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

JOSÉ ELEANDRO CUSTÓDIO

Atribuição autoral de textos digitais

São Paulo

2020

JOSÉ ELEANDRO CUSTÓDIO

Atribuição autoral de textos digitais

Versão corrigida

Dissertação apresentado 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 09 de dezembro de 2019. 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

Custódio, José Eleandro

Atribuição autoral de textos digitais / José Eleandro Custódio ;
orientador, Ivandré Paraboni. – 2020.

118 f : il.

Dissertação (Mestrado em Ciências) - Programa de Pós-
Graduação em Sistemas de Informação, Escola de Artes, Ciências
e Humanidades, Universidade de São Paulo, em 2019.

Versão corrigida

1. Processamento de linguagem natural. 2. Aprendizado
computacional. 3. Autor - Identificação. 4. Análise de texto. 5.
Reconhecimento de texto. I. Paraboni, Ivandré, orient. II. Título

CDD 22.ed.– 006.35

Dissertação de autoria de José Eleandro, sob o título “**Atribuição autoral de textos digitais**”, 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 9 de dezembro de 2019 pela comissão julgadora constituída pelos doutores:

Prof. Dr. Ivandré Paraboni

Universidade de São Paulo
Escola de Artes, Ciências e Humanidades
Presidente

Prof. Dra. Solange Oliveira Rezende

Universidade de São Paulo
Instituto de Ciências Matemáticas e Computação

Prof. Dr. Esteban Fernandez Tuesta

Universidade de São Paulo
Escola de Artes, Ciências e Humanidades

*Dedico essa dissertação aos meus pais Benedito e Idaiusa
que foram minha fonte de inspiração.*

Agradecimentos

A realização desse mestrado contou com o apoio e incentivo de inúmeras pessoas, as quais sou imensamente grato.

Aos meus pais, Benedito e Idaiusa, e irmã, Helena, que sempre me apoiaram e foram pacientes com minha ausência nos períodos de maior dedicação aos estudos.

Ao orientador, Ivandré Paraboni, pela paciência, dedicação e parceria nas publicações e eventos acadêmicos.

Aos meus velhos amigos pela paciência nesse longo período de ausência e aos novos amigos mestrando do Programa de Pós-graduação em Sistemas de Informação (PPgSI) da EACH, pela oportunidade de aprendermos juntos, compartilhando experiências e momentos.

E, finalmente, agradeço a Deus pela vida, pela esperança em momentos de cansaço e pela força para concluir esse projeto de mestrado.

*“Mera mudança não é crescimento. Crescimento é a síntese de mudança e continuidade, e
onde não há continuidade não há crescimento.”*
(C. S. Lewis)

Resumo

CUSTÓDIO, José Eleandro. **Atribuição autoral de textos digitais** 2020. 118 f. Dissertação (Mestrado em Ciências) – Escola de Artes, Ciências e Humanidades, Universidade de São Paulo, São Paulo, 2019.

A atribuição autoral de textos digitais (AA) visa identificar quem é o autor de um determinado texto a partir de um conjunto de autores possíveis. Sua aplicação pode ajudar na solução de casos de escândalos de corrupção, na identificação de abusos na utilização da internet, na detecção de notícias falsas, na detecção de pseudônimos e outros. Esse trabalho apresenta um estudo que usou n-gramas de caracteres, de palavras, de anotações linguísticas (POS), modelos de representação distribuída (*embeddings*). Foram aplicados métodos de aprendizado de máquina e proposto um método para combinar diversos modelos. Os resultados foram avaliados nos domínios literatura, letras de músicas e mensagens de microblogs.

Palavras-chaves: Atribuição autoral. identificação autoral. aprendizado de máquina. PLN. processamento de língua natural. distorções textuais. *word embedding*. *Part-of-speech*. POS.

Abstract

CUSTÓDIO, José Eleandro. **Authorship Attribution of digital texts**. 2020. 118 p. Dissertation (Master of Science) – School of Arts, Sciences and Humanities, University of São Paulo, São Paulo, 2019.

Authorship attribution (AA) of digital text is a computational task which aims to identify who is the author of a text given a set of candidate authors. Its application may help to solve corruption scandals, identification of abuses on internet usage, fake news detection or pseudonyms detection. Computational methods for AA includes multivariate statistics and machine learning. This work presents a study that used n-grams of characters, words, linguistic annotations (POS) and word embeddings models. Machine learning methods were applied and it was proposed a method to combine several models. Results were evaluated in literature, song lyrics and microblogging domains.

Palavras-chaves: authorship attribution. author identification. machine learning. natural language processing, NLP. text distortion. embedding. part-of-speech. POS.

Lista de figuras

Figura 1 – Arquitetura do modelo Word2Vec	29
Figura 2 – Arquitetura <i>Distributed Memory</i> e <i>Distributed Bag of Words</i>	31
Figura 3 – Representação distância Komogorov-Smirnov	35
Figura 4 – Regressão Logística Multinomial representada como neurônios	37
Figura 5 – Operação de convolução em 2D	38
Figura 6 – Arquitetura de convolução para sentenças e documentos	39
Figura 7 – Expansão de um neurônio de uma rede recorrente	40
Figura 8 – Arquitetura da rede neural multicabeças	56
Figura 9 – Análise da capacidade de separação da medidas de similaridade	63
Figura 10 – Arquitetura do comitê de máquina para AA multidomínio	68
Figura 11 – Comparação dos métodos baselines em AA de conjunto aberto	74
Figura 12 – Medida macro-F1 para o corpus PAN-CLEF 2019 de acordo com o número de documentos desconhecidos.	75
Figura 13 – Arquitetura do comitê para AA multidomínio	77
Figura 14 – F1 macro por idioma.	80
Figura 15 – F1 macro de acordo com o percentual de documentos de autoria desco- nhecida.	80
Figura 16 – Modelo de atribuição autoral geral	84

