UNIVERSIDADE DE SÃO PAULO
ESCOLA DE ARTES, CIÊNCIAS E HUMANIDADES
PROGRAMA DE PÓS-GRADUAÇÃO EM SISTEMAS DE INFORMAÇÃO

ARTHUR MARÇAL FLORES

**Inferência da satisfação de usuários em serviços públicos de acesso à
informação utilizando aprendizado neural**

São Paulo

2020

ARTHUR MARÇAL FLORES

**Inferência da satisfação de usuários em serviços públicos de acesso à
informação utilizando aprendizado neural**

Versão corrigida

Dissertação apresentada à Escola de Artes, Ciências e Humanidades da Universidade de São Paulo para obtenção do título de Mestre em Ciências pelo Programa de Pós-graduação em Sistemas de Informação.

Área de concentração: Metodologia e Técnicas da Computação

Versão corrigida contendo as alterações solicitadas pela comissão julgadora em 03 de novembro de 2020. A versão original encontra-se em acervo reservado na Biblioteca da EACH-USP e na Biblioteca Digital de Teses e Dissertações da USP (BDTD), de acordo com a Resolução CoPGr 6018, de 13 de outubro de 2011.

Orientador: Prof. Dr. Ivandré Paraboni

São Paulo

2020

Autorizo a reprodução e divulgação total ou parcial deste trabalho, por qualquer meio convencional ou eletrônico, para fins de estudo e pesquisa, desde que citada a fonte.

CATALOGAÇÃO-NA-PUBLICAÇÃO

(Universidade de São Paulo. Escola de Artes, Ciências e Humanidades. Biblioteca)

CRB 8 - 4936

Flores, Arthur Marçal

Inferência da satisfação de usuários em serviços públicos de acesso à informação utilizando aprendizado neural / Arthur Marçal Flores ; orientador, Ivandré Paraboni. – 2020.

102 f : il.

Dissertação (Mestrado em Ciências) - Programa de Pós-Graduação em Sistemas de Informação, Escola de Artes, Ciências e Humanidades, Universidade de São Paulo.

Versão corrigida

1. Inteligência artificial. 2. Processamento de linguagem natural. 3. Redes neurais. 4. Aprendizado computacional. 5. Usuários da informação. 6. Satisfação. 7. Governo eletrônico. I. Paraboni, Ivandré, orient. II. Título

CDD 22.ed.– 006.4

Dissertação de autoria de Arthur Marçal Flores, sob o título **“Inferência da satisfação de usuários em serviços públicos de acesso à informação utilizando aprendizado neural”**, apresentada à Escola de Artes, Ciências e Humanidades da Universidade de São Paulo, para obtenção do título de Mestre em Ciências pelo Programa de Pós-graduação em Sistemas de Informação, na área de concentração Metodologia e Técnicas da Computação, aprovada em 03 de novembro de 2020 pela comissão julgadora constituída pelos doutores:

Prof. Dr. Ivandré Paraboni
Universidade de São Paulo
Escola de Artes, Ciências e Humanidades
Presidente

Prof. Dr. Thiago Alexandre Salgueiro Pardo
Universidade de São Paulo
Instituto de Ciências Matemáticas e de
Computação

Prof. Dra. Gisele da Silva Craveiro
Universidade de São Paulo
Escola de Artes, Ciências e Humanidades

Dedico este trabalho à minha mãe Angelita Marçal Flores, que tem demonstrado todo o apoio e suporte incondicional, não apenas durante o período de realização deste mestrado mas por toda a minha jornada profissional.

Agradecimentos

Agradeço à minha família e amigos por todo o suporte e paciência durante estes dois anos e meio, vocês são minha maior fonte de força e inspiração.

Agradeço especialmente aos amigos Caio Deutsch e Clarissa Vieira, o quais acompanharam toda esta jornada sempre de perto e com muito apoio.

Aos colegas da Programa de Pós-graduação em Sistemas de Informação (PPgSI) por todo auxílio e compartilhamento de ideias e conhecimento.

Ao Prof. Dr. Elias Jacob de Menezes Neto da UFRN por disponibilizar o cópulus e-SIC, que é a base deste trabalho.

Ao Sidney Leal e à Prof. Dr. Sandra Maria Aluísio do ICMC por gentilmente me ajudar com a construção de um dos conjuntos de dados por meio da ferramenta *Coh-Matrix Port.*

E finalmente ao meu professor orientador Prof. Dr. Ivandré Paraboni, pela paciência, revisões, discussões e muitos *feedbacks* enriquecedores.

Resumo

FLORES, Arthur Marçal. **Inferência da satisfação de usuários em serviços públicos de acesso à informação utilizando aprendizado neural**. 2020. 102 f. Dissertação (Mestrado em Ciências) – Escola de Artes, Ciências e Humanidades, Universidade de São Paulo, São Paulo, 2020.

Com a rápida expansão dos sistemas de tecnologia da informação (TI) em anos recentes, surge a oportunidade de aprimorar os serviços em diversos segmentos, entre eles, os prestados pelas instituições governamentais. Dentre estes diversos serviços, a disponibilização de informações públicas é um exemplo importante, pois por meio da Lei da Transparência juntamente com a Lei de Acesso à Informação, criaram um ambiente mais democrático no Brasil. Um operacionalizador do processo de transparência, é o Sistema Eletrônico do Serviço de Informação ao Cidadão (e-SIC), sendo que os dados provenientes desta plataforma constituem um corpus de especial interesse para o presente trabalho. A disponibilidade de textos rotulados com escores de satisfação deste sistema sugere a oportunidade de utilizar métodos de processamento de línguas naturais (PLN) para inferir de forma automática a satisfação de usuários, especialmente no que diz respeito ao uso de redes neurais que têm obtido resultados positivos em diversas tarefas da área. A partir desta observação, o presente trabalho apresenta os resultados de pesquisa em nível de mestrado no campo de PLN, no domínio da satisfação de usuários, com o objetivo geral de desenvolver modelos computacionais para avaliar a satisfação dos usuários de plataformas de solicitação de acesso à informação, por meio da utilização de técnicas baseadas em aprendizado neural.

Palavras-chaves: Satisfação do usuário. Aprendizado de máquina. Processamento de língua natural. PLN. Aprendizado neural. LSTM. ELMo. BERT.

Abstract

FLORES, Arthur Marçal. **User satisfaction inference in public services of information access using neural learning**. 2020. 102 p. Dissertation (Master of Science) – School of Arts, Sciences and Humanities, University of São Paulo, São Paulo, 2020.

Given the fast expansion of information technology (IT) systems in recent years, there is the opportunity to improve services in several segments, including those provided by government institutions. Among these services, the provision of public information is an important example, as through the Transparency Law and the Access to Information Law created a more democratic environment in Brazil. An outcome of the transparency process is the Electronic System for Citizen Information Service (e-SIC), and the data from this platform constitute a corpus of special interest for the present work. The availability of labeled texts with satisfaction scores suggests the opportunity to use natural language processing (NLP) methods to automatically infer user satisfaction, especially with regard to the use of Neural Networks that have obtained positive results in several tasks in the field. Given this opportunity, the current work presents a Msc research project in the NLP field, in the domain of user satisfaction, with the goal of developing computational models to evaluate the satisfaction of users of request information access platforms through the use of techniques based on neural learning.

Keywords: User Satisfaction. Machine Learning. Natural Language Processing. NLP. Neural Learning. LSTM. ELMo. BERT.

Lista de figuras

Figura 1 – Arquiteturas do modelo <i>Word2Vec</i>	25
Figura 2 – Margens de separação do modelo SVM	27
Figura 3 – Representação de uma rede neural artificial	29
Figura 4 – Representação de uma rede neural recorrente	30
Figura 5 – Representação de uma rede <i>LSTM</i>	31
Figura 6 – Representação da arquitetura <i>Transformer</i>	33
Figura 7 – Representação mecanismo de atenção multi-cabeça	34
Figura 8 – Representação do modelo <i>LSTM</i> bidirecional com mecanismo de atenção	69
Figura 9 – Representação do modelo Stack	70
Figura 10 – Folhas da árvore de decisão para as classes intermediária e negativa do problema da satisfação com a Clareza	99
Figura 11 – Folhas da árvore de decisão para a classe positiva do problema da satisfação com a Clareza	100
Figura 12 – Árvore de decisão - Satisfação com o atendimento - Variáveis LIWC . .	101
Figura 13 – Árvore de decisão da análise de erros - Satisfação com a Clareza	102
Figura 14 – Árvore de decisão da análise de erros - Satisfação com o Atendimento .	102

Lista de tabelas

Tabela 1 – Principais campos da base e-SIC	21
Tabela 2 – Tipos de resposta da base e-SIC	22
Tabela 3 – Análise descritiva geral	23
Tabela 4 – Summary of analyzed papers	53
Tabela 5 – Distribuição das notas - Acesso Concedido	55
Tabela 6 – Distribuição das notas - Acesso Parcialmente Concedido	55
Tabela 7 – Distribuição das notas - Acesso Negado	55
Tabela 8 – Distribuição das notas - Classe Atendimento	56
Tabela 9 – Distribuição das notas binárias - Conjunto de dados 1 - Classes Clareza e Atendimento	56
Tabela 10 – Distribuição das classes do conjunto de treino e teste do experimento 1	58
Tabela 11 – Modelo de satisfação referente à clareza do atendimento - Com base nos textos de resposta	59
Tabela 12 – Modelo de satisfação referente ao pleno atendimento - Com base nos textos de resposta	59
Tabela 13 – Distribuição das notas - Conjunto de dados 2 - Classes Clareza e Atendimento	61
Tabela 14 – Distribuição das notas binárias - Conjunto de dados 2 - Classes Clareza e Atendimento	61
Tabela 15 – Distribuição das classes do conjunto de dados de treino e teste do experimento 2 - Inferência a partir das solicitações e-SIC	62
Tabela 16 – Predição da satisfação da clareza da resposta com base no texto da solicitação	62
Tabela 17 – Predição de satisfação com o atendimento com base no texto da solicitação	62
Tabela 18 – Resumo das tarefas de classificação	66
Tabela 19 – Distribuição de instâncias de treinamento e teste	73
Tabela 20 – Configuração padrão dos modelos de regressão logística <i>softmax</i>	74
Tabela 21 – Parâmetro C de regularização por tarefa	79
Tabela 22 – Parâmetros LogReg <i>softmax</i> - Classe Clareza - Textos de resposta . . .	80
Tabela 23 – Parâmetros <i>LSTM</i> com atenção - Classe Clareza - Textos de resposta .	80

Tabela 24 – Parâmetros <i>BERT</i> - Classe Clareza - Textos de resposta	80
Tabela 25 – Parâmetros LogReg <i>softmax</i> - Classe Atendimento - Textos de resposta	80
Tabela 26 – Parâmetros <i>LSTM</i> com atenção - Classe Atendimento - Textos de resposta	81
Tabela 27 – Parâmetros <i>BERT</i> - Classe Atendimento - Textos de resposta	81
Tabela 28 – Parâmetros LogReg <i>softmax</i> - Classe Atendimento - Textos de requisição	81
Tabela 29 – Parâmetros <i>LSTM</i> com atenção - Classe Atendimento - Textos de requisição	81
Tabela 30 – Parâmetros <i>BERT</i> - Classe Atendimento - Textos de requisição	81
Tabela 31 – Medida F - Tarefa Clareza com base nas respostas	82
Tabela 32 – Medida F - Tarefa Atendimento com base nas respostas	83
Tabela 33 – Medida F - Tarefa Atendimento com base nas requisições	84
Tabela 34 – Variáveis significativas do dicionário <i>LIWC</i> - Inferência do Atendimento	87
Tabela 35 – Rótulos classificados corretamente e erroneamente pelos modelos <i>base- line Coh-Metrix</i> e <i>RegLog.LIWC</i>	87
Tabela 36 – Distribuição dos conjuntos de dados utilizados na análise de erros . . .	88

Lista de abreviaturas e siglas

AS	Análise de sentimentos
BERT	<i>Bidirectional Enconders Representations from Transformers</i>
BOW	<i>Bag of words</i>
CBOW	<i>Continous bag of words</i>
ELMo	<i>Embeddings from Language Models</i>
e-SIC	Sistema Eletrônico do Serviço de Informação ao Cidadão
GRU	<i>Gated recurrent unit</i>
LIWC	<i>Linguistic Inquiry and Word Count</i>
LSTM	<i>Long short term memory</i>
ME	Máxima Entropia
MLP	<i>Multilayer perceptron</i>
NER	<i>Named-Entity Recognition</i>
PLN	Processamento de Língua Natural
ReLu	<i>Rectified linear unit</i>
RNA	Redes Neurais Artificiais
RNN	<i>Recurrent neural network</i>
SVM	<i>Support vector machine</i>
TI	Tecnologia da Informação

Sumário

1	Introdução	16
1.1	<i>O Córpus e-SIC</i>	17
1.2	<i>Inferência de satisfação a partir de textos</i>	18
1.3	<i>Hipóteses</i>	18
1.4	<i>Objetivo</i>	19
1.5	<i>Organização do documento</i>	19
2	Conceitos fundamentais	21
2.1	<i>O córpus e-SIC</i>	21
2.2	<i>Métodos de representação textual</i>	23
2.2.1	Modelos tradicionais de representação textual	23
2.2.2	Representação distribuída de palavras (<i>Embedding</i> de palavras) . .	24
2.3	<i>Métodos de aprendizado</i>	26
2.3.1	Naive-Bayes	26
2.3.2	Regressão logística <i>Softmax</i>	26
2.3.3	Máquina de Vetor Suporte (SVM)	27
2.3.4	Redes Neurais Artificiais (RNAs)	28
2.3.5	Redes Neurais Recorrentes (RNNs)	29
2.3.6	<i>Modelos de língua</i>	32
2.3.7	Considerações	35
3	Revisão Bibliográfica	36
3.1	<i>Modelagem de satisfação de usuários</i>	36
3.1.1	Satisfação do usuário em sistemas de perguntas e respostas	36
3.1.2	Mensuração de satisfação em tempo real.	38
3.1.3	Modelagem de usuários com Redes Neurais para predição de ava- liações de usuários	40
3.1.4	LSTM hierárquico com aprendizado conjunto para estimação da satisfação de clientes em chamadas de Call-Centers	42
3.2	<i>Análise de sentimentos</i>	43
3.2.1	Predição de avaliações para revisões de serviços e produtos	43

3.2.2	Detecção de sentimentos em e-mails de atendimento de clientes . . .	45
3.2.3	Análise de sentimentos de Tweets utilizando comitês de classificadores	46
3.2.4	Refinando Embedding de palavras para análise de sentimentos utilizando escores de intensidade	47
3.3	<i>Modelagem de diálogos</i>	48
3.3.1	Modelando transições na satisfação do usuário em diálogo a partir de avaliações	48
3.3.2	Identificando fatores sociais e expressivos em textos de solicitações utilizando modelos transacionais/sequenciais	50
3.3.3	Em direção à avaliação automática de diálogos em Helpdesk de clientes	51
3.4	<i>Considerações</i>	52
4	Estudo exploratório	54
4.1	<i>Experimento 1: Modelagem da satisfação com base na resposta</i>	54
4.1.1	Conjunto de dados	54
4.1.2	Modelos propostos	57
4.1.3	Resultados	59
4.1.4	Discussão	59
4.2	<i>Experimento 2: Modelagem da satisfação com base em solicitações e-SIC</i>	60
4.2.1	Conjunto de dados	60
4.2.2	Modelos propostos	61
4.2.3	Resultados	62
4.2.4	Discussão	63
4.2.5	Considerações	63
5	Métodos	64
5.1	<i>Conjunto de dados</i>	64
5.2	<i>Definição de classes de aprendizado</i>	65
5.3	<i>Classificadores propostos</i>	66
5.3.1	Regressão logística <i>softmax</i> com base em n-gramas de palavras (LogReg.word)	67
5.3.2	Regressão logística <i>softmax</i> com base em características <i>LIWC</i> (LogReg.LIWC)	67

5.3.3	Rede neural recorrente do tipo <i>LSTM</i> com mecanismo de atenção (<i>LSTM</i>)	68
5.3.4	Embeddings from Language Models (ELMo)	69
5.3.5	Bidirectional Encoders Representations from Transformers (BERT)	69
5.3.6	Regressão logística <i>softmax</i> do tipo comitê <i>stacking</i> (Stack)	70
5.4	<i>Classificadores Baseline</i>	71
5.4.1	Regressão logística <i>softmax</i> com base em característica <i>Coh-Metrix</i>	71
5.4.2	Regressão logística <i>softmax</i> com base no tempo de atendimento . .	72
6	Avaliação	73
6.1	<i>Procedimentos</i>	73
6.2	<i>Treinamento de modelos para inferência da satisfação</i>	75
6.2.1	Regressão logística <i>softmax</i> com base em n-gramas de palavras (LogReg.word)	75
6.2.2	Regressão logística <i>softmax</i> com base em características <i>LIWC</i> (LogReg.LIWC)	75
6.2.3	Rede neural recorrente do tipo <i>LSTM</i> com mecanismo de atenção (<i>LSTM</i>)	76
6.2.4	Embeddings from Language Models (ELMo)	77
6.2.5	Bidirectional Encoders Representations from Transformers (BERT)	77
6.2.6	Regressão logística <i>softmax</i> do tipo comitê <i>stacking</i> (Stack)	78
6.2.7	Modelos de baseline	79
6.2.8	Considerações	80
7	Resultados	82
7.1	<i>Experimento 1: Inferência da satisfação com a clareza com base nos textos de resposta</i>	82
7.2	<i>Experimento 2: Inferência da satisfação com o atendimento com base nos textos de resposta</i>	83
7.3	<i>Experimento 3: Inferência da satisfação com o atendimento com base nos textos de requisição</i>	84
7.4	<i>Análise de características mais relevantes para classificação</i>	85
7.4.1	Características mais relevantes para a classificação da Clareza . . .	85

7.4.2	Características mais relevantes para a classificação do Atendimento	86
7.5	<i>Análise de erros</i>	87
7.5.1	Análise de erros para a classificação da Clareza	88
7.5.2	Análise de erros para a classificação do Atendimento	88
7.6	<i>Considerações</i>	89
8	Conclusões	91
	Referências ¹	93
	Apêndice A – Árvores utilizadas para análises de variáveis e erros	99

¹ De acordo com a Associação Brasileira de Normas Técnicas. NBR 6023.

1 Introdução

Com a rápida expansão dos sistemas de tecnologia da informação (TI) em anos recentes, surge a oportunidade de aprimorar os serviços em diversos segmentos, entre eles, os serviços prestados pelas instituições governamentais. De acordo com a Pesquisa TIC Governo Eletrônico (2017) do Centro Regional de Estudos para o Desenvolvimento da Sociedade da Informação (Cetic.br¹), houve crescimento entre 2015 e 2017 nos diversos serviços eletrônicos de contato com o cidadão prestados pelo governo, seja na forma de e-mail, formulário eletrônico ou serviços de solicitação de acesso à informação. Com o crescente uso destas plataformas, torna-se imprescindível uma clara compreensão das necessidades dos cidadãos, buscando melhorar não só o atendimento mas também o processo de transparência, essencial para uma sociedade democrática.

O direito de acesso à informação governamental é fundamental nos atuais regimes democráticos (SASSO LUCIANO NURNBERG PERES, 2017). No Brasil, a Lei nº 12.527 (2011), de acesso à informação, garante que todos têm direito a receber dos órgãos públicos informações de seu interesse particular, ou de interesse coletivo ou geral, que serão prestadas no prazo da lei, sob pena de responsabilidade, ressalvadas aquelas cujo sigilo seja imprescindível à segurança da sociedade e do Estado.

Em anos recentes, a Lei da Transparência juntamente com a Lei de Acesso à Informação, criaram este ambiente mais democrático, disponibilizando as informações públicas de forma proativa (transparência ativa). Além desta, há também a questão da transparência passiva, que ocorre quando há disponibilização de informações mediante solicitação por parte dos interessados (FREIRE,).

Ainda mais relevante, a Lei nº 13.460, de 26 de Junho de 2017, dispõe sobre participação, proteção e defesa dos direitos do usuário dos serviços públicos da administração pública, regulamentando os direitos básicos e deveres dos usuários. Define ainda os aspectos de avaliação dos serviços prestados pelos órgãos e entidades públicas, por exemplo, por meio da satisfação do serviço prestado e da qualidade do atendimento ao usuário.

Um exemplo de plataforma viabilizadora do processo de transparência é o Sistema Eletrônico do Serviço de Informação ao Cidadão (e-SIC²). Dados do estudo em Falconiery Rachel Bastos (2016) mostram que, só no e-SIC, 355.797 pedidos foram registrados no

¹ <https://www.cetic.br/>

² <https://esic.cgu.gov.br>

período desde o lançamento da plataforma em maio de 2012 até março de 2016. Dados provenientes desta plataforma constituem um *córpus* de especial interesse para o presente trabalho. Este *córpus* é descrito a seguir.

1.1 O *Córpus e-SIC*

O *córpus e-SIC* é uma base contendo aproximadamente 50 mil interações entre cidadãos e governo, com requisições de informações cadastradas no período de 2015 a 2017, respostas e notas de avaliação do serviço. Este conjunto de dados foi extraído de uma plataforma online do governo federal chamada *Sistema Eletrônico do Serviço de Informações ao Cidadão (e-SIC)*, que permite que qualquer pessoa, física ou jurídica, encaminhe pedidos de acesso à informação, acompanhe o prazo e receba a resposta da solicitação realizada para órgãos e entidades do Executivo Federal (da administração direta e indireta).

Este *córpus* possui ainda, vários tipos de informações sobre a requisição (Órgãos associados ao destino, Categoria do Pedido, Origem, etc). Entretanto, para este estudo são considerados três grupos principais, as informações textuais da solicitação do usuário, as respostas também textuais dos servidores, e escores de satisfação fornecidos pelos usuários para avaliar o atendimento. Um exemplo de requisição e resposta é apresentado a seguir.

Solicito detalhamento e valor total das despesas do presidente e de sua família durante viagem para a Base Militar da Marinha da Restinga da Marambaia, no Rio, entre 29/12/16 e 02/01/17. Quanto foi gasto com deslocamentos, alimentação e segurança? Quantos servidores foram utilizados nessa viagem?

Prezado Senhor, em atenção ao pedido de acesso à informação cadastrado sob o número 00077.000264/2017-26, esclarecemos que os gastos realizados durante a referida viagem foram os seguintes: deslocamento, R\$ 107.892,02, alimentação, R\$ 103.329,35. (...) Atenciosamente, Serviço de Informações ao Cidadão Palácio do Planalto-www.planalto.gov.br/acessoainformacao

A partir desta troca de informações representada pela requisição do usuário e a resposta do servidor, o usuário então fornece respostas às perguntas de satisfação que também são integrantes do *córpus*, “A resposta fornecida foi de fácil compreensão?” e “A resposta fornecida atendeu plenamente ao seu pedido?”. As respostas são fornecidas em uma escala de um até cinco, em que um representa a menor satisfação e cinco representa a maior.

1.2 Inferência de satisfação a partir de textos

A disponibilidade de textos (tanto de perguntas como respostas) rotulados com escores de satisfação de uma grande massa de usuários do sistema e-SIC sugere a oportunidade de utilizar métodos de processamento de línguas naturais (PLN) (em especial, os que fazem uso de aprendizado de máquina supervisionado) para inferir de forma automática a satisfação de usuários a partir destes textos. Neste contexto entende-se satisfação do usuário tanto do ponto de vista da satisfação com a resposta fornecida pelos servidores, como também quanto com à clareza da resposta.

