

UNIVERSIDADE FEDERAL DE GOIÁS  
INSTITUTO DE INFORMÁTICA

AYRTON DENNER DA SILVA AMARAL

**Predição do tempo de duração de  
processos e de movimentações  
processuais na esfera trabalhista**

Goiânia  
2019

**TERMO DE CIÊNCIA E DE AUTORIZAÇÃO PARA DISPONIBILIZAR VERSÕES ELETRÔNICAS  
DE TESES E  
DISSERTAÇÕES NA BIBLIOTECA DIGITAL DA UFG**

Na qualidade de titular dos direitos de autor, autorizo a Universidade Federal de Goiás (UFG) a disponibilizar, gratuitamente, por meio da Biblioteca Digital de Teses e Dissertações (BDTD/UFG), regulamentada pela Resolução CEPEC nº 832/2007, sem ressarcimento dos direitos autorais, de acordo com a Lei nº 9610/98, o documento conforme permissões assinaladas abaixo, para fins de leitura, impressão e/ou *download*, a título de divulgação da produção científica brasileira, a partir desta data.

**1. Identificação do material bibliográfico:**     **Dissertação**     **Tese**

**2. Identificação da Tese ou Dissertação:**

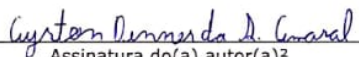
Nome completo do autor: *Cyrtan Demmer da Silva Amaral*

Título do trabalho: *Predição do tempo de duração de processos e de movimentações processuais na esfera trabalhista*


**3. Informações de acesso ao documento:**

Concorda com a liberação total do documento  SIM     NÃO<sup>1</sup>

Havendo concordância com a disponibilização eletrônica, torna-se imprescindível o envio do(s) arquivo(s) em formato digital PDF da tese ou dissertação.

  
Assinatura do(a) autor(a)<sup>2</sup>

Ciente e de acordo:

  
Assinatura do(a) orientador(a)<sup>2</sup>

<sup>1</sup> Neste caso o documento será embargado por até um ano a partir da data de defesa. A extensão deste prazo suscita justificativa junto à coordenação do curso. Os dados do documento não serão disponibilizados durante o período de embargo.

Casos de embargo:

- Solicitação de registro de patente
- Submissão de artigo em revista científica
- Publicação como capítulo de livro
- Publicação da dissertação/tese em livro

<sup>2</sup>A assinatura deve ser escaneada.

AYRTON DENNER DA SILVA AMARAL

# **Predição do tempo de duração de processos e de movimentações processuais na esfera trabalhista**

Dissertação apresentada ao Programa de Pós-Graduação do Instituto de Informática da Universidade Federal de Goiás, como requisito parcial para obtenção do título de Mestre em Ciência da Computação.

**Área de concentração:** Ciência da Computação.

**Orientador:** Prof. Dr. Anderson da Silva Soares

Goiânia  
2019

Ficha de identificação da obra elaborada pelo autor, através do Programa de Geração Automática do Sistema de Bibliotecas da UFG.

da Silva Amaral, Ayrton Denner

Predição do tempo de duração de processos e de movimentações processuais na esfera trabalhista [manuscrito] / Ayrton Denner da Silva Amaral. - 2019.  
LXVI, 66 f.

Orientador: Prof. Dr. Anderson da Silva Soares.

Dissertação (Mestrado) - Universidade Federal de Goiás, Instituto de Informática (INF), Programa de Pós-Graduação em Ciência da Computação, Goiânia, 2019.

Bibliografia.

Inclui siglas, lista de figuras, lista de tabelas.

1. Aprendizagem Profunda. 2. Redes Neurais Profundas. 3. Esfera Trabalhista. 4. Direito do Trabalho. 5. Processamento de Linguagem Natural. I. da Silva Soares, Anderson, orient. II. Título.

CDU 004



UNIVERSIDADE FEDERAL DE GOIÁS  
INSTITUTO DE INFORMÁTICA  
**ATA DE DEFESA DE DISSERTAÇÃO**

Ata nº **17/2019** da sessão de Defesa de Dissertação de **Ayrton Denner da Silva Amaral**, que confere o título de Mestre em **Ciência da Computação**, na área de concentração em Ciência da Computação.

Aos onze dias do mês de dezembro de dois mil e dezenove, a partir das quinze horas e trinta minutos, na sala 150 do Instituto de Informática, realizou-se a sessão pública de Defesa de Dissertação intitulada **“Predição do tempo de duração de processos e de movimentações processuais na esfera trabalhista”**. Os trabalhos foram instalados pelo Orientador, Professor Doutor Anderson da Silva Soares (INF/UFG) com a participação dos demais membros da Banca Examinadora: Professor Doutor Thyago Carvalho Marques (EMC/UFG), membro titular externo; e Professora Doutora Nádia Félix Felipe da Silva (INF/UFG), membra titular interna. Durante a arguição os membros da banca não fizeram sugestão de alteração do título do trabalho. A Banca Examinadora reuniu-se em sessão secreta a fim de concluir o julgamento da Dissertação, tendo sido o candidato **aprovado** pelos seus membros. Proclamados os resultados pelo Professor Doutor Anderson da Silva Soares, Presidente da Banca Examinadora, foram encerrados os trabalhos e, para constar, lavrou-se a presente ata que é assinada pelos Membros da Banca Examinadora, aos onze dias do mês de dezembro de dois mil e dezenove.

TÍTULO SUGERIDO PELA BANCA



---

Documento assinado eletronicamente por **Nadia Felix Felipe Da Silva, Professor do Magistério Superior**, em 11/12/2019, às 17:57, conforme horário oficial de Brasília, com fundamento no art. 6º, § 1º, do [Decreto nº 8.539, de 8 de outubro de 2015](#).



---

Documento assinado eletronicamente por **Anderson Da Silva Soares, Professor do Magistério Superior**, em 11/12/2019, às 17:58, conforme horário oficial de Brasília, com fundamento no art. 6º, § 1º, do [Decreto nº 8.539, de 8 de outubro de 2015](#).

---



Documento assinado eletronicamente por **Thyago Carvalho**

**Marques, Professor do Magistério Superior**, em 11/12/2019, às 18:02, conforme horário oficial de Brasília, com fundamento no art. 6º, § 1º, do [Decreto nº 8.539, de 8 de outubro de 2015](#).

---



Documento assinado eletronicamente por **Antonio Melo De Oliveira, Professor do Magistério Superior**, em 08/01/2020, às 07:38, conforme horário oficial de Brasília, com fundamento no art. 6º, § 1º, do [Decreto nº 8.539, de 8 de outubro de 2015](#).

---



A autenticidade deste documento pode ser conferida no site

<https://sei.ufg.br>

[/sei/controlador\\_externo.php?acao=documento\\_conferir&](#)

[id\\_orgao\\_acesso\\_externo=0](#), informando o código verificador **1035673**

e o código CRC **336D66CD**.

---

**Referência:** Processo nº 23070.044695/2019-21

SEI nº 1035673

Todos os direitos reservados. É proibida a reprodução total ou parcial do trabalho sem autorização da universidade, do autor e do orientador(a).

**Ayrton Denner da Silva Amaral**

Bacharel em Engenharia de Software pela Universidade Federal de Goiás. Experiência profissional em desenvolvimento web full-stack, desenvolvimento para dispositivos móveis, e uso de inteligência artificial e redes neurais para trabalhos de regressão e classificação, especialmente na área de processos trabalhistas na esfera jurídica brasileira.

Dedico esse trabalho aos meus pais, por terem confiado no meu sucesso durante todos esses anos. Dedico esse trabalho ao meu orientador, professor Anderson Soares, pelo voto de confiança e empenho em auxiliar na construção desse trabalho. Dedico esse trabalho a todos que me apoiaram e me incentivaram em cada nova etapa da minha caminhada. Esse trabalho certamente não seria o mesmo sem a existência de todos vocês.



---

## Agradecimentos

---

Agradeço primeiramente aos meus pais, que tanto me deram apoio nas decisões que eu tomei ao longo de minha vida, sendo a pós-graduação uma delas. Agradeço ao meu orientador Anderson Soares, por ter ajudado no conduzir desse projeto, tutorando não só academicamente mas também profissionalmente, ajudando a moldar esse trabalho do zero até a forma atual. Agradeço ao Instituto de Informática da Universidade Federal de Goiás por dar apoio a nível de infra-estrutura e ensino, possibilitando não só a existência desse projeto, bem como provendo toda a base necessária desde a graduação. Agradeço a Coordenação de Aperfeiçoamento de Pessoal de Nível Superior (CAPES) pela bolsa de auxílio ao mestrado. Agradeço aos colegas do grupo de estudos em Deep Learning e também do grupo de Processamento de Linguagem Natural pelas contribuições e ideias que ajudaram a pautar o desenvolvimento deste projeto, sem os quais vários aspectos e detalhes deste trabalho certamente seriam ignorados por descuido. Agradeço a equipe da Aviso Urgente pela disponibilidade dos dados fornecidos, e também a equipe Data Lawyer pelo fornecimento da infra-estrutura e equipamentos utilizados, assim como a oportunidade de aplicar os conhecimentos adquiridos em problemas atuais da realidade. Agradeço também a tantos outros nomes que foram importantes nessa trajetória de vida.

Meu muito obrigado a todos vocês.

"Skill to do comes of doing; knowledge comes by eyes always open, and working hands; and **there is no knowledge that is not power.**"

**Ralph Waldo Emerson,**  
*Society and Solitude, 1892, p. 303.*

---

## Resumo

---

Amaral, Ayrton Denner da Silva. **Predição do tempo de duração de processos e de movimentações processuais na esfera trabalhista**. Goiânia, 2019. 74p. Dissertação de Mestrado. Instituto de Informática, Universidade Federal de Goiás.

A predição de movimentação de processos judiciais é um problema relevante no contexto jurídico. A previsibilidade é uma variável importante no dimensionamento de trabalho em escritórios advocatícios. Neste trabalho é proposto uma arquitetura de rede neural artificial para predição de movimentações de processos trabalhistas na esfera jurídica brasileira. Apesar dos avanços recentes no uso de técnicas de aprendizagem de máquina e processamento de linguagem natural, o problema no contexto jurídico possui características próprias por contextos geográficos e linguísticos. Como estudo de caso, foi utilizado um banco de dados de processos do ano de 2015 de uma mesma vara da esfera trabalhista, em razão do volume de dados disponível. Apesar das dificuldades inerentes à esfera trabalhista, foi possível atingir resultados com relativo sucesso de acurácia, possibilitando o treino de um modelo capaz de modelar parcialmente o *corpus* jurídico trabalhista, bem como abstrair as relações entre os elementos que compõem os processos.

### Palavras-chave

Aprendizagem Profunda, Redes Neurais Profundas, Esfera Trabalhista, Processamento de Linguagem Natural.

---

## **Abstract**

---

Amaral, Ayrton Denner da Silva. **Prediction of duration time of proceedings and of proceedings movements in labor court.** Goiânia, 2019. 74p. MSc. Dissertation. Instituto de Informática, Universidade Federal de Goiás.

The prediction of legal proceeding movements is a relevant problem in the juridical context. Predictability is an important variable in sizing the work in attorneys offices. This works proposes an artificial neural network architecture to predict proceedings movements in Brazilian labor court. Despite the recent advances in the use of machine learning techniques and natural language processing, the problem in juridical context has its own characteristics by geographic and linguistic contexts. As a case study, a proceedings database of the year 2015 and from the same district from the labor sphere was used, due to the volume of data available. Despite the difficulties inherent in the labor sphere, it was possible to achieve results with relative accuracy success, enabling the training of a model capable of partially modeling the labor legal corpus, as well as abstracting the relationships between the elements that make up the processes.

### **Keywords**

Deep Learning, Deep Neural Networks, Labor Courts, Natural Language Processing

---

## **Lista de Abreviaturas e Siglas**

---

PLN - Processamento de Linguagem Natural

PJe - Processo Jurídico Eletrônico

PJ - Poder Judiciário

JT - Justiça do Trabalho \ Justiça Trabalhista

CNJ - Conselho Nacional de Justiça

TRT - Tribunal Regional do Trabalho

TST - Tribunal Superior do Trabalho

LSTM - Long Short-Term Memory

COR - Característica de Operação do Receptor

ASC - Área Sob a Curva

TVP - Taxa de Verdadeiro Positivo

TFP - Taxa de Falso Positivo

---

# Sumário

---

Lista de Figuras	15
Lista de Tabelas	16
1 Introdução	17
2 Definição do Problema	20
2.1 Direito do Trabalho	20
2.2 Processo e ação trabalhista	21
2.3 Atos e prazos processuais	22
2.4 Justiça do Trabalho	23
2.5 Processo Judicial eletrônico (PJe)	28
2.6 Dados de um processo trabalhista	29
3 Trabalhos relacionados	31
4 Métodos e Ferramentas	33
4.1 Limpeza e normalização do corpus	33
4.2 Representação de palavras via tokens e <i>embeddings</i>	34
4.3 Representação de partes via tokens	34
4.4 Normalização do valor de causa	36
4.5 Arquitetura da rede neural	36
4.5.1 <i>Long short-term memory</i>	36
4.5.2 Função de custo	37
4.5.3 Mecanismo de atenção	38
4.5.4 Codificação posicional	39
4.5.5 Gradiente	41
4.5.6 <i>Batch</i>	41
4.5.7 Diagrama da arquitetura	42
4.6 Equipamentos	45
5 Resultados	46
5.1 Métricas utilizadas	46
5.1.1 Acurácia	46
5.1.2 $F_1$ score	46
5.1.3 Área sob a curva	47
5.2 Resultados obtidos	50
5.2.1 Resultado das matrizes de confusão	52
5.2.2 Resultado da área sob a curva	55

5.2.3	Função de custo	57
5.2.4	Versão reduzida do modelo	60
<b>6</b>	<b>Conclusão</b>	<b>63</b>
6.1	Conclusões gerais	63
6.2	Trabalhos futuros	64
	Referências Bibliográficas	<b>67</b>

---

## Lista de Figuras

---

2.1	Tempo médio dos processos em cada grau de jurisdição na Justiça do Trabalho	25
4.1	Ilustração da matriz de codificação posicional obtida	40
4.2	Ilustração do modelo proposto	44
5.1	Exemplo de área sob a curva	49
5.2	Matriz de confusão da classe 1-7 dias	52
5.3	Matriz de confusão da classe 8-15 dias	53
5.4	Matriz de confusão da classe 16-30 dias	53
5.5	Matriz de confusão da classe 31-60 dias	54
5.6	Matriz de confusão da classe mais de 60 dias	54
5.7	Área sob a curva da classe 1-7	55
5.8	Área sob a curva da classe 8-15	55
5.9	Área sob a curva da classe 16-30	56
5.10	Área sob a curva da classe 31-60	56
5.11	Área sob a curva da classe mais de 60 dias	57
5.12	Função de custo da classe 1-7	57
5.13	Função de custo da classe 8-15	58
5.14	Função de custo da classe 16-30	58
5.15	Função de custo da classe 31-60	59
5.16	Função de custo da classe 60+	59
5.17	Ilustração da versão enxuta do modelo proposto	60
5.18	Matriz de confusão da classe 16-30 dias	61
5.19	Área sob a curva da classe 16-30 dias	62
5.20	Função de custo da classe 16-30 dias	62



---

## **Lista de Tabelas**

---

2.1	Tabela do tempo de tramitação médio de um processo em 2017	26
5.1	Resultados obtidos pelo treino	51
5.2	Resultados obtidos pelo treino da classe 16-30 dias	61

---

## Introdução

---

Técnicas de classificação e regressão com análise textual são uma parte da área de processamento de linguagem natural (ou *PLN*) utilizada em diversos trabalhos com relativo sucesso. Nesse contexto seria possível, por exemplo, analisar tarefas como estimativa de notas [61] e receita [13] de filmes, previsão de popularidade de petições on-line [80], automação de pontuação de redações [83] [8], predição de notas dadas pela mídia especializada em video-games [11], análise dimensional de sentimentos em textos [86], análise de sentimentos em conjunto de dados com textos, áudios e vídeos [68], classificação de processos quanto à decisão final e área legal do processo [81], dentre outros.

Nesse trabalho é proposto a construção de um modelo de rede neural para trabalhar com a predição do tempo de duração de processos e movimentações processuais da esfera trabalhista brasileira. A escolha do uso de redes neurais artificiais como ferramenta para esse trabalho deve-se aos recentes avanços em *benchmarks* alcançados quando comparado com abordagens utilizando algoritmos de *machine learning* clássico. Como exemplo é possível citar o problema do *benchmark* MNIST [50], um conjunto de dados clássico de 70 mil números escritos a mão, o qual apresenta em [17] uma taxa de erro de apenas 0.23%, taxa próxima de 0.2% geralmente obtida por humanos executando a mesma tarefa.

No contexto específico de processamento de textos, é possível citar alguns conjuntos de dados específicos da área textual. O *IMDB Review Dataset* [54], contendo 50 mil análises de filmes rotuladas como positivas ou negativas, apresenta resultados como [41] (95.4%), [33] (94.99%), [42] (94.1%), [59] (94.1%) e [56] (91.8%), todos trabalhos com redes neurais que atingiram mais de 91% de acurácia. Já o *DBpedia* [51] é um conjunto de dados composto de 630 mil artigos separados em 14 classes, onde resultados como [41], [42], [43], [18] e [87] atingiram menos de 1.6% de erro. Tais resultados servem como argumentos a favor da decisão de se utilizar modelos de redes neurais para trabalhos de aprendizado de máquina e processamento de linguagem natural, ao invés de algoritmos clássicos.

Como forma de oferecer maior suporte argumentativo para a decisão do uso

de redes neurais profundas em detrimento de técnicas de *machine learning* clássico, [78] apresenta resultados com diferentes abordagens de *machine learning* e mecanismos de redes neurais treinados no *Quora Question Pairs*, um conjunto de dados contendo 2.750.085 pares de questões, com o intuito de encontrar questões duplicadas. No trabalho em questão, os autores utilizaram algoritmos de *machine learning* clássico e também de redes neurais para comparar o sucesso dos modelos de diferentes categorias. Na classe de modelos considerados lineares, os autores obtiveram 80.8% de acurácia utilizando regressão linear e 80.9% de acurácia utilizando SVM; na classe de modelos utilizando algoritmos baseados em árvores de decisão, os autores obtiveram 73.2% de acurácia utilizando árvores de decisão e 75.7% de acurácia utilizando *random forest*; e na classe de redes neurais profundas, os autores obtiveram 81.4% de acurácia com células *Long short-term memory* (ou *LSTM*), 81.8% de acurácia utilizando células *LSTM* junto com mecanismo de atenção, 82.1% de acurácia com células *LSTM* com entrada bidirecional, e 82.3% de acurácia utilizando células *LSTM* com entrada bidirecional e adição de mecanismo de atenção. Tais resultados demonstram não somente a evolução de resultados obtidos com redes neurais em relação a algoritmos clássicos de *machine learning*, mas também a melhoria em resultados obtidos com outros mecanismos que agreguem com redes neurais profundas, como o mecanismo de atenção no exemplo citado.

