWD_LSTM, pretrained_fnames=lm_fns,
                                  drop_mult=1.0)
```

In [27]:
```
lr = 1e-3
lr *= bs/48
```

In [29]:
```
learn_lm.fit_one_cycle(2, lr*10, moms=(0.8,0.7))
```

| epoch | train_loss | valid_loss | accuracy | time |
|---|---|---|---|---|
| 0 | 4.049025 | 3.715031 | 0.294295 | 03:21 |
| 1 | 3.859548 | 3.546185 | 0.309279 | 03:22 |

In [30]:
```
learn_lm.unfreeze()
learn_lm.fit_one_cycle(8, lr, moms=(0.8,0.7))
```

| epoch | train_loss | valid_loss | accuracy | time |
|---|---|---|---|---|
| 0 | 3.538121 | 3.324730 | 0.336987 | 03:55 |
| 1 | 3.403326 | 3.194802 | 0.354774 | 03:55 |
| 2 | 3.272518 | 3.116966 | 0.365177 | 03:55 |
| 3 | 3.201683 | 3.061056 | 0.372967 | 03:55 |
| 4 | 3.125808 | 3.016477 | 0.379454 | 03:55 |
| 5 | 3.084919 | 2.990308 | 0.383136 | 03:55 |
| 6 | 3.054610 | 2.976258 | 0.385096 | 03:55 |
| 7 | 3.004012 | 2.973606 | 0.385916 | 03:55 |

In [32]:
```
learn_lm.save(f'{lang}fine_tuned_final')
learn_lm.save_encoder(f'{lang}fine_tuned_enc_final')
```

## Classificador

In [33]:
```
data_clas = (TextList.from_df(train_df, path, vocab=data_lm.vocab,
                              cols='manifestacao_clean')
    .split_by_rand_pct(0.1, seed=42)
    .label_from_df(cols='TEMA')
    .databunch(bs=bs, num_workers=1))

data_clas.save(f'{lang}_textlist_class_final')
```

In [44]:
```
# data_clas = load_data(path, f'{lang}_textlist_class_final',
#                       bs=bs, num_workers=1) # 16GB não foi o suficiente
data_clas = load_data(path, f'{lang}_textlist_class_final',
                      bs=int(bs/2), num_workers=1)
```

In [35]:
```
from sklearn.metrics import f1_score

@np_func
def f1(inp,targ): return f1_score(targ, np.argmax(inp, axis=-1),
                                  average='weighted')
#def f1(inp,targ): return f1_score(targ, np.argmax(inp, axis=-1)
```

In [45]:
```
learn_c = text_classifier_learner(data_clas, AWD_LSTM, drop_mult=0.5,
                                  metrics=[accuracy,f1]).to_fp16()
learn_c.load_encoder(f'{lang}fine_tuned_enc')
learn_c.freeze()
```

In [46]:
```
lr=2e-2
lr *= bs/48
```

In [47]:
```
learn_c.fit_one_cycle(2, lr, moms=(0.8,0.7))
```

| epoch | train_loss | valid_loss | accuracy | f1 | time |
|---|---|---|---|---|---|
| 0 | 1.026177 | 0.940336 | 0.674942 | 0.664635 | 00:49 |
| 1 | 0.908288 | 0.852859 | 0.699514 | 0.690218 | 00:46 |

In [48]:
```
learn_c.fit_one_cycle(2, lr, moms=(0.8,0.7))
```

| epoch | train_loss | valid_loss | accuracy | f1 | time |
|---|---|---|---|---|---|
| 0 | 1.001468 | 0.961404 | 0.659585 | 0.656178 | 00:44 |
| 1 | 0.904280 | 0.842456 | 0.704121 | 0.695551 | 00:48 |

In [49]:
```
learn_c.freeze_to(-2)
learn_c.fit_one_cycle(2, slice(lr/(2.6**4),lr), moms=(0.8,0.7))
```

| epoch | train_loss | valid_loss | accuracy | f1 | time |
|---|---|---|---|---|---|
| 0 | 0.975003 | 0.921292 | 0.681341 | 0.671340 | 00:57 |
| 1 | 0.808008 | 0.779162 | 0.725877 | 0.719074 | 00:55 |

In [50]:
```
learn_c.freeze_to(-3)
learn_c.fit_one_cycle(2, slice(lr/2/(2.6**4),lr/2), moms=(0.8,0.7))
```

| epoch | train_loss | valid_loss | accuracy | f1 | time |
|---|---|---|---|---|---|
| 0 | 0.863447 | 0.891133 | 0.705145 | 0.688052 | 01:22 |
| 1 | 0.676233 | 0.748299 | 0.739442 | 0.733197 | 01:13 |

In [51]:
```
learn_c.unfreeze()
learn_c.fit_one_cycle(1, slice(lr/10/(2.6**4),lr/10), moms=(0.8,0.7))
```

| epoch | train_loss | valid_loss | accuracy | f1 | time |
|---|---|---|---|---|---|
| 0 | 0.618851 | 0.808885 | 0.743793 | 0.739252 | 01:34 |

In [52]:
```
learn_c.save(f'{lang}clas_final')
```

## Juntar os modelos forward e backward

O modelo backward foi treinado em outro notebook igual, somente ativando opção "backwards" no primeiro passo de treinamento do modelo de linguagem, e ajustando os nomes dos arquivos com sufixo "_bwd" para distinguí-los.

In [53]:
```
data_clas = load_data(path, f'{lang}_textlist_class_final', bs=bs, num_workers=1)
learn_c = text_classifier_learner(data_clas, AWD_LSTM, drop_mult=0.5,
                                  metrics=[accuracy,f1]).to_fp16()
learn_c.load(f'{lang}clas_final', purge=False);
```

In [54]:
```
preds,targs = learn_c.get_preds(ordered=True)
accuracy(preds,targs),f1(preds,targs)
```

Out[54]:
```
(tensor(0.7440), tensor(0.7404))
```

In [55]:
```
data_clas_bwd = load_data(path, f'{lang}_textlist_class_bwd', bs=bs, num_workers=1,
                          backwards=True)
learn_c_bwd = text_classifier_learner(data_clas_bwd, AWD_LSTM, drop_mult=0.5,
                                      metrics=[accuracy,f1]).to_fp16()
learn_c_bwd.load(f'{lang}clas_bwd', purge=False);
```

In [56]:
```
preds_b,targs_b = learn_c_bwd.get_preds(ordered=True)
accuracy(preds_b,targs_b),f1(preds_b,targs_b)
```

Out[56]:
```
(tensor(0.7382), tensor(0.7360))
```

In [57]:
```
preds_avg = (preds+preds_b)/2
```

In [58]:
```
accuracy(preds_avg,targs_b),f1(preds_avg,targs_b)
```

Out[58]:
```
(tensor(0.7504), tensor(0.7470))
```

In [62]:
```
top_k_accuracy(preds, targs, 2)
```

Out[62]:
```
tensor(0.9020)
```

In [63]:
```
top_k_accuracy(preds_b, targs_b, 2)
```

Out[63]:
```
tensor(0.9066)
```

In [64]:
```
top_k_accuracy(preds_avg, targs_b, 2)
```

Out[64]:
```
tensor(0.9130)
```

In [65]:
```
top_k_accuracy(preds, targs, 3)
```

Out[65]:
```
tensor(0.9555)
```

In [66]:
```
top_k_accuracy(preds_b, targs_b, 3)
```

Out[66]:
```
tensor(0.9526)
```

In [67]:
```
top_k_accuracy(preds_avg, targs_b, 3)
```

Out[67]:
```
tensor(0.9598)
```

In [ ]:
```

```