De forma mais específica, observa-se que o uso de métodos neurais tem obtido resultados positivos em diversas tarefas da área (como Análise de sentimentos (ZHANG; WANG; LIU, 2018), Classificação de documentos (CHENG *et al.*, 2019), *Tagging* (GOPALAKRISHNAN; SOMAN; PREMJI, 2019) etc), além de ainda ser pouco encontrado na literatura. Estes métodos têm apresentado grande contribuição em estratégias mais recentes de modelos de língua, por exemplo os modelos ELMo (PETERS *et al.*, 2018) e as arquiteturas *Transformers* do modelo *BERT* (DEVLIN *et al.*, 2019), que também têm superado diversos resultados do estado da arte em tarefas de PLN.

1.3 Hipóteses

A hipótese geral deste estudo é que a satisfação de usuários em serviços de solicitação de acesso à informação por meio de plataformas digitais pode ser estimado com o uso de modelos de PLN, tanto com base nos textos das respostas fornecidas pelo servidor, quanto das requisições dos usuários. Para testar esta hipótese, pretende-se comparar modelos computacionais que avaliem a satisfação do usuário a partir do texto com uso de métricas tradicionais da área de aprendizado de máquina como acurácia e medida F. Espera-se que os modelos desenvolvidos apresentem resultados superiores aos de sistemas de *baseline* que se façam pertinentes.

Mais especificamente a partir desta hipótese geral, são enunciadas três hipóteses específicas a serem testadas no presente trabalho.

- H1: O grau de satisfação do usuário em relação à **clareza** das respostas obtidas em serviços do tipo de solicitação de acesso à informação pode ser estimado com o uso

de modelos de processamento de língua natural tomando-se como entrada os textos de **resposta** fornecidos pelo prestador do serviço.

- H2: O grau de satisfação do usuário em relação ao **atendimento** em serviços do tipo de solicitação de acesso à informação pode ser estimado com o uso de modelos de processamento de língua natural tomando-se como entrada os textos de **resposta** fornecidos pelo prestador do serviço.
- H3: O grau de satisfação do usuário em relação ao **atendimento** em serviços do tipo de solicitação de acesso à informação pode ser estimado com o uso de modelos de processamento de língua natural tomando-se como entrada os textos de **requisições** de usuários do sistema.

1.4 *Objetivo*

O objetivo deste trabalho é desenvolver modelos computacionais supervisionados para avaliar a satisfação dos usuários de plataformas de solicitação de acesso à informação, por meio da utilização de técnicas de aprendizado neural, com acurácia superior à de métodos tradicionais de classificação de documentos.

Este trabalho será conduzido seguindo as atividades desenvolvimento de modelos computacionais para pré-processamento dos dados, desenvolvimento de modelos de *baseline* e satisfação do cliente tanto quanto à clareza da resposta quanto ao atendimento, extração de características, classificação de textos usando aprendizado supervisionado, refinamento e avaliação com base nas métricas definidas.

1.5 *Organização do documento*

Este trabalho está organizado da seguinte forma, os conceitos fundamentais necessários para a compreensão desta obra são apresentados no Capítulo 2. Inicialmente é descrita a formulação elementar do problema abordado, ou seja, a modelagem de satisfação de usuários, e em seguida são apresentadas as formas de representação textual a partir de modelos tradicionais e baseados em *embeddings* de palavras, e ainda, alguns dos principais métodos de aprendizado utilizados na área. No Capítulo 3 são apresentadas as revisões bibliográficas divididas em três subseções. A primeira apresenta estudos sobre modelagem de satisfação de usuários, a segunda mostra estudos acerca de modelagem de diálogos e,

por fim, estudos a respeito de técnicas com base em AS. O Capítulo 4 retrata o estudo exploratório e seus resultados iniciais. O Capítulo 5 apresenta as soluções para os problemas de inferência de satisfação. Em seguida o Capítulo 6 detalha a construção, otimização e avaliação dos métodos apresentados no capítulo anterior. Os resultados do trabalho são tratados no Capítulo 7 e finalmente o Capítulo 8 apresenta uma conclusão da obra.

2 Conceitos fundamentais

Este capítulo apresenta os conceitos fundamentais, relacionados ao escopo deste projeto de pesquisa, acerca da base de dados e das técnicas usadas para predição de satisfação de usuários, sendo dividido em quatro partes. Descrição do *córpus*, Métodos de representação textual tradicionais e baseados em *embeddings* e Métodos de aprendizado.

2.1 O *córpus* e-SIC

Nesta seção, são descritas as principais características do *córpus* e-SIC, que é a base deste trabalho. Além de informações sobre identificadores do pedido, protocolos e datas, a base de dados é composta por informações relacionadas à solicitação, a situação do pedido, resposta do questionamento e avaliação do atendimento. A tabela 1 abaixo, mostra uma relação destas informações por categoria.

Tabela 1 – Principais campos da base e-SIC

Solicitação	OrgaoSuperiorAssociadoaoDestinatario
	OrgaoDestinatario
	ResumoSolicitacao
	DetalhamentoSolicitacao
	FormaResposta
	OrigemSolicitacao
	CategoriaPedido
	SubCategoriaPedido
	NumeroPerguntas
Situação do pedido	Situacao
	PrazoAtendimento
	FoiProrrogado
	FoiReencaminhado
Resposta	DataResposta
	Resposta
	TipoResposta
	ClassificacaoTipoResposta
Avaliação do atendimento	Pergunta1
	Resposta1
	Pergunta2
	Resposta2

Fonte: Arthur Marçal Flores 2020

Nesta representação, as colunas Pergunta1 e Pergunta2 indicam respectivamente os textos da primeira e da segunda pergunta feita ao usuário no questionário de satisfação. Da mesma forma, as colunas Resposta1 e Resposta2 representam, em uma escala de 1 a 5, a nota que o usuário deu à primeira e a segunda pergunta do questionário de satisfação.

Um conceito a ser destacado é o tipo de acesso, representado pela coluna TipoResposta. A base de dados é composta por um conjunto de 49.481 perguntas e respostas, que são divididas em diversos grupos conforme a tabela 2.

Tabela 2 – Tipos de resposta da base e-SIC

Tipo Resposta	Qtde.	%
Acesso Concedido	38.537	77,9%
Não se trata de solicitação de informação	2.413	4,9%
Acesso Negado	2.361	4,8%
Acesso Parcialmente Concedido	2.302	4,7%
Informação Inexistente	1.515	3,1%
Órgão não tem competência para responder sobre o assunto	1.458	2,9%
Pergunta Duplicada/Repetida	817	1,7%
Encaminhado para o e-Ouv	78	0,2%
Total	49.481	100%

Fonte: Arthur Marçal Flores 2020

Os questionários de avaliação possuem 6 perguntas, 2 para quando o acesso é parcialmente concedido, 2 para quando o acesso é concedido e outras 2 para quando o acesso é negado.

1. Acesso concedido

- a) A resposta fornecida foi de fácil compreensão?
- b) A resposta fornecida atendeu plenamente ao seu pedido?

2. Acesso parcialmente concedido

- a) A justificativa para a entrega parcial da informação foi satisfatória?
- b) A informação fornecida foi de fácil compreensão?

3. Acesso negado

- a) A justificativa para o não fornecimento da informação foi satisfatória?
- b) A justificativa fornecida foi de fácil compreensão?

Abaixo, segue uma relação com a análise descritiva de algumas informações da base de dados. Estas informações estão organizadas de acordo com a divisão utilizada no experimento descrito no capítulo 4 do estudo exploratório. Na segunda coluna são mostrados os números referentes à base completa. Na terceira e quarta coluna são mostrados os números referentes à base utilizada no experimento com os textos das solicitações do

usuário como entrada e, por fim, na quinta e sexta coluna são mostrados os números referentes à base utilizada no experimento com os textos das respostas como entrada. As informações são separadas em classe Clareza e Atendimento, que referem-se às duas perguntas relativas à satisfação do usuário. A distinção na quantidade de instâncias destas bases é referente ao balanceamento efetuado devido à diferença na distribuição das classes.

Tabela 3 – Análise descritiva geral

Descrição	Base completa	Textos de solicitação		Textos de resposta	
		Classe Clareza	Classe Atendimento	Classe Clareza	Classe Atendimento
Número de instâncias	49.481	19.708	23.990	3.764	7.692
Número de solicitantes únicos	25.660	12.828	15.072	2.948	5.611
Total de palavras das solicitações	3.829.028	1.572.361	1.892.126	270.633	545.136
Tamanho do vocabulário das solicitações	197.637	116.274	132.181	37.939	60.539
Tamanho médio das solicitações	484	498	492	449	443
Total de palavras das respostas	7.088.275	3.027.342	3.559.918	493.526	974.873
Tamanho do vocabulário das respostas	218.721	120.141	136.716	42.567	64.278
Tamanho médio das respostas	914	980	949	832	804

Fonte: Arthur Marçal Flores 2020

2.2 Métodos de representação textual

Nessa seção são examinadas algumas das principais formas para representação textual. A representação textual tem o objetivo de transformar as informações dos textos em um formato que os algoritmos de PLN possam entender e atuar (KO, 2012). São apresentados os modelos tradicionais de representação textual, assim como os modelos baseados em *embeddings* em consequência de sua importância nos trabalhos mais recentes do estado-da-arte para PLN.

2.2.1 Modelos tradicionais de representação textual

O modelo saco de palavras, do inglês, *bag of words (BOW)* (SALTON; WONG; YANG, 1975) adota a representação simbólica em vetores de números naturais $\vec{w}_d$ para contagem e frequência de palavras. Neste modelo, o documento é representado pelo vetor $\vec{w}$ de dimensão d , sendo d o total de palavras únicas no texto, e cada dimensão contendo a frequência das palavras no texto. Neste modelo, a semântica e a ordem das palavras não são consideradas. Ainda que utilizando uma representação textual simplificada e que possua a desvantagem de gerar matrizes esparsas em sua representação, este modelo é largamente utilizado em atividades relacionadas a PLN.

Outra desvantagem do modelo *BOW* é que todos os termos são considerados igualmente relevantes, quando na verdade alguns termos nem têm capacidade preditiva para o modelo (MANNING; RAGHAVAN; SCHUTZE, 2008). Por este motivo, é introduzida a ideia de reduzir o peso de um termo de acordo com o aumento da frequência de documentos onde este termo aparece. A combinação da representação matriz de termos $tf_{t,d}$ com o inverso da frequência nos documentos idf_t , produz uma representação que atribui um peso ao termo t em cada documento d que é:

1. O mais alto quando t ocorre muitas vezes em poucos documentos;
2. Baixo quando t ocorre poucas vezes em um documento ou ocorre em muitos documentos;
3. O menor quando t aparece em praticamente todos os documentos;

Assim, pode-se definir esta representação conforme a equação 1 citada em Manning, Raghavan e Schutze (2008).

$$tf - idf_{t,d} = tf_{t,d} \cdot \log \frac{N}{df_t} \quad (1)$$

Onde $tf_{t,d}$ representa a razão da frequência do termo em cada documento em comparação ao número total de palavras no documento e $\log \frac{N}{df_t}$ representa a ponderação do inverso da frequência nos documentos.

2.2.2 Representação distribuída de palavras (*Embedding* de palavras)

Os modelos tradicionais apresentados anteriormente, baseados em contagem, utilizam apenas uma dimensão no vetor para representar os termos, e a dimensão deste vetor é proporcional ao número de palavras únicas no corpus. Esta abordagem também não leva em consideração a ordem em que as palavras aparecem e nem a similaridade entre estas palavras, tratando-as de forma independente.

Na representação distribuída, cada palavra é associada a um vetor $\vec{w}$ em $\mathbb{R}^d$ no qual o significado da palavra é capturado nas diferentes dimensões d deste vetor. Estas dimensões não são interpretáveis, e um significado pode ser capturado pela combinação de muitas dimensões (GOLDBERG; HIRST, 2017).

Neste modelo de representação, os vetores são ditos de densos, ou seja, cada característica é embutida em um espaço dimensional d . Esta dimensão é normalmente

muito menor do que o número de características. Por exemplo, na representação *one-hot vectors*, (GOLDBERG; HIRST, 2017) onde a dimensão do vetor é proporcional à quantidade de palavras únicas naquele vocabulário, cada palavra é representada por um vetor com uma dimensão do tamanho deste vocabulário. Em contraste, na representação embutida o vetor pode ter, por exemplo, 100 ou 200 dimensões. O maior benefício da representação por meio de vetores densos é o aumento da generalização do modelo. Com a utilização de um classificador não linear (e.g.: redes neurais) é possível identificar as principais características que podem capturar similaridades entre as palavras (GOLDBERG; HIRST, 2017).

De acordo com o estudo em Mikolov *et al.* (2013), estas representações distribuídas foram amplamente utilizadas com sucesso, por exemplo, no trabalho em Bengio *et al.* (2003), onde é desenvolvido um modelo neural de língua (*Neural Natural Language Modeling - NNLM*). As arquiteturas desenvolvidas em Mikolov *et al.* (2013) são uma extensão deste modelo *NNLM*, sendo elas a *Continuous bag of words (CBOW)* e *Skip-gram*. *CBOW* considera o contexto com uma janela fixa predefinida de tamanho k onde o objetivo é prever a palavra central W_c a partir das palavras ao seu redor ($W_{c-k}, \dots, W_{c-1}, W_{c+1}, \dots, W_{c+k}$). Em contraste, a arquitetura *Skip-gram* utiliza a palavra central W_c para prever o contexto.

Suas representações são ilustradas na figura 1.

Figura 1 – Arquiteturas do modelo *Word2Vec*

Fonte: Mikolov *et al.* (2013)

2.3 Métodos de aprendizado

Nesta seção são apresentados alguns dos principais modelos de aprendizado utilizados no PLN. Começando com os mais simples, como o método de *Naive-Bayes*, até modelos mais complexos, como redes neurais e *Transformers*.

2.3.1 Naive-Bayes

O classificador *Naive-Bayes* é um modelo probabilístico baseado no *teorema de Bayes*, formulado conforme a Equação 2 citada em Maron (1961).

$$P(A|B) = \frac{P(B|A)P(A)}{P(B)} \quad (2)$$

Nesta equação $P(A|B)$ é a probabilidade condicional de A dado B ou a probabilidade a posteriori, $P(B|A)$ representa a verossimilhança, $P(A)$ é a probabilidade a priori do evento A e, finalmente, $P(B)$ é a normalização.

Uma importante suposição adotada por este algoritmo é que existe uma independência condicional entre as variáveis. Entretanto, esta suposição não é verdadeira uma vez que existe maior probabilidade de ocorrência na frequência de algumas variáveis. De qualquer modo, esta suposição torna a construção do classificador mais simples e que, ainda sim, apresenta bom desempenho nos resultados de muitos trabalhos (HEMALATHA *et al.*, 2013).

2.3.2 Regressão logística *Softmax*

Também conhecido como regressão logística multinomial ou classificador de Máxima Entropia (ME), este método é um classificador multi-classe com interpretação probabilística e diferente do método *Naive-Bayes* apresentado anteriormente, este classificador não assume a independência das variáveis de entrada. Este método também é amplamente utilizado como normalizador na saída de diversos classificadores como de redes neurais e *Transformers*, conforme apresentado na figura 6.

Uma forma de estimar a distribuição de probabilidade $p(y|x)$ é utilizar o método da Máxima Verossimilhança, encontrando assim o melhor vetor θ de parâmetros $p(y|x; \theta)$.

Para o caso binário, pode-se estimar apenas a probabilidade de uma das classes, uma vez que o valor somado das probabilidades das duas classes deve ser 1. Para isso, é utilizado a função sigmoide, a qual comprime os valores de saída no intervalo $(0,1)$, conforme a equação 3.

$$P(y = 1|x; \theta) = \sigma(\theta^T x) \quad (3)$$

Uma das formas de resolver esta equação é minimizar o log da função de custo verossimilhança negativa (*Negative Log-Likelihood*) por meio do algoritmos como o gradiente descendente. (GOODFELLOW; BENGIO; COURVILLE, 2016).

2.3.3 Máquina de Vetor Suporte (SVM)

O algoritmo de Máquina de Vetor Suporte (SVM) tem mostrado ótimos resultados em tarefas de categorização tradicional de textos, de modo geral tendo melhor desempenho do que algoritmos como *Naive-Bayes* e Máxima Entropia (JOACHIMS, 1998).

Figura 2 – Margens de separação do modelo SVM

Fonte: Cortes e Vapnik (1995)

Conforme mostrado na figura 2, em contraste com os classificadores probabilísticos *Naive-Bayes* e Máxima Entropia, SVM utiliza margens para separar e maximizar a distância entre as classes. Conforme descrito em Pang, Lee e Vaithyanathan (2002), no caso da classificação com duas categorias, o objetivo é buscar um hiperplano representado pelo vetor $\vec{w}$, o qual não apenas separa os vetores dos documentos em diferentes classes, mas que também maximiza a separação entre eles. Esta busca pela máxima separação é um problema de otimização com restrição. Sendo $c_j \in \{-1, 1\}$ a classe correta do documento

$\vec{d}_j$, a solução pode ser escrita conforme a equação 4 (PANG; LEE; VAITHYANATHAN, 2002).

$$\vec{w} := \sum j\alpha_j c_j \vec{d}_j, \alpha > 0 \quad (4)$$

onde α_j são obtidos resolvendo-se o problema de otimização dual. Estes $\vec{d}_j$ tal que α_j é maior que zero são chamados vetores de suporte, desde que sejam o único vetor contribuindo para $\vec{w}$. A classificação das instâncias consiste assim em determinar em qual lado do hiperplano de $\vec{w}$ elas se situam.

2.3.4 Redes Neurais Artificiais (RNAs)

As redes neurais artificiais (RNAs) são modelos inspirados no funcionamento do cérebro. Os componentes principais deste sistema são as unidades chamadas neurônios, capazes de processar informações por meio de impulsos elétricos e sinapses. O primeiro modelo implementado é conhecido como *Feed Forward Neural Network (FFNN)*. Nele as informações fluem apenas em uma direção, e nesta rede, cada sinal de entrada está associada a um peso. O neurônio soma o resultado da multiplicação do sinal de entrada dos pesos associados e aplica uma função não-linear, entregando um valor de saída. As redes são formadas por conexões de neurônios, que podem compor camadas.

A rede neural mais básica é conhecida como *Perceptron* e é definida conforme a equação 5 (GOLDBERG; HIRST, 2017).

$$\begin{aligned} NN_{Perceptron}(x) &= xW + b \\ x &\in \mathbb{R}^{d_{in}}, W \in \mathbb{R}^{d_{in} \times d_{out}}, b \in \mathbb{R}^{d_{out}} \end{aligned} \quad (5)$$

Onde x representa a entrada, W a matriz de pesos e b o termo de viés.

Apesar da facilidade em sua implementação e treinamento, além de uma interpretabilidade razoável, o modelo *Perceptron* é restrito ao ponto de conseguir representar apenas funções lineares. Uma forma mais complexa de rede neural é a *Multilayer perceptron (MLP)*, que possui camadas intermediárias chamadas de camadas escondidas, que permitem a distinção de dados que não são linearmente separáveis. Isso é possível porque esta rede consegue aprender representações de uma grande quantidade de funções não lineares. A figura 3 ilustra a representação de uma rede neural com múltiplas camadas.

Figura 3 – Representação de uma rede neural artificial

Fonte: Goldberg e Hirst (2017)

A utilização desta rede e de redes mais profundas (com múltiplas camadas escondidas), se tornou viável apenas com a introdução do algoritmo *Backpropagation* (RUMELHART; HINTON; WILLIAMS, 1986). Este algoritmo tem o objetivo de fazer fluir, na direção oposta à saída, a informação relacionada à função de custo, calculando de forma eficiente as derivadas parciais (gradientes) por meio da regra da cadeia da diferenciação, enquanto mantem os resultados intermediários (DENG; LIU, 2018).

2.3.5 Redes Neurais Recorrentes (RNNs)

Em PLN, uma tarefa importante é entender as relações de longa distância entre palavras ou sentenças que abrangem um valioso significado sintático e semântico. As redes neurais recorrentes, introduzidas em Rumelhart, Hinton e Williams (1986), permitiram a representação de entradas sequenciais de tamanho arbitrário em vetores de tamanhos fixos, e são capazes de capturar estas relações de longa distância com grande êxito (DENG; LIU, 2018). Sua estrutura é similar à de uma MLP, mas permite conexões entre as unidades escondidas associadas ao fluxo de informação no tempo (PASCANU; MIKOLOV; BENGIO, 2013).

Estas redes utilizam não só as informações da entrada atual u_t , mas também a saída da camada escondida no estado anterior X_{t-1} . Formalmente, o estado X_t para o tempo t pode ser definido conforme a equação 6 (PASCANU; MIKOLOV; BENGIO, 2013).

$$X_t = \sigma(\mathbf{W}_{rec}\mathbf{x}_{t-1} + \mathbf{W}_{in}\mathbf{u}_t + \mathbf{b}) \quad (6)$$

Onde σ representa uma função (elemento a elemento), geralmente tangente hiperbólica ($\tanh$) ou sigmoide, W_{rec} representa a matriz recorrente de pesos, X_{t-1} o estado anterior, W_{in} a matriz de peso de entrada, u_t a entrada do estado atual t e b o termo de viés.

A figure 4 ilustra a arquitetura de uma rede neural recorrente.

Figura 4 – Representação de uma rede neural recorrente

Fonte: Pascanu, Mikolov e Bengio (2013)

A figura 4 pode ser entendida como um desdobramento da rede no tempo. Em cada passo no tempo, uma cópia do modelo é criada e os gradientes podem ser calculados utilizando o algoritmo *Backpropagation*, que calcula as derivadas parciais nos diferentes estados. ε_t representa o erro obtido no estado t .

Variações de RNN como a *Long Short Term Memory (LSTM)* (HOCHREITER; SCHMIDHUBER, 1997) têm se destacado em tarefas de PLN, uma vez que conseguem evitar os problemas de gradiente explosivo e gradientes que se anulam, que ocorrem em sentenças de comprimento longo ao tornarem o gradiente (sinais que representam o erro) cada vez maior (explosivos) ou menor (que se anulam) durante o processo de treinamento (BENGIO; SIMARD; FRASCONI, 1994).

As redes de arquitetura *LSTM* introduzem um mecanismo de controle de memória, que definem a cada iteração quais informações (e quanto) devem ser escritas na memória e quanto do atual estado de memória deve ser esquecido (DENG; LIU, 2018). Na figura 5 é possível observar uma representação da arquitetura *LSTM*.

Figura 5 – Representação de uma rede *LSTM*

Fonte: Graves (2013)

A figura 5, representa o desdobramento da rede e seus componentes internos que têm funções de regular o fluxo de informação. Estes componentes são descritos na equação 7.

$$\begin{aligned}
 i_t &= \sigma(W_{xi}x_t + W_{hi}h_{t-1} + W_{ci}c_{t-1} + b_i) \\
 f_t &= \sigma(W_{xf}x_t + W_{hf}h_{t-1} + W_{cf}c_{t-1} + b_f) \\
 c_t &= f_t c_{t-1} + i_t \tanh(W_{xc}x_t + W_{hc}h_{t-1} + b_c) \\
 o_t &= \sigma(W_{xo}x_t + W_{ho}h_{t-1} + W_{co}c_t + b_o) \\
 h_t &= o_t \tanh c_t
 \end{aligned} \tag{7}$$

Onde σ representa a função logística, i_t , f_t , o_t representam respectivamente as três válvulas *input*, *forget* e *output*, com as funções respectivas de adicionar novas informações ao estado atual, “esquecer” informações anteriores não relevantes e definir informações para a saída. O termo c_t representa a célula de entrada, todas do mesmo tamanho que o vetor escondido (*hidden vector*) h . W representa as matrizes de peso e b os termo de viés (GRAVES, 2013).

É importante ressaltar que apesar do destaque destes métodos, quando comparados com outras áreas como processamento de imagens e vídeos, a efetividade dos modelos neurais (com muitas camadas) possui menor clareza. Em PLN, parece ser mais difícil de

obter conhecimento da linguagem natural a partir das representações aprendidas por meio dos dados textuais (DENG; LIU, 2018).