Dado a existência de poucos trabalhos envolvendo uso de técnicas de *machine learning* para textos jurídicos da área brasileira, faz-se necessário maiores investigações envolvendo textos da área jurídica com redes neurais, especialmente trabalhos envolvendo redes recorrentes e processamento de linguagem natural com dados da esfera trabalhista brasileira. Além disso, o uso de estruturas como mecanismo de atenção e codificação posicional ainda são relativamente pouco explorados, sobretudo no estudo de textos da língua portuguesa e de trabalhos em português da área jurídica, tornando necessária uma investigação do uso de tais técnicas nesse tipo de *corpus*.

Um modelo de predição temporal em processos trabalhistas é de grande valor para advogados e outros profissionais da área jurídica trabalhista, onde tal modelo poderá ajudar os respectivos profissionais a melhor se prepararem na organização e priorização de tarefas, haja visto que terão uma aproximação da duração do processo trabalhista como um todo. Os profissionais também terão um indicativo melhor para uma análise da situação atual de um processo em andamento, já que terão uma previsão do momento onde irá ocorrer a próxima movimentação de um processo trabalhista, dado o seu histórico das movimentações que já ocorreram anteriormente. Com isso, um advogado poderia, por exemplo, executar um diagnóstico de um processo trabalhista recente para avaliar se tal processo será relativamente rápido ou moroso no seu decorrer, bem como também analisar se, dado o decorrer de um processo, o mesmo está próximo de ser finalizado ou ainda irá se arrastar por um tempo considerável dentro da área trabalhista.

---

Esse trabalho apresenta como hipótese a possibilidade de modelar uma predição de acurácia aceitável acerca do tempo de duração dos processos e suas movimentações na esfera trabalhista, dado o uso de dados de processos trabalhistas e suas movimentações, assim como outros elementos disponíveis. Com o intuito de validar tal hipótese, bem como em gerar maiores investigações e estudos envolvendo redes neurais artificiais e processos trabalhistas da língua portuguesa, foram reunidas neste trabalho 114.988 movimentações oriundas de 2.554 processos da Justiça do Trabalho, especificamente do ano de 2015, referentes ao TRT 15 - situado em São Paulo - e todos da mesma vara trabalhista. Após o treino dos dados coletados em um modelo de rede neural, são apresentados os resultados obtidos durante o treino, junto com valores para nível de comparação. Também são descritos os dados coletados da Justiça Trabalhista, bem como o processo de treino envolvido. Por fim, são feitas algumas comparações de valores médios das durações em dias dos processos e suas movimentações.

No capítulo 2 é apresentado uma contextualização do Direito do Trabalho e da Justiça Trabalhista, bem como outros conceitos como processos, ações, atos, prazos, dentre outros. O capítulo 3 lista trabalhos relacionados que envolvem o uso de modelos preditivos na área jurídica. O capítulo 4 apresenta uma lista das ferramentas utilizadas e os impactos gerados no resultado atingido. O capítulo 5 contém uma discussão acerca do resultado final obtido, sob o prisma de diferentes métricas de avaliação. Por fim, no capítulo 6 estão listadas as conclusões obtidas da execução do trabalho, além de possibilidades de trabalhos futuros para evolução do estado atual do modelo.

## Definição do Problema

---

Este capítulo apresenta uma discussão acerca dos problemas envolvendo a esfera trabalhista no âmbito jurídico brasileiro. A seção 2.1 expõe conceitos teóricos do Direito do Trabalho, em relação às suas atividades e visões de diferentes autores. A seção 2.2 determina as diferenças entre processos e ações trabalhistas, bem como suas condições de existência e princípios, dentre outros. Na seção 2.3 é tratado sobre as classificações e sub-classificações de atos e prazos processuais, bem como seus cálculos de dias. A seção 2.4 discorre sobre a situação atual dos processos trabalhistas, o tempo despendido na dinâmica dos tribunais, bem como as oportunidades na área que este trabalho visa direcionar. A seção 2.5 visa dar uma breve análise do Processo Judicial Eletrônico, sendo a fonte mais importante de dados dentro da atual realidade. A seção 2.6 visa discorrer sobre como os dados dos processos trabalhistas e suas movimentações são disponibilizados atualmente pelos tribunais, comarcas e outros agentes envolvidos.

### 2.1 Direito do Trabalho

O Direito do Trabalho (também denominado como Direito Social) é um ramo do direito destinado a regular as relações entre empregadores, empregados e suas respectivas entidades representativas; estabelecer órgãos de previdência e de seguros sociais; gerir os canais necessários para resolver os conflitos entre os indivíduos em ambiente trabalhista e entre as categorias profissionais, tendo como foco principal o bem coletivo [72]. Não obstante, ainda que o Direito do Trabalho busque estabelecer a equidade entre patrões e empregados, tal ramo do Direito também se estabelece como uma proteção ao trabalhador, vendo neste o elo mais fraco das relações trabalhistas [76].

O Direito do Trabalho é visto por alguns autores como uma área de natureza mista, abarcando relações e entendimentos tanto do Direito Privado quanto do Direito Público, numa situação *sui generis* [28], enquanto outros autores entendem o Direito do Trabalho como uma área unicamente do Direito Público [72]. O Direito do Trabalho é composto de normas imperativas e nacionais [76], ou seja, normas cogentes, as quais pressupõem um caráter de cumprimento obrigatório, e que incidem em todo o território

nacional, sendo de competência da União. Em suma, o Estado atua exercendo seu poder soberano, impondo medidas que visam solucionar as contendas entre os envolvidos, em um vínculo claro de subordinação e submissão.

## 2.2 Processo e ação trabalhista

O termo *ação trabalhista* refere-se ao ato de um indivíduo contatar o Poder Judiciário com o intuito de receber uma resposta do mesmo, enquanto o termo *processo trabalhista* refere-se ao sistema de atos envolvidos no decorrer da ação, do seu início até a resposta do Poder Judiciário [77] [10].

Uma ação trabalhista só pode existir dada a satisfação de 3 condições fundamentais:

- Interesse em agir: representa a existência de utilidade, necessidade e adequação em se provocar a jurisdição;
- Legitimidade "*ad causam*": representa a pertinência lógica entre o que está sendo pedido, quem pede e em relação ao que se pede;
- Possibilidade jurídica do pedido: representa a condição do pedido em não estar defeso em lei e nem ofender a moral ou os bons costumes.

Dada uma ação trabalhista, o pedido formulado pelo autor da mesma será julgada procedente ou improcedente por um magistrado da Justiça do Trabalho.

Uma ação trabalhista é composta pelos seguintes elementos:

- Partes: representa os sujeitos do processo, podendo compor como reclamantes (polo ativo), reclamados (polo passivo) ou terceiros (como testemunhas, por exemplo);
- Pedido: representa a providência jurisdicional desejada pela ação, de acordo com o contexto da petição inicial;
- Causa de pedir: sub-dividida em fatos (ou exposição dos fatos) e fundamentos jurídicos para a ação trabalhista.

Um processo trabalhista representa o conjunto de regras e procedimentos em relação ao trâmite e resolução das contendas entre indivíduos e/ou coletivos em relação ao contexto trabalhista. Além disso, o processo do trabalho observa princípios como:

- Acesso facilitado ao judiciário, incluindo gratuidade de justiça e custas processuais pagas pelo vencido após o trânsito em julgado da decisão
- Busca pela conciliação, permitindo acordo entre as partes para findar o processo, além de proferimento da decisão apenas após rejeição de proposta de conciliação entre as partes

- Celeridade, velando pelo rápido andamento das causas, permitindo reclamação escrita ou verbal e acumulação de várias reclamações no mesmo processo quando se tratarem de funcionários do mesmo estabelecimento
- Devido processo legal, garantindo sentença proferida apenas por autoridade competente, imparcialidade do juiz envolvido, direito ao contraditório e ampla-defesa, bem como proibição de proferimento de sentença diversa das questões suscitadas
- Busca da verdade real, dando liberdade ao juiz para determinar as provas a serem produzidas

## 2.3 Atos e prazos processuais

Os atos processuais podem ser classificados como atos das partes, atos do juiz e atos dos servidores e auxiliares da justiça [10]. Os atos do juiz podem ser sub-classificados em:

- Sentença: ato processual onde o juiz trata de alguma das situações previstas no Código de Processo Civil
- Decisão interlocutória: ato processual onde o juiz resolve questão incidente do processo trabalhista
- Despacho: demais atos do juiz cujo a lei não estabelece outra forma

Tais atos processuais devem ser realizados em dias úteis, no horário entre 06:00 horas e 20:00 horas. Os atos iniciados durante o horário normal poderão ser concluídos após as 20:00 horas quando o adiamento da conclusão do ato prejudicar a execução do processo ou causar dano grave. Poderão ainda haver casos excepcionais onde a notificação inicial e a penhora poderão ser realizadas em um sábado, domingo ou feriado.

Os prazos processuais poderão ser classificados segundo destinatários do prazo, origem do prazo e natureza do prazo. Os prazos também podem ser sub-divididos em:

- Quanto a destinatários do prazo:
  - Prazos próprios: prazos destinados as partes e sempre sujeitos à preclusão
  - Prazos impróprios: prazos destinados aos juízes ou servidores, e que não são sujeitos à preclusão trabalhista
- Quanto a origem do prazo:
  - Prazos legais: prazos estabelecidos pela lei
  - Prazos judiciais: prazos estabelecidos pelo magistrado
  - Prazos convencionais: prazos estabelecidos pelas partes
- Quanto a natureza do prazo:

- Prazos dilatatórios: prazos passíveis de alteração pela vontade das partes
- Prazos peremptórios: prazos impassíveis de alteração

Os prazos estão dispostos da seguinte maneira:

- Atos processuais: devem ser realizados nos prazos prescritos em lei, ou de acordo com determinação do juiz no caso de omissão da lei, ou ainda o prazo será de **cinco dias** para prática de ato processual quando não houver também a assinatura do juiz.
- Intimações: devem obrigar comparecimento após decorridas **vinte e quatro horas** quando não houver outra marcação por parte da lei
- Suspensão do processo: os processos podem ser suspensos em até **seis meses**, dada convenção das partes.
- Despachos: devem ser proferidos pelo juiz no prazo de **dois dias**
- Decisões: devem ser proferidos pelo juiz no prazo de **dez dias**

Havendo motivo justificado, o juiz poderá exceder por tempo igual os prazos previamente estabelecidos pela legislação.

Em relação aos prazos processuais trabalhistas, os mesmos são contados a partir da data em que foram feitos pessoalmente, ou a partir da data em que fora recebida a notificação do processo. No caso de processos eletrônicos, a data da publicação é considerada sendo o primeiro dia útil seguinte ao da disponibilização do processo eletrônico, enquanto a data de início da contagem do prazo será sempre o primeiro dia útil subsequente a intimação. Já no caso do vencimento do prazo, o mesmo será prorrogado no caso de vencimento nos dias de sábado, domingo, feriado, caso de vencimento em dia em que foi determinado fechamento do fórum ou expediente encerrado antes do horário normal, sendo sempre prorrogados para o primeiro dia útil seguinte.

Por fim, será computado em quádruplo o prazo para contestar e em dobro o prazo para recorrer quando a parte em questão for a Fazenda Pública ou o Ministério Público.

## 2.4 Justiça do Trabalho

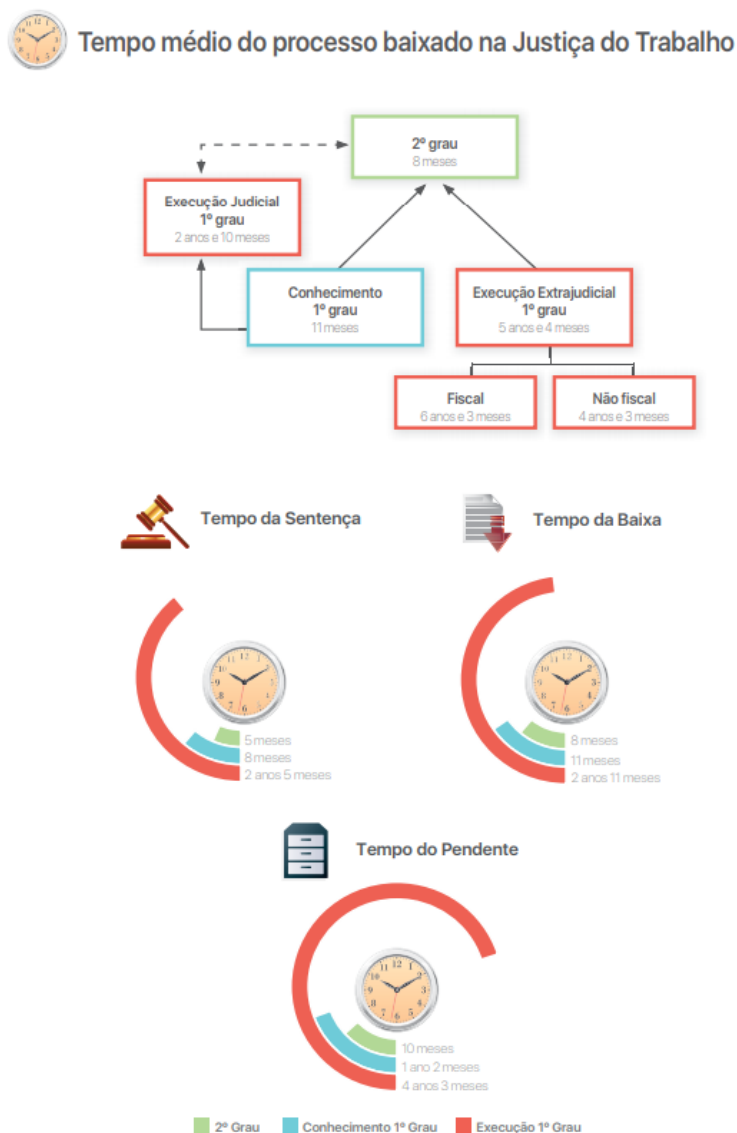
De acordo com o relatório Justiça em Números, produzido pelo CNJ no ano de 2018 [2], o Brasil atualmente conta com 24 TRTs (também denominados como "regiões"), tendo o TST como órgão máximo de tal ramo da justiça. A Justiça do Trabalho têm como objetivo conciliar e julgar as ações judiciais entre empregados e empregadores, bem como outras controvérsias decorrente de tais relações de trabalho, além das demandas que tenham origem no cumprimento de suas próprias sentenças. Cada região é sub-dividida em dois graus de jurisdição: 1º grau sendo sendo composto pelas varas de trabalho, onde atuam os juízes do trabalho; 2º grau sendo compostos pelos TRTs em si, julgando



recursos ordinários contra as decisões da respectiva vara do trabalho, dissídios coletivos ou outros desencontros que possam ocorrer na justiça trabalhista e que não tenham sido previamente resolvidos no 1º grau de jurisdição. Já o TST é a entidade máxima da Justiça do Trabalho, composto por 27 ministros, responsáveis por uniformizar as decisões sobre ações trabalhistas, consolidando a nível nacional a jurisprudência na esfera trabalhista.

Atualmente, a Justiça do Trabalho apresenta 1.572 das 15.398 unidades judiciárias do primeiro grau (ou 10.2% do total). A Justiça do Trabalho apresentou uma despesa total de R\$ 18.283.148.816 reais no ano de 2017, sendo responsável por aproximadamente 20.1% das despesas totais do Poder Judiciário em 2017. Como força de trabalho responsável, a Justiça do Trabalho atualmente emprega 3.658 magistrados (sendo 3.102 destes apenas na jurisdição de 1º grau), além de 39.585 servidores e 12.343 auxiliares.

Para o ano de 2017, as estimativas de prazos como tempo médio para cada grau de jurisdição do CNJ estão exemplificadas na Figura 2.1:



**Figura 2.1:** Tempo médio dos processos em cada grau de jurisdição na Justiça do Trabalho

Em relação ao tempo de tramitação dos processos, a tabela 2.1 apresenta os dados do CNJ com o cálculo da média do tempo em cada fase do processo considerando cada instância do Poder Judiciário:

	Sentença	Baixa	Pendente	Categoria
Tribunal Superior do Trabalho	1 ano e 2 meses	1 ano e 8 meses	2 anos e 6 meses	Tribunal superior
Tribunal Regional do Trabalho	5 meses	8 meses	10 meses	2º Grau
Varas do Trabalho (execução)	2 anos e 5 meses	2 anos e 11 meses	4 anos e 3 meses	1º Grau
Varas do Trabalho (conhecimento)	8 meses	11 meses	1 ano e 2 meses	1º Grau

**Tabela 2.1:** *Tabela do tempo de tramitação médio de um processo em 2017*

Além da tabela 2.1, o CNJ também calcula um tempo médio de duração dos processos nas seguintes categorias:

- Acervo de processos: 2 anos e 4 meses
- Processos baixados: 1 ano e 3 meses
- Sentenças de processos: 11 meses

Em relação ao ano de 2017, a Justiça do Trabalho (ou JT) recebeu 4.321.842 de casos novos, além de 4.622.521 sentenças, 4.481.991 baixados e 5.517.25 processo pendentes, sendo que no total, aproximadamente 82% das movimentações ocorreram no 1º grau de jurisdição, e o restante 18% das movimentações processuais ocorrendo no 2º grau da JT. No total, a Justiça do Trabalho é responsável por 7% dos processos existentes no Poder Judiciário como um todo, além de 14.8% dos casos novos inseridos no ano de 2017.

De acordo com o CNJ, a Justiça do Trabalho apresenta uma concentração no assunto "verbas rescisórias de rescisão do contrato de trabalho", o que ocorre em razão da JT possuir um menor quantitativo de assuntos cadastrados junto as Tabelas Nacionais, o que acarreta numa maior concentração de processos em um único item. Numericamente, a Justiça Trabalhista apresenta 241 assuntos cadastrados, em contraste dos 2.286 assuntos da Justiça Estadual por exemplo. Além disso, a Justiça Trabalhista apresenta um padrão mais homogêneo que outras esferas, pois a mesma apresenta vários tribunais que estão vinculados ao mesmo assunto.