Lista de algoritmos

Algoritmo 1 – Atribuição autoral usando o método impostor	51
Algoritmo 2 – Atribuição autoral usando o método <i>unmasking</i>	52
Algoritmo 3 – Comitê dinâmico para atribuição autoral	86

Lista de tabelas

Tabela 1 – Representação do vetor <i>one-hot</i> da palavra avião para um dado vocabulário.	26
Tabela 2 – Matriz de contagem da representação <i>bag-of-words</i>	26
Tabela 3 – Histórico das edições da competição PAN	42
Tabela 4 – Resumo dos trabalhos correlatos	60
Tabela 5 – Matriz de correlação das medidas de similaridade textual.	64
Tabela 6 – Verificação autoral - AUC e acurácia médias em 5-partições.	65
Tabela 7 – Características do problemas do PAN 2018	67
Tabela 8 – Exemplo de distorção de texto aplicada ao primeiro documento do problema 09 da base de treinamento.	69
Tabela 9 – Parâmetros analisados pelo processo de classificação do PAN2018 . . .	70
Tabela 10 – Valores ótimos encontrados para PAN2018	70
Tabela 11 – PAN-CLEF 2018 - F1 para AA no córpis de desenvolvimento	71
Tabela 12 – PAN2018 - Visão geral	72
Tabela 13 – PAN2018 - F1 reportada pelos organizadores - Por língua	73
Tabela 14 – PAN2018 - F1 reportada pelos organizadores - Por número de autores .	73
Tabela 15 – Parâmetros do modelo	78
Tabela 16 – Valores ótimos encontrados para PAN2019	78
Tabela 17 – F1 macro baseado do córpis de desenvolvimento do PAN-CLEF 2019 AA	79
Tabela 18 – Exemplo de anotações sintáticas	84
Tabela 19 – Estrutura do comitê de máquina para AA para um documento e cinco autores.	85
Tabela 20 – Estatística descritiva do córpis PAN-EVAL-2018	89
Tabela 21 – Estatística descritiva do córpis Lyrics	90
Tabela 22 – Estatística descritiva do córpis Social	92
Tabela 23 – Medida F1 macro para o córpis PAN-EVAL-2018	93
Tabela 24 – Medida F1 macro para o córpis PAN-EVAL-2018 consolidado por número de autores e língua.	94
Tabela 25 – Medida F1 macro para o corpus Lyrics	94

Tabela 26 – Medida F1 macro para o cópús Lyrics consolidado por número de autores e língua.	94
Tabela 27 – Medida F1 macro para o corpus SocialAA	95
Tabela 28 – Medida F1 macro para o cópús Social consolidado por número de autores e número de caracteres por documento.	96
Tabela 29 – Categorias de n-gramas de caracteres analisados	100
Tabela 30 – Relevância dos n-gramas de caracteres avaliados por categoria de con- teúdo aplicados ao cópús de desenvolvimento do PAN-2018.	102
Tabela 31 – Características textuais relevantes para <i>Std.charN</i>	117
Tabela 32 – Características textuais relevantes para <i>Dist.charN</i>	118
Tabela 33 – Características textuais relevantes para <i>Std.wordN</i>	118

Lista de abreviaturas e siglas

AUC	Área sob a curva ROC
PLN	Processamento de língua natural
LSTM	<i>Long short term memory</i> - Redes neurais recorrentes de memória curta e longa
CNN	<i>Convolutional Neural Nets</i> - Redes neurais de convolução
Word2vec	Modelo de <i>embeddings</i> para palavras
Doc2vec	Modelo de <i>embeddings</i> para documentos
BERT	Bidirectional Encoder Representations from Transformers
BOW	<i>Bag of Word</i> ou modelo de espaço de vetores
KLD	Kullback–Leibler
EN	Língua Inglesa
ES	Língua Espanhola
DU	Língua holandesa
GE	Língua alemã
PL	Língua polonesa
PT	Língua portuguesa
FR	Língua francesa
IT	Língua Italiana
NV	Domínio do texto - Romances
CT	Domínio do texto - Contos
EM	Domínio do texto - Ensaios
AR	Domínio do texto - Artigos de jornais
RV	Domínio do texto - Críticas e avaliações de produtos
SMS	Domínio do texto - Mensagens curtas

Sumário

1	Introdução	18
1.1	<i>Hipóteses</i>	19
1.2	<i>Objetivo</i>	19
1.3	<i>Organização do documento</i>	20
2	Conceitos fundamentais	21
2.1	<i>Atribuição Autoral</i>	21
2.2	<i>Métodos computacionais para atribuição autoral</i>	25
2.3	<i>Modelos computacionais de representação textual</i>	25
2.3.1	Representações tradicionais <i>one-hot</i> , <i>bag of words</i> e n-gramas	26
2.3.2	Representação distribuída de palavras (<i>Word Embeddings</i>)	27
2.4	<i>Métodos computacionais baseados em vizinho mais próximo</i>	30
2.4.1	Medidas de distâncias textuais de uso geral	31
2.4.2	Medidas de distância e técnicas específicas da AA	33
2.5	<i>Seleção de variáveis com Kolmogorov-Smirnov</i>	34
2.6	<i>Método de aprendizado de máquina</i>	36
2.6.1	Regressão logística <i>softmax</i> ou classificador de máxima entropia . .	36
2.6.2	Redes neurais de convolução	38
2.6.3	Redes neurais recorrentes	39
2.7	<i>Considerações</i>	40
3	Revisão bibliográfica	41
3.1	<i>Competições PAN para atribuição autoral</i>	41
3.2	<i>Modelos de AA usando bag of n-grams, embeddings e variações</i>	42
3.2.1	Uso de n-gramas de caracteres (<i>Char-NG</i>)	43
3.2.2	Uso de métodos de distorção textual	44
3.2.3	AA aplicada a mídia sociais	45
3.2.4	Modelos de AA usando representação distribuída	47
3.3	<i>Modelos de AA baseados similaridade textual</i>	49
3.3.1	Uso de estatística multivariada	49
3.3.2	Modelos usando Regra Delta de Burrows	50