2.3.6 Modelos de língua

A criação das redes *LSTM* a partir das representações distribuídas de palavras motivaram o desenvolvimento de novos modelos e representações textuais, como por exemplo as representações contextuais (*contextual embeddings*). Estas, são capazes de representar palavras iguais com significados distintos (polissemia), dependendo do seu contexto.

O modelo de língua *Embedding from Language Models (ELMo)* utiliza redes *LSTM* bidirecionais profundas para analisar sentenças antes de atribuir um vetor de representação fixa às palavras, capturando assim seu contexto e criando representações dinâmicas (PETERS *et al.*, 2018). Seu treinamento ocorre por meio da predição da próxima palavra a partir de uma sequência de palavras e esta atividade é chamada de modelagem de língua (*Language Modeling*), sendo muito vantajosa dada que é uma tarefa que não exige rótulos (não supervisionada).

Um outro modelo de língua é o *Bidirectional Encoders Representations from Transformers (BERT)*. Recentemente desenvolvido por pesquisadores do Google, difere de todos seus predecessores pois foi treinado considerando mutualmente os contextos esquerdos e direitos, gerando resultados do estado da arte em diversas tarefas de processamento de língua natural (DEVLIN *et al.*, 2019).

Este modelo utiliza o conceito de *Transformers* bidirecionais de diversas camadas introduzido em Vaswani *et al.* (2017), que foi disponibilizado tanto na versão *Base* com 12 camadas (L), 768 camadas escondidas (H) e 12 cabeças de atenção (A), totalizando 110 milhões de parâmetros, quanto na versão *Large*, com L, H e A com os respectivos valores 24, 1024 e 16, totalizando 340 milhões de parâmetros.

A figura 6 ilustra a arquitetura completa do *Transformer*.

Figura 6 – Representação da arquitetura *Transformer*

Fonte: Vaswani *et al.* (2017)

Nesta arquitetura, é introduzido o conceito de atenção multi-cabeça. Mecanismos de atenção têm demonstrado a melhora da performance de modelos em diversas tarefas de PLN, por exemplo, caracterização autoral, classificação de sentimentos e *textual entailment* (VASWANI *et al.*, 2017). Uma representação desta arquitetura é ilustrada na figura 7.

Figura 7 – Representação mecanismo de atenção multi-cabeça

Fonte: Vaswani *et al.* (2017)

Mais especificamente, o mecanismo de atenção introduzido em Vaswani *et al.* (2017) é chamado de *Scaled Dot-Product Attention*. Este mecanismo é projetado múltiplas vezes permitindo ao modelo mutualmente avaliar informações a partir de representações de subespaços diferentes e em diferentes posições.

O mecanismo *Scaled Dot-Product Attention* é descrito por meio da equação 8.

$$Attention(Q, K, V) = softmax(\frac{QK^T}{\sqrt{d_k}})V \quad (8)$$

Este tipo de mecanismo auxilia no aprendizado de dependências em longas sequências. Conforme demonstrado, um fator chave para aprender estas dependências é a distância que os sinais *forward* e *backward* têm que percorrer na rede. Quanto menor este caminho, mais fácil de aprender estas dependências (VASWANI *et al.*, 2017).

O mecanismo multi-cabeça é descrita pela equação 9.

$$MultiHead(Q, K, V) = Concat(head_1, ..., head_h)W^O \quad (9)$$

$$Onde head_i = Attention(QW_i^Q, KW_i^K, VW_i^V)$$

Nas equações 8 e 9, Q , K e V são abstrações representadas por vetores respectivamente nomeados de *Query*, *Key* e *Value* e $\sqrt{d_k}$ representa o fator de normalização. A função de atenção faz o mapeamento entre o vetor *Query* e o par chave-valor (*key-value*). O sinal de saída é calculado a partir da soma ponderada destes valores, onde o valor

do peso da ponderação é atribuído a partir da compatibilidade entre a *Query* e chave correspondente.

Ao contrário dos modelos tradicionais, *BERT* não é treinado por meio da inferência de palavras dado um contexto unidirecional, mas por meio de duas outras tarefas *Masked LM* (MLM) e *Masked Sentence Prediction* (NSP).

Na primeira tarefa, 15% de todos os *tokens* do corpus são mascarados aleatoriamente com o objetivo de serem inferidos durante o pré-treino. A ideia é que este mecanismo permita o aprendizado por meio de um contexto bidirecional verdadeiro, ao contrário de métodos que concatenam um modelo treinado da esquerda para a direita com outro treinado da direita para esquerda. Na segunda tarefa, o modelo é treinado para inferir se uma sentença é subsequente à outra, sendo que 50% das vezes a sentença B é de fato subsequente à sentença A e nos outros 50% é escolhida uma sentença aleatória (VASWANI *et al.*, 2017).

É possível ainda, efetuar um ajuste fino (*fine-tuning*) nos pesos do modelo para tarefas ou corpus específicos por meio da substituição dos sinais de entrada e saída do modelo *BERT*. Este procedimento é relativamente pouco custoso quando comparado ao processo de pré-treino, mas ainda possui um custo considerável em relação à modelos mais tradicionais.

2.3.7 Considerações

Neste capítulo foram definidos os principais conceitos tomados como base para este trabalho. Foi apresentado uma descrição inicial do corpus e-SIC, o qual será utilizado no estudo exploratório apresentados no capítulo 4 e no desenvolvimento da pesquisa proposta. Por fim, foram apresentados os métodos de representações textuais e de aprendizado.

É importante ressaltar que existem outras formas de contornar a questão dos gradientes explosivos ou que se anulam. Uma arquitetura recentemente desenvolvida, considerada uma versão simplificada das redes *LSTM*, é conhecida como *Gated Recurrent Unit*(GRU) (CHO *et al.*, 2014). A diferença principal entre estas redes é que uma única válvula controla simultaneamente a informação a ser esquecida e a decisão de atualizar o estado atual. Outras variações podem ser exploradas, entretanto, nenhuma possui melhor desempenho que *LSTM* e *GRU* em diversas tarefas (GOODFELLOW; BENGIO; COURVILLE, 2016).

3 Revisão Bibliográfica

Este capítulo apresenta uma revisão bibliográfica exploratória sobre estudos de satisfação de usuários a partir de textos. Os trabalhos aqui considerados foram selecionados a partir dos repositórios ACL Anthology, Scopus e IEEE e foram privilegiados artigos mais recentes e de maior fator de impacto. Para facilitar a discussão, esta revisão é dividida em estudos sobre a modelagem de satisfação de usuários (Seção 3.1), estudos de modelagem de diálogos (Seção 3.2) e técnicas com base em AS (Seção 3.2). É apresentado ainda, um resumo dos estudos e considerações finais (Seção 3.4) sobre esta revisão.

Por fim, vale mencionar que a escolha dos temas para revisão bibliográfica, satisfação (de modo mais amplo), diálogos e análise de sentimentos se deu com o objetivo de explorar diferentes formas de modelar o problema, uma vez que a modelagem da satisfação de usuários em serviços públicos de solicitação de acesso por meio de técnicas de aprendizado neural é um tema pouco explorado na literatura.

3.1 *Modelagem de satisfação de usuários*

Nesta seção são apresentados os estudos que propõem modelar o problema de satisfação de usuários. São abordadas diversas técnicas de aprendizado bem como conjuntos de dados de domínios distintos.

3.1.1 Satisfação do usuário em sistemas de perguntas e respostas

No estudo em Liu, Bian e Agichtein (2008) é apresentado um modelo para a predição da satisfação de usuários de plataformas de perguntas e respostas (CQA), como *Yahoo! Answers*. Além disso, são exploradas as características mais relevantes para a satisfação a partir do ponto de vista do questionador.

Nestas plataformas, o questionador faz a pergunta e aguarda os usuários responderem, os quais podem votar e comentar nas respostas de outros usuários. Uma vez que o questionador avalia as respostas, ele pode selecionar a que considera melhor e, dependendo da plataforma, atribuir uma nota e comentário, encerrando o *post*. Neste ponto, é considerado que o usuário está satisfeito. Por outro lado, alguns questionadores nunca encerram ou avaliam as respostas. Neste caso, a resposta mais votada por outros

usuários poderia ser considerada a correta, mas não necessariamente refletindo o ponto de vista do questionador. Portanto, a resposta destes casos é considerada como insatisfeito. Em resumo, o questionador é considerado satisfeito quando encerra o *post* selecionando e avaliando a melhor resposta com uma nota mínima de três estrelas. Caso contrário, é considerado insatisfeito.

Para a modelagem do problema, são considerados seis grupos de características.

1. **Perguntas:** Composta por quantidade de caracteres da pergunta, do título e da descrição, horário do *post* e avaliação recebida pela pergunta.
2. **Relação entre Perguntas e Respostas:** Composta por quantidade de caracteres da resposta, respostas, votos positivos e negativos, máximo de votos positivos ou negativos, entre outras.
3. **Histórico do questionador:** Nesta categoria foi avaliado o histórico de satisfação do usuário em relação a respostas passadas.
4. **Histórico do usuário provedor da respostas:** Neste caso, avaliou-se a quantidade de respostas dadas, de *posts* resolvidos e de respostas marcadas como as melhores. Além disso, também foi avaliada a reputação do usuário, média de respostas e a resposta com mais votos positivos.
5. **Categoria da pergunta:** Partindo da hipótese de que existem variações de comportamento de acordo com a categoria da pergunta, a média da nota de satisfação do questionador a priori é dividida por categoria.
6. **Textuais:** Nesta categoria são utilizados unigramas e bigramas (apenas os 1000 mais frequentes) tanto para as perguntas quanto para as respostas.

No experimento, é utilizado o *software* WEKA implementando os classificadores Máquina de Vetor Suporte (SVM), Árvores de Decisão (C4.5 e Floresta Aleatória), *AdaBoost* e *Naive-Bayes*. As métricas de avaliação utilizadas são acurácia, precisão, revogação, e medida F.

Além da nota do questionador, considerada como verdadeira, são definidas outras duas referências para as respostas. Uma por meio de um serviço *Mechanical Turk* e outra definida pelos próprios pesquisadores. A nota atribuída pelos terceiros obteve maior correlação com a nota do questionador, e portanto este *baseline* foi utilizado.

Uma primeira análise foi efetuada comparando-se as características e os classificadores. O classificador de Árvore de Decisão C4.5 apresentou o melhor desempenho na

medida F em todas as variações de características e atingindo o desempenho máximo de 0,77% quando selecionado, por meio da técnica de Ganho de Informação, as 15 melhores características. Por meio desta técnica, a análise dos resultados mostra que a característica de histórico de avaliação do questionador é a mais importante.

Em uma segunda análise, o tamanho do conjunto de dados é modificado. Foram testadas desde 50 até 5000 perguntas, sendo obtido 0,7 na medida F com uma amostra de apenas 500 perguntas e 0,75 para uma amostra de 2000 perguntas.

Também é testada a abordagem *Offline vs Online*. Os experimentos anteriores são considerados como *offline* pois as respostas já estavam disponíveis, permitindo assim explorar uma diversa quantidade de características. Nesta análise, tenta-se prever a satisfação em tempo real, sem nenhuma resposta. Neste caso, os resultados mostram uma queda no desempenho do modelo de 0,77 para 0,74 na medida F. Apesar da redução, os resultados ainda foram superiores aos modelos *baselines*.

Com o objetivo de avaliar a importância de cada categoria de característica, testa-se a remoção de cada um dos grupos. A análise dos resultados mostra que as características relacionadas à pergunta e ao histórico do questionador são as que mais sensibilizam o desempenho do modelo, ao contrário das características do histórico do usuário e relação pergunta-resposta. Por fim, outras análises em relação às características textuais e ao histórico do questionador também foram efetuadas, resultando em um aumento significativo do desempenho do modelo para grupos de questionadores com pelo menos de uma pergunta na plataforma.

3.1.2 Mensuração de satisfação em tempo real.

No estudo em Park e Gates (2009) é proposto um modelo para avaliar a satisfação de clientes por meio dos textos transcritos automaticamente a partir de chamadas em centrais de atendimento. Os principais objetivos são identificar as característica mais relevantes associadas à satisfação e também os classificadores com melhor desempenho e alta acurácia. Para tanto, são testados diversos classificadores de aprendizado como Árvore de Decisão, *Naive-Bayes*, Regressão Logística e Máquina de Vetor Suporte (SVM).

Como motivação, é avaliada a frequência de palavras positivas e negativas e é concluído que apenas esta informação é insuficiente para prever a satisfação, sendo

portanto, necessário identificar as principais características que, de acordo com as hipóteses enunciadas, possuem alta correlação com a satisfação. Foi analisada a quantidade de palavras consideradas positivas e negativas nas ligações de clientes satisfeitos e insatisfeitos, concluindo que menos de 10% dos insatisfeitos utilizam o mesmo número ou mais de palavras negativas.

Diferente da avaliação de um produto ou serviço, a opinião dos usuários não está necessariamente explícita nos diálogos. Assim, é necessário explorar outros fatores que impactam na satisfação dos clientes, como a qualidade do serviço e o tempo para resolução do problema e recebimento de ofertas. Estes fatores são capturados de diversas fontes.

No estudo, são avaliados dois cenários referentes às notas de satisfação. No primeiro cenário, as chamadas são avaliadas com uma nota de um a cinco, representando respectivamente Completamente insatisfeito, Um pouco insatisfeito, Neutro, Um pouco satisfeito, Completamente satisfeito. No segundo cenário, uma avaliação binária é definida, empregando apenas as notas Satisfeito ou Insatisfeito.

Com o objetivo de identificar as características mais relevantes para o problema, são avaliados 16500 comentários relacionados a resultados acerca de pesquisas de satisfação em uma companhia automotiva. Outras características a respeito das ligações também são avaliadas, como por exemplo, tempo de ligação, palavras relacionadas a sentimentos, nomes de competidores, velocidade da fala, entre outras. Ao todo, são utilizadas quatro categorias de características: estruturais, prosódicas, léxicas e contextuais.

No experimento, são utilizadas configurações padrões do *software RapidMiner* no conjunto de dados transcrito automaticamente pela ferramenta *IBM Attila Speech recognition*. Dado que o processo de anotação manual da variável resposta é muito custoso e ainda pode agregar erros de viés, é utilizado o resultado dos questionários de satisfação respondidos pelos próprios clientes. Foram utilizados dois *baselines* para a nota de satisfação. O primeiro *baseline* é conhecido como Classe Dominante, com base nas classes mais frequentes, sendo 5 para satisfeito e 2 para insatisfeito. A nota do segundo *baseline* foi determinada pelo julgamento do próprio atendente ao fim da chamada.

No modelo de cinco pontos na nota de satisfação, o *baseline* de Classe Dominante apresentou uma acurácia de 0,57. A análise dos resultados mostra uma melhora de aproximadamente 15% utilizando o classificador SVM com núcleo radial, atingindo um total de 0,66 na métrica acurácia. Já no modelo com resposta binária, os *baselines* de Classe Dominante e julgamento do atendente apresentaram uma acurácia de 0,76 e 0,86

respectivamente. Os classificadores que obtiveram o melhor desempenho, com acurácia de 0,89, foram Árvore de Decisão e SVM.

Com o objetivo de avaliar a contribuição de cada categoria de característica, são executados diversos testes por meio da estratégia *leave-one-out*, que remove um dos tipos por vez, estruturais, prosódicas, léxicas e contextuais. Os resultados mostram que em todos os testes, exceto Árvore de Decisão para classificação binárias, o modelo que utiliza todas as características obteve o melhor desempenho.

Por fim, partindo da hipótese de que é possível medir a satisfação dos clientes no decorrer da ligação, ou seja, em tempo real, são utilizados textos transcritos a partir da primeira metade dos diálogos. Foram utilizados apenas os classificadores Árvore de Decisão e SVM, uma vez que estes obtiveram o melhor desempenho no primeiro experimento. As análises dos resultados mostram que ambos atingiram maior acurácia ao utilizar o diálogo completo. Entretanto, quando comparados o modelo *baseline* de julgamento do atendente, obtiveram melhores resultados utilizando apenas a primeira metade do diálogo.

3.1.3 Modelagem de usuários com Redes Neurais para predição de avaliações de usuários

No estudo em Tang *et al.* (2015) é proposto um modelo de redes neurais para predição da satisfação de usuários considerando não apenas as informações semânticas dos textos mas também as relacionadas aos usuários. Além de utilizar técnicas baseadas em léxicos, também é apresentado um novo modelo vetorial denominado *Userword Composition Vector Model (UWCVM)*.

Com o objetivo de extrair automaticamente as melhores características a partir dos dados, é argumentado que aprender as representações textuais a partir de redes neurais. Contudo, estas arquiteturas não apresentam bom desempenho quando utilizadas diretamente na tarefa de predição da satisfação de usuários, uma vez que diferentes usuários podem utilizar palavras distintas para representar os mesmos sentimentos. Da mesma forma, palavras similares podem representar sentimentos distintos.

Em síntese, é proposto modificar léxicos semânticos por meio da inclusão de informações dos usuários. Estas modificações são executadas mediante operações de multiplicações entre os vetores de cada palavra e as matrizes que representam cada usuário. Estas matrizes mapeiam a representação original das palavras e as suas respectivas

modificações. Cada representação de palavra modificada é utilizada como entrada em outro modelo, o qual constrói a representação vetorial do documento (a avaliação). Por fim, esta representação é utilizada como entrada em uma rede neural que irá desempenhar a predição da satisfação do usuário.

No experimento, são utilizados dois conjuntos de dados, um de avaliação de filmes, *Rotten Tomatoes*, e outro de restaurantes, *Yelp Dataset Challenge 2013*. O problema é tratado como uma classificação com diversos rótulos e são utilizadas as métricas de Erro absoluto médio (MAE) e Raiz do erro quadrado médio (RMSE) para mensurar a diferença entre as notas preditas e reais. Para comparação dos resultados são definidos diversos *baselines*.

1. *Majority*: A nota mais frequente no conjunto de dados de treinamento é atribuída para cada uma das observações no conjunto de dados de teste.
2. *BOW*: Classificador de Máquina de Vetor Suporte (SVM) com as revisões representadas como *Bag of words*.
3. *BOW + BOO*: Além da representação *Bag of words*, é incorporado *bag-of-opinion* utilizando léxicos de sentimentos.
4. *VecAvg*: Classificador SVM com as revisões são representadas por meio do cálculo da média dos seus respectivos vetores de palavras aprendidos por meio do método *word2vec*.
5. *RAE*: Autoencoders recursivos treinados com vetores de palavras aprendidos com *word2vec*.
6. *PVDM*: Técnica de estado da arte conhecida como *Distributed Memory Model of Paragraph Vectors*, a qual apresenta alto desempenho em diversas tarefas de análise de sentimentos.
7. *CNN*: Arquitetura redes neurais convolucionais, as quais também têm demonstrado alto desempenho.

A análise dos resultados mostra que o modelo proposto superou todos os *baselines*, tanto na métrica MAE quanto na RMSE, para os dois conjuntos de dados.

3.1.4 LSTM hierárquico com aprendizado conjunto para estimação da satisfação de clientes em chamadas de Call-Centers

No estudo em Ando *et al.* (2017) é apresentada uma estratégia para a predição da satisfação de clientes em *Call-Centers* a partir da modelagem conjunta em dois níveis do diálogo. A abordagem se resume a primeiramente modelar a parte do diálogo onde apenas o cliente se comunica (*turn-level*) e aplicar as probabilidades estimadas resultantes no modelo que utiliza a informação do diálogo completo (*call-level*). Diferente de métodos tradicionais, a abordagem proposta não trata os segmentos do diálogo de forma independente. Ao contrário, utiliza o modelo de *Memória de longo e curto prazo (LSTM)* para capturar as relações sequenciais de longa distância. Além disso, com o objetivo de melhorar as estimativas dos dois modelos conjuntamente, é utilizada uma técnica de *joint learning* (HASHIMOTO *et al.*, 2017).

O conjunto de dados utilizado é composto por chamadas gravadas de operadores em *Call-Centers* e também por notas escritas destas chamadas. O tema abordado nas ligações é referente à venda de comida congelada, havendo tarefas de compras, solicitações e cancelamentos. A partir das gravações, são criados cenários com diferentes situações em termos de contexto e emoções expressas durante os diálogos. Além disso, os rótulos são definidos manualmente com cinco níveis, muito positivo, positivo, neutro, negativo e muito negativo. Diversas técnicas são utilizadas para segmentação entre cliente e operador a partir dos áudios das ligações.

Prévio à modelagem, duas análises são efetuadas no conjunto de dados. Na primeira, são analisadas as polaridades dos turnos com base em sua continuidade, ou seja, a frequência de chamadas identificadas com polaridade positivas ou negativas dada uma sequência de turnos positivos ou negativos. Nesta análise, é confirmada a importância da utilização da informação dos turnos de forma dependente, capturando os contextos. Na segunda análise, é avaliada a relação entre a polaridade das chamadas apenas no nível do turno do cliente e no nível de toda a chamada. São identificadas diferentes relações entre estes níveis, as quais levaram a proposição de duas hipóteses. A primeira é que a informação referente ao turno do cliente pode ser utilizada para estimação da satisfação da chamada como um todo, e a segunda é que a utilização da modelagem mútua aumentaria o desempenho da estimação nos dois níveis.

O método utilizado para a modelagem do problema é a sobreposição (*stacking*) de dois tipos de arquitetura *LSTM-RNN*. A primeira rede tem como objetivo estimar a satisfação do cliente utilizando apenas o seu turno no diálogo. Esta rede utiliza uma arquitetura com dois tipos de *LSTM*, uma unidirecional e outra bidirecional, e a ideia é capturar informações sequenciais de longa distância. A segunda rede tem como objetivo estimar a satisfação do cliente utilizando a informação de todo o diálogo, e sua arquitetura *LSTM-RNN* utiliza múltiplas entradas (inclusive informações de saída da primeira rede) e apenas uma saída. Os dois modelos combinados são otimizados por meio de uma técnica de aprendizado conjunto de múltiplas tarefas. Devido à diferenças na convergência durante a otimização, a rede inferior é primeiramente treinada adicionando um peso λ na função de perda.

Para o experimento, é definido um modelo *baseline* a partir do classificador de Máquina de Vetor Suporte com núcleo radial. Para a avaliação, os modelos desenvolvidos utilizam 21 características prosódicas, 12 léxicas e 11 interativas, sendo 4 classificadores variando entre *LSTM* unidirecional e bidirecional e com inclusão ou não de aprendizado conjunto. Os resultados mostram que todos os modelos propostos obtiveram melhor desempenho que o *baseline*. O aprendizado conjunto para otimização também melhorou o desempenho dos classificadores, tanto em nível de predição da satisfação do cliente utilizando apenas o seu diálogo, como na predição da satisfação do cliente com base no diálogo completo.

3.2 Análise de sentimentos

Nesta seção são apresentados estudos abordando algumas das estratégias para modelagem do problema da satisfação de usuários, que é frequentemente tratado por meio de técnicas de análise de sentimentos.

3.2.1 Predição de avaliações para revisões de serviços e produtos

No estudo em Gupta, Fabbri e Haffner (2010) é proposto um modelo para predição de satisfação baseado na análise de sentimentos orientada a diferentes aspectos da revisão textual, pois de acordo com os autores, as revisões baseadas apenas em estrelas falham em capturar diversas opiniões. Para o estudo foram desenvolvidos modelos de regressão, que atribuem uma nota numérica para os aspectos avaliados, e modelos de

classificação. No corpus utilizado as notas variam de 1 a 5, sendo que os aspectos examinados foram comida, serviço, ambiente, valor e experiência geral.