Os assuntos mais demandados podem ser especificados da seguinte maneira:

- Assuntos mais demandados em 2017 (JT):
  - Remuneração, Verbas Indenizatórias e Benefícios / Adicional - 5.847.967 demandas (11.51%)
  - Rescisão do Contrato de Trabalho / Seguro Desemprego - 833.466 demandas (1.64%)
  - Remuneração, Verbas Indenizatórias e Benefícios / Salário / Diferença Salarial - 636.148 demandas (1.25%)
  - Responsabilidade Civil do Empregador / Indenização por Dano Moral - 538.757 demandas (1.06%)
  - Rescisão do Contrato de Trabalho / Verbas Rescisórias - 375.092 demandas (0.74%)
- Assuntos mais demandados no 2º grau em 2017 (JT):
  - Rescisão do Contrato de Trabalho / Verbas Rescisórias - 717.983 demandas (7.57%)
  - Responsabilidade Civil do Empregador / Indenização por Dano Moral demandas - 155.869 (1.64%)
  - Remuneração, Verbas Indenizatórias e Benefícios / Salário / Diferença Salarial demandas - 142.299 (1.50%)
  - Rescisão do Contrato de Trabalho / Seguro Desemprego demandas - 72.919 (0.77%)
  - Remuneração, Verbas Indenizatórias e Benefícios / Adicional demandas - 72.794 (0.77%)
- Assuntos mais demandados no 1º grau em 2017 (JT):
  - Rescisão do Contrato de Trabalho / Verbas Rescisórias - 5.129.984 demandas (16.29%)
  - Responsabilidade Civil do Empregador / Indenização por Dano Moral - 677.597 demandas (2.15%)
  - Remuneração, Verbas Indenizatórias e Benefícios / Salário / Diferença Salarial - 493.849 demandas (1.57%)
  - Rescisão do Contrato de Trabalho / Seguro Desemprego - 465.838 demandas (1.48%)
  - Remuneração, Verbas Indenizatórias e Benefícios / Adicional - 302.298 demandas (0.96%)

Além disso, o CNJ também apresenta uma lista das classes mais recorrentes no ano de 2017:

- Classes mais demandadas em 2017 (JT):
  - Processo de Conhecimento / Procedimento de Conhecimento - 2.738.482 demandas (7.62%)
  - Recursos / Recursos Trabalhistas - 846.767 demandas (2.36%)
  - Recursos / Embargos - 75.945 demandas (0.21%)
  - Outros Procedimentos / Cartas - 59.331 demandas (0.17%)
  - Processo de Execução / Embargos - 18.608 demandas (0.05%)
- Classes mais demandadas no 2º grau em 2017 (JT):
  - Recursos / Recursos Trabalhistas - 846.760 demandas (10.36%)
  - Recursos / Embargos - 22.088 demandas (0.27%)
  - Processo de Conhecimento / Procedimento de Conhecimento - 19.855 demandas (0.24%)
  - Processo Administrativo - 16.424 demandas (0.20%)
  - Precatório - 5.740 demandas (0.07%)
- Classes mais demandadas no 1º grau em 2017 (JT):
  - Processo de Conhecimento / Procedimento de Conhecimento - 2.718.627 demandas (14.01%)
  - Outros Procedimentos / Cartas - 59.330 demandas (0.31%)
  - Recursos / Embargos - 53.857 demandas (0.28%)
  - Processo de Execução / Embargos - 18.608 demandas (0.10%)
  - Processo de Execução / Processo de Execução Trabalhista - 15.977 demandas (0.08%)

Considerando a evolução da Justiça Trabalhista entre os anos de 2016 e 2017, o número de processos baixados cresceram em 6.6%. Comparando o volume de processos pendentes e o volume de novos processos ingressados, há uma razão de 1.3 processos pendentes por caso novo na Justiça do Trabalho. Tais números influenciam no chamado "tempo de giro de acervo", métrica que mede quanto tempo é necessário para zerar o estoque de processos de uma esfera da Justiça, considerando que não houvesse ingresso de novas demandas, e que tal esfera mantivesse a mesma produtividade dos magistrados e servidores. No caso da Justiça do Trabalho, o tempo de giro do acervo é de 1 anos e 2 meses.

## 2.5 Processo Judicial eletrônico (PJe)

O PJe (sigla para Processo Judicial Eletrônico) é um sistema de tramitação de processos judiciais, sendo um projeto fruto da união entre a Justiça Federal e o Conselho

Nacional de Justiça (sigla para CNJ), tendo como foco atender as necessidades dos diversos segmentos do Poder Judiciário brasileiro (Justiça Militar da União e dos Estados, Justiça do Trabalho e Justiça Comum, Federal e Estadual). Com isso, o CNJ visa adotar uma solução única e gratuita para todos os tribunais, assim como o acompanhamento dos processos, independentemente de qual esfera do Justiça o respectivo processo tramita [3]. O PJe foi lançado oficialmente em 21 de junho de 2011, e desde 3 de fevereiro de 2014, o CNJ está utilizando exclusivamente o PJe para o trâmite de novos processos, além de estarem migrando os processos que ainda estão em andamento [5].

## 2.6 Dados de um processo trabalhista

Atualmente, os processos digitais do PJe seguem o respectivo padrão:

- Número sequencial do processo
- Dígito verificador do processo
- Ano do processo
- Código da justiça do processo
- Código da região do processo
- Código da vara do processo
- Sistema de origem do processo
- Indicador se o processo está ou não em segredo de justiça
- Indicador se o documento irá trazer os movimentos em segredo do processo
- Data da próxima audiência do processo
- Nível da consulta do processo
- Classe da última movimentação do processo, contendo os respectivos dados:
  - Grau do processo
  - Data de autuação do processo
  - Código da classe do processo
  - Data da última movimentação do processo
- Lista das classes do processo, com cada objeto contendo os respectivos dados:
  - Grau do processo
  - Data de autuação
  - Código da classe do processo
  - Nome da classe do processo
  - Siga da classe do processo
  - Código do órgão julgador do processo
  - Nome do órgão julgador do processo
  - Índice indicando tramitação preferencial ou não do processo

- Lista das partes da classe do processo, com cada objeto contendo os respectivos dados:
  - \* Polo do processo
  - \* Nome da parte
  - \* Tipo de pessoa
  - \* Número do documento da parte
  - \* Lista de advogados da parte, com cada objeto contendo os respectivos dados:
    - Código OAB do advogado
    - Dígito OAB do advogado
    - Unidade federativa da OAB do advogado
    - Nome do advogado
    - Número do documento do advogado
  - \* Data de nascimento da parte
  - \* Sexo da parte
  - \* Outros detalhes da parte
- Conta judicial do processo
- Data de início da busca de tramitação do processo
- Lista das movimentações do processo, com cada objeto contendo os respectivos dados:
  - Sigla da classe da movimentação
  - Data da movimentação
  - Descrição da movimentação
  - Grau da movimentação
  - Lista dos documentos da movimentação do processo, com cada objeto contendo os respectivos dados:
    - \* ID do documento
    - \* Extensão do documento
    - \* Descrição do documento
    - \* Visibilidade do documento
    - \* ID único do documento
  - Código nacional da movimentação do processo
- Número do processo em formato simples
- Número do processo formatado
- Valor da causa

---

## Trabalhos relacionados

---

Este capítulo apresenta uma relação de trabalhos que envolvem modelos preditivos aplicados na área jurídica para diversos objetivos. Também é apresentado um resumo para cada trabalho, dispondo tanto sobre o objetivo como discorrendo sobre as ferramentas utilizadas em cada um.

Especificamente na área jurídica, [82] utilizou o algoritmo de Máquinas de Suporte Vetorial (*SVM - Support Vector Machine*) para analisar processos na corte jurídica francesa, propondo prever a área legal de um processo, a decisão final de um processo, e por fim a década de um processo, utilizando bigramas para representação de palavras. Uma evolução de [82] é apresentada em [81], que também aborda a análise de processos na corte jurídica francesa, novamente trabalhando na predição do resultado de um caso processual, predição da área legal de um processo, e predição temporal da década do processo, utilizando um sistema *ensemble* que combina os resultados de vários classificadores do tipo *SVM*.

Também na área jurídica, é possível citar [62], que utiliza um *corpus* do Tribunal Europeu dos Direitos Humanos para classificar sentenças como argumentativas ou não-argumentativas. [14] utiliza técnicas como *TF-IDF* (ou *frequência do termo–inverso da frequência nos documentos*) para seleção de características e *SVM* para classificação de um texto legal de acordo com um domínio jurídico. Os autores do trabalho [53] propõem um modelo para predição de decisões de processos, além da construção de um conjunto próprio de dados, utilizando *GRUs* com um mecanismo de atenção mútua organizado em pares. [7] utiliza *n-gramas* como características para representação textual em um modelo *SVM*, que tem como propósito uma classificação binária de casos onde ocorreram ou não uma violação de algum artigo da Convenção Europeia dos Direitos Humanos. [46] analisa dois séculos de dados da Suprema Corte dos Estados Unidos utilizando um classificador *Random forest* pra predição da decisão dos processos e dos votos da corte.

Dado a seleção de tais trabalhos relacionados, a proposição de um trabalho que vise trabalhar com publicações na justiça trabalhista e predição do tempo de processos e de movimentações processuais diferencia-se dos trabalhos relacionados previamente desenvolvidos. Ainda que também se desenvolva na área de classificação, o modelo



proposto por esse trabalho se distingue no alvo da classificação em si, buscando prever duração em tempo de movimentos dos processos, ao invés do resultado das decisões do processo em si. A inovação do modelo proposto por esse trabalho se apresenta não somente no alvo da predição, mas também na língua utilizada, haja visto que os trabalhos relacionados situam-se no contexto jurídico europeu e norte-americano, sendo nenhum especificamente em português.

## Métodos e Ferramentas

---

Este capítulo apresenta os métodos e ferramentas utilizados neste trabalho. A seção 4.1 apresenta um método utilizado para limpeza e normalização do corpus do trabalho, bem como explicando as razões de fazê-lo. A seção 4.2 apresenta como as palavras são representadas na arquitetura do modelo proposto. A seção 4.3 estende a representação de palavras via tokens também para a representação das partes do processo, expondo os motivos de tal representação, bem como seus resultados. A seção 4.4 explica os impactos do uso da normalização de valores numéricos durante o treino de modelos, assim como os valores resultantes calculados. A seção 4.5 descreve detalhadamente as camadas, conceitos e outros elementos que compõem a construção e execução do modelo utilizado, como *LSTM* (4.5.1), função de custo (4.5.2), mecanismo de atenção (4.5.3), codificação posicional (4.5.4), gradiente (4.5.5), *batch* (4.5.6), bem como um diagrama do modelo construído (4.5.7). Por fim, a seção 4.6 apresenta os equipamentos utilizados para a execução deste trabalho.

### 4.1 Limpeza e normalização do corpus

A normalização do corpus é uma forma de reduzir o tamanho do vocabulário, de forma a substituir diferentes instâncias textuais por uma única instância, sem uma perda substancial de informação. Além disso, a limpeza do corpus têm como objetivo remover erros e inconsistências que podem ocorrer por qualquer motivo nos textos que compõem o corpus. Para isso, é utilizado um script de normalização de texto descrito em [37] que pode ser livremente acessado e utilizado em [4]. As ações realizadas pelo script são:

- Conversão do texto para caixa baixa
- Remoção de reticências
- Remoção de textos dentro de colchetes
- Remoção de texto em HTML
- Substituição de todos os números por 0 (zero)
- Substituição de todas as ocorrências de email pelo token `EMAIL`

- Substituição de todas as ocorrências de URL pelo token URL
- Normalização de diferentes tipos de aspas e hífen para um único tipo de cada
- Remoção de outros sinais de pontuações

## 4.2 Representação de palavras via tokens e *embeddings*

*Embeddings* são representações dimensionais em forma de vetores, com o intuito de representar categorias e dados discretos como vetores numéricos, que podem ser projetados em um espaço dimensional.

No modelo proposto, *embeddings* serão utilizados para representar as informações das movimentações trabalhistas, bem como tokens para indicar informações como código da movimentação, órgão julgador de 1º e 2º grau, fase da movimentação, comarca da movimentação, e informação de partes na descrição das movimentações. Tal abordagem foi selecionada em detrimento de outras alternativas como *one-hot encoding*, haja visto que o alto número de instâncias para classes como órgão julgador produziria um vetor extremamente disperso em relação a informações, o que penalizaria a performance final do modelo.

Como forma de facilitar o treino, evitando a inicialização aleatória dos valores numéricos dos *embeddings* do vocabulário utilizado, [37] fornece vários *embeddings* para uso. No caso deste trabalho, foram utilizados os *embeddings* treinados via *GloVe* [66]. [37] disponibiliza os *embeddings* do *GloVe* em tamanhos de 50, 100, 300, 600 e 1.000 dimensões. Na predição de movimentações trabalhistas, os *embeddings* de tamanho 100 foram escolhidos arbitrariamente.

*GloVe* (sigla para *Global Vector for Word Representation*, ou Vetores Globais para Representação de Palavras) é um algoritmo não-supervisionado para a obtenção de representação vetoriais de palavras. O treino do algoritmo é feito utilizando estatísticas de co-ocorrência de pares de palavras em um corpus. Tais estatísticas são treinadas via um modelo log-bilinear de mínimo quadrado ponderado. Com isso, o treino do algoritmo obtém vetores que podem ser mapeados em um plano cartesiano, onde a distância euclidiana entre duas palavras relaciona-se com a similaridade semântica de ambas, fruto da composição dos valores dos vetores finais.

## 4.3 Representação de partes via tokens

Em [71], uma das atividades realizadas no pré-treino do modelo é a utilização de tokens para delimitação e representação de informações. Esta atividade delimita informações previamente estruturadas que passam a ser concatenadas em forma de texto, além de ser uma maneira de executar um *fine-tuning* facilmente implementável, porém

efetivo. Tal atividade de delimitação de estruturas de informações no texto feita em [71] tem inspirações em [74].

Para o contexto atual, este trabalho utiliza um token único para representar as informações de partes. Tal representação é uma forma de evitar problemas que possam acontecer com o nome das partes, pois nomes próprios apresentam palavras diferentes e que podem aparecer pouquíssimas vezes durante os dados de treino, influenciando negativamente a performance do modelo para representações de *embeddings* de tais palavras.

A alternativa implementada para tal problema é representar as partes por um token único, como uma forma de generalizar a existência do nome da parte para um treino em comum em relação as instâncias de partes. Com isso, o modelo é apresentado a um token que representa a instância de "parte", contextualizando a informação, ao invés de um nome próprio em específico que pode ocorrer em uma exposição insuficiente em representar informações o treino. Em suma, o modelo é sempre apresentado ao mesmo token único, ao invés de diferentes nomes próprios que, ainda que estejam no mesmo contexto, não apresentam ligações por via do nome.

Como um exemplo de parte que se beneficiará com o uso de tokens para representação, podemos utilizar as representações da empresa *Burger King*:

- Burger King do Brasil Assessoria A Restaurantes LTDA.
- Bk Brasil Operação e Assessoria A Restaurantes S.A.
- Fast Burger Comércio de Alimentos S/A.
- King Food Comércio de Alimentos S/A.

Caso tais partes fossem representadas pelos *embeddings* de cada uma das palavras, as palavras que tivessem uma aparição em menor número seriam subrepresentadas em relação com as palavras que apresentam uma aparição maior. Todos os exemplos apresentados representam a franquia *Burger King* no Brasil, porém palavras como *Burger* e *King* aparecem mais do que palavras como *Bk*, gerando um problema de subrepresentação e, com isso, no treino de *embeddings* menos assertivos.

Além disso, a junção dos *embeddings* não necessariamente representaria a parte em questão. A junção dos *embeddings* das palavras *Fast*, *Burger*, *Comércio*, *Alimentos* e *S/A* não necessariamente representaria a franquia *Burger King* nos processos trabalhistas, e muito menos garantiria a mesma representação que a junção dos *embeddings* das palavras *King*, *Food*, *Comércio*, *Alimentos* e *S/A*.

## 4.4 Normalização do valor de causa

Normalização de características é uma técnica utilizada para transformar um universo de valores expressivamente grandes para uma faixa de valores significativamente menor, geralmente dentro de um universo de valores do tipo  $[0, 1]$ .

Os valores pedidos na causa são uma característica que se encaixam nessa condição, podendo ser de algumas dezenas de reais, ou até em alguns casos na casa de milhões (geralmente reservados para ações ajuizadas em coletivo). Tal dinâmica de valores discrepantes pode penalizar a acurácia final do modelo, o que faz necessário o uso de normalização para melhor acomodar os valores fornecidos.

Dentro os normalizadores disponíveis, dado o espaço existente de valores possíveis e a forte presença de *outliers* nos dados de treino, neste trabalho fora utilizado o normalizador *RobustScaler* da biblioteca *scikit-learn* [65]. Tal normalizador remove a mediana dos valores e escala os valores de acordo com a amplitude interquartil da distribuição dos valores resultantes.

Com isso, o espaço inicial de valores que se encontrava na ordem de  $[1.000; 70.993.010]$ , foi transformado para uma ordem de  $[-0,93210465; 1502,97659146]$ . Ou seja, ao invés de mapear um espaço de aproximadamente 71 mil valores, o modelo necessitará trabalhar com um espaço de aproximadamente 1.500 valores de diferença, ou seja, uma redução de 47 vezes no espaço de valores.

## 4.5 Arquitetura da rede neural

### 4.5.1 *Long short-term memory*

*Long Short-Term Memory* (ou *LSTM*) [40] é uma arquitetura de redes neurais recorrentes, como uma evolução de redes neurais recorrentes clássicas [75]. Uma rede neural recorrente é uma categoria de redes neurais onde as conexões entre os neurônios conseguem formar uma representação temporal entre os dados de entrada. A arquitetura *LSTM* utiliza um novo tipo de unidade denominada "células de memória", uma característica que se mostra bastante útil para atividades como reconhecimento de fala [47] e modelagem de linguagem [27]. As células de memória servem como um controle para a quantidade de informação que será esquecida ou mantida durante as etapas do treino. O cálculo das células de memória podem ser descritas pelas equações (4-1) até (4-6):

$$\mathbf{i}_t = \sigma(\mathbf{W}_i \mathbf{h}_{t-1} + \mathbf{U}_i \mathbf{x}_t + \mathbf{b}_i) \quad (4-1)$$

$$\mathbf{f}_t = \sigma(\mathbf{W}_f \mathbf{h}_{t-1} + \mathbf{U}_f \mathbf{x}_t + \mathbf{b}_f) \quad (4-2)$$

$$\tilde{\mathbf{c}}_t = \tanh(\mathbf{W}_c \mathbf{h}_{t-1} + \mathbf{U}_c \mathbf{x}_t + \mathbf{b}_c) \quad (4-3)$$

$$\mathbf{c}_t = \mathbf{f}_t \cdot \mathbf{c}_{t-1} + \mathbf{i}_t \cdot \tilde{\mathbf{c}}_t \quad (4-4)$$

$$\mathbf{o}_t = \sigma(\mathbf{W}_o \mathbf{h}_{t-1} + \mathbf{U}_o \mathbf{x}_t + \mathbf{b}_o) \quad (4-5)$$

$$\mathbf{h}_t = \mathbf{o}_t \cdot \tanh(\mathbf{c}_t) \quad (4-6)$$

Em relação ao problema de sumiço do gradiente (ou *vanishing gradient problem*) [38] [39] [30] [64], as camadas de arquitetura *LSTM* abordam tal problema com um algoritmo para um gradiente *constante*, barrando assim o desaparecimento do gradiente durante o treino do modelo.

No modelo proposto, camadas *LSTM* foram utilizadas como forma de processar listas de tokens, referente as informações das movimentações trabalhistas, de modo a trabalhar listas de dados de tamanhos variáveis, bem como as relações temporais entre tais movimentações.

## 4.5.2 Função de custo

O modelo proposto trata o problema como uma sequência de problemas do tipo *one-vs-one*, ou seja, um problema multi-classe contendo  $N$  classes é subdividido em  $N$  modelos binários, cada um prevendo se um caso de exemplo pertence ou não a uma determinada classe [12]. Com isso, este trabalho irá gerar 1 modelo para classe do problema, onde cada modelo preditivo terá como objetivo classificar um grupo de informações como pertencente ou não a uma determinada classe.