3.3.3	O uso método impostor	50
3.3.4	Método <i>unmasking</i>	52
3.4	<i>Modelos de AA usando redes neurais artificiais</i>	52
3.4.1	Modelos de AA usando CNN e <i>embeddings</i> pré-treinados	53
3.4.2	Modelo de AA usando CNN aplicada a dados de mídias sociais . . .	53
3.4.3	Modelo de AA usando CNN e informação <i>POS</i>	54
3.4.4	Modelo de AA usando redes recorrentes	55
3.4.5	Outras abordagens	56
3.5	<i>Modelos de AA para o português</i>	57
3.6	<i>Considerações</i>	58
4	Estudos exploratórios	61
4.1	<i>Experimento 1: Verificação autoral</i>	61
4.1.1	Conjunto de dados	61
4.1.2	Análise das medidas de similaridade	62
4.1.3	Métodos propostos	64
4.1.4	Resultados	65
4.1.5	Considerações	65
4.2	<i>Experimento 2: Atribuição autoral de conjunto fechado e multidomínio no desafio PAN-CLEF 2018</i>	66
4.2.1	Conjunto de dados	66
4.2.2	O sistema <i>baseline</i> e o sistema de avaliação PAN-CLEF 2018	66
4.2.3	Método proposto	67
4.2.4	Experimento	69
4.2.5	Resultados	71
4.2.6	Resultados publicados pela PAN-CLEF 2018	72
4.3	<i>Experimento 2: Atribuição autoral de conjunto aberto e multidomínio no desafio PAN-CLEF 2019</i>	73
4.3.1	Conjunto de dados	74
4.3.2	Método proposto para AA de conjunto aberto	75
4.3.3	Método proposto para AA de conjunto aberto	76
4.3.4	Avaliação	77
4.3.5	Resultados	78

4.3.6	Considerações sobre a AA de conjunto aberto	80
5	Método proposto: Atribuição autoral dinâmica	82
5.1	<i>Modelo de atribuição autoral geral</i>	83
5.2	<i>Modelo baseado em anotações POS e dependência sintática</i>	83
5.3	<i>Modelo baseado em word embeddings</i>	84
5.4	<i>Modelo de atribuição autoral dinâmica - DynAA</i>	85
5.5	<i>Considerações</i>	87
6	Avaliação	88
6.1	<i>Conjunto de dados</i>	88
6.1.1	Corpus de avaliação PAN-CLEF-2018 - <i>PAN-EVAL-2018</i>	88
6.1.2	Corpus de letras de músicas - <i>Lyrics</i>	89
6.1.3	Corpus de redes sociais - Social	90
6.2	<i>Resultados</i>	93
6.3	<i>Discussões sobre os resultados</i>	96
6.3.1	Discussões sobre os resultados para PAN-EVAL-2018	96
6.3.2	Discussões sobre os resultados para Lyrics	97
6.3.3	Discussões sobre os resultados para Social	97
6.3.4	Considerações sobre o método proposto	98
6.4	<i>Análise complementar - relevância dos n-gramas de caracteres</i>	99
7	Conclusões	103
7.1	<i>Sobre as hipóteses de pesquisa</i>	103
7.2	<i>As variações do problema de AA</i>	104
7.3	<i>Os diferentes domínios</i>	104
7.4	<i>Os diferentes idiomas</i>	104
7.5	<i>As técnicas de aprendizado de máquina</i>	105
7.6	<i>Trabalhos futuros</i>	105
	Referências¹	106
	Apêndice A – Apêndice	117

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

<i>A.1</i>	<i>Relevância das características para o experimento de AA multidomínio</i>	
	<i>PAN2018</i>	<i>117</i>

1 Introdução

A atribuição autoral de textos digitais (AA) (do inglês, *Authorship Attribution*) visa identificar quem é o autor de um determinado texto a partir de um conjunto de autores possíveis (POTTHAST et al., 2017). A premissa principal da AA é que o autor deixa rastros de seu estilo, sendo que esses rastros podem ser a preferência por certas palavras, o tamanho do vocabulário, a utilização de pontuação e a repetição de certos elementos gramaticais. Quantificar essas informações é uma tarefa conhecida por estilometria (STAMATATOS, 2009).

A AA é uma tarefa importante para inúmeras atividades cotidianas e, em especial, para análise forense. Sua aplicação pode ajudar na solução de casos de escândalos de corrupção, como no caso Enron (KLIMT; YANG, 2004; CHEN et al., 2011; JUOLA; STAMATATOS, 2013), na identificação de abusos na utilização da internet (VARTAPETITANCE; GILLAM, 2012), na detecção de notícias falsas (PENG; CHOO; ASHMAN, 2016), na detecção de casos onde uma pessoa tenta se passar por outra (KOPPEL; SEIDMAN, 2018), na atribuição autoral de código-fonte (ALSULAMI et al., 2017), na detecção de pseudônimos (JUOLA, 2015) e outros.