O problema é abordado por meio de dois modelos. O primeiro é um modelo de regressão ordinal, a qual utiliza um espaço discreto de 1 a 5 como variável resposta e *rank loss* como função a ser minimizada. O segundo é um modelo de classificação, utilizando cada nota como uma classe a ser predita. Para os experimentos, os algoritmos de regressão são implementados utilizando uma técnica de regressão numérica baseada em rede neural treinada através de *back-propagation* e uma de *Ranking* baseada em uma rede *Perceptron*, chamada de *PRank*. O algoritmo de classificação é implementado utilizando classificadores binários do tipo Máxima Entropia.

Também são utilizadas diferentes características como unigramas, bigramas, *Word chunks* e *Parts-of-Speech Chunk*. *Chunks* representam um conjunto maior de palavras, normalmente caracterizando uma frase nominal, a qual não apresenta um verbo na sua composição. Além disso, como muitos sistemas de revisão utilizam apenas uma única nota geral para toda a avaliação, esta nota é utilizada no experimento como uma característica para prever a avaliação dos aspectos individuais. Por fim, para a seleção de características, e objetivando reduzir a esparsidade, filtros como ocorrência mínima de palavras são aplicados para os unigramas e bigramas. Apesar da simplicidade da representação de características por meio de unigramas, estes apresentaram o melhor desempenho no experimento inicial, assim como a utilização da característica da nota geral. Para os demais experimentos é utilizada apenas a representação por unigramas e para que um viés não fosse incluído, a característica nota geral não é utilizada.

Para avaliação e comparação dos resultados, primeiramente é gerado um *baseline* calculado por meio da função *rank loss*, utilizando a nota máxima (cinco) e a nota real da revisão. Como resultado do modelo *baseline*, a média do *rank loss* nos aspectos avaliados foi superior a 1.

Nos resultados, os aspectos ambiente e valor obtiveram o pior desempenho. Isso se deve à falta de informação nos textos relacionadas a estes aspectos, sendo observado manualmente que apenas 62% dos usuários avaliaram estes aspectos. No resultado geral, a técnica Máxima Entropia superou os outros algoritmos em todos os aspectos, apresentando resultados coerentes com estudos similares.

3.2.2 Detecção de sentimentos em e-mails de atendimento de clientes

No estudo em Gupta, Gilbert e Fabbrizio (2010) é proposto um modelo para identificação de sentimentos negativos como insatisfação e frustração em e-mails de clientes, auxiliando por exemplo, na redução de evasão. Este problema é abordado como sendo de classificação textual e o objetivo é melhorar a acurácia quanto ao modelo *baseline*, o qual obteve o desempenho de 0,52 de medida F.

De acordo com os autores existem muitos trabalhos similares fundamentados em técnicas baseadas em léxicos, contudo, estas técnicas falham ao capturar contexto semântico e, conseqüentemente, seu desempenho não tem sido satisfatório. Outros trabalhos têm obtido melhores resultados utilizando unigramas de palavras, conseguindo identificar mensagens hostis com sucesso. Entretanto, este estudo propõe uma análise mais profunda ao tentar identificar a causa destas mensagens hostis.

A estratégia proposta para o problema abordado é utilizar uma técnica conhecida como *Boostexter*. Trata-se de um algoritmo baseado em *Boosting*, que combina diferentes variações de classificadores mais fracos para produzir melhores resultados.

São definidas oito características com o objetivo de identificar e-mails hostis, Emoções Negativas, Emoções negativas em relação à empresa, Ameaças de deixar o negócio, Ameaça de informar autoridades, Ameaça de ação legal, Justificativas para um melhor tratamento, Termos de desassociação da companhia e Descrição do erro cometido pela companhia. Para cada uma destas características, listas são criadas manualmente com uma série de frases extraídas do conjunto de dados. Além disso, estas listas são enriquecidas criando variações das frases e palavras-chaves encontradas em e-mails críticos.

No experimento, são utilizados 620 e-mails no treinamento e 457 no teste. Devido ao tamanho limitado do conjunto de dados, a avaliação cruzada é empregada para apurar os resultados a partir das métricas de acurácia, precisão, revogação e medida F.

Os resultados de menor desempenho foram mensurados no modelo *baseline*, onde foi utilizada apenas a característica n-gramas. Isso se deve à esparsidade dos dados. Também foi testado um modelo baseado em regra, o qual caracterizava um e-mail como hostil com base na frequência de manifestação no termos das oito características pré-definidas, utilizando um limite de duas, três ou quatro manifestações ou mais. O melhor desempenho

foi mensurado quando um e-mail possui três ou mais características, atingindo 0,68 na medida F.

Por fim, é demonstrado o resultado de classificadores treinados apenas com as oito características e também com a adição de n-gramas. Os resultados mostram que o classificador *Boostexter* é capaz de aprender melhores regras quando comparado ao modelo que conta a frequência das características pré-definidas, atingindo 0,72 na medida F.

3.2.3 Análise de sentimentos de Tweets utilizando comitês de classificadores

No estudo em Silva, Hruschka e Hruschka (2014) é proposta uma abordagem para classificação automática de *tweets* utilizando comitês de classificadores e Léxicos. A hipótese do estudo é de que pode-se obter uma classificação acurada de *tweets* por meio da utilização destes comitês juntamente com técnicas mais adequadas de construção de características. Além disso, a utilização de comitês de múltiplos classificadores, combinados com escores obtidos por léxicos, pode melhorar a acurácia do classificador final.

No problema apresentado, normalmente são empregados modelos tradicionais de representação de *tweets*, como *Bag of words*. Entretanto, eles geram matrizes muito esparsas. Para solucionar esta dificuldade o modelo proposto emprega a técnica de *Hashing*, a qual tem produzido bons resultados quando o custo computacional é elevado. Além disso, também é utilizado um léxico de opinião criado a partir de uma lista de 4783 palavras negativas e 2006 positivas e *Emoticons*, que agregam informações sobre os sentimentos dos textos.

Também são utilizadas técnicas de pré-processamento como remoção de *stop words*, links, pontuações e outros, além da execução de *stemming*, contribuindo para diminuição da esparsidade das matrizes.

São utilizados os classificadores Florestas Aleatórias, Máquinas de Vetor Suporte, Naive-Bayes Multinomial e Regressão Logística. Uma vez treinados, o modelo da combinação é formado pela média das probabilidades das classes obtidas por cada classificador individualmente ou pelo voto majoritário destes classificadores.

Nos experimentos é utilizado um conjunto de dados de diferentes domínios, *Tecnologia: Sanders* — *Twitter Sentiment Corpus*¹, *Sem domínio: Stanford* — *Twitter Sentiment Corpus* (GO; BHAYANI; HUANG, 2009), *Política: Obama-McCain Debate (OMD)*

¹ <http://www.sananalytics.com/lab/twitter-sentiment/>

(SHAMMA; KENNEDY; CHURCHILL, 2009) e *Saúde: Health care reform (HCR)* (SPERIOSU *et al.*, 2011). Além disso, os testes foram conduzidos por meio da plataforma *Weka*.

Nos resultados são avaliados os classificadores individuais e também as combinações. O modelo de combinação mostra-se superior na acurácia, assim como os modelos que combinavam léxicos. Além disso, são analisadas diferentes agregações de *Bag of words* (*BoW*), Hashing e Léxicos. É concluído que, quando a acurácia é uma métrica mais importante, deve-se dar preferência à representação *Bag of words*. Já quando o custo computacional é relevante, a representação Hashing é preferida.

3.2.4 Refinando Embedding de palavras para análise de sentimentos utilizando escores de intensidade

No estudo em Yu *et al.* (2018) é apresentado uma estratégia para o refinamento do aprendizado de *Embeddings* de sentimentos. Um dos motivadores deste trabalho é que as técnicas tradicionais falham ao capturar informações suficientes sobre os sentimentos, criando vetores similares de palavras com polaridades distintas. Além disso, pesquisas recentes sugerem melhorias significativas no desempenho dos modelos de classificação de sentimentos ao incluir *Embeddings* de sentimentos às informações já representadas pelos modelos tradicionais de *Embeddings* de palavras.

A melhoria dos modelos, através do aprendizado de *Embeddings* de sentimentos, decorre do emprego de léxicos de sentimento, ou seja, bases que associam escores e quantificam o sentimento de cada palavra. A comparação entre o modelo desenvolvido (refinado) com os tradicionais é efetuada utilizando os conjuntos de dados *SemEval-2013 Task 2: Sentiment analysis in Twitter* (NAKOV *et al.*, 2013) e o *Stanford Sentiment Treebank (SST)* (SOCHER *et al.*, 2013).

Além disso, são definidos e implementados quatro classificadores de redes neurais profundas com diferentes arquiteturas, Redes Neurais Convolucionais, *Deep Averaging Networks*, *Memória de longo e curto prazo (LSTM)* e *Tree-LSTM*. Para fins de comparação, cada classificador é implementado utilizando *Embeddings* convencionais de palavras, *Embeddings* de sentimentos e o modelo refinado proposto.

O autor propõe a seguinte sequência de passos, representando o algoritmo utilizado no refinamento de cada palavra.

1. A distância cosseno é calculada entre os vetores da palavra alvo e das palavras no léxico.
2. As palavras mais similares são escolhidas e ordenadas de acordo com seus escores de intensidade.
3. Os vizinhos mais próximos e com um escore de intensidade similar serão ordenados de modo decrescente.
4. O modelo então atribui diferentes pesos de acordo com a ordenação e ajusta a representação vetorial da palavra-alvo.

Nos resultados dos dois conjuntos de dados, tanto *Embeddings* de sentimentos quanto o modelo proposto superaram os modelos tradicionais de *Embeddings* de palavras. Entretanto, os modelos refinados resultam em um melhor desempenho devido à sua capacidade em capturar dissimilaridades entre os sentimentos de cada palavra, mesmo quando estas são semanticamente análogas.

3.3 Modelagem de diálogos

Dado que a natureza do problema abordado pode ser tratada como um diálogo, no sentido de que pode haver uma comunicação entre um solicitante e um atendente, nesta seção são apresentados estudos que abordam a modelagem da satisfação do usuário através de diálogos.

3.3.1 Modelando transições na satisfação do usuário em diálogo a partir de avaliações

No estudo em Higashinaka *et al.* (2010) é proposto um modelo para a predição da satisfação de usuários a partir de uma nota de avaliação. A estratégia é empregar diversos modelos ocultos de *Markov* para cada sequência de diálogo com suas respectivas notas, e então combinar todos estes modelos em um único para obter a predição final. De acordo com os autores, o modelo oculto de *Markov* é utilizado pois é apropriado para sequências que mudam com o tempo. Para treinar os modelos, é utilizado o algoritmo de maximização da expectativa.

Para o experimento foram empregados dois domínios, e nos diálogos foram utilizados textos para evitar problemas intrínsecos à voz. O primeiro domínio trata de diálogos a

respeito de animais, onde os usuários escreviam através de uma interface de *chat* textual o que gostavam ou não sobre este tema. Foram coletados 1000 diálogos entre 50 usuários, contentado três rótulos, Suavidade da conversa, Proximidade percebida orientada ao sistema e Tendência a continuar a conversa. Para cada rótulo é definida uma nota de 1 à 7. O segundo domínio trata de diálogos sobre escuto ativa, ou seja, o ouvinte dialoga com o usuário com o propósito de fazer ele ou ela continuar sendo ouvido. Da mesma forma, a captura dos dados se deu por meio de uma interface de comunicação de *chat* textual e os diálogos também foram rotulados.

São propostas cinco variações de topologias para os Modelos ocultos de *Markov*.

1. *Ergodic0*: O modelo ergótico mais simples. Possui sete *SHMM* - *Speaker Hidden Markov Model*, definida a partir de dois estados, cada um emitindo diálogos de um dos participantes. Todas as probabilidades iniciais e de transição são iguais.
2. *Ergodic1*: Este modelo ergótico difere do anterior apenas por possuir um *SHMM* adicional, totalizando oito.
3. *Ergodic2*: Similar ao modelo anterior *Ergodic1*, entretanto, para que haja uma maior acurácia dos resultados, possui o dobro de estados.
4. *Concat1*: Similar ao modelo *Ergodic1*, entretanto, os modelos são combinados utilizando treinamento concatenado (LEE, 1988).
5. *Concat2*: Mesmo modelo do *Concat1*, entretanto, possui o dobro de estados.

Também para o experimento, são criados dois modelos *baselines*, um com as notas definidas aleatoriamente (de 1 a 7), e outro com a topologia similar ao modelo *Ergodic0*, porém, com as probabilidades definidas a partir de um treinamento supervisionado.

Para avaliação dos modelos é utilizada a técnica de validação cruzada com dez partições e também dois critérios, um para avaliar *matches* individuais e outro para avaliar as distribuições. A avaliação do *match* é calculada por meio do somatório de *matches* entre a nota prevista e a nota real. Como métrica de avaliação, também é calculada o Erro absoluto médio. Para avaliação da distribuição entre as notas, é utilizada a métrica da divergência de *Kullback-Leibler* (KULLBACK; LEIBLER, 1951).

Na avaliação dos resultados, o modelo supervisionado se sobressaiu em todas as métricas examinadas e também nos dois domínios. Entretanto, comparando os modelos desenvolvidos, é concluído que a adição de estados na topologia, assim como o treinamento concatenado, melhoram a acurácia da predição.

3.3.2 Identificando fatores sociais e expressivos em textos de solicitações utilizando modelos transacionais/sequenciais

No estudo em Munkova, Munk e Fraterova (2013) é proposto um modelo para identificação de características textuais que reflitam polidez, cortesia e boas maneiras durante uma solicitação escrita via formulário. Os dados utilizados no estudo foram coletados em um ambiente universitário. De acordo com os autores, polidez é uma forte característica da linguagem e comunicação, a qual também está relacionada com fatores culturais através de expressões e construções gramaticais, podendo inclusive servir como insumo para comparação de diferentes línguas.

A avaliação das solicitações é baseada na teoria da polidez de Brown e Levinson (1987), a qual afirma que cada indivíduo participante da conversa cria sua própria forma de expressão, utilizando e mesclando diversos fatores de polidez. Por este motivo, é proposto avaliar as regras para construção destas formas de expressão. Para isso, é formulado um modelo transição/sequencial para investigar as similaridades e diferenças destas formas, tanto na língua mãe (Eslováquia) quanto na língua estrangeira (Inglês).

É explicado a natureza da requisição e seus principais elementos, alertas, modificadores externos e modificadores internos. Estes elementos são conhecidos como *modality markers* e são divididos em dois grupos, os que intensificam a força da requisição e os que diminuem o seu impacto. Além disso, é explicado que a requisição do solicitante pode ser percebida em diferentes perspectivas, Orientada ao solicitante, Orientada ao receptor, Orientada ao solicitante e ao receptor, ou ainda, Impessoal.

Para efetuar a extração de informação dos textos, é utilizado um modelo baseado em regra e definiram nove fatores de polidez, Alertas, Perspectiva do solicitante, Perspectiva do receptor, Marcas de polidez, Pré-Sequências, Pós-Sequências, Dispositivos de mitigação, Minimizadores, Elogios.

Para a modelagem do problema, é utilizada uma técnica similar ao *Bag of words* a qual chamaram de Transação/Sequencial. Uma transação é definida como um conjunto de fatores observados nos textos. A análise foi efetuada por meio de regras de associação, onde a ordem dos fatores utilizados não são consideradas.

Os dados utilizados foram coletados de formulários preenchidos por estudantes de linguística, os quais necessitavam se comunicar entre si e seus professores, tanto de forma falada quanto escrita. Foram coletados 1000 formulários, sendo uma metade em

Inglês e a outra em Eslováquia. Usando tabulação cruzada foram investigadas as diferenças nos aspectos das solicitações nas duas línguas. As análises dos resultados mostram que a maioria dos fatores observados ocorrem em diferentes frequências com relação às línguas avaliadas.

Na avaliação por regra de associação, as interações entre os pares de fatores são avaliadas por meio da probabilidade de ocorrência conjunta e individual dos pares, representada por um modelo em grafos, nas duas línguas. Foram identificadas diferenças significativas entre as frequências de ocorrência dos fatores nas línguas avaliadas, provando que os modelos transação/sequencial para representação de textos curtos são uma ferramenta adequada na avaliação das relações dos atributos examinados.

3.3.3 Em direção à avaliação automática de diálogos em Helpdesk de clientes

No estudo em Zeng *et al.* (2018) é proposto um modelo para avaliar automaticamente a qualidade de diálogos no contexto de atendimento ao cliente. É definido dois meios de avaliação. A primeira é baseada na qualidade subjetiva, onde é avaliado se o problema foi claramente declarado pelo cliente, se o problema foi resolvido, a satisfação do usuário, a abordagem do atendente e a abordagem do cliente. A segunda forma de avaliação é definida a partir de *nuggets*, ou seja, sequências de diálogos que auxiliam na resolução do problema. De acordo com os autores, marcações de *nuggets* nos diálogos podem auxiliar na identificação de sua qualidade. Para estimação da qualidade dos diálogos, é desenvolvida uma terceira abordagem baseada em redes neurais *Neural Evaluation Machine (NEM)*, com o objetivo de capturar aspectos como contexto e sentimentos.

Os dados utilizados no estudo foram extraídos de diálogos reais de sistemas de atendimento ao cliente. Os rótulos referentes à qualidade do diálogo foram definidos manualmente por três diferentes colaboradores, assim como os tipos *nuggets*.

1. *CNUG0*: Refere-se a partes do diálogo do cliente que o fizeram entrar em contato com o atendimento. Basicamente são as declarações referentes ao problema inicial.
2. *HNUG*: Refere-se a partes do diálogo consideradas úteis para resolução do problema do cliente.
3. *CNUG*: Refere-se a partes do diálogo do cliente que são úteis para esclarecer o problema ao atendente.

4. *HNUG**: Refere-se a partes do diálogo do atendente que especificamente resolvem o problema do cliente.
5. *CNUG**: Refere-se a partes do diálogo do cliente que explicitamente declaram ao atendente que o problema foi resolvido.

Para calcular a qualidade dos diálogos, é mensurado aspectos como o esforço necessário para resolução do problema, que é medido com base no comprimento dos diálogos, calculado de acordo com a quantidade máxima de turnos do cliente ou do atendente. Também é mensurada a posição dos *nuggets* no texto, partindo do princípio de que a qualidade do texto é maior o quanto antes estas partes forem declaradas no diálogo. Para o caso de múltiplos *nuggets*, duas abordagens foram definidas. Na primeira, a média é calculada para obtenção da nota final, e na segunda ocorre uma consolidação de todas as partes do diálogo definidas como *nuggets* e então uma heurística é aplicada.

Na abordagem *NEM*, é utilizado o *embeddings* de palavras pré-treinado *word2vec* com 256 dimensões como entrada para uma arquitetura de redes neurais recorrentes com células unidirecionais de *LSTM*.

Na avaliação dos resultados são utilizadas as métricas de *correlação de Pearson* e erro quadrado médio. O modelo proposto *NEM* mostrou o melhor desempenho em quase todos os aspectos avaliados, exceto no aspecto de qualidade da informação disponibilizada pelo cliente, sugerindo que os modelos propostos são apropriados para diferentes tarefas.

3.4 Considerações

Esse capítulo apresentou uma revisão bibliográfica a respeito dos estudos de satisfação de usuários na literatura de PLN e áreas afins. A tabela 4 apresenta uma visão geral desta revisão, contendo as colunas com a citação de cada artigo, o conjunto de dados e seus domínios, o idioma e a técnica de aprendizado utilizada em cada estudo.

Os conjuntos de dados considerados nos estudos foram em grande parte extrações próprias, tanto de redes sociais como *Twitter* como também questionários e formulários desenvolvidos pelos próprios pesquisadores. No caso dos estudos acerca da modelagem de diálogos, os conjuntos de dados foram transcritos a partir de conversas. Os domínios apresentados nos conjuntos de dados são diversos, incluindo mais de um domínio no mesmo conjunto (Domínio Cruzado - DC).

Os idiomas considerados nos conjuntos de dados dos estudos foram, em sua maioria, o Inglês (EN), com exceção do Higashinaka *et al.* (2010) o qual utiliza transcrições de diálogos na língua japonesa (JP) e Zeng *et al.* (2018), o qual utiliza diálogos em Chinês (CN), traduzindo 30% destes para o Inglês.

Os métodos de aprendizado utilizados nos estudos também foram diversificados. A maior parte dos estudos avaliados utilizou Redes Neurais (RN), seguido de Naive-Bayes (NB). Outros métodos utilizados incluem Árvores de Decisão (AD), *Boosting*, Cadeias de *Markov* (CM), Máxima Entropia (ME), Regressão Logística (RL) e Máquinas de Vetor Suporte (SVM).

Os estudos avaliados foram divididos nas categorias modelagem de satisfação de usuários (3.1), estudos de modelagem de diálogos (3.3) e técnicas com base em AS (3.2). É possível observar que os estudos mais recentes, dos últimos três anos, em qualquer uma das categorias utilizaram variações de técnicas de redes neurais e todos estes estudos utilizaram a arquitetura *LSTM*, a qual tem sido aplicada com sucesso em muitas tarefas de PLN Zeng *et al.* (2018), assim como descrito na seção 2.3.5. Isso demonstra a importância do contexto e da informação de sequência para as tarefas de modelagem de satisfação de usuários.

Tabela 4 – Summary of analyzed papers

Study	Problem	Data set	Domain	Language	Method
Liu, Bian e Agichtein (2008)	Satisfaction	QA Websites	DC	EN	SVM, AD, Boosting, NB
Park e Gates (2009)	Satisfaction	Questionnaires	Services	EN	AD, NB, RL, SVM
Tang <i>et al.</i> (2015)	Satisfaction	Websites	Movies	EN	RN
Ando <i>et al.</i> (2017)	Satisfaction	Call-Center	Services	EN	SVM, RN
Gupta, Fabbri e Haffner (2010)	SA	Websites	Services and Products	EN	ME
Gupta, Gilbert e Fabbri (2010)	SA	E-mails	Services	EN	CM
Silva, Hruschka e Hruschka (2014)	SA	<i>Twitter</i>	DC	EN	NB
Yu <i>et al.</i> (2018)	SA	SemEval e STB	<i>Twitter</i> and Movies	EN	RN
Higashinaka <i>et al.</i> (2010)	Dialog	Dialogues	DC	JP	CM
Munkova, Munk e Fraterova (2013)	Dialog	Questionnaires	Linguistic	EN	N/A
Zeng <i>et al.</i> (2018)	Dialog	Dialogues	Services	EN/CN	RN

Fonte: Arthur Marçal Flores 2020

4 Estudo exploratório

Neste capítulo, são apresentados dois estudos exploratórios propondo modelos de classificação binária com foco no problema da inferência da satisfação de usuários de serviços públicos de acesso à informação. O primeiro estudo aborda a modelagem da satisfação do usuário no contexto da resposta do atendente a partir dos textos da base de dados e-SIC. Já o segundo estudo aborda a modelagem da satisfação do usuário no contexto de solicitação de um serviço, com textos também provenientes da mesma base de dados (e-SIC), explorando a possibilidade da predição sem nem mesmo considerar a resposta do servidor.

4.1 *Experimento 1: Modelagem da satisfação com base na resposta*

Nesta seção é apresentado um experimento desenvolvido para avaliar de forma preliminar parte da hipótese geral, assim como definida na seção 1.3, a respeito da inferência da satisfação de usuários de um serviço de solicitação de acesso à informações. O objetivo do experimento foi estimar a satisfação, representado pelos escores de avaliação do serviço, com base em duas perguntas que são utilizadas como classes a serem aprendidas.

1. Classe Clareza: A resposta fornecida foi de fácil compreensão?
2. Classe Atendimento: A resposta fornecida atendeu plenamente ao seu pedido?