A função de custo de entropia cruzada binária mede a performance de um modelo de classificação pela sua saída, que representa a probabilidade das entradas pertencerem ou não a uma determinada classe, sendo tal probabilidade emitida por um valor entre 0 e 1. O custo da entropia cruzada em geral diminui na mesma proporção que o modelo consegue convergir as probabilidades calculadas com as apresentadas no mundo real pelos objetos selecionados para treino, até atingir o valor zero.

A função de custo de entropia cruzada binária é composta por uma função de custo de entropia cruzada [60] e uma ativação de uma função sigmoide sobre o valor retornado pela função de custo. Sendo  $p$  o valor previsto pelo modelo, e  $y$  o valor real dos dados de treino, a função de custo de entropia cruzada pode ser descrita pela equação

(4-7):

$$H(p, y) = -(y \cdot \log(p) + (1 - y) \cdot \log(1 - p)) \quad (4-7)$$

A sigmoide pode ser definida como uma função logística e monótona, que quando desenhada apresenta um formato de "S" no plano cartesiano [34]. Por questão da função sigmoide, é garantido que os valores resultantes do modelo estarão limitados pelo intervalo  $[0, 1]$ . Sendo  $x$  o valor numérico a ser transformado pela sigmoide, a sigmoide pode ser descrita pela equação (4-8):

$$S(x) = \frac{1}{1 + e^{-x}} = \frac{e^x}{e^x + 1} \quad (4-8)$$

A função sigmoide também pode ser reescrita em função da tangente hiperbólica, como demonstrado pelas equações (4-9), (4-10), (4-11) e (4-12):

$$S(x) = \frac{e^{x/2}}{e^{x/2} + e^{-x/2}} \quad (4-9)$$

$$S(x) = \frac{1}{2} + \frac{1}{2} \cdot \frac{e^{x/2} - e^{-x/2}}{e^{x/2} + e^{-x/2}} \quad (4-10)$$

$$S(x) = \frac{1}{2} + \frac{1}{2} \cdot \frac{e^x - 1}{e^x + 1} \quad (4-11)$$

$$S(x) = \frac{1}{2} + \frac{1}{2} \cdot \tanh\left(\frac{x}{2}\right) \quad (4-12)$$

### 4.5.3 Mecanismo de atenção

[85] introduziu o conceito de mecanismo de atenção para trabalhos de processamento de linguagem natural, inicialmente como uma alternativa para evitar o uso de redes recorrentes e convolucionais, com inspiração em [9] [48] [63]. Mecanismos de atenção são partes integrais no tratamento de sequências de tokens, estabelecendo dependências entre todos os tokens de uma sequência, independente de sua posição ou distância de outras entidades.

O mecanismo de atenção pode ser descrito como uma função de atenção que mapeia uma consulta e um grupo de pares entre chaves e valores em um valor de saída, sendo tanto a consulta como o grupo de pares representados via vetores. O valor mapeado de saída é calculado como uma soma ponderada dos valores apresentados, sendo o peso de cada valor computado via uma função de concordância entre a consulta e sua chave correspondente.

O cálculo da função de atenção recebe como entrada vetores de consultas e chaves de dimensão  $d_k$ , e valores de dimensão  $d_v$ . A função então opera uma multiplicação entre a consulta e todas as chaves, divide cada resultado por  $\sqrt{d_k}$  e aplica uma função *softmax* [32] [12] (semelhante a função de sigmoide descrita na seção 4.5.2) para obter os pesos de cada valor informado.

Sendo assim, a função de atenção é calculada em uma sequência de consultas, gerando uma matriz  $Q$  de resultados. As chaves e valores também são usadas para construir as matrizes  $K$  e  $V$  respectivamente, permitindo calcular a matriz de valores finais por meio da equação (4-13):

$$\text{Atenção}(Q, K, V) = \text{softmax}\left(\frac{Q \cdot K^T}{\sqrt{d_k}}\right) \cdot V \quad (4-13)$$

A função *softmax* (também conhecida como função exponencial normalizada) é uma função que recebe um vetor  $K$  como entrada e normaliza tal vetor em uma distribuição probabilística de  $K$  probabilidades. Com isso, enquanto os valores originais do vetor  $K$  poderiam estar negativos, maiores do que 1, e com a soma de todos eles resultando em um valor qualquer, após a aplicação da função *softmax* em tal valor, todos os valores passam a estar presentes dentro do intervalo  $[0, 1]$ , além da soma de todos os valores do mesmo vetor passarem a sempre ser igual a 1, possibilitando a interpretação de tais valores como probabilidades, e com a soma de todas as probabilidades se igualando a 100%.

Sendo  $X$  um vetor de  $K$  elementos a serem transformados, a função *softmax* pode ser descrita pela equação (4-14):

$$\text{Softmax}(X_i) = \frac{e^{x_i}}{\sum_{j=1}^K e^{x_j}}, \text{ para } i = 1, 2, \dots, k - 1, k \quad (4-14)$$

No modelo proposto, o mecanismo de atenção fora utilizado como um aditivo no processamento textual dos tokens das informações das movimentações, utilizado como uma maneira de contribuir para a acurácia final do modelo.

#### 4.5.4 Codificação posicional

A codificação posicional é uma abordagem para adicionar informações numéricas a *embeddings* de tokens, como forma de informar ao modelo valores que possam estar acoplados aos *embeddings* de tokens. Em [85], os *embeddings* das informações textuais são gerados junto com os *embeddings* de codificação posicional, onde o *embedding* de um token é somado junto ao *embedding* da informação numérica daquele token em específico, e com isso o modelo consegue ser informado de uma palavra e a posição daquela palavra em um texto, por exemplo. A codificação posicional é originalmente construída



por meio de funções de seno e cosseno em diferentes frequências, permitindo que o modelo represente sequências de tamanhos diferentes. Com isso, as funções formam uma progressão geométrica entre  $2\pi$  e  $2\pi \cdot 10^4$ . Finalmente, após a matriz de valores da codificação posicional ser construída, os valores não são mais atualizados durante o tempo de treino do modelo [29].

Sendo  $p$  a posição na matriz de dias de um número específico que representa a diferença em dias, sendo  $i$  a dimensão dos *embeddings*, e sendo  $d_e$  o tamanho dos *embeddings*, as funções de codificação posicional podem ser descritas de acordo com a equação (4-15) para geração via seno de índices pares e com a equação (4-16) para geração via cosseno de índices ímpares:

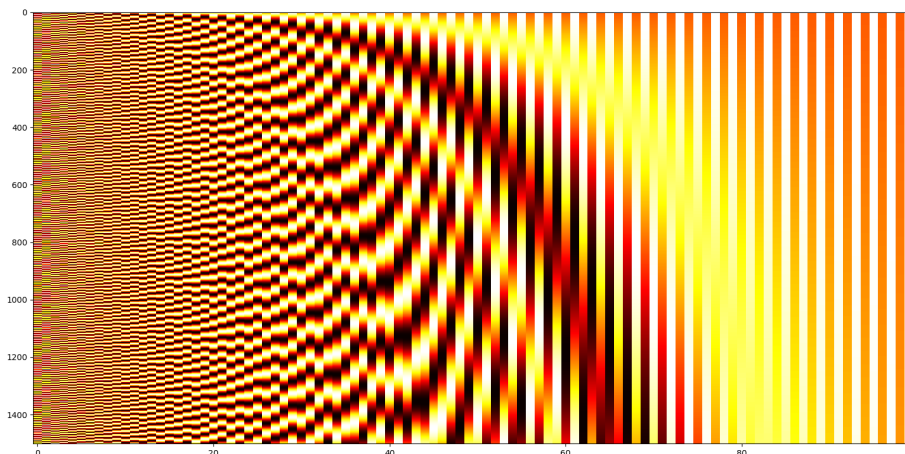
$$CP(p, 2i) = \sin\left(\frac{p}{10000^{2i/d_e}}\right) \quad (4-15)$$

$$CP(p, 2i + 1) = \cos\left(\frac{p}{10000^{2i/d_e}}\right) \quad (4-16)$$

As informações das movimentações, ao serem representadas como um token para cada palavra ou outra informação e ao serem concatenadas em um vetor único, perdem a representação em dias da data do acontecimento de cada movimentação. A codificação posicional é utilizada como uma adição aos *embeddings* de cada token, como forma de acoplar a cada token a informação em dias de sua respectiva data de acontecimento.

A codificação posicional tem como objetivo

A Figura 4.1 exibe o resultado final de uma matriz de codificação posicional, composta por 100 posições por *embedding* e janela temporal de 1500 dias:



**Figura 4.1:** Ilustração da matriz de codificação posicional obtida

A Figura 4.1 exibe uma representação gráfica da matriz de codificação posi-

cional. O padrão de ondas verificado na imagem indica que a construção da matriz de codificação posicional foi feita com sucesso, pois esse padrão é o resultado do uso alternado das funções de seno e cosseno nos valores de dias. Na horizontal, a Figura 4.1 representa as posições do vetor de cada *embedding* de 100 posições, enquanto na vertical estão representados os dias possíveis para cada movimentação trabalhista, limitando a 1500 dias. Por exemplo, a linha de número 1400 indica a representação em *embeddings* do dia 1400, após a aplicação alternada das funções de seno e cosseno. Durante o treino, caso alguma movimentação tenha acontecido 1400 dias após o início do processo, essa linha seria selecionada como representação vetorial da data da movimentação.

### 4.5.5 Gradiente

O gradiente *Adam* [49] [73] (abreviação de *Adaptive Moment Estimation* - Estimativa de Momento Adaptivo) é uma evolução de dois gradientes estocásticos, combinando vantagens de ambos: *RMSProp* [84], que apresenta sucesso em relação a gradientes esparsos, e *AdaGrad* [22], que apresenta sucesso em ambientes não-estacionários. O gradiente *Adam* é uma otimização estocástica eficiente criada com o intuito de exigir um baixo requerimento de memória, onde o gradiente calcula taxas de aprendizado individuais e adaptativas para os *momentos* do gradiente. No trabalho proposto, o gradiente *Adam* foi utilizado com a taxa de aprendizado igual a  $10^{-3}$ .

### 4.5.6 Batch

Um *batch* (ou "lote" em tradução livre) pode ser exemplificado como um conjunto de exemplos. Durante o processo de treino, os exemplos dos dados de treino são subdivididos em lotes de mesmo tamanho, onde cada mini-divisão de *batch* é inserida no modelo, a predição gerada é utilizada para calcular um erro e então atualizar o modelo de acordo com o erro encontrado. Cada iteração do modelo com um lote é denominada de etapa, e a junção de todas as etapas dos lotes compõe uma época.

No modelo proposto, foram escolhidos *batches* de tamanho 128 para treino. Trabalhos como [79] argumentam que, ainda que trabalhos atuais utilizem valores de *batch* de tamanho inicial como 32, 64 e 128, (e a partir desse valor experimentarem outros valores próximos), é possível atingir melhorias de resultado com valores incomuns e excepcionalmente grandes para *batch*, como por exemplo 65.536 itens por *batch*. Porém, dado as limitações técnicas de memória em replicar tais resultados, fora decidido uma abordagem mais tradicional na montagem do tamanho dos *batches* de treino.

### 4.5.7 Diagrama da arquitetura

Após os trabalhos de pré-processamento no corpus utilizado, as seguintes informações foram utilizadas para a modelagem de informação do conjunto de dados de movimentações da esfera trabalhista:

- Movimentações:
  - Diferença em dias da movimentação
  - Código da movimentação
  - Grau da movimentação
  - Descrição da movimentação
- Comarca do processo
- Fase do processo
- Órgão Julgador em 1º grau do processo
- Órgão Julgador em 2º grau do processo
- Valor de causa normalizado

As informações de diferença em dias, código, grau e descrição são informações que todas as movimentações de um processo apresentam, enquanto informações de comarca, fase, órgão julgador de 1º e 2º grau e valor de causa são informações que o processo apresenta. O valor de causa é inserido com o valor normalizado, como descrito na seção 4.4. As informações de comarca do processo, fase do processo, órgão julgador do 1º grau do processo e órgão julgador do 2º grau do processo são inseridos via *embeddings*, conforme descrito na seção 4.2. Sendo  $n$  a quantidade de classes existentes para uma informação categórica, o tamanho do vetor de dimensões do *embedding* de cada informação categórica é igual a  $\lceil n^{1/4} \rceil$  [1]. Por exemplo, em um caso onde houvessem 80 classes diferentes para uma categoria, o cálculo do tamanho do vetor de *embeddings* seria como definido pela equação (4-17):

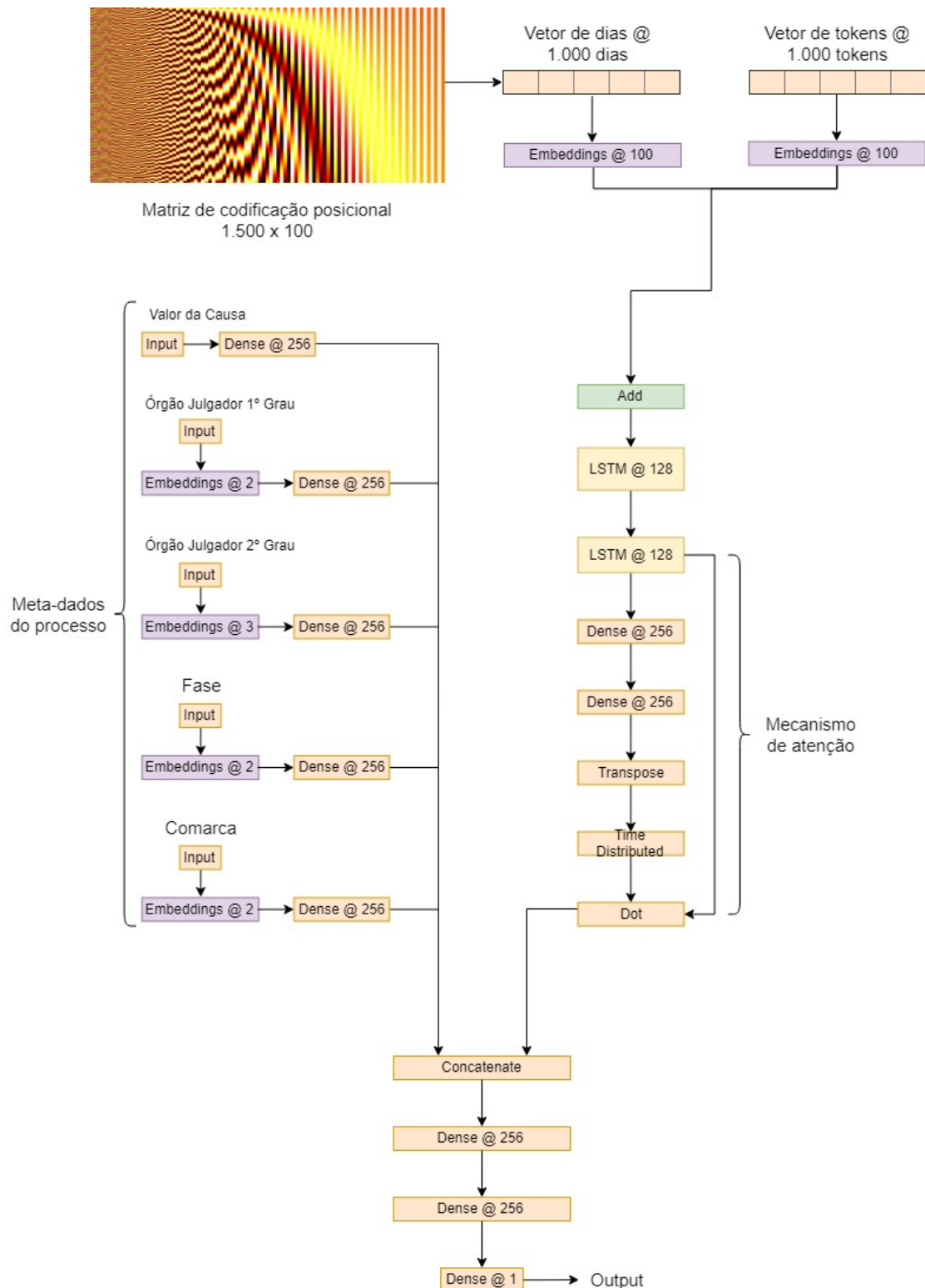
$$\lceil 8^{1/4} \rceil = \lceil 2.99069 \rceil = 3 \quad (4-17)$$

As informações das movimentações são tratadas por meio de dois vetores de inteiros: o primeiro representando os dias do acontecimento de cada token, e o segundo representando os tokens da concatenação das informações das movimentações. Os dois vetores das informações de movimentação são tratados por um combo de células de *embedding* e *LSTM*, conforme demonstrado em 4.5.1 e 4.2 respectivamente. Os *embeddings* gerados por cada entrada são então somados, formando um vetor único resultante das somas dos *embeddings*, de acordo com o mecanismo de codificação posicional em 4.5.4. O vetor de *embedding* resultante é então inserido em um mecanismo de atenção, de acordo com a descrição em 4.5.3. As entradas restantes são inseridas a parte, com as entradas

categóricas sendo tratadas por meio de *embeddings*, e os resultados das entradas são concatenadas em uma única camada, que então culmina numa célula final do resultado previsto.

As informações das movimentações são fixas como vetores de tamanho mil. Esse valor de limitação foi escolhido de forma arbitrária, e foi usado como um valor limite que pudesse abarcar o máximo de movimentações possíveis pertencentes ao conjunto de dados de treino, ao mesmo tempo que não fosse um impeditivo a nível de recursos técnicos disponíveis. No caso de uma concatenação resultar em um vetor maior que mil tokens, apenas os últimos mil tokens serão utilizados. No caso da concatenação resultar em um vetor menor que os mil tokens, o vetor será preenchido com valores igual a zero até atingir o tamanho de mil tokens. O tratamento de tamanho ocorre tanto no vetor de dias do acontecimento dos tokens das movimentações, como também no vetor dos tokens das informações das movimentações.

A Figura 4.2 exhibe uma exemplificação da arquitetura do modelo proposto:



**Figura 4.2:** Ilustração do modelo proposto

Na figura 4.2, há uma diagramação da arquitetura do modelo proposto. A entrada dos dados é subdividida em dois grupos. O primeiro grupo de entradas compete a parte de valor de causa, órgão julgador de 1º e 2º grau, fase do processo, e comarca. Nesse primeiro grupo, apenas o valor de causa é representado como valor numérico, sendo todas as

outras informações dados categóricos representados via *embeddings*. O segundo grupo de entradas recebe as informações dos vetores dos tokens das movimentações, bem como os vetores dos dias de cada movimentações. Ambas as informações são adicionadas entre si via codificação posicional, como descrito na seção 4.5.4. A codificação posicional é então processada por um mecanismo de atenção, como descrito na seção 4.5.3. Os resultados do processamento de ambos os grupos são concatenados em uma única camada, que então resulta na saída final do modelo.

## 4.6 Equipamentos

Os resultados obtidos foram alcançados em uma máquina da seguinte configuração:

- Processador: 128 x Power9 16-core @ 2.7 GHz
- Memória: 1.02 TB Ram
- GPU: NVIDIA Tesla V100 16GB

---

## Resultados

---

Neste Capítulo são apresentados e discutidos os resultados obtidos no trabalho. A seção 5.1 explica as métricas utilizadas - acurácia (5.1.1),  $F_1$  score (5.1.2) e área sob a curva (5.1.3) - enquanto a seção 5.2 expõe os resultados obtidos.