A AA é uma tarefa de interesse das áreas de humanidades digitais, da análise forense de textos e da linguística computacional. Os métodos computacionais para atribuição autoral utilizam análise estatística multivariada (SAVOY, 2016; EVERT et al., 2017), métodos baseados em vizinho mais próximo (KOCHER; SAVOY, 2017; KOPPEL; SEIDMAN, 2018; VARELA et al., 2016), aprendizado de máquina com SVM (SCHWARTZ et al., 2013; STAMATATOS, 2017), redes neurais recorrentes (BAGNALL, 2016), redes neurais de convolução (SHRESTHA et al., 2017; SARI; STEVENSON, 2016), modelos de compressão (HALVANI; GRANER, 2018) e outros.

Apesar desta ampla gama de iniciativas, a AA é um problema de pesquisa não totalmente resolvido (POTTHAST et al., 2017), sendo inclusive o tema da série de competições PAN-CLEF (KESTEMONT et al., 2018). De forma mais específica, estudos desta área exploram técnicas independentes de idioma e de domínio, e assim subutilizam os recursos linguístico-computacionais. Por exemplo, no trabalho em Tschuggnall et al. (2017) pode ser visto que as mesmas técnicas de AA apresentam desempenhos diferentes quando aplicadas a idiomas diferentes. Além disso, os trabalhos existentes, de modo geral,

são voltados para o idioma inglês, sendo raros os trabalhos para o português brasileiro (VARELA et al., 2016).

Um dos pontos chaves para os estudos da AA é a forma de extração de características do textos, sendo n-gramas de caracteres um dos métodos mais difundidos. Além dele, é possível encontrar n-grama de palavras, anotações linguísticas, de representações distribuídas (*word embeddings*, cf. Bengio et al. (2003)). No geral, essas fontes de conhecimentos são estudadas de forma independentes umas das outras.

Com base nestas observações, o restante deste documento apresenta uma pesquisa em nível de mestrado acadêmico tratando do problema computacional de AA em textos digitais.

1.1 Hipóteses

Este trabalho considera as seguintes hipóteses:

- H1: A combinação de mais de um tipo de extração de característica aumentará o desempenho de sistema.
- H2: O uso de modelos dependentes de idioma do tipo *part-of-speech* aumentará o desempenho de sistemas de AA.
- H3: O uso de modelos dependentes de idioma do tipo representação distribuída (*embeddings*) aumentará o desempenho de sistemas de AA.

Estas hipóteses serão testadas comparando-se os modelos propostos com sistemas de *baseline* que se façam pertinentes. A avaliação será feita por meio de medidas tradicionais em aprendizado de máquina, como a medida F, acurácia e outros. Espera-se que o resultado médio seja superior ao dos modelos de *baseline* de acordo com as métricas estipuladas.

1.2 Objetivo

O objetivo geral deste trabalho é enriquecer modelos de atribuição autoral de texto digitais com conjunto fechado de autores utilizando conhecimentos dependentes e independentes de idioma combinados com técnicas de aprendizados de máquina, de modo a obter resultados superiores ao estabelecido em trabalhos anteriores.

Os modelos enriquecidos com tais técnicas serão avaliados nos domínios literatura, microblog e letras de músicas, e em diversos idiomas, sendo de especial interesse os idiomas português brasileiro e inglês. Os corpuses considerados são: PAN-CLEF2014 (STAMATATOS et al., 2014), PAN-CLEF2018 (KESTEMONT et al., 2018), PAN-CLEF2018 (KESTEMONT et al., 2019), microblogs (ROCHA et al., 2017), e Lyrics (desenvolvido para esse trabalho). Tendo suporte a atribuição autoral conforme será visto no capítulo 6.

1.3 Organização do documento

O restante deste documento está organizado da seguinte forma. Os capítulos seguintes apresentam as revisões bibliográficas de conceitos fundamentais (Capítulo 2), discussões acerca de estudos correlatos encontrados na literatura (Capítulo 3), estudo exploratório (Capítulo 4), método proposto (Capítulo 5), avaliação (Capítulo 6) e, por fim, as conclusões (Capítulo 7).

2 Conceitos fundamentais

Esse capítulo apresenta os conceitos fundamentais, relacionados ao escopo deste projeto de pesquisa, acerca das técnicas usadas na área de atribuição autoral.

2.1 Atribuição Autoral

A atribuição autoral (AA) é uma técnica computacional que visa a identificar o autor de um texto a partir de um conjunto de autores possíveis, baseando-se em padrões de estilo deixados pelos autores (POTTHAST et al., 2017). A quantificação do estilo de escrita, ou estilometria, compreende um vasto conjunto de medidas e técnicas que buscam extrair uma “biometria” textual (NEAL et al., 2017).

Do ponto de vista de aprendizado de máquina, a AA pode ser vista como um problema de classificação multi-classes (STAMATATOS, 2009). Em certas situações, é interessante ter um grau de confiança, ou probabilidade, de um documento pertencer a um autor (NEAL et al., 2017). Apesar da similaridade com a tradicional tarefa de classificação de documentos, a AA busca elementos inconscientes da escrita que sejam independentes do conteúdo semântico do texto (KEŠELJ et al., 2003).