Nota-se entretanto que a base original é composta por notas em uma escala de 1 a 5, e que neste experimento as avaliações passaram por um processo de conversão para valores binários (positivas e negativas) com o objetivo de simplificar o estudo. A metodologia deste processo será explicada nas próximas seções.

Além disso, também foi desenvolvido um terceiro modelo *baseline* com base unicamente no tempo entre a solicitação e a resposta. Pressupõe-se que quanto menor este período, maior a nota de satisfação dada pelo usuário.

4.1.1 Conjunto de dados

O conjunto de dados abordado neste estudo foi extraído da plataforma e-SIC descrita na seção 2.1.

Contudo, neste experimento foi utilizado apenas o tipo de resposta Acesso Concedido, uma vez que as distribuições das notas possuem grande variação de acordo com o tipo de resposta. As tabelas 5, 6 e 7, exemplificam esta variação na distribuição da Classe Clareza respectivamente para os tipos de resposta Acesso Concedido, Acesso Parcialmente Concedido e Acesso Negado.

Tabela 5 – Distribuição das notas - Acesso Concedido

Nota	Qtde.	%
5	27.845	73,8%
4	3.380	8,9%
1	3.119	8,3%
3	2.118	5,6%
2	1.255	3,3%
Total	37.717	100%

Fonte: Arthur Marçal Flores 2020

Tabela 6 – Distribuição das notas - Acesso Parcialmente Concedido

Nota	Qtde.	%
5	1.276	56,4%
1	384	16,9%
4	244	10,8%
3	213	9,4%
2	147	6,5%
Total	2.264	100%

Fonte: Arthur Marçal Flores 2020

Tabela 7 – Distribuição das notas - Acesso Negado

Nota	Qtde.	%
5	1.009	43,5%
1	706	30,4%
3	222	9,6%
4	193	8,3%
2	190	8,2%
Total	2.320	100%

Fonte: Arthur Marçal Flores 2020

Com base no tipo de resposta escolhido, Acesso Concedido, avaliou-se as classes descritas na seção anterior. A distribuição das notas na classe Clareza foi mostrada na tabela 5. A tabela 8 mostra a distribuição das notas na classe Atendimento.

Tabela 8 – Distribuição das notas - Classe Atendimento

Nota	Qtde.	%
5	22.726	60,3%
1	7.091	18,7%
4	3.696	9,8%
3	2.495	6,7%
2	1.709	4,5%
Total	37.717	100%

Fonte: Arthur Marçal Flores 2020

Com base nas notas de avaliação disponíveis, classes binárias foram definidas da seguinte forma. Primeiramente, dividiu-se as classes em Satisfeito (notas 5 e 4) e Insatisfeito (notas 1 e 2). A nota 3, intermediária, foi excluída da avaliação.

Após este processo de conversão em valores binários, as classes passaram a apresentar as seguintes distribuições conforme tabela 9.

Tabela 9 – Distribuição das notas binárias - Conjunto de dados 1 - Classes Clareza e Atendimento

Instâncias	Classe Clareza		Classe Atendimento	
	Quantidade	%	Quantidade	%
Positivas	31.225	87,7%	26.422	75,0%
Negativas	4.374	12,3%	8.800	25,0%

Fonte: Arthur Marçal Flores 2020

Além das informações da base de dados apresentadas anteriormente, existem outras variáveis que foram utilizadas para homogeneização do conjunto de dados dos modelo e também na definição da característica utilizada no modelo *baseline*, explicada na próxima seção.

- DataRegistro: Data da solicitação
- FoiProrrogado: Informa se houve prorrogação do prazo
- FoiReencaminhado: Informa se o pedido foi reencaminhado
- FormaResposta: Tipo de resposta escolhida pelo solicitante (via sistema, e-mail, etc)
- OrigemSolicitacao: Informa se o pedido foi aberta em um balcão, SIC ou internet.
- NumeroPerguntas: Número de perguntas feitas no pedido
- DataResposta: Data da resposta ao pedido
- ClassificacaoTipoResposta: Subtipo da resposta dada ao pedido

4.1.2 Modelos propostos

Nesta seção são apresentados os modelos com base no texto de resposta da solicitação e o modelo *baseline*.

Para o primeiro modelo, onde pretende-se avaliar a influência do texto de resposta do servidor na satisfação do usuário solicitante, foi utilizada apenas o tipo de resposta de Acesso Concedido, uma vez que este representa praticamente 80% do conjunto de dados. Além disso, as distribuições das classes das respostas possuem grande variação nos outros tipos de resposta. Após esta separação, optou-se por remover instâncias com variáveis onde a distribuição fosse menor que 80%, mantendo-se assim um conjunto de dados mais homogêneo. Portanto, para as seguintes características, foi filtrada apenas a resposta com maior distribuição.

- FoiProrrogado: “Não” - 90%
- FoiReencaminhado: “Não” - 90%
- FormaResposta: “Sistema” - 99%
- OrigemSolicitacao: “Internet” - 99%
- NumeroPerguntas: “1” - 80%
- ClassificacaoTipoResposta: “Inserida e-SIC” - 80%

Uma vez filtrada a base original obteve-se um conjunto de dados com 18.185 instâncias. Como as distribuições das duas classes de respostas estão desbalanceadas, optou-se por aplicar uma técnica de balanceamento por meio da redução da classe majoritária (satisfeito) (DRUMMOND; HOLTE, 2003) e também, exclusão da nota intermediária (3). Assim, obteve-se um conjunto de dados para o modelo com base nos textos de resposta, com um total de 3.688 instâncias para a classe Clareza e outro conjunto de dados para a classe Atendimento com 7.554 instâncias.

Após o balanceamento dos conjuntos de dados para as classes Clareza e Atendimento, foi definido o conjunto de treino e teste, na proporção 70/30. A tabela 10, mostra a distribuição das notas de avaliação para cada uma das classes nos respectivos conjuntos de treino e teste.

Tabela 10 – Distribuição das classes do conjunto de treino e teste do experimento 1

Descrição	Classe Clareza		Classe Atendimento	
	Treino	Teste	Treino	Teste
Escore positivos	1.269	575	5.154	2.209
Escore negativos	1.312	532	5.154	2.209

Fonte: Arthur Marçal Flores 2020

Neste experimento, os textos foram representados por matrizes de *TF-IDF* com contagem por bigrama, conforme apresentado no capítulo 2. Como técnica de aprendizado foram utilizadas regressão logística, do pacote *sklearn* da linguagem *Python*, com regularização L2 (HOERL; KENNARD, 2000) onde é adicionado à função de custo um termo de penalidade na soma dos quadrados dos coeficientes. Para definição do parâmetro C da regularização foi utilizado o procedimento tradicional de busca em grade (*GridSearch*), onde ocorre uma busca exaustiva de um subconjunto de parâmetros definidos manualmente. Para esta busca, utilizou-se os valores de C iguais a 0,1 e 1,0, a técnica de validação cruzada (ARLOT; CELISSE, 2010) com 10 conjuntos e buscando otimizar a medida F. Por fim, utilizou-se o otimizador *liblinear* (FAN *et al.*, 2008). Como resultado da busca em grade, obteve-se um valor de C igual a 0,1.

Para fins de comparação, também foi implementada a técnica de rede neural Multilayer perceptron (MLP). Da mesma forma que na técnica anterior, também foi utilizada a busca em grade para definir os melhores parâmetros. Entretanto, para o modelo de MLP buscou-se pela melhor arquitetura relacionada à quantidade de neurônios (50 ou 100) na camada escondida. Os outros parâmetros foram mantidos como os valores padrão da função *MLPClassifier* do pacote *sklearn* na linguagem *Python*. A função de ativação padrão é a retificadora (ReLU) (NAIR; HINTON, 2010), o otimizador Adam (KINGMA; BA, 2015), o termo de regularização L2 padrão é 0,0001, a taxa de aprendizado 0,001 e por fim, o número máximo padrão de iterações do otimizador é 150.

Partindo da premissa de que o tempo de atendimento pode ser uma variável importante na satisfação do usuário (ou seja, que usuários com rápido atendimento podem apresentar escores de satisfação mais altos), a ponto de talvez obscurecer outros fatores, foi criado um modelo de *baseline* que utiliza unicamente o tempo de resposta (medido como a diferença em dias entre as variáveis *DataResposta* e *DataRegistro*) para representar este cenário. A expectativa assim é a de que as predições realizadas com base no texto sejam superiores às do *baseline* de estimativa de tempo.

Os procedimentos efetuados foram os mesmos a fim de se utilizar a mesma base de dados descrita anteriormente neste experimento. Como técnica de aprendizado foi utilizada regressão logística com regularização L2 e otimizador *liblinear*. Os mesmos parâmetros descritos nos modelo de regressão logística anterior foram aplicados. Para cada tarefa de aprendizado foi calculado o desempenho de um modelo *baseline*.

4.1.3 Resultados

A tabela 11 mostra a comparação entre os modelos da classe Clareza, onde pretende-se investigar se a resposta do servidor foi de fácil compreensão. A tabela 12 mostra a comparação entre os modelos para a classe Atendimento, onde pretende-se investigar se a resposta do servidor atendeu plenamente a solicitação. Os melhores resultados aparecem destacados em negrito.

Tabela 11 – Modelo de satisfação referente à clareza do atendimento - Com base nos textos de resposta

Modelo	Precisão	Revogação	Medida F	Acurácia
<i>Baseline</i> tempo atendimento	0,56	0,56	0,56	0,56
Resposta - LR	0,61	0,61	0,60	0,61
Resposta - MLP	0,59	0,58	0,58	0,58

Fonte: Arthur Marçal Flores 2020

Tabela 12 – Modelo de satisfação referente ao pleno atendimento - Com base nos textos de resposta

Modelo	Precisão	Revogação	Medida F	Acurácia
<i>Baseline</i> tempo atendimento	0,56	0,55	0,54	0,55
Resposta - LR	0,63	0,63	0,63	0,63
Resposta - MLP	0,62	0,62	0,62	0,62

Fonte: Arthur Marçal Flores 2020

4.1.4 Discussão

Os modelos desenvolvidos no experimento tiveram o objetivo de avaliar a satisfação do usuário referente à fácil compreensão da resposta do atendente (classe Clareza) e a satisfação do usuário referente ao pleno atendimento de sua solicitação (Classe Atendimento).

Tanto o modelo desenvolvido para a classe Clareza quanto para a classe Atendimento, foram capazes de prever com uma margem de acerto a satisfação dos usuários e-SIC considerando unicamente os textos de resposta. Já o modelo *baseline*, com base unicamente no tempo, teve seu desempenho superado por estes modelos, sugerindo que é possível inferir a satisfação do usuário com maior assertividade utilizando as informações textuais.

É importante notar que neste estudo preliminar não foram exploradas técnicas de pré-processamento, nem outras formas de representação textuais. Pressupõe-se que a modelagem por meio de representações distribuídas possa melhorar o desempenho de classificadores em relação aos modelos *baseline* pertinentes.

4.2 Experimento 2: Modelagem da satisfação com base em solicitações e-SIC

Em complemento ao experimento anterior, tratando da inferência de satisfação a partir de respostas textuais, esta seção apresenta um experimento semelhante com o objetivo de obter esta mesma informação a partir das perguntas textuais dos usuários, tal qual disponibilizadas também no corpus e-SIC.

As classes a serem aprendidas são as mesmas do experimento 1 anterior.

1. Classe Clareza: A resposta fornecida foi de fácil compreensão?
2. Classe Atendimento: A resposta fornecida atendeu plenamente ao seu pedido?

Da mesma forma, as notas com escala de 1 a 5, também passaram pelo mesmo processo de conversão para valores binários, positivos e negativos.

4.2.1 Conjunto de dados

O conjunto de dados abordado neste estudo foi extraído da mesma plataforma (e-SIC) do estudo 1 anterior. Da mesma forma, o problema de modelagem abordado neste experimento também foi transformado em classificação binária, dividindo-se as classes em Satisfeito (notas 5 e 4) e Insatisfeitos (notas 1 e 2). A nota 3, intermediária, também foi excluída da avaliação. A tabela 13, mostra a distribuição das notas na classe Clareza e na classe Atendimento.

Tabela 13 – Distribuição das notas - Conjunto de dados 2 - Classes Clareza e Atendimento

Nota	Clareza		Atendimento	
	Quantidade	%	Quantidade	%
5	33.648	69,5%	26.403	54,5%
1	5.402	11,2%	11.826	24,4%
4	4.378	9,0%	4.617	9,5%
3	3.060	6,3%	3.325	6,9%
2	1.961	4,0%	2.278	4,7%

Fonte: Arthur Marçal Flores 2020

Após a conversão das notas para valores binários, as classes Clareza e Atendimento apresentaram as distribuições conforme tabela 14.

Tabela 14 – Distribuição das notas binárias - Conjunto de dados 2 - Classes Clareza e Atendimento

Instâncias	Clareza		Atendimento	
	Quantidade	%	Quantidade	%
Positivas	38.026	83,8%	31.020	68,7%
Negativas	7.363	16,2%	14.104	31,3%

Fonte: Arthur Marçal Flores 2020

4.2.2 Modelos propostos

Nesta seção são apresentados os dois modelos com base nos textos de solicitação, um buscando estimar a de satisfação em relação à fácil compreensão da respostas (classe Clareza), outro em relação ao pleno atendimento da solicitação (classe Atendimento) e um terceiro modelo *baseline*, com base no tempo de atendimento. Uma vez que o primeiro e segundo o modelo utilizam como entrada apenas os textos de solicitação, não é possível adotar nenhuma premissa que utilize como referência outras informações da base de dados, que em sua maioria estão intrinsecamente associadas à resposta do atendimento. Por este motivo, como ponto de partida do experimento foi utilizada toda a base de dados com 48.449 instâncias (solicitações em formato textual).

Uma vez que as distribuições das duas classes de respostas estão desbalanceadas, optou-se por aplicar uma técnica de balanceamento por meio da redução da classe majoritária (satisfeito) (DRUMMOND; HOLTE, 2003). Assim, obteve-se um conjunto de dados para o modelo com base na solicitação com um total de 14.726 instâncias para a classe Clareza e outro conjunto de dados para a classe Atendimento com 28.208 instâncias.

Após o balanceamento dos conjuntos de dados, foi definido o conjunto de treino e teste. Para tal, optou-se por manter 70% dos dados no conjunto de treino e 30% no conjunto de testes. A tabela 15, mostra a distribuição das classes nos respectivos conjuntos de treino e teste.

Tabela 15 – Distribuição das classes do conjunto de dados de treino e teste do experimento 2 - Inferência a partir das solicitações e-SIC

Instâncias	Clareza		Atendimento	
	Treino	Teste	Treino	Teste
Positivo	5.154	2.209	9.857	4.247
Negativo	5.154	2.209	9.888	4.216

Fonte: Arthur Marçal Flores 2020

Foram utilizados os mesmos modelos de Regressão logística e *Multilayer perceptron* do experimento anterior, tanto para os modelos baseados nas informações textuais, quanto para o modelo *baseline* com base unicamente no tempo de atendimento.

4.2.3 Resultados

A tabela 16 mostra a comparação entre os modelos da classe Clareza, onde pretende-se investigar a fácil compreensão da resposta da solicitação e o modelo *baseline* do tempo de atendimento. A tabela 17 mostra a comparação entre os modelos para a classe Atendimento, onde pretende-se investigar se o servidor atendeu plenamente a solicitação e também o modelo *baseline*. Os melhores resultados aparecem destacados em negrito.

Tabela 16 – Predição da satisfação da clareza da resposta com base no texto da solicitação

Modelo	Precisão	Revogação	Medida F	Acurácia
<i>Baseline</i> tempo atendimento LR	0,56	0,56	0,56	0,56
Solicitação - LR	0,69	0,69	0,69	0,69
Solicitação - MLP	0,63	0,63	0,63	0,63

Fonte: Arthur Marçal Flores 2020

Tabela 17 – Predição de satisfação com o atendimento com base no texto da solicitação

Modelo	Precisão	Revogação	Medida F	Acurácia
<i>Baseline</i> tempo atendimento LR	0,56	0,55	0,54	0,55
Solicitação - LR	0,67	0,67	0,66	0,67
Solicitação - MLP	0,65	0,65	0,64	0,65

Fonte: Arthur Marçal Flores 2020

4.2.4 Discussão

Da mesma forma que no experimento 1, os modelos baseados em texto apresentaram resultados superiores ao *baseline* de tempo de atendimento, tanto no quesito Clareza quanto no Atendimento.

Mais uma vez, pressupõe-se que a modelagem por meio de representações distribuídas possa contribuir para melhorar ainda mais o desempenho de classificadores em relação aos modelos *baseline* pertinentes.

4.2.5 Considerações

Este capítulo apresentou dois experimentos para inferência da satisfação de usuários, quanto a Clareza e o Atendimento, no domínio de plataformas de solicitação de acesso à informação. O primeiro experimento utilizou os textos do corpus e-SIC referentes às respostas dos servidores e o segundo os textos referentes às requisições dos usuários .

O estudo sugere que é possível estimar a satisfação do tipo Clareza e Atendimento com base em texto de respostas e perguntas com resultados superiores ao de um *baseline* de estimativa de tempo de atendimento.

Apesar de não serem diretamente comparáveis, os resultados dos experimentos 1 e 2 sugerem que não há uma grande margem no desempenho dos classificadores, seja utilizando os textos de resposta ou solicitação.

5 Métodos

Os resultados obtidos no estudo exploratório sugerem que é possível estimar a satisfação do tipo Clareza e Atendimento com base em textos de respostas e perguntas com resultados superiores ao de um *baseline* de estimativa de tempo de atendimento. Estes resultados motivaram a continuação do desenvolvimento de métodos, que serão apresentados neste capítulo, para expansão e aprimoramento destes modelos. Para este fim, foram examinados os modelos de aprendizado de máquina supervisionado de regressão logística, redes neurais recorrentes do tipo *LSTM* (HOCHREITER; SCHMIDHUBER, 1997) com mecanismo de atenção, *Embeddings from Language Models (ELMo)* (PETERS *et al.*, 2018) e *Bidirectional Encoder Representations from Transformers (BERT)* (DEVLIN *et al.*, 2019).

Os modelos desenvolvidos têm especificamente o objetivo de estimar a satisfação em relação à fácil compreensão das respostas, representado pela classe Clareza, e também em relação ao pleno atendimento da solicitação, representado pela classe Atendimento, conforme definido na seção 4.1. Com exceção do modelo *baseline* com base no tempo de atendimento, todos os demais utilizam estritamente informações textuais para extração de características de entrada.

5.1 Conjunto de dados

Conforme descrito da seção 1.1, o corpus e-SIC é composto a partir de uma base de aproximadamente 50 mil interações entre cidadãos e governo, contendo textos de requisições de informações, respostas destas requisições e notas de avaliação, tanto referente à satisfação com o pleno atendimento, quanto a satisfação em relação à clareza da resposta. Um exemplo¹ de requisição e resposta é apresentado a seguir.

Bom dia. Fui aprovado para o Cargo de Analista em Serviços Logísticos, vaga para Goiânia/Brasília. No edital diz que preciso ter registro profissional para atuar na área. A dúvida é a seguinte: sou formado em Administração e pretendo fazer o registro no CRA de Goiás. Se eu for chamado para Brasília, minha carteirinha e registro profissional daqui terão validade para que eu possa exercer minhas atividades na DATAPREV/DF? Ou terei que ter dois registros, um em Goiânia ou outro no Brasília?

¹ http://www.consultaesic.cgu.gov.br/busca/_layouts/15/DownloadPedidos/DownloadDados.aspx

Prezado Senhor, Em resposta ao seu pedido realizado pelo SIC/DATAPREV, Informamos que: A sua comprovação de registro no Conselho Regional de Administração - CRA de Goiás tem validade em todo território nacional. Sendo assim, não haverá necessidade de solicitar transferência para o CRA do Distrito Federal. Ressaltamos também que é necessário o comprovante de quitação da anuidade junto ao respectivo Conselho. Atenciosamente, Serviço de Informações ao Cidadão - SIC EMPRESA DE TECNOLOGIA E INFORMAÇÕES DA PREVIDÊNCIA SOCIAL – DATAPREV

Originalmente as notas de avaliação, tanto para a clareza da resposta quanto para satisfação com o atendimento, são apresentadas como escores de 1 a 5, sendo 1 e 2 representando a insatisfação, 3 intermediário e 4 e 5 representando a satisfação. Como exemplificado na tabela 13, este conjunto de dados possui um grande desbalanceamento, sendo que aproximadamente 80% das notas representam escores positivos.

A distribuição original das notas de satisfação do corpus, contendo apenas aproximadamente 7% da classe intermediária, inviabiliza a capacidade de generalização dos classificadores desenvolvidos no presente trabalho. Por este motivo, o conjunto de dados foi reconfigurado de acordo com duas estratégias distintas. Na primeira estratégia, a nota de avaliação intermediária foi removida e as instâncias de notas 1 e 2 somadas para compor a classe negativa em relação à satisfação, enquanto as instâncias de notas 4 e 5 foram somadas para compor a classe positiva. Na segunda estratégia, para abordar o problema mantendo a classe intermediária, foi aplicado ao conjunto de dados uma técnica de balanceamento por meio da redução das classes majoritárias positivas e negativas. Este procedimento foi aplicado tanto no problema em relação à clareza quanto à satisfação com o atendimento.

5.2 Definição de classes de aprendizado

O problema de classificação enunciado nesta seção será dividido em três partes, cada uma associada a um experimento, e correspondendo a cada uma das hipóteses de pesquisa H1, H2 e H3 enunciadas na seção 1.3. O primeiro experimento, para a inferência da classe Clareza, utiliza apenas os textos com base nas respostas, uma vez que este aspecto está especificamente relacionado à compreensão da resposta fornecida pelo atendente do serviço. O segundo experimento, para a inferência da classe Atendimento utiliza apenas os textos das respostas e o terceiro experimento os textos das requisições.

Nos três experimentos realizados, foram consideradas duas abordagens do problema de classificação. A primeira abordagem procura tratar o problema mantendo a proporção original das classes durante a validação dos modelos, mas estimando apenas as notas positivas e negativas (ou seja, definindo um problema de classificação binária). Na segunda abordagem as notas intermediárias foram mantidas no problema (ou seja, definindo um problema de classificação multi-classe ternário) mas, conforme discutido na seção anterior, a distribuição destas classes é desbalanceada, e por este motivo foi aplicado um balanceamento na base de dados.

A tabela 18 sumariza as hipóteses (H) a serem verificadas, as tarefas de classificação a serem desenvolvidas, os textos, abordagem, e a seção desta monografia descrevendo cada um dos três experimentos.

Tabela 18 – Resumo das tarefas de classificação

Hipótese	Tarefa	Texto	Abordagem	Balanceamento	Seção
H1	Clareza	Resposta	Binária Ternária	Não Sim	7.1
H2	Atendimento	Resposta	Binária Ternária	Não Sim	7.2
H3	Atendimento	Requisição	Binária Ternária	Não Sim	7.3

Fonte: Arthur Marçal Flores 2020

5.3 Classificadores propostos

No presente trabalho consideramos cinco estratégias computacionais baseadas em modelos de classificador único, e uma sexta estratégia que faz uso de um comitê de classificadores. Estes modelos são relacionados a seguir, juntamente com os sistemas de *baseline* a serem considerados para fins de avaliação.

- **LogReg.word**: Regressão logística *softmax* com base em n-gramas de palavras
- **LogReg.LIWC**: Regressão logística *softmax* com base em características *LIWC*
- **LSTM**: Rede neural recorrente do tipo *LSTM* com mecanismo de atenção
- **ELMo**: *Embeddings from Language Models*
- **BERT**: *Bidirectional Encoders Representations from Transformers*
- **Stack**: Regressão logística *softmax* do tipo comitê *stacking*

- **Baseline Coh-Metrix:** Regressão logística *softmax* com base nas variáveis Coh-Metrix
- **Baseline tempo:** Regressão logística *softmax* com base no tempo de atendimento

5.3.1 Regressão logística *softmax* com base em n-gramas de palavras (LogReg.word)

Este classificador foi constituído com base no método de regressão logística *softmax*. Por ser um classificador multi-classe de rápida execução foi possível aplicá-lo nas duas abordagens do problema, binária e ternária. As variáveis de entrada deste classificador foram definidas a partir de unigramas e bigramas de palavras transformados em uma matriz de termos *TF-IDF*.