### 5.1 Métricas utilizadas

Para avaliação de resultados do modelo proposto, foram utilizados três métricas: acurácia,  $F_1$  score e área sob a curva.

#### 5.1.1 Acurácia

A acurácia pode ser descrita como a fração de elementos relevantes selecionados dentro de um todo, e também como o grau de proximidade entre as medições de um sistema e o valor real dos itens medidos [26]. A acurácia é definida pela equação (5-1):

$$Acurácia = \frac{\sum \text{Positivos verdadeiros} + \sum \text{Negativos verdadeiros}}{\sum \text{Total de elementos}} \quad (5-1)$$

#### 5.1.2 $F_1$ score

O  $F_1$  score é uma métrica estatística que leva em consideração a precisão e o *recall* de modelos preditivos como forma de pontuação da acurácia de tais modelos.

A precisão pode ser definida como a fração de elementos relevantes selecionados dentro de um grupo de elementos selecionados. Ou seja, em uma classificação binária, a precisão é a razão entre o número de verdadeiros positivos selecionados por um modelo e o número de elementos que foram classificados como positivos pelo modelo. Sendo assim, a precisão pode ser definida pela equação (5-2):

$$Precisão = \frac{\sum \text{Positivos verdadeiros}}{\sum \text{Positivos previstos}} \quad (5-2)$$

O *recall* é definido como a razão entre os elementos relevantes selecionados e todos os itens relevantes dentro do universo de elementos disponíveis. Em outras palavras, em uma classificação binária, o *recall* é definido como a fração de verdadeiros positivos selecionados dentro de todos os positivos disponíveis no grupo de elementos. O *recall* pode ser definido pela equação (5-3):

$$Recall = \frac{\sum Positivos\ verdadeiros}{\sum Positivos\ totais} \quad (5-3)$$

Sendo  $p$  a precisão e  $r$  o *recall* de um modelo, o cálculo do  $F_1$  score de um modelo pode ser definido pela equação (5-4):

$$F_1(p, r) = \left( \frac{r^{-1} + p^{-1}}{2} \right)^{-1} = 2 \cdot (p^{-1} + r^{-1})^{-1} = \frac{2 \cdot p \cdot r}{p + r} \quad (5-4)$$

### 5.1.3 Área sob a curva

A área sob a curva é o nome dado para a área abaixo da Curva Característica de Operação do Receptor (ou curva COR). A curva COR pode ser definida como uma representação gráfica que ilustra a performance de um modelo binário a partir do limite de decisão projetado pelo modelo [69].

A área sob a curva representa a probabilidade de um classificador binário retornar um valor maior para um elemento positivo do que um elemento negativo, sendo ambos escolhidos aleatoriamente e sendo um classificador que utilize para elementos positivos valores maiores do que elementos negativos, como 0 para negativo e 1 para positivo, por exemplo [24].

A curva COR é criada pela projeção da razão entre taxa de positivos verdadeiros e a taxa de positivos falsos. A taxa de positivos verdadeiros, também conhecidos como *recall*, pode ser descrito pela equação (5-3), como apresentado na seção 5.1.2. Já a taxa de positivos falsos, pode ser descrita como a razão entre os elementos não-relevantes que foram erradamente selecionados como relevantes (ou seja, elementos erroneamente assinalados como positivos) e todos os elementos não-relevantes do universo trabalhado. A taxa de falso positivo pode ser descrita pela equação (5-5):

$$Taxa\ de\ falso\ positivo = \frac{\sum Positivos\ falsos}{\sum Negativos\ totais} \quad (5-5)$$

Sendo  $X$  a probabilidade de uma determinada instância de elemento, sendo  $T$  o limite de decisão, um elemento é classificado como positivo se  $X > T$ , ou seja, se a posição de  $X$  estiver acima do limite de decisão  $T$  no espaço cartesiano. De contrário, caso  $X < T$ , temos que  $X$  é um elemento negativo para a dada classe do modelo binário. Sendo  $f_1(x)$  a densidade de probabilidade caso o elemento  $X$  seja classificado como positivo pelo limite



de decisão  $T$ , e sendo  $f_0(x)$  a densidade de probabilidade do mesmo elemento caso ele tenha sido classificado como negativo pelo mesmo limite de decisão, é possível definir a taxa de verdadeiro positivo pela equação (5-6) e a taxa de falso positivo pela equação (5-7):

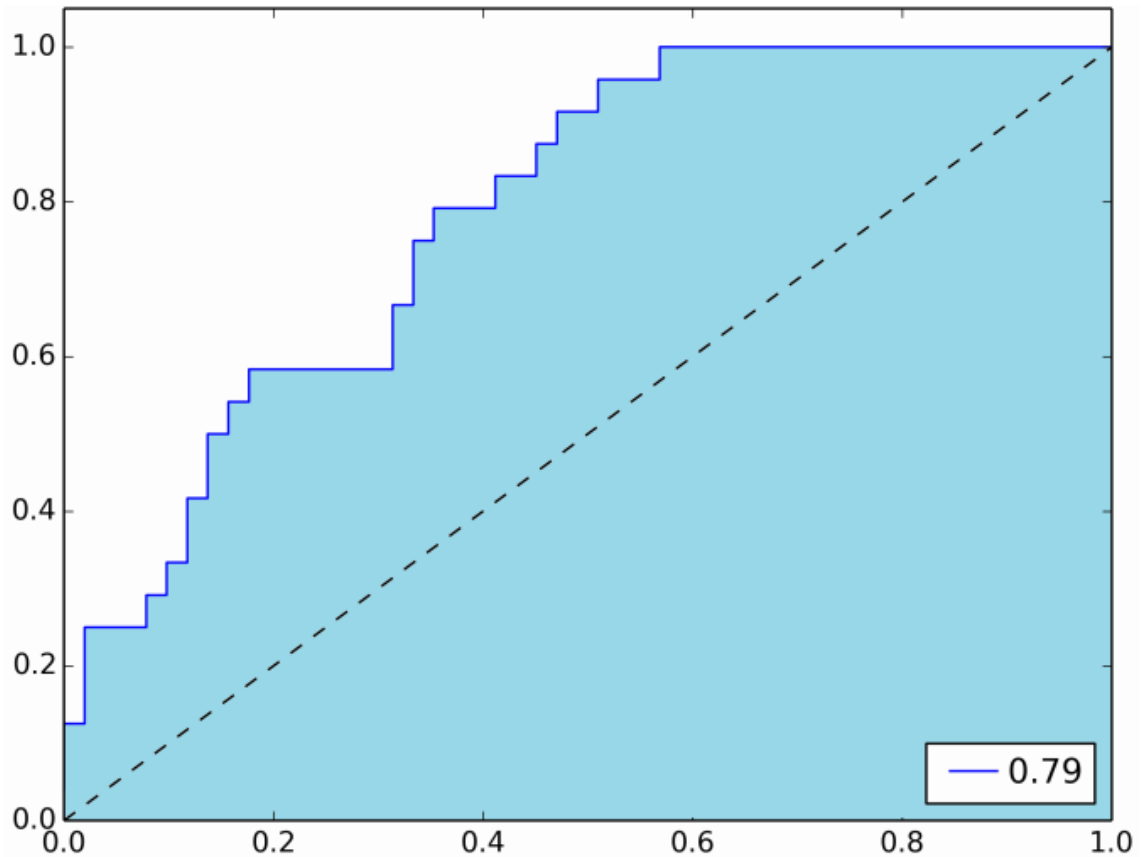
$$TVP(T) = \int_T^{\infty} f_1(x) dx \quad (5-6)$$

$$TFP(T) = \int_T^{\infty} f_0(x) dx \quad (5-7)$$

Com isso, a projeção da curva COR no plano cartesiano representa parametricamente  $TVP(T)$  versus  $TFP(T)$ , tendo o limite de decisão  $T$  como parâmetro de variável.

Dado a projeção da curva COR sobre um plano cartesiano, é possível calcular a área sob a curva projetada. A área sob a curva representa a probabilidade do classificador analisado de classificar positivamente um elemento positivo qualquer que se situe acima de um elemento negativo qualquer no plano cartesiano, supondo que a classe positiva se situe acima da classe negativa no plano cartesiano em relação ao limite de decisão projetado pelo modelo escolhido [24].

A Figura 5.1 apresenta um exemplo de curva COR, tendo também colorida a área sob a curva. No caso da figura, a área sob a curva apresenta um valor de 0.79:



**Figura 5.1:** Exemplo de área sob a curva

Além disso, o cálculo da área sob a curva é fortemente relacionado ao Teste U de Mann-Whitney [36] [55], um teste estatístico e não paramétrico que compara dois grupos, buscando as chances de um elemento aleatório de um primeiro grupo apresentar um valor superior ou não a um elemento aleatório de um segundo grupo, calculando por meio da igualdade das medianas de dois grupos. O coeficiente de Gini, uma medida de dispersão estatística que busca calcular a igualdade da distribuição de valores para elementos de um mesmo grupo [35], também se relacionado ao cálculo da área sob a curva. Sendo  $G_1$  o coeficiente de Gini, a equação (5-8) descreve a área sob a curva em função do coeficiente de Gini, permitindo o cálculo da área sob a curva por meio da média de aproximações trapezoidais [35]:

$$G_I = 1 - \sum_{k=1}^n (X_k - X_{k-1}) \cdot (Y_k + Y_{k-1}) = 2 \cdot A(x) - 1 \quad (5-8)$$

O cálculo da área sob a curva resulta em valores variando de 0 a 1, sendo 0 para modelos que erraram todos os casos apresentados e 1 para modelos que acertaram todos os casos apresentados [69] [70]. Além disso, em alguns casos é possível calcular parcialmente a área sob a curva [57], o que pode ser útil em certas situações para a análise específica de uma região da área sob a curva COR.

## 5.2 Resultados obtidos

Inicialmente, fora feito uma análise envolvendo 2.554 processos da esfera trabalhista, especificadamente do TRT 15 (presente em São Paulo), o que contempla 114.988 movimentações, todas geradas no ano de 2015.

Como forma de analisar separadamente a acurácia da arquitetura proposta para cada espaço delimitado de dias, foram delimitadas 5 classes que pudessem distribuir de maneira consistente as movimentações em relação a sua diferença em dias da data da movimentação imediatamente anterior, sendo as classes as seguintes:

- 1-7 dias
- 8-15 dias
- 16-30 dias
- 31-60 dias
- Mais de 60 dias

Durante um treino de 5 épocas para cada classe, foram utilizadas 80% das movimentações para treino e 20% das movimentações para teste. Cada época durou aproximadamente 11 minutos, o que contabilizou um tempo aproximado de 275 minutos (ou 4 horas e meia) para um treino completo envolvendo todas as classes.

Após o período de treino de cada classe, cada modelo executou uma etapa de predição dos dados fornecidos, tanto de treino como de teste, como forma de avaliar o resultado final, utilizando as métricas estabelecidas na seção 5.1 para avaliar a exatidão de cada modelo treinado, além de permitir a comparação de todos entre si. Usando a delimitação das classes propostas, os resultados obtidos foram organizados de acordo com a tabela 5.1:

Métricas	Máximo	Mínimo	Média	Desvio padrão
<b>1-7 dias</b>				
Acurácia de treino	85.96%	83.08%	84.83%	0.83%
$F_1$ score de treino	0.855	0.808	0.845	0.015
Acurácia de teste	76.19%	73.36%	73.85%	0.86%
$F_1$ score de teste	0.749	0.719	0.728	0.009
Área sob a curva	0.812	0.789	0.802	0.008
<b>8-15 dias</b>				
Acurácia de treino	84.84%	82.58%	83.95%	0.77%
$F_1$ score de treino	0.854	0.812	0.831	0.009
Acurácia de teste	66.62%	58.24%	61.31%	2.65%
$F_1$ score de teste	0.678	0.633	0.656	0.014
Área sob a curva	0.683	0.614	0.645	0.026
<b>16-30 dias</b>				
Acurácia de treino	86.18%	84.47%	85.11%	0.48%
$F_1$ score de treino	0.848	0.827	0.837	0.007
Acurácia de teste	62.07%	55.76%	57.36%	01.71%
$F_1$ score de teste	0.665	0.615	0.651	0.015
Área sob a curva	0.647	0.618	0.631	0.010
<b>31-60 dias</b>				
Acurácia de treino	86.95%	85.22%	85.97%	0.59%
$F_1$ score de treino	0.864	0.838	0.852	0.008
Acurácia de teste	66.16%	55.46%	58.76%	2.91%
$F_1$ score de teste	0.681	0.639	0.651	0.012
Área sob a curva	0.656	0.575	0.621	0.028
<b>Mais de 60 dias</b>				
Acurácia de treino	89.56%	87.03%	88.95%	0.88%
$F_1$ score de treino	0.892	0.873	0.884	0.007
Acurácia de teste	71.02%	61.34%	62.05%	02.90%
$F_1$ score de teste	0.739	0.681	0.687	1.017
Área sob a curva	0.682	0.663	0.673	0.006

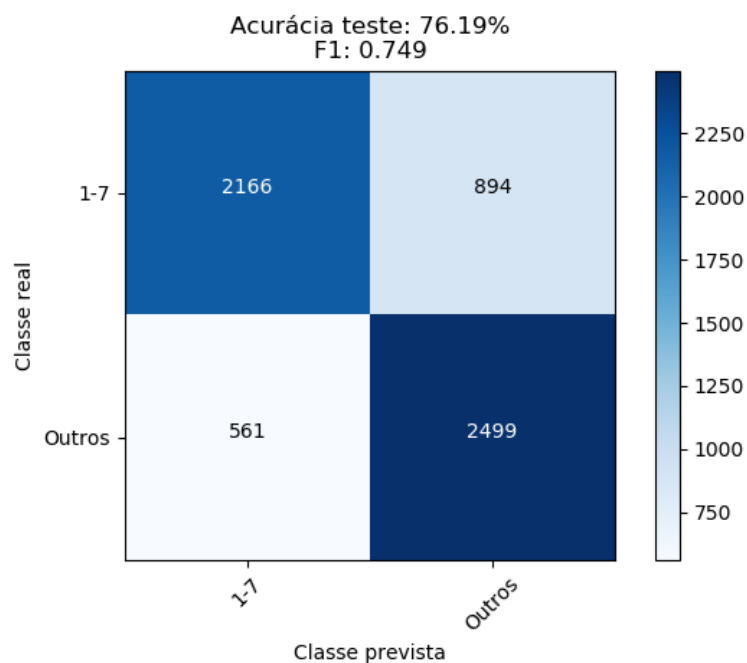
Tabela 5.1: Resultados obtidos pelo treino

Na tabela 5.1, percebe-se que quanto mais extremo em nível de prazo for um caso analisado, mais chances de ser assertivo a classificação terá: as classes 1-7 dias e mais de 60 dias são as que apresentam os melhores valores de teste para acurácia e  $F_1$  score. Comparando somente os valores médios em teste, a classe 1-7 dias alcançou 73.85% de acurácia e 0.728 de  $F_1$  score, enquanto a classe mais de 80 dias alcançou 62.05% de acurácia e 0.687 de  $F_1$  score. As classes 16-30 dias e 31-60 dias foram as únicas que apresentaram valores de acurácia de teste abaixo da marca de 60%: 57.36% na acurácia de teste para a classe 16-30 dias, e 58.76% para a classe 31-60 dias. Em nenhuma outra classe foi observada o mesmo comportamento.

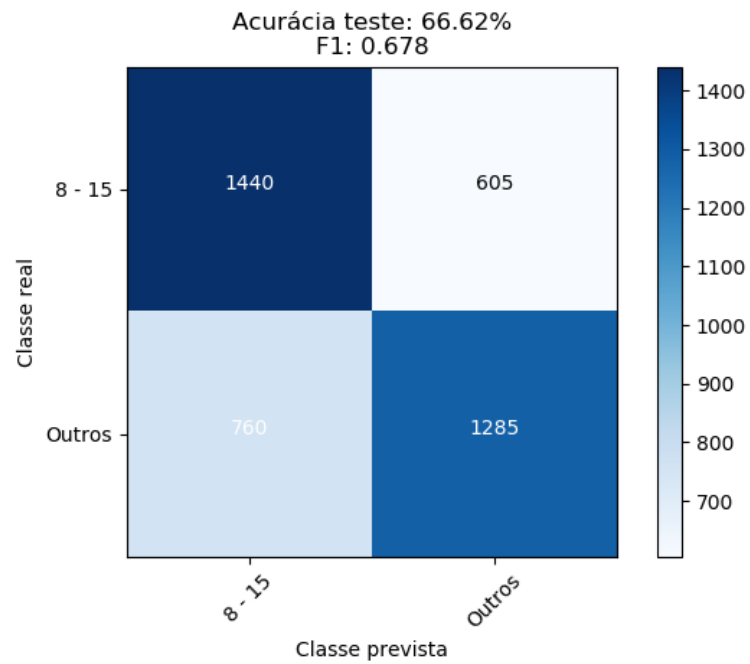
No tocante da área sob a curva, a classe 1-7 dias alcançou o valor de 0.802, sendo a única a apresentar um valor destoante da faixa entre 0.6 a 0.7. Todas as outras classes apresentaram valores da área sob a curva presentes nessa faixa, sendo o menor valor em 0.621 (classe 31-60 dias) e o maior valor em 0.673 (mais de 60 dias).