Os avanços tecnológicos aumentaram e diversificaram o número de fontes de dados textuais. Isso gerou a necessidade de expansão da AA para textos originados em blogs, redes sociais e sistemas de mensagens curtas, motivando o avanço das técnicas existentes que foram criadas para textos literários. Entre os fatores que impactam o desempenho de sistema de AA destacam-se:

- **Canal:** O meio de comunicação, ou a ferramenta utilizada, influencia o modo de escrita, como é o caso dos sistemas de mensagens curtas. Estes impõem um estilo enxuto, são tolerantes a erros de ortografia e utilizam símbolos gráficos como *emojicons* e *hashtags* (SHRESTHA et al., 2017).
- **Idioma:** As complexidades lexicais, morfológicas e sintáticas são diferentes por língua, dando aos autores um leque maior, ou menor, de possibilidades para expressar seus estilos. O trabalho em Tschuggnall et al. (2017) apontou resultados com patamares diferentes para um mesmo método, quando aplicado a *corpus* em diferentes línguas.

- **Domínio ou Gênero do texto:** Gênero para AA refere-se a formatos narrativos. Exemplos desses formatos são contos, romances, artigos de jornais e avaliações de produtos (KESTEMONT et al., 2012).
- **Tópico:** refere-se ao assunto tratado no texto. Por exemplo, um autor utiliza um conjunto de palavras para falar sobre a guerra no Iraque, e outro para falar sobre legalização das drogas (SAPKOTA et al., 2014).
- **Tamanho do texto:** Os métodos de representação de textos são baseados em contagem de palavras e caracteres, o que produz representações esparsas e com alta dimensionalidade (TURNERY; PANTEL, 2010). Dado isso, a AA de textos curtos é mais desafiadora quando comparada a textos longos (SCHWARTZ et al., 2013; SHRESTHA et al., 2017).
- **Número de autores:** Os métodos de aprendizado de máquina são afetados pelo número de classes de saída, exigindo um conjunto maior de dados de entrada (CHAWLA et al., 2002). Diversos estudos analisados no capítulo 3 apresentaram redução no desempenho com o aumento no número de autores.

Esses fatores motivam diferentes modalidades para a AA. Essas abordagens, ou subtarefas, são enumeradas a seguir:

- **Atribuição autoral de conjunto fechado:** nessa tarefa, os textos do conjunto de teste pertencem a um dos autores candidatos presentes no corpus de treinamento (KOPPEL; SCHLER; ARGAMON, 2009; KESTEMONT et al., 2018). Essa tarefa é explorada no presente trabalho.
- **Atribuição autoral de conjunto aberto:** nessa tarefa, os textos do conjunto de teste não necessariamente foram escritos por um dos autores do corpus de treinamento (STAMATATOS et al., 2014). Nesse caso, o classificador deverá ser capaz de lidar com a classe *autor desconhecido*.
- **Caracterização autoral:** busca-se extrair informações demográficas do autor do texto. Dados como idade, língua materna, gênero e outros podem ajudar a reduzir a lista de autores candidatos (STAMATATOS et al., 2014).
- **k-Atribuição autoral:** também conhecida por ordenação autoral ou k-vizinhos, é uma forma mais flexível do problema de atribuição, onde as saídas do classificador são ordenadas pela probabilidade e são retornados os k autores mais prováveis. De

forma similar à caracterização autoral, essa abordagem ajuda a reduzir o espaço de busca (AFROZ et al., 2014).

- **Verificação autoral:** busca verificar se dois documentos foram escritos pelo mesmo autor (KOPPEL et al., 2012), não sendo necessário saber quem são os autores. Essa abordagem é utilizada indiretamente na tarefa de AA por meio de métodos de distância e será detalhada na seção 2.4.
- **Agrupamento autoral:** tem como objetivo agrupar uma coleção de documentos em conjuntos, de tal forma que, cada conjunto contenha textos de apenas um autor (ROSSO et al., 2016).
- **Ligação autoral:** é uma versão mais flexível do agrupamento autoral, em que pares de documentos são ordenados pela confiança de pertencerem ao mesmo autor (TSCHUGGNALL et al., 2017).
- **Quebra de estilo:** busca identificar se um documento possui autoria única ou múltipla e, conseqüentemente, deseja-se saber em quais pontos houve quebra de estilo (KESTEMONT et al., 2018). Esse problema pode ser visto como uma tarefa de agrupamento autoral de parágrafos.

O trabalho em Kestemont et al. (2019) define formalmente a atribuição autoral como um tupla (A, K, U) , sendo A o conjunto de autores conhecidos, K o conjunto de textos dos autores conhecidos, e U o conjunto de textos de autoria desconhecida. Para cada $a \in A$ é possível definir os $k_a \in K$ tal que contenha os textos escritos pelo autor $a \in A$. Cada documento em U deve ser associado a um $a \in A$. No caso da AA de conjunto fechado sempre haverá um $a \in A$. Já na atribuição de conjunto aberto a pode ser vazio. Do ponto de vista de categorização de textos, K é o conjunto de treinamento e U é o conjunto de testes.

Os estudos de AA adotam diversos tipos de conhecimento para extrair o estilo de escrita de um autor. Esses tipos são definidos por características dependentes e independentes de idioma. Em Neal et al. (2017) e em Rocha et al. (2017) são apresentados os principais tipos de características utilizadas. De modo geral, são baseadas em cinco categorias de informação.

- **Categoria lexical:** são formadas por estatísticas baseadas em caracteres e palavras. Exemplos desta categoria são tamanho médio das palavras, número de letras maiúsculas, tamanho das sentenças, etc (GHAEINI, 2013).

- **Categoria sintática:** são formadas por estatísticas que medem a organização das sentenças. Exemplos desta categoria são frequência da pontuação (STAMATATOS, 2017), palavras de função (KESTEMONT, 2014), etc.
- **Categoria semântica:** são formadas por técnicas que buscam capturar o significado das palavras e das sentenças. Exemplos desta categoria são contagem das palavras, analisadores semânticos (VARELA et al., 2016), *word embeddings* (SARI; STEVENSON, 2016), etc.
- **Categoria específica de domínio:** são formadas por estatísticas baseadas em palavras-chave, *tags* HTML (ALSULAMI et al., 2017), *emoticons* (SHRESTHA et al., 2017), etc.