5.3.2 Regressão logística *softmax* com base em características *LIWC* (LogReg.LIWC)

Neste classificador as características de entrada foram extraídas a partir das categorias de palavras *Linguistic Inquiry and Word Count (LIWC)* (TAUSCZIK; PENNEBAKER, 2010). *LIWC* é um dicionário que categoriza palavras em classes psicolinguísticas como dinheiro, emoções negativas, pronomes pessoais, etc. É amplamente utilizado em diversas tarefas como análise de sentimentos (FILHO; PARDO; ALUÍSIO, 2013), reconhecimento de personalidade (SANTOS; RAMOS; PARABONI, 2019) e outros.

A versão disponível para o português é uma tradução do dicionário de língua inglesa discutida em Filho, Pardo e Aluísio (2013) e conta com 64 categorias de palavras. Dado um texto de entrada, o dicionário *LIWC* é utilizado para construir uma representação vetorial de 64 características que refletem a proporção de palavras existentes em cada categoria da seguinte forma. Mais especificamente, consulta-se todas as categorias a que cada palavra pertence, atualizando-se os respectivos contadores. Ao final do texto, estes contadores são divididos pelo total de palavras examinadas, obtendo assim um vetor de porcentagens.

5.3.3 Rede neural recorrente do tipo *LSTM* com mecanismo de atenção (*LSTM*)

Aqui são descritas as camadas que compõe o classificador *LSTM*, começando pela camada de *Embedding*, a qual é responsável pelo mapeamento dos *tokens* em representação distribuída de vetores de palavras. Os vetores extraídos do repositório NILC² possuem 300 dimensões e foram treinados com o algoritmo *Word2Vec* utilizando a estratégia *Skip-Gram*, conforme descrito na seção 2.2.2.

Em seguida, os vetores de representação distribuída são submetidos à uma camada *LSTM* Bidirecional, que por sua vez tem a capacidade de aprender representações de longos termos considerando o contexto de estados anteriores e posteriores ao estado avaliado da informação sequencial, constituindo o sinal de saída por meio de uma função, por exemplo de concatenação.

A seguir, é introduzida uma camada de atenção multi-cabeça. A quantidade de cabeças é obtida por meio de uma busca em grade que resulta na melhor performance da métrica Medida F. Por fim, a saída da camada de atenção multi-cabeça é achatada e submetida à camada densa final. Por meio da função *softmax* as probabilidades são normalizadas e distribuídas de acordo com a quantidade de rótulos do problema.

Estas probabilidades são obtidas a partir da função de perda probabilística *Binary Cross-Entropy*. Também referida como *logistic loss* ou *log loss*, esta função faz o mapeamento entre as probabilidades $\hat{y}$ inferidas pelo classificador e os rótulos verdadeiros y . A rede é treinada para maximizar o *log* da probabilidade condicional $P(y = 1|x)$ para cada exemplo de treino (GOLDBERG; HIRST, 2017). A função de perda é definida pela equação 10, onde $\hat{y}$ representa o vetor de probabilidades de saída do classificador e y o rótulo da classe correta.

$$L_{logistic}(\hat{y}, y) = -y \log \hat{y} - (1 - y) \log(1 - \hat{y}) \quad (10)$$

A figura 8 ilustra a arquitetura final deste modelo, sendo representada por X com os valores de entrada, E os vetores de *embeddings*, W o vetor com os pesos de entrada da camada *LSTM* e H os vetores de saída da camada *LSTM*. A camada do mecanismo de atenção multi-cabeça é explicada na figura 7.

² <http://nilc.icmc.usp.br/nilc/index.php/repositorio-de-word-embeddings-do-nilc>

Figura 8 – Representação do modelo *LSTM* bidirecional com mecanismo de atenção

Fonte: Arthur Marçal Flores 2020

5.3.4 Embeddings from Language Models (ELMo)

Conforme citado no capítulo 2 o modelo de língua ELMo (PETERS *et al.*, 2018) representa palavras de acordo com o seu contexto e é treinado por meio de redes *LSTM* bidirecionais em corpú textuais muito grandes. Para a língua portuguesa um modelo pré-treinado no corpú *brWaC* (FILHO *et al.*, 2018) foi gerado a partir do trabalho em Felix, Soares e Castro (2020), que inicialmente treina o modelo para a tarefa de reconhecimento de entidades nomeadas (NER). No presente trabalho, os pesos (representações das palavras) deste modelo são inicialmente utilizados e ajustados posteriormente por meio de um treinamento com gradiente descendente para a tarefa de classificação no corpú e-SIC.

5.3.5 Bidirectional Encoders Representations from Transformers (BERT)

Conforme descrito no capítulo 2, *BERT* (DEVLIN *et al.*, 2019) é um modelo de língua e a sua utilização, a partir de *Transformers*, tem apresentado ganhos expressivos em diversas tarefas de PLN. No trabalho em Souza, Nogueira e Lotufo (2019) foram treinados

dois modelos para a língua portuguesa, *base* e *large*, por meio do corpus *brWaC* (FILHO *et al.*, 2018) e então aplicados à tarefa NER. Os resultados superaram a performance do modelo *BERT Multilingual*, treinado em 104 línguas, entre elas a Portuguesa.

No presente trabalho, todos os experimentos com exceção do problema de classificação binária para inferência da clareza (devido à *overfitting*), utilizaram o modelo versão *large* disponibilizado em Souza, Nogueira e Lotufo (2019). A fase de ajuste fino foi executada utilizando como entrada os textos do corpus e-SIC e as classes de satisfação como saída. Da mesma maneira que a versão *base* do modelo *BERT* original, este modelo possui 24 camadas, 1024 camadas escondidas e 16 cabeças de atenção.

5.3.6 Regressão logística *softmax* do tipo comitê *stacking* (Stack)

Conforme demonstrado em Custódio e Paraboni (2018), o uso de múltiplos classificadores pode auxiliar determinadas tarefas de PLN, como por exemplo na atribuição autoral. Com o objetivo de melhorar o desempenho dos classificadores apresentados na seção anterior, é proposto desenvolver um modelo aplicando uma arquitetura que segue a estratégia de comitê de máquinas a partir das probabilidades geradas pelos classificadores individuais.

Dentre os métodos mais comuns na abordagem de comitê de máquinas o modelo aplicado no presente trabalho é do tipo *stacking*. Originalmente apresentado em Wolpert (1992), o autor conclui que para muitos problemas de generalização o método *stacking* pode reduzir as taxas de erros. A figura 9 ilustra esta arquitetura.

Figura 9 – Representação do modelo Stack

Os meta-classificadores desenvolvidos são do tipo regressão logística *softmax* e utilizaram como entrada as probabilidades de dois modelos individuais. A primeira entrada foi gerada a partir das probabilidades de saída y_1 e y_2 do modelo de regressão logística com base em n-gramas de palavras, enquanto a segunda entrada a partir das probabilidades y_3 e y_4 do modelo *LSTM* com mecanismo de atenção.

5.4 Classificadores Baseline

Nesta seção são propostos dois classificadores *baseline* de regressão logística *softmax*. O primeiro classificador, aplica a ferramenta *Coh-Metrix* (MCNAMARA *et al.*, 2014) para extração de variáveis de compreensão de textual, sendo utilizado como referência para o problema da satisfação com a clareza da resposta. O segundo classificador, com base no tempo de atendimento, é utilizado como referência para o problema da satisfação com o atendimento e foi definido da mesma forma que o classificador da seção 4.1.

5.4.1 Regressão logística *softmax* com base em característica *Coh-Metrix*

Com o objetivo de tratar o problema de inferência da classe clareza foi definido um classificador *baseline*, também utilizando o método de regressão logística *softmax*, mas que emprega como entrada características extraídas a partir da ferramenta *Coh-Metrix*. *Coh-Metrix* é um programa computacional que calcula diversas métricas de coesão, compreensão e sintaxe, que teriam a função de auxiliar o leitor a conectar as ideias de um texto, portanto, a premissa é que um classificador com base nestas características possa ser definido como um *baseline* para o experimento.

Por tratar-se de um experimento que dispõe de um conjunto de dados na língua portuguesa, para extração³ das características foi utilizada a versão adaptada desta ferramenta para o Português (SCARTON; ALUÍSIO, 2010). Exemplos destas características são métricas com base nos tamanhos médios das sentenças e contagens de letras ou sílabas por palavra a partir dos estudos desenvolvidos em Flesch (1948), além de métricas que mensuram a complexidade semântica e sintática com base no *framework* desenvolvido em (LENNON; BURDICK, 2014) para avaliar a inteligibilidade de textos.

³ Resultados da ferramenta *Coh-Metrix* gentilmente cedidos pela equipe NILC da USP São Carlos

Foram extraídos 178 características a partir dos textos de resposta. Alguns exemplos destas características são Proporção de substantivos em relação à quantidade de palavras do texto, Quantidade média de adjuntos adverbiais por oração do texto, Quantidade média de orações por sentença, Proporção de Verbos no Pretérito Mais que Perfeito do Indicativo em relação à quantidade de verbos flexionados no texto e Quantidade de Sentenças no texto.

5.4.2 Regressão logística *softmax* com base no tempo de atendimento

Assim como no experimento do estudo exploratório no capítulo 4, para o problema da satisfação com o atendimento foi desenvolvido um modelo *baseline* com base no tempo de atendimento. A premissa é que o tempo de atendimento pode ser uma variável significativa na determinação da satisfação do usuário, quanto menor o tempo entre a solicitação da informação e o recebimento da resposta, mais satisfeito estaria o cliente.

O classificador desenvolvido utiliza o método de regressão logística *softmax* e a variável de entrada foi definida unicamente a partir da diferença, em dias, entre a data de recebimento da resposta e a data de solicitação do pedido.

6 Avaliação

Este capítulo compreende os métodos utilizados para construção, otimização e avaliação dos problemas, desde a definição do conjunto de dados até os métodos de treinamento, bem como a escolha dos parâmetros dos modelos.

Para cada um dos problemas avaliados - clareza da resposta e a satisfação com o atendimento - foram definidos dois conjuntos de dados, que posteriormente foram separados em treino e teste. O primeiro conjunto contém três classes de avaliação, positiva, intermediária e negativa, caracterizando assim um problema de abordagem ternária. O segundo conjunto contém duas classes de avaliação, positiva e negativa, caracterizando um problema de abordagem binária. Para definição dos hiper-parâmetros foi utilizada a estratégia de busca em grade.

O problema da clareza é avaliado em um experimento que considera apenas os textos das respostas, e o problema do atendimento é avaliado em dois experimentos, um considerando como entrada os textos das respostas e outro os textos das requisições. A tabela 19 mostra distribuição das classes nos problemas avaliados.

Tabela 19 – Distribuição de instâncias de treinamento e teste

Abordagem	Classe	Clareza		Atendimento	
		Treino	Teste	Treino	Teste
Ternária	Positiva	1.581	720	2.296	1.029
	Intermediária	1.623	678	2.319	1.006
	Negativa	1.628	673	2.267	958
Binária	Positiva	2.736	1.204	21.682	9.338
	Negativa	1.632	669	9.904	4.200

Fonte: Arthur Marçal Flores 2020

6.1 Procedimentos

Nesta seção são apresentados os procedimentos executados para atacar o problema da inferência da satisfação, abordando as tarefas de Clareza e Atendimento. Assim como no experimento do estudo exploratório do capítulo 4, todos os classificadores do tipo regressão logística *softmax* foram obtidos a partir do pacote *sklearn* da linguagem *Python*. Os parâmetros deste classificador, como otimizador, tipo de regularização, fator de regularização e número de iterações variam de acordo com cada modelo e quando não es-

pecificados, seguiram as definições padrões do pacote na versão 0.23.1. Estas configurações apresentadas na tabela 20.

Tabela 20 – Configuração padrão dos modelos de regressão logística *softmax*

Configurações	
Otimizador	<i>lbfgs</i>
Regularização	L2
Fator de regularização C	0,1
Peso das classes (<i>Class Weight</i>)	<i>balanced</i>
Número máximo de iterações	100

Fonte: Arthur Marçal Flores 2020

Para a abordagem de classificação binária, onde as classes são desbalanceadas, o parâmetro *class_weight*¹ é definido com o valor *balanced*. Este parâmetro ajusta o peso das classes de acordo com as proporções de distribuição, funcionando como um penalizador. O peso de uma dada classe é ajustado de acordo com o tamanho da amostra, o número de classes e a frequência da classe em questão.

Assim como no experimento do estudo exploratório, as otimizações dos parâmetros do classificador de regressão logística *softmax* foram executadas utilizando a função *GridSearchCV*, que faz uma busca em grade a partir de um dicionário de parâmetros e respectivos valores. A métrica utilizada para avaliar os resultados no conjunto de dados de validação durante o processo de busca foi o *f1_macro*, o qual aplica uma média simples a partir do calculo da medida F de cada classe.

Na construção dos modelos de redes neurais recorrentes do tipo *LSTM*, foi utilizado o pacote *Keras* na versão 2.3.1. Este pacote facilita a construção de arquiteturas de redes neurais artificiais utilizando como base o *framework Tensorflow*. Nestes experimentos a versão utilizada do *framework* foi a 2.1.0.

Para o modelo ELMo, foi utilizado o *framework AllenNLP* na versão 0.8.2, do próprio criador do modelo. Como tokenizador de palavras foi utilizado o modelo *core* em português *pt_core_news_sm* do pacote *spacy* na versão 2.0.18. Devido a limitações computacionais não foi possível treinar este modelo para a abordagem binária, e portanto para este modelo será considerada apenas a abordagem ternária.

Finalmente, no caso do modelo (*BERT*), observamos que faz-se necessária uma infraestrutura computacional sofisticada devido à sua grande quantidade de parâmetros, mesmo na fase de ajuste fino. Por este motivo, estes modelos foram desenvolvidos em um

¹ https://scikit-learn.org/stable/modules/generated/sklearn.linear_model.LogisticRegression.html

ambiente na nuvem² com placas gráficas de até 16Gb de memória, e utilizando a biblioteca *Transformers*, versão 2.11.0 (WOLF *et al.*, 2019).

6.2 Treinamento de modelos para inferência da satisfação

Nesta seção são descritos os procedimentos gerais utilizados na construção de todos os modelos³. Isso inclui os modelos de inferência da satisfação em relação à clareza, com base apenas nos textos das respostas e também os modelos de inferência da satisfação com o atendimento, que contempla tanto os textos das respostas quanto os das requisições. Inclui também, os procedimentos efetuados tanto para construção dos modelos ternários, que consideram as notas de avaliação positivas, intermediárias e negativa, quanto para construção dos modelos binários, que consideram apenas as notas positivas e negativas.

6.2.1 Regressão logística *softmax* com base em n-gramas de palavras (LogReg.word)

Este modelo utilizou como características de entrada os unigramas e bigramas de palavras extraídos dos textos. Para as extrações com base nos textos das respostas foram extraídas aproximadamente 135 mil características. Para as extrações com base nos textos das requisições foram extraídas aproximadamente 120 mil características. A partir da busca em grade, foram definidos outros parâmetros como a frequência mínima e máxima de termos em documentos durante a construção do vocabulário.

A partir desta extrações, uma matriz de termos *TF-IDF* foi definida, a configuração padrão foi mantida. Com o objetivo de normalizar os valores da matriz *TF-IDF* foi aplicado a função *MaxAbsScaler*. O valor do parâmetro de regularização também foi definido a partir de uma busca em grade.

6.2.2 Regressão logística *softmax* com base em características *LIWC* (LogReg.LIWC)

Este modelo também utiliza o classificador de regressão logística *softmax*. Suas características de entrada foram extraídas a partir do software *LIWC*, como explicado na seção 5.3. Para cada fonte de texto, respostas e requisições, foram extraídas 64 carac-

² <https://colab.research.google.com/>

³ <https://github.com/arthurmarcal/user-satisfaction-public-services>

terísticas. Assim como no modelo *baseline* COH-METRIX, as características extraídas foram normalizadas por meio da função *MaxAbsScaler* e o parâmetro de regularização também foi encontrada a partir de uma busca em grade.

6.2.3 Rede neural recorrente do tipo *LSTM* com mecanismo de atenção (*LSTM*)

Este classificador foi construído utilizando a arquitetura de uma rede neural recorrente do tipo *LSTM* com inclusão do mecanismo de atenção multi-cabeça. Como pré-processamento, os textos foram *tokenizados* em palavras e truncados à esquerda com o valor zero (*zero padding*). Diferentes valores de truncamento foram testados, sendo definidos como 128, 256 e 512 palavras por documento.

Conforme explicado a seção 5.3 a arquitetura deste classificador é composta de uma camada de entrada com *tokens* de palavras, seguidas de uma camada de *embedding* utilizando uma matriz de pesos de 300 dimensões, pré-treinada com o algoritmo *word2vec* e a estratégia *skip-gram* a partir de um vocabulário em português. Em seguida é aplicada uma camada *LSTM* bidirecional com inclusão do mecanismo de atenção multi-cabeça. Uma camada *dropout* é inclusa para evitar *overfitting*. Por fim, os vetores são achatados utilizando uma camada *Flatten* e enviados para a camada final *Dense* com normalização por meio da função *softmax*.

O treinamento da rede ocorre por meio do otimizador *Adam* (KINGMA; BA, 2015) com a função de perda probabilística *Binary Cross-Entropy*. A otimização foi definida para executar em 50 épocas para cada conjunto de parâmetros. Contudo, um parâmetro de parada (*EarlyStopping*) foi estabelecido com o objetivo de interromper o treinamento caso o valor avaliado da perda no conjunto de validação não decrementasse após o fim de três épocas.

A busca em grade dos parâmetros foi efetuada considerando a quantidade de células, o *dropout* e o *dropout recorrente* da camada *LSTM*, o número de cabeças no mecanismo de atenção e a taxa de aprendizado *lr* do otimizador.

6.2.4 Embeddings from Language Models (ELMo)

Na construção deste classificador foi empregado o *framework AllenNLP*, do próprio criador do modelo. A partir de representações já disponibilizadas para português, conforme descrito no início desta seção, utilizou-se uma arquitetura *LSTM* bidirecional para ajustar os pesos para a tarefa de classificação do corpus e-SIC. Esta rede *LSTM* utilizou 1024 células de entrada (a partir da camada de *embedding*) e 64 células na camada escondida.

Todos os textos foram *tokenizados* e limitados a 128 caracteres devido às limitações computacionais. Para todos os experimentos, foram utilizados 10 épocas no parâmetro de treinamento do modelo e otimizador definido como Adam (KINGMA; BA, 2015), com taxa de aprendizado de 0,0003.

6.2.5 Bidirectional Encoders Representations from Transformers (BERT)

Antes de definir o processo ajuste fino deste modelo, é importante observar que o corpus e-SIC é caracterizado por conter sentenças longas. Conforme demonstrado na tabela 3, os textos de requisição possuem em média aproximadamente 450 à 500 caracteres. Quando avaliada a quantidade média de palavras, os textos de requisição e resposta mostram respectivamente 75 e 150 palavras, em média.

O treinamento do modelo *BERT* se torna especialmente custoso não apenas devido à grande quantidade de parâmetros, mas também em consequência dos cálculos efetuados pelo mecanismo de atenção. No modelo original, uma sequência de n *tokens* requer n^2 operações, escalando o custo quadraticamente em proporção ao comprimento das sentenças (BELTAGY; PETERS; COHAN, 2020). Assim, em consequência da arquitetura do modelo e das características do próprio corpus, o processo de experimentação se torna restrito, mesmo em um ambiente computacional mais sofisticado.

No processo de ajuste fino, o modelo utiliza primordialmente seu próprio *tokenizer* para mapear as palavras das sentenças em *tokens* pré-definidos em seu vocabulário, o qual é construído durante o treinamento do modelo. Após o mapeamento é necessário incluir determinados *tokens* especiais. Ao final de cada sentença é requerido incluir o *token* [SEP], que é utilizado na tarefa de treinamento NSP, conforme especificado na seção 2.3.6. Para tarefas de classificação é necessário incluir o *token* [CLS] no início de cada sentença. Isso se

faz necessário pois na arquitetura *BERT* são utilizados 12 camadas subsequentes do bloco *Transformers* e em cada camada os vetores respectivos à cada *token* são transformados agregando informações contextuais. Na última camada o vetor correspondente ao *token* [CLS] é utilizado para agregar a representação de toda a sequência e servindo de fato como única entrada para o classificador.

Após inclusão destes *tokens* especiais, é necessário definir uma janela de tamanho fixo com no máximo 512 *tokens*, ou seja, a sentença será truncada caso possua uma quantidade de *tokens* maior do que este limite. Caso a sentença possua uma quantidade menor do que o limite, o *token* especial [PAD] é inserido a direita com o objetivo de preencher os espaços remanescentes da janela. Este procedimento é chamado de *Padding*.

Por fim, é necessário definir uma máscara de atenção (*Attention Mask*). Esta máscara representa um vetor com os valores zero e um, indicando com o valor zero todos os *tokens* onde o procedimento de *padding* foi aplicado.

Após pré-processamento dos textos do conjunto de dados de treino e de teste, os vetores contendo os *tokens* podem ser submetidos ao classificador pré-treinado para executar o procedimento de ajuste fino. Para todos os experimentos, utilizou-se uma política de taxa de aprendizado com decaimento linear, sem período de aquecimento.

6.2.6 Regressão logística *softmax* do tipo comitê *stacking* (Stack)

Após a construção de todos os classificadores individuais foi realizado o desenvolvimento do meta-classificador de comitê. Este classificador foi composto por dois classificadores individuais, o primeiro de regressão logística *softmax* e o segundo *LSTM* com atenção, conforme explicado na seção 5.3.6.

Após a fase de treino dos modelos individuais foi possível salvá-los guardando suas estruturas e pesos para uso posterior sem precisar retreiná-los. Em seguida os textos foram pré-processados gerando *tokens* para entrada do modelo *LSTM*, seguindo os mesmos procedimentos de pré-processamento efetuados para este modelo conforme explicado na seção 6.2.3. Os dois modelos foram carregados em memória e as probabilidades de predição y_1 , y_2 , y_3 e y_4 , para o caso da abordagem binária, foram geradas no conjunto de dados de treino conforme ilustrado na figura 9. Para a abordagem ternária, três probabilidades são geradas, uma para cada classe e para cada modelo individual.

Os vetores com as probabilidades das classes de cada modelo foram então concatenados em uma nova tabela de variáveis que foi utilizada como entrada para o meta-classificador do tipo regressão logística *softmax*. Por fim, foi executado um procedimento de busca em grade com 10 dobras com o objetivo de encontrar o melhor parâmetro C de regularização no conjunto de treino. Estes procedimentos foram executados para todos os seis meta-classificadores de comitê desenvolvidos.

A tabela 21 mostra a relação de parâmetros de regularização (C) para cada um dos modelos Stack.