### 5.2.1 Resultado das matrizes de confusão

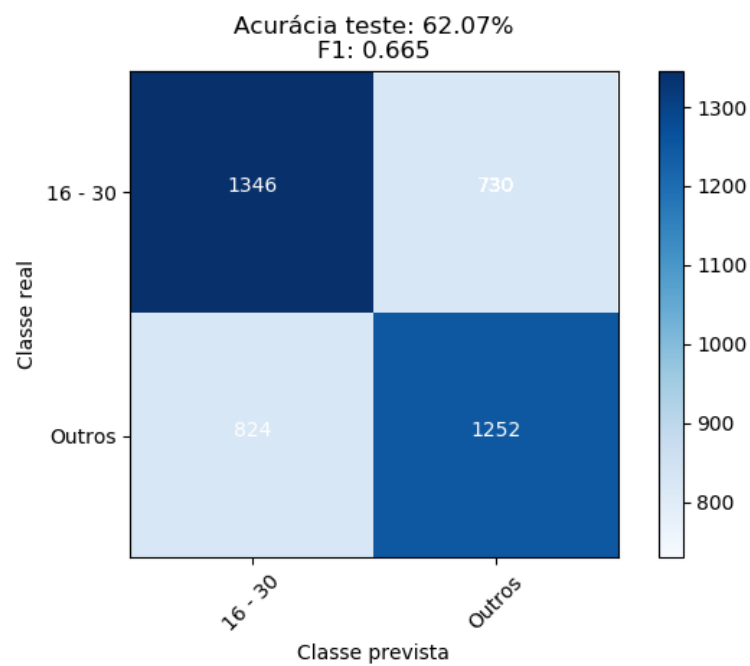
As matrizes 5.2, 5.3, 5.4, 5.5 e 5.6 representam as matrizes de confusão das classes 1-7 dias, 8-15 dias, 16-30 dias, 31-60 dias, e mais de 60 dias, respectivamente:



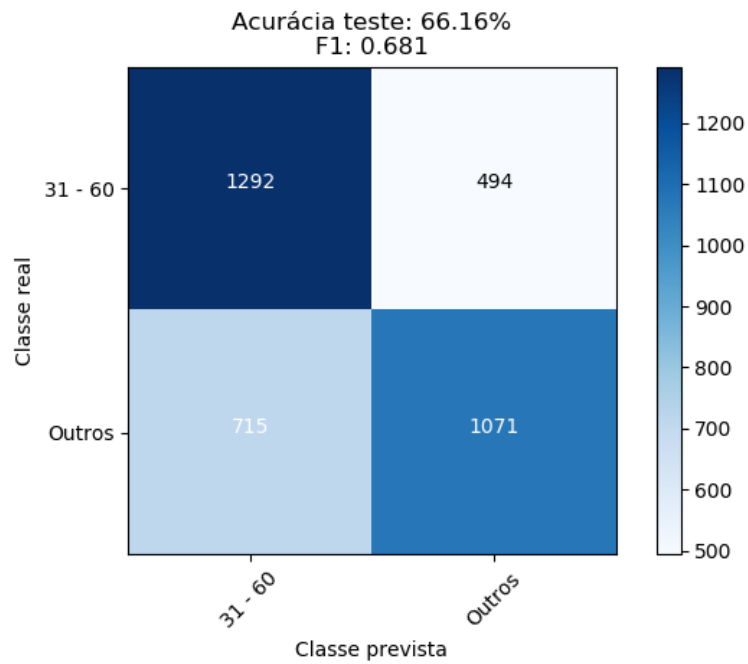
**Figura 5.2:** Matriz de confusão da classe 1-7 dias



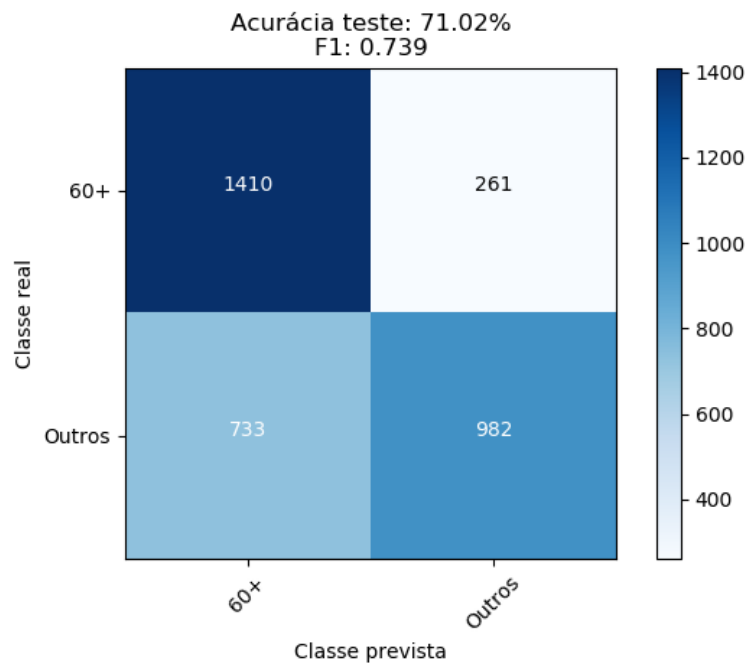
**Figura 5.3:** Matriz de confusão da classe 8-15 dias



**Figura 5.4:** Matriz de confusão da classe 16-30 dias



**Figura 5.5:** *Matriz de confusão da classe 31-60 dias*



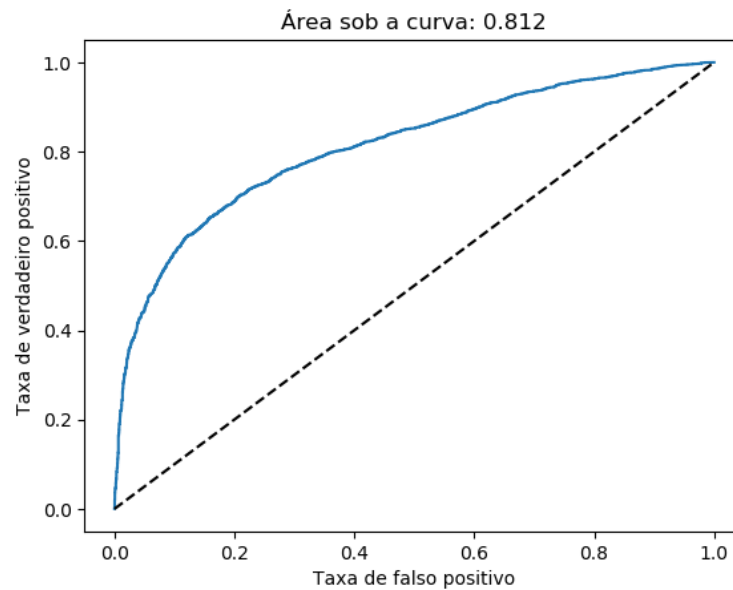
**Figura 5.6:** *Matriz de confusão da classe mais de 60 dias*

Sendo a matriz de confusão uma representação gráfica dos valores previamente exibidos na tabela 5.1. A classe 1-7 dias foi a única a acertar mais casos que não pertencem na classe dos que os casos que pertencem a classe: 2499 exemplos não pertencentes a classe 1-7 dias foram previstos corretamente e 2166 exemplos pertencentes a classe 1-7 dias também foram previstos corretamente. Em todas as outras classes testemunha-se o

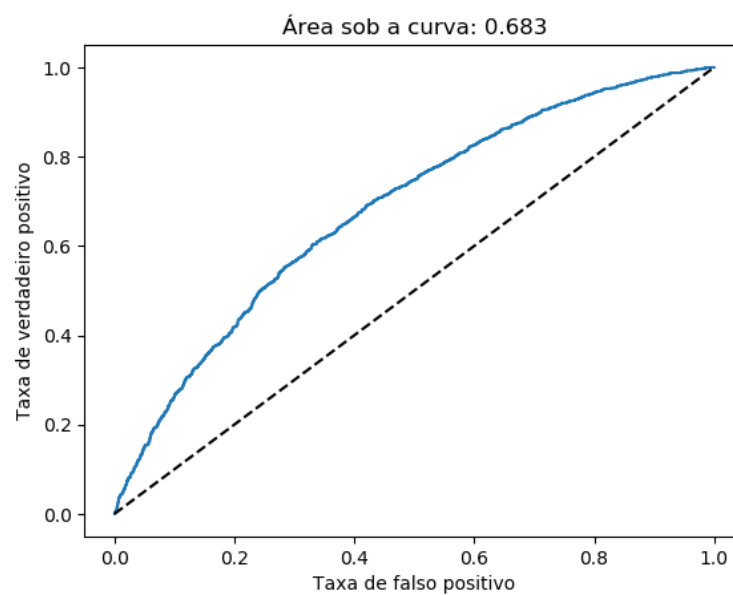
caso oposto: o acerto em exemplos pertencentes as classes é maior do que o acerto em exemplos não pertencentes as classes.

### 5.2.2 Resultado da área sob a curva

As figuras 5.7, 5.8, 5.9, 5.10 e 5.11 representam os gráficos da curva COR das classes 1-7 dias, 8-15 dias, 16-30 dias, 31-60 dias, e mais de 60 dias, respectivamente:

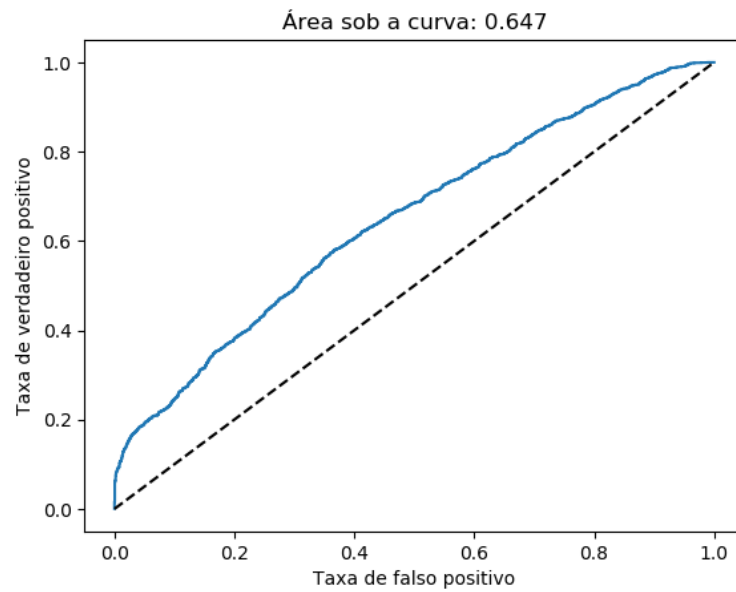


**Figura 5.7:** Área sob a curva da classe 1-7

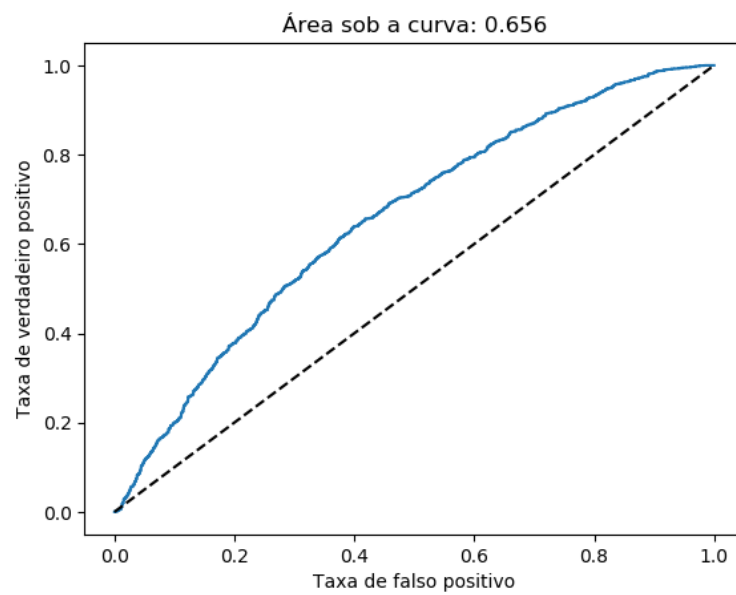


**Figura 5.8:** Área sob a curva da classe 8-15

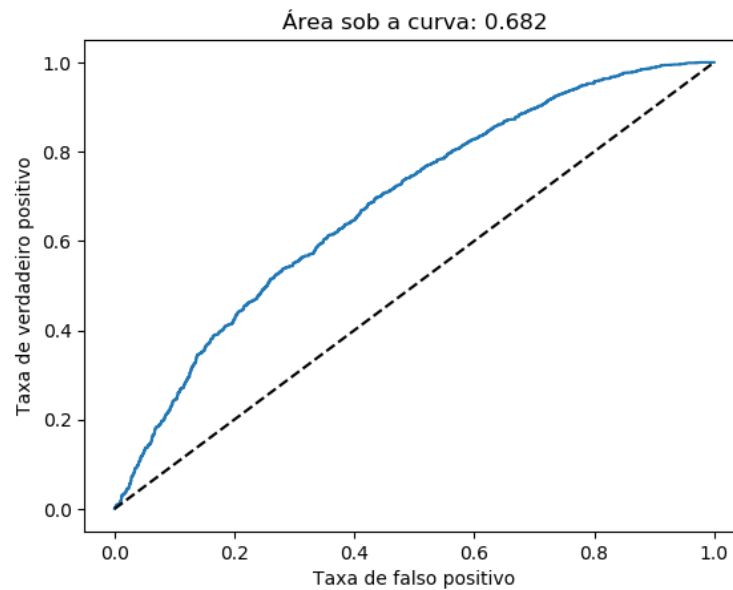




**Figura 5.9:** Área sob a curva da classe 16-30



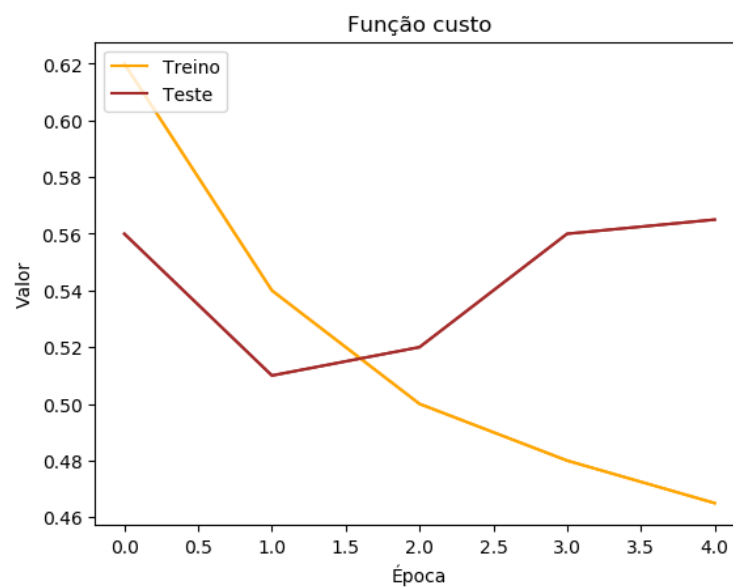
**Figura 5.10:** Área sob a curva da classe 31-60



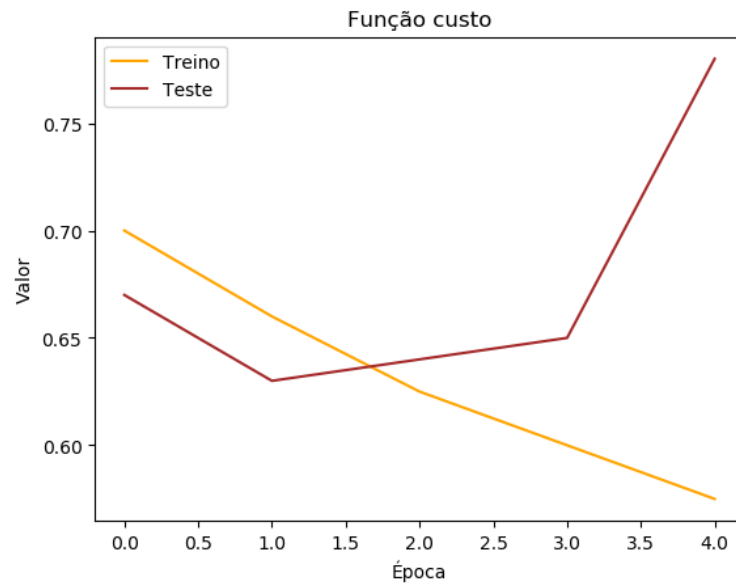
**Figura 5.11:** Área sob a curva da classe mais de 60 dias

### 5.2.3 Função de custo

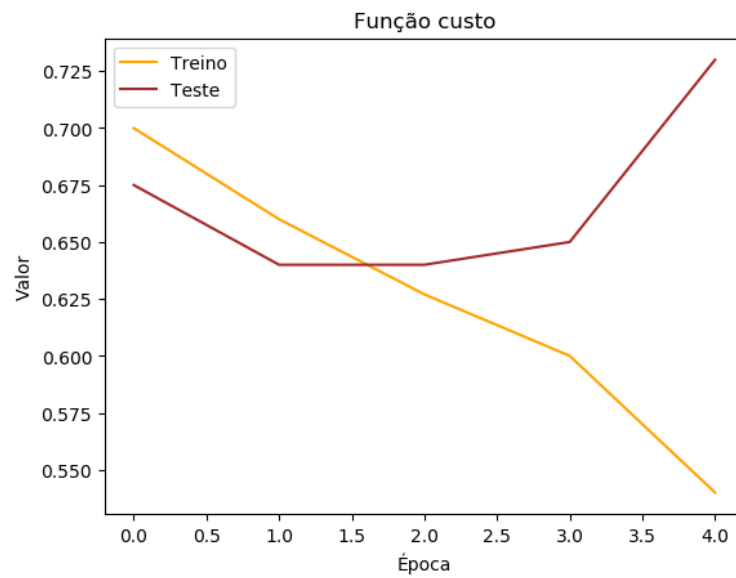
As figuras 5.12, 5.13, 5.14, 5.15 e 5.16 representam os gráficos da função de custo das classes 1-7 dias, 8-15 dias, 16-30 dias, 31-60 dias, e mais de 60 dias, respectivamente:



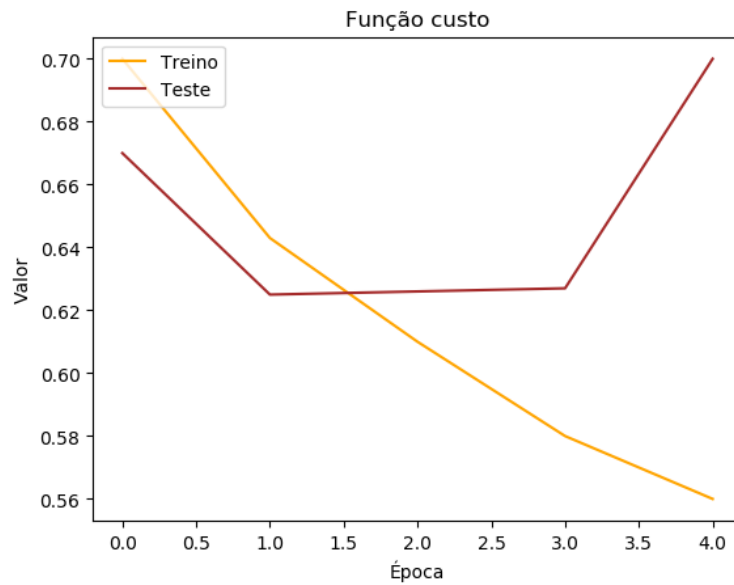
**Figura 5.12:** Função de custo da classe 1-7



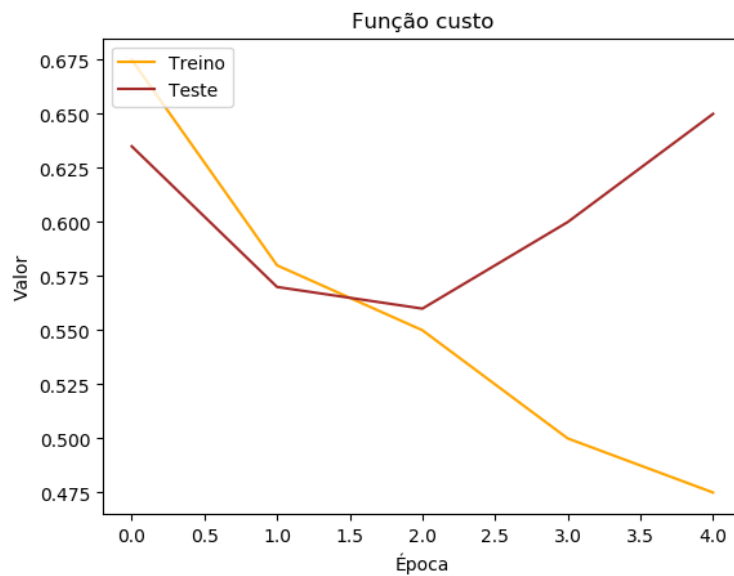
**Figura 5.13:** Função de custo da classe 8-15