Além dessa classificação, é possível organizar o conhecimento em dois grandes grupos: palavras e caracteres. As palavras mais frequentes de um texto são consideradas fontes confiáveis para a AA por serem independentes de domínio e por serem utilizadas de forma inconsciente (KESTEMONT, 2014). Um subconjunto delas é conhecido por *palavras de função* (do inglês, *function words*), e compreende artigos, preposições, locuções adverbiais, e outros.

As sequências de caracteres são considerados os modelos mais efetivos para AA (KJELL; WOODS; FRIEDER, 1994; KESTEMONT, 2014; NEAL et al., 2017), sendo que uma contagem simples pode produzir modelos próximos ao estado-da-arte (NEAL et al., 2017). Assim como as palavras, os caracteres mais frequentes (CNG) (do inglês, *common n-grams*) são considerados as fontes de conhecimento mais confiáveis (KEŠELJ et al., 2003; SAPKOTA et al., 2014) e capturam aspectos sintáticos e morfológicos.

2.2 Métodos computacionais para atribuição autoral

Os métodos para AA podem ser agrupados em famílias (KOPPEL; SCHLER; ARGAMON, 2009; NEAL et al., 2017; ROCHA et al., 2017). Os primeiros métodos utilizados foram os variantes unitários, seguidos por estatística multivariada, e mais recentemente por técnicas de similaridade e aprendizado de máquina.

Os variantes unitários extraem métricas únicas para representar todo o texto. Um exemplo de métrica é a inclinação da curva Zipf, que é um fator entre curvas da distribuição da frequências das palavras no texto e curva da distribuição de probabilidades Zipf. Apesar de seus resultados promissores, essas abordagens se demonstraram limitadas (SAVOY, 2012). Os métodos de estatística multivariada nascem com o trabalho de Mosteller e Wallace (1963) em uma versão rudimentar do algoritmo Naïve Bayes aplicado no corpus *The Federalist Papers*¹, e são considerados o início da AA assistida por computador. As técnicas de similaridade compreendem os métodos baseados em distância e, devido a suas aplicações em várias subtarefas da AA, serão vistos em detalhes na seção 2.4. Os métodos de aprendizado de máquina ganharam popularidade a partir dos anos 2000, e métodos desse tipo serão utilizados no capítulo 4 deste trabalho.

Além dessas abordagens, podem ser encontrados modelos utilizando redes complexas, cadeias de Markov, técnicas de compressão de texto e outros. Mais informações podem ser encontradas nas revisões de literatura em Rocha et al. (2017), que possui viés computacional, em Neal et al. (2017), com viés de análise estilométrica, e em Savoy (2012) com viés de estatística multivariada.

O presente trabalho irá analisar os modelos de representação tradicional na seção 2.3, onde será visto o modelo tradicional *Bag of Words*, BOW, e os modelos recentes de representação distribuída. Em seguida, serão vistos os métodos baseados em vizinhos mais próximos na seção 2.4 e aprendizado de máquina.

2.3 Modelos computacionais de representação textual

Nessa seção são explorados os métodos computacionais para representação textual. São vistos os modelos tradicionais devido à consistência dos resultados, e os modelos baseados em *word embeddings* devido ao destaque recebido na literatura mais recente.

¹ Disponível em <<http://www.gutenberg.org/ebooks/18>>.

2.3.1 Representações tradicionais *one-hot*, *bag of words* e n-gramas

O modelo *one-hot* (GOLDBERG, 2017) representa um símbolo, ou termo, de um texto utilizando a forma vetorial $\vec{t}_d \in \mathbb{N}$ indicando a ocorrência, ou não, do termo. Nesse modelo, o termo é representado pelo vetor $\vec{t}$ de dimensão d igual ao total de palavras únicas no vocabulário do texto, sendo todas as posições preenchidas com zeros, exceto a posição i referente ao índice do termo no vocabulário.

Tabela 1 – Representação do vetor *one-hot* da palavra avião para um dado vocabulário.

Termo	O	avião	decolou	hoje
Avião	0	1	0	0

Fonte: José Eleandro Custódio, 2020

O modelo *bag of words* (BOW) (SALTON; WONG; YANG, 1975; TURNEY; PANTEL, 2010) 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 número de palavras únicas no vocabulário do texto, e cada dimensão contendo a contagem das palavras no texto. Esse modelo desconsidera elementos gramaticais e a ordem das palavras. O modelo BOW pode, também, ser definido pela soma dos vetores *one-hot* dos termos presentes no documento (GOLDBERG, 2017). Na tabela 2 pode ser visto um exemplo da matriz de contagem para dois documentos.

Tabela 2 – Matriz de contagem da representação *bag-of-words*

Documento	Maria	João	foi	para	Natal	Salvador	e	depois	Recife
Maria foi para Natal	1	0	1	1	1	0	0	0	0
João foi para Salvador e depois foi para Recife	0	1	2	2	0	1	1	1	1

Fonte: José Eleandro Custódio, 2020

Apesar da representação simplificada, o modelo BOW é utilizado com sucesso em diversas tarefas relacionadas ao processamento de língua natural (PLN) (TURNEY; PANTEL, 2010). No entanto, uma das desvantagens desse modelo é sua representação esparsa, sendo comum atingir de 10 a 40 mil dimensões e descarta a ordem das palavras.