Tabela 21 – Parâmetro C de regularização por tarefa

Tarefa	Texto	Abordagem	Parâmetro C
Clareza	Resposta	Binária	1E-04
Clareza	Resposta	Ternária	1E-1
Atendimento	Resposta	Binária	1E+02
Atendimento	Resposta	Ternária	1
Atendimento	Requisição	Binária	1E+02
Atendimento	Requisição	Ternária	1E+02

Fonte: Arthur Marçal Flores 2020

6.2.7 Modelos de baseline

O modelo *baseline* para o problema da satisfação com a clareza foi definido a partir de um classificador de regressão logística *softmax*. As características de entrada deste modelo foram extraídas a partir da ferramenta *COH-METRIX*, como explicado na seção 5.3. No total, foram extraídas 178 características a partir dos textos de resposta. Como pré-processamento, foi efetuada a normalização destas características a partir da função *MaxAbsScaler*. Esta função normaliza os valores das característica tal que o valor absoluto máximo seja 1. O melhor parâmetro de regularização foi encontrado partir de uma busca em grade.

O modelo *baseline* para o problema da satisfação com o atendimento também foi definido a partir de um classificador de regressão logística *softmax*. Assim como no experimento do estudo exploratório, a variável de entrada deste modelo foi definida a partir da diferença em dias entre as variáveis *DataResposta* e *DataRegistro*. A otimização do parâmetro de regularização C foi determinada utilizando uma busca em grade com validação cruzada de 10 dobras.

6.2.8 Considerações

Esta seção sumariza os valores ótimos obtidos para cada modelo desenvolvido de acordo com o procedimento detalhado nas seções anteriores.

As tabelas 22, 23 e 24, mostram os parâmetros dos modelos que tratam o problema da satisfação com a clareza utilizando os textos de resposta.

Tabela 22 – Parâmetros LogReg *softmax* - Classe Clareza - Textos de resposta

Abordagem	Modelo	Parâmetros					
		analyzer	min_df	max_df	max_iter	ngram_range	C
Binária	<i>Baseline</i> Coh-Metrix	NA	NA	NA	10.000	NA	1E+02
	n-gramas de palavras	word	1	75%	10.000	(1,2)	1E-01
	LIWC	NA	NA	NA	10.000	NA	1
Ternária	<i>Baseline</i> Coh-Metrix	NA	NA	NA	10.000	NA	1E-03
	n-gramas de palavras	word	1	75%	10.000	(1,2)	1E-01
	LIWC	NA	NA	NA	10.000	NA	1E-01

Fonte: Arthur Marçal Flores 2020

Tabela 23 – Parâmetros *LSTM* com atenção - Classe Clareza - Textos de resposta

Abordagem	Modelo	Parâmetros					
		max_length	lstm cells	dropout	rec_dropout	lr	att_heads
Binária	<i>LSTM</i>	512	128	10%	10%	5E-05	4
Ternária	<i>LSTM</i>	256	256	10%	50%	5E-04	4

Fonte: Arthur Marçal Flores 2020

Tabela 24 – Parâmetros *BERT* - Classe Clareza - Textos de resposta

Abordagem	Modelo	Parâmetros				
		max_length	batch_size	lr	epochs	rate policy
Binária	<i>BERT</i>	128	16	3E-05	1	decaimento linear
Ternária	<i>BERT</i>	128	16	2E-05	2	decaimento linear

Fonte: Arthur Marçal Flores 2020

As tabelas 25, 26 e 27, mostram os parâmetros dos modelos que tratam o problema da satisfação com o atendimento com base nos textos de resposta.

Tabela 25 – Parâmetros LogReg *softmax* - Classe Atendimento - Textos de resposta

Abordagem	Modelo	Parâmetros					
		analyzer	min_df	max_df	max_iter	ngram_range	C
Binária	<i>Baseline</i> tempo de atendimento	NA	NA	NA	100	NA	1E-04
	n-gramas de palavras	word	1	100%	10.000	(1,2)	1
	LIWC	NA	NA	NA	10.000	NA	1E+01
Ternária	<i>Baseline</i> tempo de atendimento	NA	NA	NA	100	NA	1E-04
	n-gramas de palavras	word	1	75%	10.000	(1,2)	1E-01
	LIWC	NA	NA	NA	10.000	NA	1E+01

Fonte: Arthur Marçal Flores 2020

Tabela 26 – Parâmetros *LSTM* com atenção - Classe Atendimento - Textos de resposta

Abordagem	Modelo	Parâmetros					
		max_length	lstm cells	dropout	rec_dropout	lr	att_heads
Binária	<i>LSTM</i>	256	256	10%	25%	5E-04	4
Ternária	<i>LSTM</i>	128	256	10%	10%	5E-04	16

Fonte: Arthur Marçal Flores 2020

Tabela 27 – Parâmetros *BERT* - Classe Atendimento - Textos de resposta

Abordagem	Modelo	Parâmetros				
		max_length	batch_size	lr	epochs	rate policy
Binária	<i>BERT</i>	128	8	3E-05	1	decaimento linear
Ternária	<i>BERT</i>	128	128	2E-05	2	decaimento linear

Fonte: Arthur Marçal Flores 2020

As tabelas 28, 29 e 30, mostram os parâmetros dos modelos que tratam o problema da satisfação com o atendimento com base nos textos de requisição.

Tabela 28 – Parâmetros LogReg *softmax* - Classe Atendimento - Textos de requisição

Abordagem	Modelo	Parâmetros					
		analyzer	min_df	max_df	max_iter	ngram_range	C
Binária	<i>Baseline</i> tempo de atendimento	NA	NA	NA	100	NA	1E-04
	n-gramas de palavras	word	1	75%	10.000	(1,2)	1
	LIWC	NA	NA	NA	10.000	NA	1E+02
Ternária	<i>Baseline</i> tempo de atendimento	NA	NA	NA	100	NA	1E-04
	n-gramas de palavras	word	1	75%	10.000	(1,2)	1E-01
	LIWC	NA	NA	NA	10.000	NA	1

Fonte: Arthur Marçal Flores 2020

Tabela 29 – Parâmetros *LSTM* com atenção - Classe Atendimento - Textos de requisição

Abordagem	Modelo	Parâmetros					
		max_length	lstm cells	dropout	rec_dropout	lr	att_heads
Binária	<i>LSTM</i>	128	128	10%	10%	5E-04	2
Ternária	<i>LSTM</i>	128	128	10%	10%	1E-04	16

Fonte: Arthur Marçal Flores 2020

Tabela 30 – Parâmetros *BERT* - Classe Atendimento - Textos de requisição

Abordagem	Modelo	Parâmetros				
		max_length	batch_size	lr	epochs	rate policy
Binária	<i>BERT</i>	128	8	2E-05	1	decaimento linear
Ternária	<i>BERT</i>	128	64	2E-05	2	decaimento linear

Fonte: Arthur Marçal Flores 2020

7 Resultados

Este capítulo apresenta os resultados de três experimentos independentes, cada qual associado à verificação de uma das hipóteses de pesquisa deste trabalho, enunciadas na seção 1.3. Cada experimento foi conduzido utilizando os modelos que se fazem pertinentes conforme detalhado no capítulo 5, e a verificação da hipótese é realizada com uso do teste de significância estatística de *McNemar* (MCNEMAR, 1947), entre os modelos *baseline* de cada experimento e o modelo *BERT*. Para todos os experimentos, apresentamos resultados na forma de medida F ponderada por classe, e os melhores resultados para cada tarefa são destacados.

7.1 Experimento 1: Inferência da satisfação com a clareza com base nos textos de resposta

O objetivo do experimento de classificação da clareza com base nos textos de resposta foi verificar a validade da hipótese de pesquisa H1 deste trabalho. Esta hipótese sugere que o grau de satisfação do usuário em relação à clareza das respostas obtidas em serviços do tipo de solicitação de acesso à informação pode ser estimado com o uso de modelos de processamento de língua natural tomando-se como entrada os textos de resposta fornecidos pelo prestador do serviço. A tabela 31 apresenta os resultados das abordagens binária e ternária dos modelos de inferência.

Tabela 31 – Medida F - Tarefa Clareza com base nas respostas

Abordagem	Modelo	Negativo	Intermediário	Positivo	Ponderado
Binária	baseline Coh-Metrix	0,51	-	0,65	0,60
	LogReg.word	0,52	-	0,74	0,66
	LogReg.LIWC	0,53	-	0,61	0,58
	LSTM	0,38	-	0,75	0,62
	BERT	0,15	-	0,78	0,74
	Stack	0,40	-	0,80	0,74
Ternária	baseline Coh-Metrix	0,43	0,34	0,37	0,38
	LogReg.word	0,43	0,38	0,40	0,40
	LogReg.LIWC	0,41	0,30	0,41	0,38
	LSTM	0,38	0,41	0,43	0,41
	ELMo	0,40	0,35	0,43	0,40
	BERT	0,43	0,39	0,42	0,42
	Stack	0,36	0,31	0,50	0,41

Fonte: Arthur Marçal Flores 2020

No caso da abordagem do problema como classificação binária, observamos que o modelo de melhor resultado foi *BERT*. A vantagem deste modelo sobre o *baseline* é significativa de acordo com o teste de *McNemar* ($\alpha = 0,05, \chi = 12,6, p < 0,0001$). No caso da classificação ternária, a diferença entre os modelos *baseline* e *BERT* também é significativa ($\alpha = 0,05, \chi = 7,9, p = 0,00497$). Estes resultados oferecem suporte à hipótese de pesquisa H1.

7.2 Experimento 2: Inferência da satisfação com o atendimento com base nos textos de resposta

O objetivo do experimento de classificação do atendimento com base nos textos de resposta foi verificar a validade da hipótese de pesquisa H2 deste trabalho. Esta hipótese sugere que o grau de satisfação do usuário em relação ao atendimento em serviços do tipo de solicitação de acesso à informação pode ser estimado com o uso de modelos de processamento de língua natural tomando-se como entrada os textos de resposta fornecidos pelo prestador do serviço. A tabela 32 apresenta os resultados das abordagens binária e ternária dos modelos de inferência.

Tabela 32 – Medida F - Tarefa Atendimento com base nas respostas

Abordagem	Modelo	Negativo	Intermediário	Positivo	Ponderado
Binária	Baseline tempo	0,40	-	0,67	0,58
	LogReg.word	0,58	-	0,74	0,72
	LogReg.LIWC	0,50	-	0,61	0,61
	LSTM	0,51	-	0,75	0,72
	BERT	0,48	-	0,83	0,76
	Stack	0,54	-	0,80	0,73
Ternária	Baseline tempo	0,13	0,35	0,47	0,32
	LogReg.word	0,44	0,36	0,45	0,42
	LogReg.LIWC	0,44	0,33	0,45	0,41
	LSTM	0,40	0,36	0,50	0,42
	ELMo	0,44	0,25	0,48	0,39
	BERT	0,46	0,37	0,52	0,45
	Stack	0,43	0,37	0,43	0,41

Fonte: Arthur Marçal Flores 2020

No caso da modelagem do problema como classificação binária, observamos que o modelo de melhor resultado foi *BERT*. A vantagem deste modelo sobre o *baseline* é significativa de acordo com o teste de *McNemar* ($\alpha = 0,05, \chi = 34,77, p < 0,0001$). No caso da classificação ternária, o melhor modelo também foi *BERT* e com diferença

significativa ($\alpha = 0.05, \chi = 45, 47, p < 0,0001$). Estes resultados oferecem suporte à hipótese de pesquisa H2.

7.3 Experimento 3: Inferência da satisfação com o atendimento com base nos textos de requisição

O objetivo do experimento de classificação do atendimento com base nos textos de resposta foi verificar a validade da hipótese de pesquisa H2 deste trabalho. Esta hipótese sugere que o grau de satisfação do usuário em relação ao atendimento em serviços do tipo de solicitação de acesso à informação pode ser estimado com o uso de modelos de processamento de língua natural tomando-se como entrada os textos de requisições de usuários do sistema. A tabela 33 apresenta os resultados das abordagens binária e ternária dos modelos de inferência.

Tabela 33 – Medida F - Tarefa Atendimento com base nas requisições

Abordagem	Modelo	Negativo	Intermediário	Positivo	Ponderado
Binária	Baseline tempo	0,40		0,67	0,58
	LogReg.word	0,55		0,75	0,68
	LogReg.LIWC	0,47		0,63	0,58
	LSTM	0,46		0,79	0,69
	BERT	0,38		0,82	0,76
	Stack	0,54		0,76	0,69
Ternária	Baseline tempo	0,13	0,35	0,47	0,32
	LogReg.word	0,45	0,41	0,47	0,44
	LogReg.LIWC	0,42	0,35	0,41	0,39
	LSTM	0,47	0,37	0,50	0,45
	ELMo	0,42	0,28	0,51	0,41
	BERT	0,42	0,41	0,51	0,46
	Stack	0,45	0,38	0,45	0,43

Fonte: Arthur Marçal Flores 2020

No caso da modelagem do problema como classificação binária, observamos que o modelo de melhor resultado foi *BERT*. A vantagem deste modelo sobre o *baseline* é significativa de acordo com o teste de *McNemar* ($\alpha = 0.05, \chi = 305, 22, p < 0,0001$). No caso da classificação ternária, o melhor modelo também foi *BERT* e com diferença significativa ($\alpha = 0.05, \chi = 55, 28, p < 0,0001$). Estes resultados oferecem suporte à hipótese de pesquisa H3.

7.4 Análise de características mais relevantes para classificação

Com o objetivo de identificar as características mais relevantes para alguns dos modelos apresentados no capítulo 5, foram conduzidas análises de importância por meio da construção de um modelo de classificação de árvore. Um classificador de árvore apresenta o maior índice Gini, o qual está associado ao grau de impureza, nas características selecionadas nos primeiros nós. Para fins de simplificação os modelos avaliados nestas análises serão o *baseline Coh-Metrix* e o *LogReg.LIWC*, uma vez que estes modelos utilizam variáveis com significado menos abstrato, conforme explicado no capítulo 5. Os experimentos escolhidos para avaliação foram os que utilizam variáveis extraídas a partir das respostas do atendimento e da abordagem ternária, uma vez que os classificadores construídos para análise da abordagem binária não demonstraram um bom desempenho.

Na construção do modelo de avaliação, aplicou-se o mesmo conjunto de dados de treinamento utilizado nos experimentos dos modelos *baseline Coh-Metrix* no caso da Clareza e o *RegLog.LIWC* no caso do Atendimento. O classificador de árvore utilizado foi o *RandomForest* do pacote *sklearn* e os parâmetros profundidade máxima e número de estimadores foram respectivamente fixados em 7 e 5 para o caso da Clareza e 3 e 5 para o caso do Atendimento. Para fins de simplificação, nestas análises são mostrados apenas os resultados do melhor estimador gerado pelo algoritmo.

7.4.1 Características mais relevantes para a classificação da Clareza

No experimento da satisfação com a Clareza, a visualização da árvore é mais complexa dado que o parâmetro de profundidade foi definida como 7 devido a uma maior quantidade de variáveis no conjunto de dados. No conjunto de testes, as métricas deste modelo na medida F para as classes negativa, intermediária e positiva foram avaliadas respectivamente em 23%, 29% e 56%.

A variável com maior índice Gini foi *sentences*, a qual calcula a quantidade de sentenças no texto. A partir do maior número de amostras em cada folha, foi possível extrair as variáveis mais significativas para cada nota de satisfação, negativa (161 amostras), intermediária (353 amostras) e positiva (273 amostras). Para as notas negativa, intermediária e positiva as variáveis¹ mais significativas, de acordo com o melhor estimador

¹ Conceitos em: (<https://simpligo.sidle.al/nilcmatrixdoc>)

da árvore de decisão, foram respectivamente (*content words* > 0,609), (*brunet* ≤ 12,64) e (*brunet* > 6,861).

Na figura 10 do Apêndice A é possível observar a parte da árvore de decisão contendo as folhas com maior número de amostras para as classes intermediária (a esquerda) e negativa (a direita). E na figura 11, também do Apêndice A, é possível observar a parte da árvore de decisão contendo as folhas com maior número de amostras para a classe positiva. Além das variáveis das folhas, outras variáveis em nós superiores também foram selecionadas.

- *medium_short_sentence_ratio*
- *coreference_pronoun_ratio*
- *honore*: Estatística de *Honoré* (HONORÉ, 1979)
- *familiaridade_mean*
- *named_entity_ratio_text*
- *syllables_per_content_word*

7.4.2 Características mais relevantes para a classificação do Atendimento

Para o experimento da satisfação com o atendimento, no conjunto de testes as métricas deste modelo na medida F para as classes negativa, intermediária e positiva foram avaliadas respectivamente em 38%, 31% e 42%. A figura 12 do Apêndice A mostra a árvore do melhor estimador gerado para este experimento. É possível observar que a variável *LIWC* com maior índice Gini foi a *discrep*, a qual está associada a discrepância. No dicionário *LIWC* em português, aproximadamente 2,3% das palavras são classificadas nesta categoria. Exemplos de palavras do dicionário nesta classificação são aguardar, comum, embora, pretender, se e também.

A partir do maior número de amostras em cada folha, é possível observar as variáveis mais significativas para cada classe de satisfação, negativa, intermediária e positiva. Na tabela 34 é possível observar as variáveis, seu significado e exemplos de palavras no dicionário *LIWC*.

Tabela 34 – Variáveis significativas do dicionário *LIWC* - Inferência do Atendimento

Classe	Variável	Significado	Exemplos
Negativa	ipron	Pronomes impessoais	a, acontecimento, aquele, aquela, cujo, essa, esse, idéia, ninguém, traste, qual
Intermediária	ingest	ingestão	abandonam, bebida
Positiva	shehe	terceira pessoa do singular	desaparecer, frequentemente, macarrão, substituir
			a, dele, ele, fêmea, homem, macho, moça, mulher, seu, sua

Fonte: Arthur Marçal Flores 2020

7.5 Análise de erros

Com o objetivo analisar os casos onde alguns dos modelos apresentados no capítulo 5 não classificaram corretamente os rótulos nos conjuntos de dados de teste, foram conduzidas análises de importância por meio da construção de um modelo de classificação de árvore. Construiu-se um conjunto de dados a partir da comparação dos rótulos inferidos pelos modelos analisados e os rótulos verdadeiros do conjunto de dados de treinamento.

O modelos escolhidos para a análise são o *baseline Coh-Metrix* no caso da Clareza e o *RegLog.LIWC* no caso do Atendimento, utilizando a abordagem binária para fins de simplificação. Especificamente para o caso do atendimento, apenas as variáveis do dicionário *LIWC* relacionadas a linguística e gramática foram selecionadas, uma vez que são puramente textuais, sendo provavelmente mais relacionadas aos erros do classificador do que, por exemplo, características psicolinguísticas de emoção. Além disso, foi criada uma variável chamada de *Word Count (WC)* com a contagem de palavras por documento, uma vez que esta variável não foi extraída do dicionário *LIWC*. Na tabela 35 é possível verificar a quantidade de erros e acertos de cada modelo.

Tabela 35 – Rótulos classificados corretamente e erroneamente pelos modelos *baseline Coh-Metrix* e *RegLog.LIWC*

Resultado	baseline Coh-Metrix	RegLog.LIWC
Acertos	1.108	8.034
Erros	765	5.504

Fonte: Arthur Marçal Flores 2020

Com base no conjunto de dados da tabela 35, gerou-se outros dois conjuntos para cada um dos problemas a partir de um balanceamento por *undersampling*. Para a análise de erros da Clareza, um conjunto com 765 instâncias classificadas corretamente e 765 instâncias não classificadas corretamente e da mesma forma para o Atendimento, um conjunto com 5.504 instâncias classificadas corretamente e 5.504 não classificadas

corretamente. Cada um destes conjuntos foi dividido em treino e teste com uma proporção de 66/33. A distribuição final dos conjuntos de dados podem ser visualizados na tabela 36.

Tabela 36 – Distribuição dos conjuntos de dados utilizados na análise de erros

Conjunto de dados		Clareza	Atendimento
Treino	Acertos	520	3.724
	Erros	505	3.651
Teste	Acertos	260	1.853
	Erros	245	1.780

Fonte: Arthur Marçal Flores 2020

O classificador de árvore utilizado foi o *RandomForest* do pacote *sklearn* e o parâmetros profundidade máxima e número de estimadores foram respectivamente fixados em 5 e 6 para o caso da Clareza e 3 e 5 para o caso do Atendimento. Para fins de simplificação, nestas análises também serão mostrados apenas o resultado do melhor estimador gerado pelo algoritmo.

7.5.1 Análise de erros para a classificação da Clareza

No experimento da análise de erro da satisfação com a Clareza, no conjunto de testes as métricas deste modelo na medida F para as classes de rótulos classificados com erro e rótulos classificados corretamente foram avaliadas respectivamente em 54% e 53%. A figura 13 do Apêndice A mostra a árvore do melhor estimador gerado para este experimento e observa-se que a variável *Coh-Matrix* com maior índice Gini foi a *nouns_ambiguity*. Além disso, outras variáveis significativas são listadas.

- *sentences_with_zero_clause*
- *verbs_ambiguity*
- *tmp_pos_conn_ratio*
- *idade_aquisicao_mean*

7.5.2 Análise de erros para a classificação do Atendimento

Para o experimento da análise de erro da satisfação com o atendimento, no conjunto de testes as métricas deste modelo na medida F para as classes de rótulos classificados com erro e rótulos classificados corretamente foram avaliadas respectivamente em 61% e

54%. A figura 14 do Apêndice A mostra a árvore do melhor estimador gerado para este experimento e observa-se que a variável *LIWC* com maior índice Gini foi a *conj*, a qual está associada a conjunções. Exemplos de palavras do dicionário nesta classificação são: contudo, entretanto, mas, pois, porque, porém, quanto e todavia.

Outras variáveis de destaque de acordo com a figura 14 são *pronoun*, a qual está associada aos pronomes na sentença e também a variável de contagem de palavras (*WC*), uma vez que mais de 50% das instâncias são classificadas como erros quando as sentenças possuem mais de 44 palavras.

7.6 Considerações

De acordo com os resultados apresentados neste capítulo, o modelo *BERT* se destacou em todos os experimentos, com significância nos resultados em relação aos modelos *baseline* pertinentes à cada tarefa de acordo com o teste de *McNemar*. Estes resultados dão suporte as hipóteses descritas na seção 1.3.

A partir das análises efetuadas na seção 7.4, observa-se que características como *sentences*, *content words* e *brunet* possuem relevância para a tarefa de classificação da satisfação com a Clareza das respostas, enquanto as características *discrep*, *ipron*, *ingest* e *shehe* são exemplos de maior relevância para a tarefa de classificação da satisfação com o Atendimento.

Nas análises efetuadas na seção 7.5 observa-se que as características *nouns_ambiguity*, *sentences_with_zero_clause*, *verbs_ambiguity*, *tmp_pos_conn_ratio* e *idade_aquisicao_mean* têm maior relevância para a classificação incorreta da satisfação com a Clareza, enquanto as características *conj*, *pronoun* e *WC* têm maior relevância para a classificação incorreta da satisfação com o Atendimento.

Observa-se ainda que por questões de limitações computacionais e dos próprios modelos, como é o caso do *BERT* que não trabalha com mais de 512 *tokens*, grande parte das sentenças sofreu um truncamento, que poderia potencialmente ter levado a uma perda de informação (BELTAGY; PETERS; COHAN, 2020). Durante o processo de experimentação, onde foram utilizadas sentenças maiores (ex: 512 *tokens* ao invés de 128 ou 256), a aplicação de alguns parâmetros se tornou limitada (ex: *batch size*). A partir das experimentações deste trabalho, pode-se dizer que a utilização de modelos com

muitos parâmetros, como é o caso do *BERT*, se torna muito limitada, trazendo dificuldades principalmente no processo de ajuste fino dos hiper-parâmetros.