**Figura 5.14:** Função de custo da classe 16-30



**Figura 5.15:** Função de custo da classe 31-60

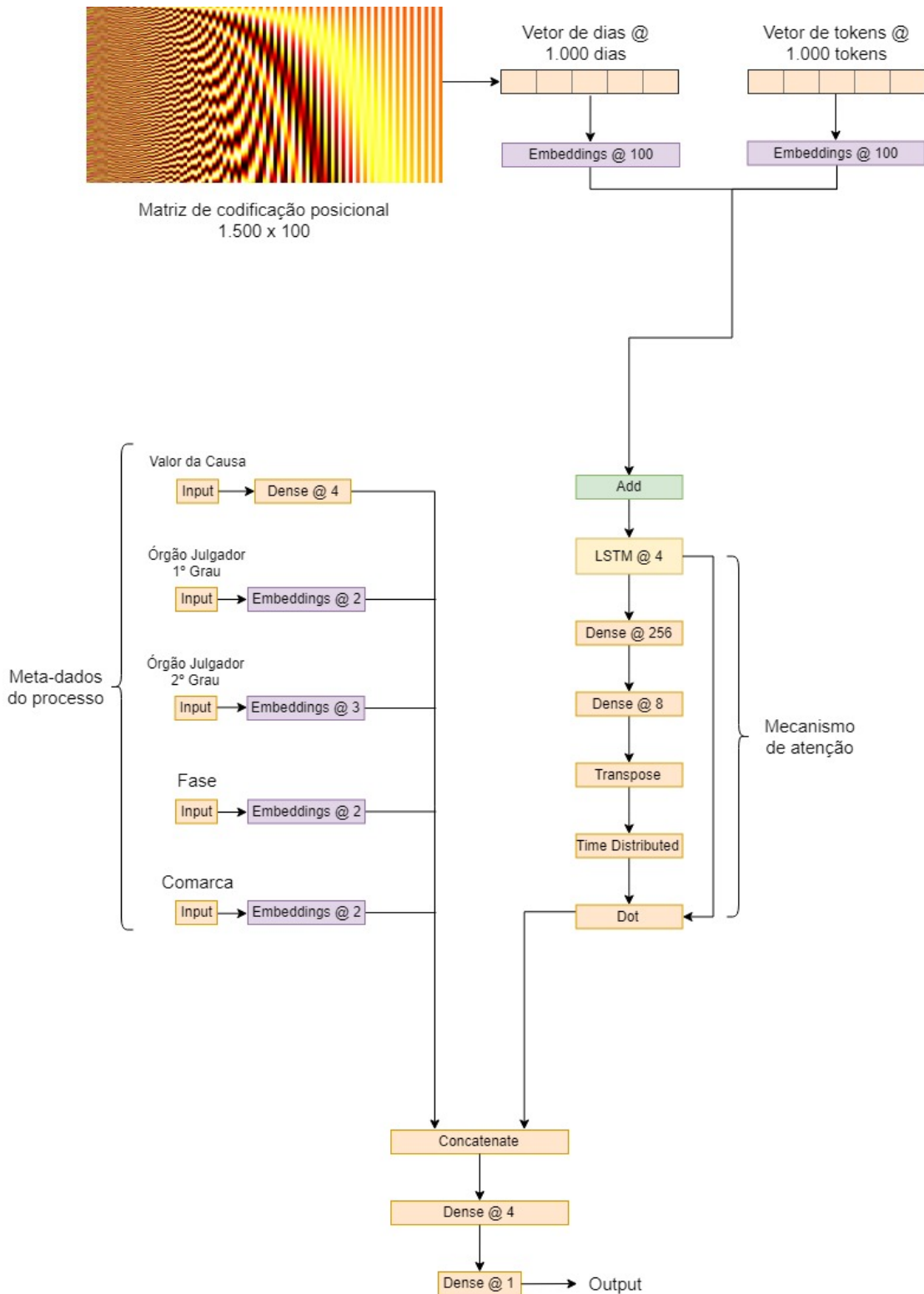


**Figura 5.16:** Função de custo da classe 60+

De acordo com as imagens da função de custo, fica visível a situação de *overfit* que aflige o modelo treinado. Os modelos apresentam um comportamento persistente onde a função custo de teste diminui o valor inicialmente, porém com algumas poucas épocas passa a aumentar esse valor, até mesmo ultrapassando o valor inicial na primeira época, enquanto o valor da função de custo no treino diminui durante todo o processo. Tal comportamento é visível independente da classe a ser treinada no modelo.

### 5.2.4 Versão reduzida do modelo

Uma segunda versão do modelo fora construída, como uma forma mais enxuta da arquitetura exibida em 4.2. A Figura 5.17 exemplifica a versão reduzida do modelo originalmente proposto:



**Figura 5.17:** Ilustração da versão enxuta do modelo proposto

A versão enxuta do modelo original é o resultado de uma série de modificações:

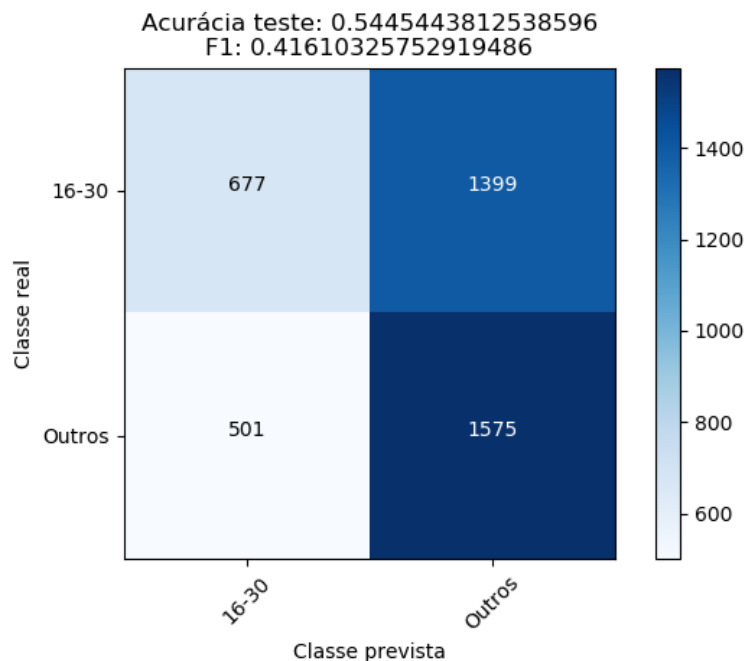
- Remoção de uma camada do tipo *LSTM*
- Remoção de cinco camadas de neurônios densamente conectados
- Redução na quantidade de neurônios nas camadas densamente conectadas restantes

Utilizando os mesmos dados e parâmetros relatados na seção 5.2, um novo treino foi executado. A tabela 5.2 apresenta os resultados novamente avaliados pelas métricas definidas em 5.1, e obtidos com o treino dos dados da classe 16-30 dias utilizando o modelo de configuração reduzida:

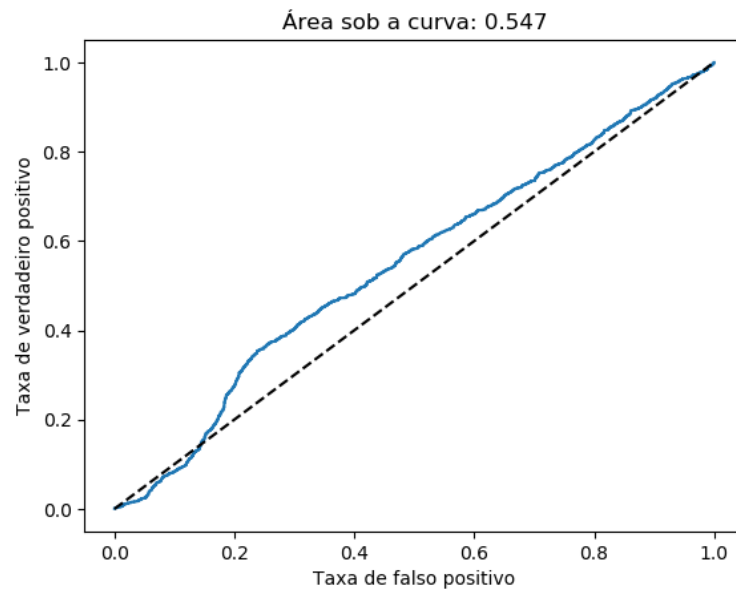
Métricas	Máximo	Mínimo	Média	Desvio padrão
Acurácia de treino	75.37%	74.08%	74.47%	0.45%
$F_1$ score de treino	0.737	0.718	0.727	0.007
Acurácia de teste	54.45%	51.28%	53.46%	0.98%
$F_1$ score de teste	0.606	0.557	0.595	0.016
Área sob a curva	0.592	0.517	0.532	0.023

**Tabela 5.2:** Resultados obtidos pelo treino da classe 16-30 dias

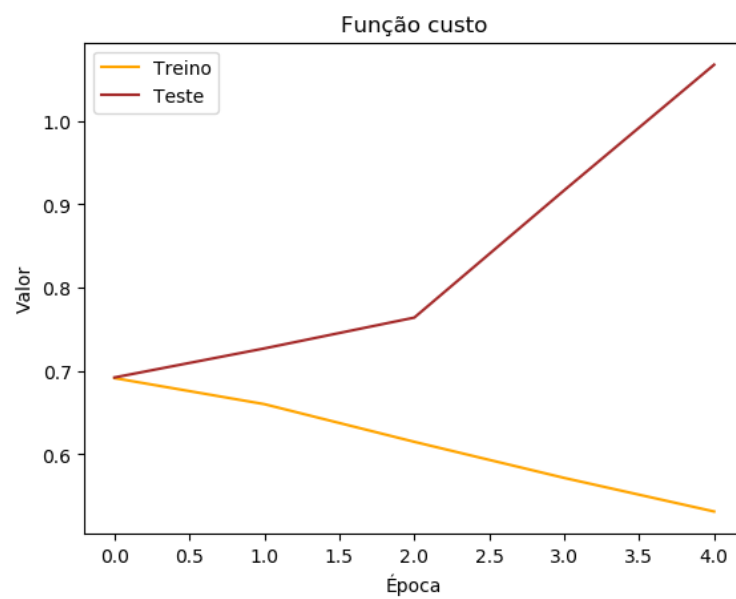
As figuras 5.18, 5.19 e 5.20 demonstram a matriz de confusão, o gráfico da curva COR e a função de custo dos resultados alcançados, respectivamente:



**Figura 5.18:** Matriz de confusão da classe 16-30 dias



**Figura 5.19:** Área sob a curva da classe 16-30 dias



**Figura 5.20:** Função de custo da classe 16-30 dias

---

## Conclusão

---

Neste capítulo serão apresentadas as conclusões a partir dos resultados obtidos. A seção 6.1 discorre sobre as conclusões gerais observadas a partir dos resultados obtidos, enquanto a seção 6.2 apresenta a relação de sugestões de trabalhos futuros possíveis de serem realizados com o intuito de melhorar os resultados obtidos.

### 6.1 Conclusões gerais

Em relação aos períodos de dias selecionados como classes dos modelos preditivos, é visível maior facilidade da arquitetura para períodos em específico: as classes de 1-7 dias e de mais de 60 dias foram as que obtiveram resultados acima de 70% de acurácia de teste (76.19% e 71.02%, aproximadamente). Por serem classes situadas nos extremos dos períodos propostos, e serem tão díspares entre si e entre as classes definidas, as características de tais períodos os tornam mais evidentes em relação ao todo do que as outras classes, fazendo com que os modelos específicos desses períodos tenham uma performance melhor. Em suma, processos que apresentam um estado onde a próxima movimentação irá demorar menos de uma semana ou mais de dois meses são mais evidentes do que outros processos, apresentando características de comportamento que os tornam mais óbvios em relação a próxima movimentação que o restante.

Em acordo com o entendimento do desempenho das classes mais ao extremo, as classes de 8-15 dias e 30-60 dias, além de serem as classes imediatamente vizinhas as classes de extremo, são as classes que apresentam os valores de acurácia de teste abaixo das classes de extremo. Em suma, enquanto as classes mais ao extremo (ou seja, as classes de 1-7 dias e mais de 60 dias) são as que apresentam o primeiro e segundo melhor resultado dos treinos executados, as classes imediatamente após as classes de extremo (ou seja, as classes de 8-15 dias e de 30-60 dias) apresentam o terceiro e quarto melhor resultado do treino: 66.62% e 66.16% de acurácia de teste, respectivamente. Por último, a classe mais ao centro dos valores propostos também é a classe de pior performance resultada: a classe 16-30 dias apresenta 62.07% de acurácia de teste.



Por outro lado, os resultados obtidos em 5.2.4 com o treino do modelo reduzido apresentam que, mesmo com uma remoção substancial de camadas e neurônios envolvidos, houve uma redução de apenas 10% em aproximado nos valores das métricas utilizadas para avaliação de desempenho. Não obstante, o comportamento de *overfit* presente no treino do modelo original também é perceptível após o treino do modelo reduzido, como é possível na comparação entre os resultados do treino com a classe 16-30 dias nas tabelas 5.1 e 5.2. A presença do comportamento de *overfit* mesmo após a redução expressiva de neurônios e de camadas na rede é um indicativo de deficiência quanto a representação dos dados como entrada nos modelos treinados. Haja visto que a representação dos dados permaneceu a mesma tanto no modelo proposto quanto na versão reduzida, e a mudança ocorreu na diminuição do número de camadas e na quantidade de neurônios em algumas camadas, uma possível explicação para tal resultado é que o empecilho esteja justamente na maneira como os dados estão representados, e não na arquitetura presente após os *inputs* designados.

Apesar dos contratempos e das limitações nos resultados obtidos, é visível a possibilidade de melhoria no modelo proposto. Não obstante, o modelo conseguiu obter uma predição de razoável sucesso em nível de acurácia, delimitando uma base promissora para evolução em melhores modelos preditivos e estabelecendo um mínimo de precisão que, dado o ineditismo de tal tipo de ferramenta no contexto jurídico, apresenta um protagonismo necessário para a realidade da área. Os resultados também demonstram uma capacidade do modelo de entender e interpretar parcialmente as entrelinhas dos processos trabalhistas e suas movimentações, indicando uma base sólida no que tange em abstrair as diferentes relações que os processos trabalhistas apresentam em suas movimentações.

Com isso, verifica-se um progresso efetivo do modelo neural em entender como a evolução histórica de um processo trabalhista influencia no seu culminar, considerando como os diferentes agentes envolvidos se fazem presentes nesse procedimento. Ao abarcar as relações entre entidades tão díspares entre si, como advogados, partes, comarcas, órgãos julgadores, valores de causa, dentre outros, o modelo proposto demonstra um escrutínio baseado em uma complexidade inerente ao domínio jurídico da esfera trabalhista, ainda que com possibilidade de maior avanço futuro.

## 6.2 Trabalhos futuros

Diante dos resultados obtidos, há uma plethora de possibilidades para a conquista de melhores resultados. Uma das alternativas de melhoria poderá ser o uso de mais dados no momento do treino. Atualmente estão disponíveis 686.549 processos da esfera trabalhista para treino, contendo movimentações referentes ao ano de 2015 no TRT 15 e presentes em uma única vara. Dado o baixo número de processos envolvidos no treino

até o momento (114.988 movimentações oriundas de 2.554 processos - 0.37% do total), espera-se que haja uma melhora expressiva nos resultados obtidos e uma diminuição no erro do modelo treinado.

Uma possível melhoria na acurácia dos modelos seria uma decisão de escolha de quais movimentações deveriam ser previstas. Atualmente, todas as movimentações indiscriminadamente são selecionadas como objeto de predição, porém com a seleção de tipos de movimentações em específico - como arquivamento, recurso ordinário, recurso de revista, agravo de instrumento em recurso de revista, embargo, decisão, acórdão ou trânsito em julgado - é possível uma melhoria nos resultados de acurácia obtidos, dado o universo mais preciso de movimentações a serem utilizadas como objeto de precisão.

Outra possibilidade seria a possibilidade de substituição de outras entidades por tokens, expandindo a atividade feita em 4.3 para outras entidades presentes nos textos das descrições das movimentações, como pessoas, organizações ou locais, dentre outros. [20] é uma possibilidade de uso de extração de entidades na língua portuguesa, facilitando o treino de extração de entidades nas descrições das movimentações, e permitindo a substituição de entidades encontradas por tokens em específico, como uma forma de diminuir a quantidade de *embeddings* envolvidos e melhorando o conteúdo dos *embeddings* utilizados.

No trabalho apresentado, a única informação textual utilizada fora a descrição das movimentações trabalhistas. É possível utilizar também outras informações textuais como a descrição dos documentos das movimentações, a descrição do anexo das movimentações e o corpo dos anexos das movimentações trabalhistas. Por um lado, a inserção de mais informação textual possibilitará os modelos a serem expostos a uma quantidade maior de informações. Por outro lado, o aumento de tokens inseridos no modelo exigirá uma mudança na arquitetura em relação a quantidade de tokens inseridos, haja visto que a arquitetura proposta está direcionada para descrições de movimentações, que comumente são apenas algumas frases, ao passo que textos de anexos de movimentações geralmente são compostas de páginas de textos, aumentando consideravelmente o tamanho das informações textuais a serem inseridas.

Como descrito na seção 4.2, este trabalho utilizou representações via tokens e *embeddings* em conjunto com camadas *LSTM* para simbolizar as informações de movimentações dentro do cálculo da arquitetura da rede neural, em contraste de outras arquiteturas como ELMo [67] [16] [23], BERT [21] e ULMFiT [41]. Dado que tais arquiteturas apresentam performance e resultado mais promissores do que arquiteturas mais convencionais como as utilizadas no trabalho proposto, uma possibilidade de trabalho futuro é a substituição do uso de *embeddings* e *LSTM* por outra arquitetura com melhores resultados. Além disso, pelo fato do uso de *embeddings* do tipo *GloVe* terem sido utilizados no trabalho, é possível também a substituição por *embeddings* oriundos de outros algoritmos,

como *word2Vec* [58] [31], *fastText* [15] [45] [44] e *wang2vec* [52], dentre outros.

É possível expandir a quantidade de informações utilizadas no treino dos modelos. Por exemplo, comparando as características utilizadas na seção 4.5.7 com todas as características disponíveis para uso descritas na seção 2.6, é perceptível que existem uma plethora de características disponíveis para uso no treino, permitindo oportunidades de experimentação entre as características disponíveis para inserção nos modelos durante o treino. Levando em consideração o fato de que os modelos treinados foram alimentados com processos do TRT da 15ª região do ano de 2015 e de uma única vara, também é possível avançar o treino para outras varas do mesmo TRT, bem como outros TRTs e com outros anos.