O modelo n-gramas é uma extensão do modelo BOW e codifica sequências consecutivas de termos em um termo único. Esses novos termos podem ser palavras (n-gramas de palavras), caracteres (n-gramas de caracteres), anotações de classes gramaticais (POS)

(n-gramas de POS), palavras concatenadas com POS (n-gramas de palavra com POS), etc. Sequências com $n = 1$ são chamadas de unigramas, com $n = 2$ de bigramas, etc.

Modelos do tipo BOW e n-gramas utilizando frequências absolutas não são comparáveis entre si. Ao mesmo tempo, termos muito frequentes no cópua costumam ter baixa capacidade preditiva. Para contornar ambos os problemas, utiliza-se o modelo TF-IDF (SALTON; WONG; YANG, 1975), onde a contagem do termo é substituída pela frequência no documento (TF) (do inglês, *term frequency*) ponderada pela frequência no cópua (IDF) (do inglês, *inverse document frequency*). A normalização TF é feita pela norma L1 e consiste da divisão de cada termo pela frequência total, e é dada na equação $TF(t, d) = \frac{Contagem(t,d)}{\sum_{j=1}^N Contagem(j,d)}$. A normalização IDF penaliza o termo t considerando o número de documentos onde ele ocorre, dado por $DF(t, D)$, considerando o número total de documento D . Na equação 1 é apresentado a definição original definida em Salton, Wong e Yang (1975) e na equação 2 é apresentada uma versão suavizada definida em Pedregosa et al. (2011).

$$IDF(t, D) = \log \left(\frac{D}{1 + DF(t)} \right) \quad (1)$$

$$IDF_{Suavizado}(t, D) = \log \left(\frac{D}{DF(t)} \right) + 1 \quad (2)$$

2.3.2 Representação distribuída de palavras (*Word Embeddings*)

Os modelos baseados em n-gramas utilizam uma única dimensão para representar a palavra e como consequência, palavras similares ganham representações completamente diferentes. Por exemplo, os vetores que representam as palavras “prédio”, “gato” e “artigo” são diferentes do ponto de vista matemático. Além disso, a informação de sequência não é capturada completamente pelo modelo de n-gramas, devido ao alcance limitado de subsequências.

Diversos modelos tentaram superar essas deficiências valendo-se da hipótese distribucional (HARRIS, 1954; TURNEY; PANTEL, 2010). Essa hipótese estabelece que o significado de uma palavra pode ser inferido pelo contexto, e que palavras que ocorrem em um mesmo contexto tendem a ter significado similares. Os métodos mais conhecidos que utilizam a hipótese distribucional são análise de semântica latente (LSA) (DEERWESTER

et al., 1990), Latent Dirichlet Allocation (LDA) (BLEI; NG; JORDAN, 2003) e, mais recentemente, modelos neurais de língua natural (BENGIO et al., 2003; MIKOLOV et al., 2013). Outros modelos baseados nessa premissa são analisados em Turian, Ratinov e Bengio (2010).

Os modelos baseados na hipótese distribucional utilizam vetores densos para representar cada palavra. Os vetores $\vec{w}_d \in \mathbb{R}^d$ geralmente possuem número de dimensões d menor que o número de símbolos distintos no vocabulário V . O processo matemático de mapear um espaço de $|V|$ dimensões em um espaço de $d \leq |V|$ é conhecido como *embedding*. Já o processo de mapear um termo t baseado em seus vizinhos é conhecido como *word embeddings*. Por serem utilizados como entrada para outros modelos de aprendizado de máquina, esses modelos são classificados como aprendizado não supervisionados, semi-supervisionados e, algumas vezes, aprendizados de representação (TURIAN; RATINOV; BENGIO, 2010). Os vetores de *embeddings* são sumarizados geralmente por média, e utilizados como entrada para algoritmos de classificação como SVM, regressão logística e outros (SARI; STEVENSON, 2016).

Em Bengio et al. (2003) são apresentados os modelos neurais de língua natural. Esses modelos utilizam redes neurais artificiais (RNA) para geração de *embeddings* de palavras por meio da maximização da probabilidade de uma palavra ocorrer dado as palavras precedentes. Nesse trabalho foi utilizado a representação *one-hot*, função de ativação tanh e saída com função *softmax*. Esse modelo foi base para diversas propostas, e discutiremos a seguir dois modelos baseados nessa premissa, Word2Vec (MIKOLOV et al., 2013) e Doc2vec (LE; MIKOLOV, 2014). Além desses, outros modelos podem ser encontrados. O trabalho em Pennington, Socher e Manning (2014) propôs o método *Glove* que constrói os *embeddings* a partir da fatoração da matriz de co-ocorrência. O trabalho em Joulin et al. (2017), Bojanowski et al. (2017a) propôs o método *fasttext*² (BOJANOWSKI et al., 2017b) que considera partes das palavras na construção dos *embeddings*.

Word2vec - Representação distribuída de palavras

O trabalho em Mikolov et al. (2013) apresentou o modelo Word2Vec nas arquiteturas CBOW e Skip-gram, e as abordagens de treinamento da rede com *hierarquical softmax* (RUMELHART; HINTON; WILLIAMS, 1988) e *negative sampling*. Ambas as arquiteturas

² Disponível em: <https://fasttext.cc/>

valem-se de uma tarefa “falsa” que busca calcular a probabilidade de uma palavra e , após o treinamento, tanto a entrada como a saída são descartadas e os pesos da rede são utilizados como *embeddings* para representação das palavras. As arquiteturas estão ilustradas na figura 1.