8 Conclusões

Este trabalho apresentou uma pesquisa em nível de mestrado dedicada à tarefa de classificação supervisionada, inferindo a satisfação de usuários de plataformas de solicitação de acesso à informação. A satisfação foi avaliada tanto no âmbito da clareza das respostas, quanto no âmbito do pleno atendimento, utilizando como base unicamente os textos das respostas e das requisições do corpus e-SIC. Foi apresentado como principal contribuição a aplicação de modelos computacionais, para um problema de inferência de satisfação em uma plataforma pública de grande utilização e na língua portuguesa, sendo possivelmente uma iniciativa inédita neste aspecto. As principais contribuições apresentadas foram:

- Modelos do tipo Regressão Logística *Softmax* e modelos de língua baseados em *Transformers BERT* e *ELMo* para o problema da inferência da satisfação
- Resultados de avaliação em cenários de classificação binária e ternária
- Uso dos modelos *BERT* e *ELMo* em Português, a serem reaproveitados em outros trabalhos deste tipo

Consideramos que os experimentos realizados oferecem suporte à hipótese de pesquisa central deste trabalho, de que a satisfação de usuários em serviços de solicitação de acesso à informação pode ser estimada com o uso de modelos de PLN, tanto com base nos textos das respostas fornecidas pelo servidor, quanto das requisições do usuário, com performance superior aos modelos *baselines* e nas métricas estabelecidas. Em especial, foi demonstrado que modelos do tipo *BERT* muitas vezes proporcionam resultados consideravelmente superiores às alternativas consideradas para o problema de classificação tanto da Clareza quanto do Atendimento.

Mais especificamente para o problema da satisfação com a clareza da resposta fornecida, os resultados mostraram que a abordagem com modelos puramente baseados em dados, o quais não fazem nenhuma suposição sobre a linguagem em questão, superou métodos estatísticos tradicionais e bem estabelecidos, como o do *COH-METRIX*. Os experimentos ainda sugerem é possível inferir a satisfação com o atendimento não apenas tomando-se como entrada os textos de respostas, mas também os textos das requisições. De certo modo, a satisfação parece ser pré-determinada não apenas pela qualidade do serviço prestado, mas também pelas palavras dos usuário durante o preenchimento da solicitação.

O trabalho realizado possui também algumas limitações importantes. O conjunto de dados do *cópus e-SIC* extraído em sua maioria de registros até o ano de 2017 possui um grande desbalanceamento nas classes, não sendo possível utilizar a classe intermediária a partir da distribuição original da base. No caso ternário, foi necessário remover muitas instâncias para se obter um conjunto balanceado. Também, o *cópus* coletado *e-SIC*, não é totalmente público, já que a plataforma do serviço mudou sua política de divulgação a partir de 2016. Por fim, durante a execução dos modelos *Bert* e *ELMo* não foi possível testar muitas variações de hiper-parâmetros por limitações computacionais.

Como sugestão para trabalhos futuros consideramos aumentar o conjunto de dados a partir da extração de bases mais recentes. Uma nova versão do *cópus e-sic*, agora de acordo com as novas normas da plataforma e 100% pública, foi recentemente coletada e os atuais experimentos serão repetidos para maior transparência. Outras possibilidades de trabalhos futuros incluem o uso de características não-linguísticas como demográficas e órgãos e meios de recebimento da solicitação, aproximando-se assim de uma aplicação mais completa na área de avaliação automática da satisfação de usuários.

Referências¹

- ANDO, A.; MASUMURA, R.; KAMIYAMA, H.; KOBASHIKAWA, S.; AONO, Y. Hierarchical LSTMs with joint learning for estimating customer satisfaction from contact center calls. In: *Proc. Interspeech 2017*. [s.n.], 2017. p. 1716–1720. Disponível em: <http://dx.doi.org/10.21437/Interspeech.2017-725>. Citado 2 vezes nas páginas 42 e 53.
- ARLOT, S.; CELISSE, A. A survey of cross-validation procedures for model selection. *Statist. Surv.*, The American Statistical Association, the Bernoulli Society, the Institute of Mathematical Statistics, and the Statistical Society of Canada, v. 4, p. 40–79, 2010. Citado na página 58.
- BELTAGY, I.; PETERS, M. E.; COHAN, A. Longformer: The long-document transformer. *ArXiv*, abs/2004.05150, 2020. Citado 2 vezes nas páginas 77 e 89.
- BENGIO, Y.; DUCHARME, R.; VINCENT, P.; JANVIN, C. A neural probabilistic language model. *J. Mach. Learn. Res.*, v. 3, p. 1137–1155, 2003. Citado na página 25.
- BENGIO, Y.; SIMARD, P. Y.; FRASCONI, P. Learning long-term dependencies with gradient descent is difficult. *IEEE Trans. Neural Networks*, v. 5, n. 2, p. 157–166, 1994. Citado na página 30.
- CHENG, Y.; YE, Z.; WANG, M.; ZHANG, Q. Document classification based on convolutional neural network and hierarchical attention network. *Neural Network World*, Czech Technical University in Prague, Faculty of Transportation Sciences, v. 29, n. 2, p. 83–98, 2019. Citado na página 18.
- CHO, K.; MERRIENBOER, B. van; GULCEHRE, C.; BAHDANAU, D.; BOUGARES, F.; SCHWENK, H.; BENGIO, Y. Learning phrase representations using rnn encoder–decoder for statistical machine translation. *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2014. Citado na página 35.
- CORTES, C.; VAPNIK, V. Support-vector networks. *Mach. Learn.*, v. 20, n. 3, p. 273–297, 1995. Citado na página 27.
- CUSTÓDIO, J. E.; PARABONI, I. EACH-USP ensemble cross-domain authorship attribution. In: *Working Notes Papers of the Conference and Labs of the Evaluation Forum (CLEF-2018) vol.2125*. Avignon, France: [s.n.], 2018. p. 7. Citado na página 70.
- DENG, L.; LIU, Y. *Deep Learning in Natural Language Processing*. Singapore: Springer, 2018. ISBN 9811052085. Citado 3 vezes nas páginas 29, 30 e 32.
- DEVLIN, J.; CHANG, M.; LEE, K.; TOUTANOVA, K. BERT: pre-training of deep bidirectional transformers for language understanding. In: BURSTEIN, J.; DORAN, C.; SOLORIO, T. (Ed.). *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*. [S.l.]: Association for Computational Linguistics, 2019. p. 4171–4186. Citado 4 vezes nas páginas 18, 32, 64 e 69.

¹ De acordo com a Associação Brasileira de Normas Técnicas. NBR 6023.

DRUMMOND, C.; HOLTE, R. C4.5, class imbalance, and cost sensitivity: Why under-sampling beats oversampling. *Proceedings of the ICML'03 Workshop on Learning from Imbalanced Datasets*, 01 2003. Citado 2 vezes nas páginas 57 e 61.

FALCONIERY RACHEL BASTOS, W. O. A. Raio x da transparência passiva. *FGV DAPP - Análises Políticas*, 2016. Citado na página 16.

FAN, R.-E.; CHANG, K.-W.; HSIEH, C.-J.; WANG, X.-R.; LIN, C.-J. Liblinear: A library for large linear classification. *J. Mach. Learn. Res.*, JMLR.org, v. 9, p. 1871–1874, jun. 2008. ISSN 1532-4435. Citado na página 58.

FELIX, N.; SOARES, A.; CASTRO, P. *Deep Learning for Named Entity Recognition in Legal Domain*. Tese (Doutorado) — Universidade Federal de Goiás, 01 2020. Citado na página 69.

FILHO, J. A. W.; WILKENS, R.; IDIART, M.; VILLAVICENCIO, A. The brwac corpus: A new open resource for brazilian portuguese. In: *Proceedings of the Eleventh International Conference on Language Resources and Evaluation, LREC 2018, Miyazaki, Japan, May 7-12, 2018*. [S.l.]: European Language Resources Association (ELRA), 2018. p. 4339–4344. Citado 2 vezes nas páginas 69 e 70.

FILHO, P. P. B.; PARDO, T. A. S.; ALUÍSIO, S. M. An evaluation of the brazilian portuguese LIWC dictionary for sentiment analysis. In: *Proceedings of the 9th Brazilian Symposium in Information and Human Language Technology, STIL 2013, Fortaleza, Brazil, October 21-23, 2013*. [S.l.]: Brazilian Computer Society, 2013. Citado na página 67.

FLESCH, R. A new readability yardstick. *Journal of Applied Psychology*, v. 32, June 1948. ISSN 0021-9010. Citado na página 71.

FREIRE, C. M. B. F. R. Como o cidadão avalia o portal? um estudo com os usuários do portal da transparência do governo federal. Citado na página 16.

GO, A.; BHAYANI, R.; HUANG, L. Twitter sentiment classification using distant supervision. *Processing*, v. 150, 01 2009. Citado na página 46.

GOLDBERG, Y.; HIRST, G. *Neural Network Methods for Natural Language Processing*. Toronto, ON, Canada: Morgan & Claypool Publishers, 2017. ISBN 1627052984, 9781627052986. Citado 5 vezes nas páginas 24, 25, 28, 29 e 68.

GOODFELLOW, I.; BENGIO, Y.; COURVILLE, A. *Deep Learning*. Cambridge, MA, USA: MIT Press, 2016. Citado 2 vezes nas páginas 27 e 35.

GOPALAKRISHNAN, A.; SOMAN, K. P.; PREMJI, B. Part-of-speech tagger for biomedical domain using deep neural network architecture. In: *10th International Conference on Computing, Communication and Networking Technologies, ICCCNT 2019, Kanpur, India, July 6-8, 2019*. [S.l.]: IEEE, 2019. p. 1–5. Citado na página 18.

GRAVES, A. Generating sequences with recurrent neural networks. *CoRR*, abs/1308.0850, 2013. Citado na página 31.

GUPTA, N.; FABBRIZIO, G. di; HAFFNER, P. Capturing the stars: Predicting ratings for service and product reviews. In: *Proceedings of the NAACL HLT 2010 Workshop on Semantic Search*. Stroudsburg, PA, USA: Association for Computational Linguistics, 2010. (SS '10), p. 36–43. Citado 2 vezes nas páginas 43 e 53.

GUPTA, N. K.; GILBERT, M.; FABBRIZIO, G. di. Emotion detection in email customer care. *Computational Intelligence*, v. 29, p. 489–505, 2010. Citado 2 vezes nas páginas 45 e 53.

HASHIMOTO, K.; XIONG, C.; TSURUOKA, Y.; SOCHER, R. A joint many-task model: Growing a neural network for multiple nlp tasks. *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, Association for Computational Linguistics, 2017. Citado na página 42.

HEMALATHA, I.; LATHA, I.; VARMA, G.; GOVARDHAN, D. Sentiment classification in online reviews using frn algorithm. In: . [S.l.: s.n.], 2013. p. 357–362. Citado na página 26.

HIGASHINAKA, R.; MINAMI, Y.; DOHSAKA, K.; MEGURO, T. Modeling user satisfaction transitions in dialogues from overall ratings. In: *Proceedings of the 11th Annual Meeting of the Special Interest Group on Discourse and Dialogue*. Stroudsburg, PA, USA: Association for Computational Linguistics, 2010. (SIGDIAL '10), p. 18–27. ISBN 978-1-932432-85-5. Citado 2 vezes nas páginas 48 e 53.

HOCHREITER, S.; SCHMIDHUBER, J. Long short-term memory. *Neural Comput.*, MIT Press, Cambridge, MA, USA, v. 9, n. 8, p. 1735–1780, nov. 1997. ISSN 0899-7667. Citado 2 vezes nas páginas 30 e 64.

HOERL, A. E.; KENNARD, R. W. Ridge regression: Biased estimation for nonorthogonal problems. *Technometrics*, v. 42, n. 1, p. 80–86, 2000. Citado na página 58.

HONORÉ, A. Some simple measures of richness of vocabulary. *Association for literary and linguistic computing bulletin*, v. 7, n. 2, p. 172–177, 1979. Citado na página 86.

JOACHIMS, T. Text categorization with support vector machines: Learning with many relevant features. In: NEDELLEC, C.; ROUVEIROL, C. (Ed.). *Machine Learning: ECML-98, 10th European Conference on Machine Learning, Chemnitz, Germany, April 21-23, 1998, Proceedings*. [S.l.]: Springer, 1998. (Lecture Notes in Computer Science, v. 1398), p. 137–142. Citado na página 27.

KINGMA, D. P.; BA, J. Adam: A method for stochastic optimization. In: BENGIO, Y.; LECUN, Y. (Ed.). *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*. [S.l.: s.n.], 2015. Citado 3 vezes nas páginas 58, 76 e 77.

KO, Y. A study of term weighting schemes using class information for text classification. In: *Proceedings of the 35th International ACM SIGIR Conference on Research and Development in Information Retrieval*. New York, NY, USA: ACM, 2012. (SIGIR '12), p. 1029–1030. ISBN 978-1-4503-1472-5. Citado na página 23.

KULLBACK, S.; LEIBLER, R. A. On information and sufficiency. *Ann. Math. Statist.*, The Institute of Mathematical Statistics, v. 22, n. 1, p. 79–86, 03 1951. Citado na página 49.

LEE, K. *Automatic Speech Recognition: The Development of the SPHINX System*. NY, US: Springer US, 1988. (The Springer International Series in Engineering and Computer Science). ISBN 9780898382969. Citado na página 49.

LENNON, C.; BURDICK, H. *The Lexile Framework as an Approach for Reading Measurement and Success*. Durham, North Carolina, US: MetaMetrics, 2014. Citado na página 71.

LIU, Y.; BIAN, J.; AGICHTEIN, E. Predicting information seeker satisfaction in community question answering. In: *Proceedings of the 31st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*. New York, NY, USA: ACM, 2008. (SIGIR '08), p. 483–490. ISBN 978-1-60558-164-4. Citado 2 vezes nas páginas 36 e 53.

MANNING, C. D.; RAGHAVAN, P.; SCHUTZE, H. *Introduction to Information Retrieval*. New York, NY, USA: Cambridge University Press, 2008. ISBN 0521865719, 9780521865715. Citado na página 24.

MARON, M. E. Automatic indexing: An experimental inquiry. *J. ACM*, v. 8, n. 3, p. 404–417, 1961. Citado na página 26.

MCNAMARA, D. S.; GRAESSER, A. C.; MCCARTHY, P. M.; CAI, Z. *Automated Evaluation of Text and Discourse with Coh-Metrix*. [S.l.]: Cambridge University Press, 2014. Citado na página 71.

MCNEMAR, Q. Note on the sampling error of the difference between correlated proportions or percentages. *Psychometrika*, v. 12, n. 2, 1947. Citado na página 82.

MIKOLOV, T.; CHEN, K.; CORRADO, G.; DEAN, J. Efficient estimation of word representations in vector space. In: BENGIO, Y.; LECUN, Y. (Ed.). *1st International Conference on Learning Representations, ICLR 2013, Scottsdale, Arizona, USA, May 2-4, 2013, Workshop Track Proceedings*. [S.l.: s.n.], 2013. Citado na página 25.

MUNKOVA, D.; MUNK, M.; FRATEROVA, Z. Identifying social and expressive factors in request texts using transaction/sequence model. In: *Proceedings of the International Conference Recent Advances in Natural Language Processing RANLP 2013*. [S.l.]: INCOMA Ltd. Shoumen, Bulgaria, 2013. p. 496–503. Citado 2 vezes nas páginas 50 e 53.

NAIR, V.; HINTON, G. E. Rectified linear units improve restricted boltzmann machines. In: *Proceedings of the 27th International Conference on International Conference on Machine Learning*. [S.l.]: Omnipress, 2010. p. 807–814. ISBN 9781605589077. Citado na página 58.

NAKOV, P.; ROSENTHAL, S.; KOZAREVA, Z.; STOYANOV, V.; RITTER, A.; WILSON, T. Semeval-2013 task 2: Sentiment analysis in twitter. In: *Proceedings of the 7th International Workshop on Semantic Evaluation, SemEval@NAACL-HLT 2013, Atlanta, Georgia, USA, June 14-15, 2013*. [S.l.]: The Association for Computer Linguistics, 2013. p. 312–320. Citado na página 47.

PANG, B.; LEE, L.; VAITHYANATHAN, S. Thumbs up?: Sentiment classification using machine learning techniques. In: *Proceedings of the ACL-02 Conference on Empirical Methods in Natural Language Processing - Volume 10*. [S.l.: s.n.], 2002. Citado 2 vezes nas páginas 27 e 28.

PARK, Y.; GATES, S. C. Towards real-time measurement of customer satisfaction using automatically generated call transcripts. In: *Proceedings of the 18th ACM Conference on Information and Knowledge Management*. New York, NY, USA: ACM, 2009. (CIKM '09), p. 1387–1396. ISBN 978-1-60558-512-3. Citado 2 vezes nas páginas 38 e 53.

PASCANU, R.; MIKOLOV, T.; BENGIO, Y. On the difficulty of training recurrent neural networks. In: *Proceedings of the 30th International Conference on Machine Learning, ICML 2013, Atlanta, GA, USA, 16-21 June 2013*. [S.l.]: JMLR.org, 2013. (JMLR Workshop and Conference Proceedings, v. 28), p. 1310–1318. Citado 2 vezes nas páginas 29 e 30.

PETERS, M.; NEUMANN, M.; IYYER, M.; GARDNER matt; CLARK, C.; LEE, K.; ZETTLEMOYER, L. Deep contextualized word representations. In: *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*. New Orleans, Louisiana: Association for Computational Linguistics, 2018. p. 2227–2237. Citado 4 vezes nas páginas 18, 32, 64 e 69.

RUMELHART, D. E.; HINTON, G. E.; WILLIAMS, R. J. Learning representations by back-propagating errors. *Nature*, v. 323, n. 6088, p. 533–536, 1986. Citado na página 29.

SALTON, G.; WONG, A.; YANG, C. S. A vector space model for automatic indexing. *Commun. ACM*, ACM, New York, NY, USA, v. 18, n. 11, p. 613–620, nov. 1975. ISSN 0001-0782. Citado na página 23.

SANTOS, W. R. dos; RAMOS, R. M. S.; PARABONI, I. Computational personality recognition from facebook text: psycholinguistic features, words and facets. *New Review of Hypermedia and Multimedia*, v. 25, n. 4, p. 268–287, 2019. Citado na página 67.

SASSO LUCIANO NURNBERG PERES, L. M. S. N. D. D. C. B. H. L. S. V. M. Análise descritiva dos pedidos de acesso à informação ao portal da transparência do governo federal brasileiro. *Revista da controladoria-geral da união*, v. 9, 2017. Citado na página 16.

SCARTON, C. E.; ALUÍSIO, S. maria. Análise da inteligibilidade de textos via ferramentas de processamento de língua natural: adaptando as métricas do coh-metrix para o português. *Linguamática*, v. 2, n. 1, p. 45–61, 2010. Citado na página 71.

SHAMMA, D. A.; KENNEDY, L.; CHURCHILL, E. F. Tweet the debates: Understanding community annotation of uncollected sources. In: . [S.l.]: ACM, 2009. p. 3–10. ISBN 978-1-60558-759-2. Citado na página 47.

SILVA, N. F. da; HRUSCHKA, E. R.; HRUSCHKA, E. R. Tweet sentiment analysis with classifier ensembles. *Decision Support Systems*, v. 66, p. 170 – 179, 2014. ISSN 0167-9236. Citado 2 vezes nas páginas 46 e 53.

SOCHER, R.; PERELYGIN, A.; WU, J.; CHUANG, J.; MANNING, C. D.; NG, A.; POTTS, C. Recursive deep models for semantic compositionality over a sentiment treebank. In: *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*. Seattle, Washington, USA: Association for Computational Linguistics, 2013. p. 1631–1642. Citado na página 47.

SOUZA, F.; NOGUEIRA, R.; LOTUFO, R. Portuguese named entity recognition using bert-crf. *arXiv preprint arXiv:1909.10649*, 2019. Citado 2 vezes nas páginas 69 e 70.

SPERIOSU, M.; SUDAN, N.; UPADHYAY, S.; BALDRIDGE, J. Twitter polarity classification with label propagation over lexical links and the follower graph. In: *Proceedings of the First Workshop on Unsupervised Learning in NLP*. Stroudsburg, PA, USA: Association for Computational Linguistics, 2011. (EMNLP '11), p. 53–63. ISBN 978-1-937284-13-8. Citado na página 47.

TANG, D.; QIN, B.; LIU, T.; YANG, Y. User modeling with neural network for review rating prediction. In: *Proceedings of the 24th International Conference on Artificial Intelligence*. [S.l.]: AAAI Press, 2015. (IJCAI'15), p. 1340–1346. ISBN 978-1-57735-738-4. Citado 2 vezes nas páginas 40 e 53.

TAUSCZIK, Y. R.; PENNEBAKER, J. W. The psychological meaning of words: LIWC and computerized text analysis methods. *Journal of Language and Social Psychology*, v. 29, n. 1, p. 24–54, 2010. Citado na página 67.

VASWANI, A.; SHAZEER, N.; PARMAR, N.; USZKOREIT, J.; JONES, L.; GOMEZ, A. N.; KAISER, L. u.; POLOSUKHIN, I. Attention is all you need. In: GUYON, I.; LUXBURG, U. V.; BENGIO, S.; WALLACH, H.; FERGUS, R.; VISHWANATHAN, S.; GARNETT, R. (Ed.). *Advances in Neural Information Processing Systems 30*. [S.l.]: Curran Associates, Inc., 2017. p. 5998–6008. Citado 4 vezes nas páginas 32, 33, 34 e 35.

WOLF, T.; DEBUT, L.; SANH, V.; CHAUMOND, J.; DELANGUE, C.; MOI, A.; CISTAC, P.; RAULT, T.; LOUF, R.; FUNTOWICZ, M.; BREW, J. Huggingface's transformers: State-of-the-art natural language processing. *ArXiv*, abs/1910.03771, 2019. Citado na página 75.

WOLPERT, D. H. Stacked generalization. *Neural networks*, v. 5, n. 2, p. 241–259, 1992. Citado na página 70.

YU, L.; WANG, J.; LAI, K. R.; ZHANG, X. Refining word embeddings using intensity scores for sentiment analysis. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, v. 26, n. 3, p. 671–681, March 2018. ISSN 2329-9290. Citado 2 vezes nas páginas 47 e 53.

ZENG, Z.; LUO, C.; SHANG, L.; LI, H.; SAKAI, T. Towards automatic evaluation of customer-helpdesk dialogues. *Journal of Information Processing*, v. 26, p. 768–778, 01 2018. Citado 2 vezes nas páginas 51 e 53.

ZHANG, L.; WANG, S.; LIU, B. Deep learning for sentiment analysis: A survey. *Wiley Interdiscip. Rev. Data Min. Knowl. Discov.*, v. 8, 2018. Citado na página 18.

Apêndice A – Árvores utilizadas para análises de variáveis e erros

1 Características mais relevantes para a classificação da Clareza

As figuras 10 e 11 apresentam as árvores geradas pelo melhor estimador de análise das características mais relevantes para a classificação da Clareza.

Figura 10 – Folhas da árvore de decisão para as classes intermediária e negativa do problema da satisfação com a Clareza

Fonte: Arthur Marçal Flores 2020

Figura 11 – Folhas da árvore de decisão para a classe positiva do problema da satisfação com a Clareza

Fonte: Arthur Marçal Flores 2020

2 Características mais relevantes para a classificação do Atendimento

A figura 12 apresenta a árvore gerada pelo melhor estimador de análise das características mais relevantes para a classificação do Atendimento.

Figura 12 – Árvore de decisão - Satisfação com o atendimento - Variáveis LIWC

Fonte: Arthur Marçal Flores 2020

3 Análise de erros para a classificação da Clareza

A figura 13 apresenta a árvore gerada pelo melhor estimador de análise de erros para a classificação da Clareza.

Figura 13 – Árvore de decisão da análise de erros - Satisfação com a Clareza

Fonte: Arthur Marçal Flores 2020

4 Análise de erros para a classificação do Atendimento

A figura 14 apresenta a árvore gerada pelo melhor estimador de análise de erros para a classificação do Atendimento.

Figura 14 – Árvore de decisão da análise de erros - Satisfação com o Atendimento

Fonte: Arthur Marçal Flores 2020