Mais especificadamente em relação ao ano, é importante notar os impactos das mudanças causadas pela reforma trabalhista em 2017 [6]. Somente nos 6 primeiros meses após a aprovação da reforma, observou-se uma redução de 40% no número do ingresso de novas ações trabalhistas [25], bem como uma redução de 36% no número de ações trabalhistas ajuizadas no ano de 2018 em relação ao ano de 2017, onde apenas no TRT da 15ª região constatou-se uma queda de 32% no número de processos ajuizados [19].

Por fim, tendo em mente as considerações levantadas na seção 6.1 referentes aos possíveis problemas encontrados na representação dos dados, também há uma possibilidade de melhoria envolvendo a busca de uma maneira alternativa de se representar os dados presentes.

---

## Referências Bibliográficas

---

- [1] **Google developers blog: Introducing tensorflow feature columns.** <https://developers.googleblog.com/2017/11/introducing-tensorflow-feature-columns.html>. "Acessado em 17/07/2019". 42
- [2] **Justiça em números 2018.** <http://www.cnj.jus.br/files/conteudo/arquivo/2018/08/44b-7368ec6f888b383f6c3de40c32167.pdf>. "Acessado em 02/09/2018". 23
- [3] **Pje wiki.** [http://www.pje.jus.br/wiki/index.php/Página\\_principal](http://www.pje.jus.br/wiki/index.php/Página_principal). "Acessado em 02/09/2018". 29
- [4] **Portuguese word embeddings: Evaluating on word analogies and natural language tasks - preprocessing.py.** [https://github.com/nathanshartmann/portuguese\\_word\\_embeddings/blob/master/-preprocessing.py](https://github.com/nathanshartmann/portuguese_word_embeddings/blob/master/-preprocessing.py). "Acessado em 20/10/2018". 33
- [5] **Processo judicial eletrônico (pje) - portal cnj.** <http://www.cnj.jus.br/tecnologia-da-informacao/processo-judicial-eletronico-pje>. "Acessado em 02/09/2018". 29
- [6] ALESSI, D.; OTHERS. **Relações de trabalho docente no brasil à luz do neoliberalismo: Uma análise da reforma trabalhista de 2017.** 2019. 66
- [7] ALETRAS, N.; TSARAPATSANIS, D.; PREOȚIUC-PIETRO, D.; LAMPOS, V. **Predicting judicial decisions of the european court of human rights: A natural language processing perspective.** *PeerJ Computer Science*, 2:e93, 2016. 31
- [8] ALIKANIOTIS, D.; YANNAKOUDAKIS, H.; REI, M. **Automatic text scoring using neural networks.** *arXiv preprint arXiv:1606.04289*, 2016. 17
- [9] BAHDANAU, D.; CHO, K.; BENGIO, Y. **Neural machine translation by jointly learning to align and translate.** *arXiv preprint arXiv:1409.0473*, 2014. 38
- [10] BASILE, C. R. O. **Processo do trabalho: justiça do trabalho e dissídios trabalhistas.** Saraiva, 2012. 21, 22

- [11] BATCHU, V.; BATTU, V.; REDDY, M. K.; MAMIDI, R. **"how to rate a video game?- a prediction system for video games based on multimodal information.** *arXiv preprint arXiv:1805.11372*, 2018. 17
- [12] BISHOP, C. M. **Pattern recognition and machine learning.** springer, 2006. 37, 39
- [13] BITVAI, Z.; COHN, T. **Non-linear text regression with a deep convolutional neural network.** In: *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, volume 2, p. 180–185, 2015. 17
- [14] BOELLA, G.; DI CARO, L.; HUMPHREYS, L. **Using classification to support legal knowledge engineers in the eunomos legal document management system.** In: *Fifth international workshop on juris-informatics (jurisin)*, 2011. 31
- [15] BOJANOWSKI, P.; GRAVE, E.; JOULIN, A.; MIKOLOV, T. **Enriching word vectors with subword information.** *Transactions of the Association for Computational Linguistics*, 5:135–146, 2017. 66
- [16] CHE, W.; LIU, Y.; WANG, Y.; ZHENG, B.; LIU, T. **Towards better UD parsing: Deep contextualized word embeddings, ensemble, and treebank concatenation.** In: *Proceedings of the CoNLL 2018 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies*, p. 55–64, Brussels, Belgium, October 2018. Association for Computational Linguistics. 65
- [17] CIREŞAN, D.; MEIER, U.; SCHMIDHUBER, J. **Multi-column Deep Neural Networks for Image Classification.** *ArXiv e-prints*, Feb. 2012. 17
- [18] CONNEAU, A.; SCHWENK, H.; BARRAULT, L.; LECUN, Y. **Very deep convolutional networks for text classification.** *arXiv preprint arXiv:1606.01781*, 2016. 17
- [19] DE CARVALHO, A. X. **A reforma trabalhista e o mito da geração de empregos.** *CARTA SOCIAL*, p. 12, 2018. 66
- [20] DE CASTRO, P. V. Q.; DA SILVA, N. F. F.; DA SILVA SOARES, A. **Portuguese named entity recognition using lstm-crf.** In: *International Conference on Computational Processing of the Portuguese Language*, p. 83–92. Springer, 2018. 65
- [21] DEVLIN, J.; CHANG, M.-W.; LEE, K.; TOUTANOVA, K. **Bert: Pre-training of deep bidirectional transformers for language understanding.** *arXiv preprint arXiv:1810.04805*, 2018. 65

- [22] DUCHI, J.; HAZAN, E.; SINGER, Y. **Adaptive subgradient methods for online learning and stochastic optimization.** *Journal of Machine Learning Research*, 12(Jul):2121–2159, 2011. 41
- [23] FARES, M.; KUTUZOV, A.; OEPEN, S.; VELLDAL, E. **Word vectors, reuse, and replicability: Towards a community repository of large-text resources.** In: *Proceedings of the 21st Nordic Conference on Computational Linguistics*, p. 271–276, Gothenburg, Sweden, May 2017. Association for Computational Linguistics. 65
- [24] FAWCETT, T. **An introduction to roc analysis.** *Pattern recognition letters*, 27(8):861–874, 2006. 47, 48
- [25] FERNANDES, C. P.; SILVA, J. B.; GUILHERME, G. S. **Reforma trabalhista brasileira e suas consequências na atualidade.** *Rede Iberoamericana de Pesquisa em Seguridade Social*, 1(1):380–397, 2019. 66
- [26] FOR GUIDES IN METROLOGY, J. C. **International vocabulary of metrology—basic and general concepts and associated terms (vim)**, 2008. 46
- [27] GAL, Y.; GHARAMANI, Z. **A theoretically grounded application of dropout in recurrent neural networks.** In: *Advances in neural information processing systems*, p. 1019–1027, 2016. 36
- [28] GARCIA, G. F. B. **Introdução ao Estudo de Direito.** Editora Método LTDA, 2015. 20
- [29] GEHRING, J.; AULI, M.; GRANGIER, D.; YARATS, D.; DAUPHIN, Y. N. **Convolutional sequence to sequence learning.** In: *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, p. 1243–1252. JMLR. org, 2017. 40
- [30] GOH, G. B.; HODAS, N. O.; VISHNU, A. **Deep learning for computational chemistry.** *Journal of computational chemistry*, 38(16):1291–1307, 2017. 37
- [31] GOLDBERG, Y.; LEVY, O. **word2vec explained: deriving mikolov et al.’s negative-sampling word-embedding method.** *arXiv preprint arXiv:1402.3722*, 2014. 66
- [32] GOODFELLOW, I.; BENGIO, Y.; COURVILLE, A. **Deep learning.** MIT press, 2016. 39
- [33] GRAY, S.; RADFORD, A.; KINGMA, D. P. **Gpu kernels for block-sparse weights.** Technical report, Technical report, OpenAI, 2017. 17
- [34] HAN, J.; MORAGA, C. **The influence of the sigmoid function parameters on the speed of backpropagation learning.** In: *International Workshop on Artificial Neural Networks*, p. 195–201. Springer, 1995. 38

- [35] HAND, D. J.; TILL, R. J. **A simple generalisation of the area under the roc curve for multiple class classification problems.** *Machine learning*, 45(2):171–186, 2001. 49
- [36] HANLEY, J. A.; MCNEIL, B. J. **The meaning and use of the area under a receiver operating characteristic (roc) curve.** *Radiology*, 143(1):29–36, 1982. 49
- [37] HARTMANN, N.; FONSECA, E.; SHULBY, C.; TREVISIO, M.; RODRIGUES, J.; ALUISIO, S. **Portuguese word embeddings: Evaluating on word analogies and natural language tasks.** *arXiv preprint arXiv:1708.06025*, 2017. 33, 34
- [38] HOCHREITER, S. **Untersuchungen zu dynamischen neuronalen netzen.** *Diploma, Technische Universität München*, 91(1), 1991. 37
- [39] HOCHREITER, S.; BENGIO, Y.; FRASCONI, P.; SCHMIDHUBER, J.; OTHERS. **Gradient flow in recurrent nets: the difficulty of learning long-term dependencies**, 2001. 37
- [40] HOCHREITER, S.; SCHMIDHUBER, J. **Long short-term memory.** *Neural computation*, 9(8):1735–1780, 1997. 36
- [41] HOWARD, J.; RUDER, S. **Universal language model fine-tuning for text classification.** In: *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, volume 1, p. 328–339, 2018. 17, 65
- [42] JOHNSON, R.; ZHANG, T. **Supervised and semi-supervised text categorization using lstm for region embeddings.** *arXiv preprint arXiv:1602.02373*, 2016. 17
- [43] JOHNSON, R.; ZHANG, T. **Deep pyramid convolutional neural networks for text categorization.** In: *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, volume 1, p. 562–570, 2017. 17
- [44] JOULIN, A.; GRAVE, E.; BOJANOWSKI, P.; DOUZE, M.; JÉGOU, H.; MIKOLOV, T. **Fasttext.zip: Compressing text classification models.** *arXiv preprint arXiv:1612.03651*, 2016. 66
- [45] JOULIN, A.; GRAVE, E.; BOJANOWSKI, P.; MIKOLOV, T. **Bag of tricks for efficient text classification.** In: *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*, p. 427–431. Association for Computational Linguistics, April 2017. 66
- [46] KATZ, D. M.; BOMMARITO II, M. J.; BLACKMAN, J. **A general approach for predicting the behavior of the supreme court of the united states.** *PloS one*, 12(4):e0174698, 2017. 31

- [47] KAWAKAMI, K. **Supervised sequence labelling with recurrent neural networks.** PhD thesis, Ph. D. thesis, Technical University of Munich, 2008. 36
- [48] KIM, Y.; DENTON, C.; HOANG, L.; RUSH, A. M. **Structured attention networks.** *arXiv preprint arXiv:1702.00887*, 2017. 38
- [49] KINGMA, D. P.; BA, J. **Adam: A method for stochastic optimization.** *arXiv preprint arXiv:1412.6980*, 2014. 41
- [50] LECUN, Y.; CORTES, C.; BURGES, C. **Mnist handwritten digit database.** *AT&T Labs [Online]. Available: <http://yann.lecun.com/exdb/mnist>*, 2, 2010. 17
- [51] LEHMANN, J.; ISELE, R.; JAKOB, M.; JENTZSCH, A.; KONTOKOSTAS, D.; MENDES, P. N.; HELLMANN, S.; MORSEY, M.; VAN KLEEF, P.; AUER, S.; OTHERS. **Dbpedia—a large-scale, multilingual knowledge base extracted from wikipedia.** *Semantic Web*, 6(2):167–195, 2015. 17
- [52] LING, W.; DYER, C.; BLACK, A.; TRANCOSO, I. **Two/too simple adaptations of word2vec for syntax problems.** In: *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies.* Association for Computational Linguistics, 2015. 66
- [53] LONG, S.; TU, C.; LIU, Z.; SUN, M. **Automatic judgment prediction via legal reading comprehension**, 2018. 31
- [54] MAAS, A. L.; DALY, R. E.; PHAM, P. T.; HUANG, D.; NG, A. Y.; POTTS, C. **Learning word vectors for sentiment analysis.** In: *Proceedings of the 49th annual meeting of the association for computational linguistics: Human language technologies-volume 1*, p. 142–150. Association for Computational Linguistics, 2011. 17
- [55] MASON, S. J.; GRAHAM, N. E. **Areas beneath the relative operating characteristics (roc) and relative operating levels (rol) curves: Statistical significance and interpretation.** *Quarterly Journal of the Royal Meteorological Society: A journal of the atmospheric sciences, applied meteorology and physical oceanography*, 128(584):2145–2166, 2002. 49
- [56] MCCANN, B.; BRADBURY, J.; XIONG, C.; SOCHER, R. **Learned in translation: Contextualized word vectors.** In: *Advances in Neural Information Processing Systems*, p. 6294–6305, 2017. 17
- [57] MCCLISH, D. K. **Analyzing a portion of the roc curve.** *Medical Decision Making*, 9(3):190–195, 1989. 49



- [58] MIKOLOV, T.; CHEN, K.; CORRADO, G.; DEAN, J. **Efficient estimation of word representations in vector space.** *arXiv preprint arXiv:1301.3781*, 2013. 66
- [59] MIYATO, T.; DAI, A. M.; GOODFELLOW, I. **Adversarial training methods for semi-supervised text classification.** *arXiv preprint arXiv:1605.07725*, 2016. 17
- [60] MURPHY, K. P. **Machine learning: a probabilistic perspective.** MIT press, 2012. 37
- [61] OGHINA, A.; BREUSS, M.; TSAGKIAS, M.; DE RIJKE, M. **Predicting imdb movie ratings using social media.** In: *European Conference on Information Retrieval*, p. 503–507. Springer, 2012. 17
- [62] PALAU, R. M.; MOENS, M.-F. **Argumentation mining: the detection, classification and structure of arguments in text.** In: *Proceedings of the 12th international conference on artificial intelligence and law*, p. 98–107. ACM, 2009. 31
- [63] PARIKH, A. P.; TÄCKSTRÖM, O.; DAS, D.; USZKOREIT, J. **A decomposable attention model for natural language inference.** *arXiv preprint arXiv:1606.01933*, 2016. 38
- [64] PASCANU, R.; MIKOLOV, T.; BENGIO, Y. **On the difficulty of training recurrent neural networks.** In: *International conference on machine learning*, p. 1310–1318, 2013. 37
- [65] PEDREGOSA, F.; VAROQUAUX, G.; GRAMFORT, A.; MICHEL, V.; THIRION, B.; GRISSEL, O.; BLONDEL, M.; PRETTENHOFER, P.; WEISS, R.; DUBOURG, V.; OTHERS. **Scikit-learn: Machine learning in python.** *Journal of machine learning research*, 12(Oct):2825–2830, 2011. 36
- [66] PENNINGTON, J.; SOCHER, R.; MANNING, C. **Glove: Global vectors for word representation.** In: *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, p. 1532–1543, 2014. 34
- [67] PETERS, M. E.; NEUMANN, M.; IYYER, M.; GARDNER, M.; CLARK, C.; LEE, K.; ZETTMLOYER, L. **Deep contextualized word representations.** *arXiv preprint arXiv:1802.05365*, 2018. 65
- [68] PORIA, S.; CAMBRIA, E.; GELBUKH, A. **Deep convolutional neural network textual features and multiple kernel learning for utterance-level multimodal sentiment analysis.** In: *Proceedings of the 2015 conference on empirical methods in natural language processing*, p. 2539–2544, 2015. 17
- [69] POWERS, D. M. **Evaluation: from precision, recall and f-measure to roc, informedness, markedness and correlation.** 2011. 47, 49

- [70] POWERS, D. M. **Roc-concert: Roc-based measurement of consistency and certainty.** In: *2012 Spring Congress on Engineering and Technology*, p. 238–241. IEEE, 2012. 49
- [71] RADFORD, A.; NARASIMHAN, K.; SALIMANS, T.; SUTSKEVER, I. **Improving language understanding by generative pre-training.** URL [https://s3-us-west-2.amazonaws.com/openai-assets/research-covers/language-unsupervised/language\\_understanding\\_paper.pdf](https://s3-us-west-2.amazonaws.com/openai-assets/research-covers/language-unsupervised/language_understanding_paper.pdf), 2018. 34, 35
- [72] REALE, M. **Lições Preliminares de Direito. 2.ª tiragem.** 2001. 20
- [73] REDDI, S. J.; KALE, S.; KUMAR, S. **On the convergence of adam and beyond.** 2018. 41
- [74] ROCKTÄSCHEL, T.; GREFENSTETTE, E.; HERMANN, K. M.; KOČISKÝ, T.; BLUNSON, P. **Reasoning about entailment with neural attention.** *arXiv preprint arXiv:1509.06664*, 2015. 35
- [75] RUMELHART, D. E.; HINTON, G. E.; WILLIAMS, R. J.; OTHERS. **Learning representations by back-propagating errors.** *Cognitive modeling*, 5(3):1, 1988. 36
- [76] SAMPAIO, T.; JR., F. **Introdução ao estudo do Direito.** Grupo Gen-Atlas, 2018. 20
- [77] SCHIAVI, M. **Manual de direito processual do trabalho.** LTr, 2016. 21
- [78] SHARMA, L.; GRAESSER, L.; NANGIA, N.; EVCI, U. **Natural language understanding with the quora question pairs dataset.** *arXiv preprint arXiv:1907.01041*, 2019. 18
- [79] SMITH, S. L.; KINDERMANS, P.-J.; YING, C.; LE, Q. V. **Don't decay the learning rate, increase the batch size.** *arXiv preprint arXiv:1711.00489*, 2017. 41
- [80] SUBRAMANIAN, S.; BALDWIN, T.; COHN, T. **Content-based popularity prediction of online petitions using a deep regression model.** *arXiv preprint arXiv:1805.06566*, 2018. 17
- [81] SULEA, O.-M.; ZAMPIERI, M.; MALMASI, S.; VELA, M.; DINU, L. P.; VAN GENABITH, J. **Exploring the use of text classification in the legal domain.** *arXiv preprint arXiv:1710.09306*, 2017. 17, 31
- [82] SULEA, O.-M.; ZAMPIERI, M.; VELA, M.; VAN GENABITH, J. **Predicting the law area and decisions of french supreme court cases.** *arXiv preprint arXiv:1708.01681*, 2017. 31

- [83] TAGHIPOUR, K.; NG, H. T. **A neural approach to automated essay scoring.** In: *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, p. 1882–1891, 2016. 17
- [84] TIELEMAN, T.; HINTON, G. **Lecture 6.5-rmsprop: Divide the gradient by a running average of its recent magnitude.** *COURSERA: Neural networks for machine learning*, 4(2):26–31, 2012. 41
- [85] VASWANI, A.; SHAZEER, N.; PARMAR, N.; USZKOREIT, J.; JONES, L.; GOMEZ, A. N.; KAISER, Ł.; POLOSUKHIN, I. **Attention is all you need.** In: *Advances in neural information processing systems*, p. 5998–6008, 2017. 38, 39
- [86] WANG, J.; YU, L.-C.; LAI, K. R.; ZHANG, X. **Dimensional sentiment analysis using a regional cnn-lstm model.** In: *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, volume 2, p. 225–230, 2016. 17
- [87] ZHANG, X.; ZHAO, J.; LECUN, Y. **Character-level convolutional networks for text classification.** In: *Advances in neural information processing systems*, p. 649–657, 2015. 17