A arquitetura CBOW considera o contexto contendo as c palavras anteriores ($w_{-c:1}$) e as c posteriores ($w_{1:c}$) para calcular a probabilidade $P(w|w_{-c:1}, w_{1:c})$ de uma palavra w ao centro. Os vetores *one-hot* são consolidados por média, o neurônio central utiliza ativação linear e a saída usa a função *hierarchical softmax*, ou *negative sampling*. Por não considerar a ordem das palavras, esse modelo foi chamado de *Continuous Bag of words*. A arquitetura *Skip-gram* utiliza processo similar, mas usa a palavra no centro para prever o contexto. A representação da entrada é dado em *one-hot*.

Figura 1 – Arquitetura do modelo Word2Vec

Fonte: Mikolov et al. (2013)

CBOW e *Skipgram* são relevantes para PLN por uma série de fatores. Primeiramente, os modelos propostos apresentaram algumas propriedades algébricas e semânticas possibilitando operações como, o vetor(*biggest*) - vetor(*big*) + vetor(*small*) $\sim$ vetor(*smallest*). Além disso, foi proposto o novo algoritmo de otimização *negative sampling* como uma alternativa ao *hierarchical softmax*. Mais detalhes desse método pode ser encontrado em

Goldberg e Levy (2014). Finalmente, o método Word2Vec foi avaliado em uma base com bilhões de palavras, sendo o resultado disponibilizado publicamente³.

Doc2vec - Representação distribuída de parágrafos

O trabalho em Le e Mikolov (2014) propõe uma representação distribuída para sentenças, parágrafos ou documentos chamada Doc2Vec. De forma análoga ao modelo Word2vec, foram propostos duas arquiteturas. A arquitetura Memória Distribuída de Vetores de Parágrafos (*PV-DM*) (do inglês, *Paragraph Vectors of Distributed Memory*) concatena o vetor *one-hot* dos parágrafos (ou documentos) com os vetores *one-hot* para cada palavra e calcula a probabilidade da palavra ao centro. A arquitetura *Bag of Words* Distribuída de Vetores de Parágrafos (*PV-DBOW*) (do inglês, *Paragraph Vectors of Distributed Bag of words*) adota o vetor *one-hot* do parágrafo para calcular a probabilidade das palavras do contexto, e descartando a ordem das palavras. A arquitetura DM apresentou resultados superiores ao DBOW quando aplicado à tarefa de análise de sentimento. Os métodos descritos estão ilustradas na figura 2.

Por meio das arquiteturas PV-DM e PV-DBOW é possível extrair a representação distribuída das palavras e também dos documentos. Entre suas aplicações encontram-se análise de sentimento e classificação de documentos.

2.4 Métodos computacionais baseados em vizinho mais próximo

A atribuição autoral pode ser feita usando métodos baseados em vizinho mais próximo (STAMATATOS, 2009). Nessa abordagem, os documentos são transformados em vetores, e os pares são comparados por uma medida de distância, ou de dissimilaridade (KOPPEL et al., 2012). Os documentos cuja autoria se deseja validar são atribuídos ao autor do texto que obteve menor distância entre os documentos de autoria conhecida.

Caso haja mais de um documento por autor, é possível usar os paradigmas *por perfil* ou *por instância*. No paradigma por perfil, concatena-se todos os textos de um mesmo autor. No paradigma por instância, são gerados múltiplos resultados que exigem uma etapa adicional para sumarização por média, votos e outros.

³ A implementação original e a base pré-treinada para língua inglesa está disponível em: <https://code.google.com/archive/p/word2vec/>

Figura 2 – Arquitetura *Distributed Memory* e *Distributed Bag of Words*

(a) Modelo PV-DM

(b) PV-DBOW

Fonte: Le e Mikolov (2014)

Os métodos baseados em distância podem ser aplicados na atribuição de conjunto fechado e de conjunto aberto. Essencialmente, são os métodos centrais para a verificação, ligação e clusterização autoral. A seguir, é conceituada a definição de distância, são listadas as principais funções deste tipo e são explorados os métodos específicos da área de análise autoral.

2.4.1 Medidas de distâncias textuais de uso geral

As medidas, ou funções, de distância devem obedecer quatro propriedades (DEZA; DEZA, 2009):

1. $D(a, b) \geq 0$ para todo a e b .
2. $D(a, b) = 0$ se e, somente se, $a = b$.
3. $D(a, b) = D(b, a)$, d é uma função simétrica.
4. $D(a, c) \leq D(a, b) + D(b, c)$, a desigualdade triangular.

Diversas medidas encontradas na literatura seguem essas propriedades. Mais informações sobre as propriedades matemáticas e uma lista extensa de medidas deste tipo pode ser encontrada em Deza e Deza (2009). A seguir, são listadas as medidas utilizadas nos trabalhos de AA que serão discutidos no capítulo 3, suas definições informais, e suas definições matemáticas em termos dos vetores de documentos a a b :

- *Distância de cossenos* é baseada na medida de *similaridade cossenos* (Cos), uma medida usada na área de recuperação da informação para comparação de documentos (GROSSMAN; FRIEDER, 2012). Cos é definida como o cosseno do ângulo entre dois vetores. Cossenos próximos a 1 indicam que os vetores apontam na mesma direção, e portanto significa concordância. Valores próximos a 0 indicam que os vetores são ortogonais, significando neutralidade. Valores próximos a -1 indicam que os vetores apontam para direções oposta e significam discordância:

$$\text{Cossenos}(a, b) = 1 - \text{Cos}(a, b) = 1 - \frac{a \cdot b}{\|a\|_2 \|b\|_2} \quad (3)$$

- *Manhattan*, também conhecida como quarteirão ou $L1$: é a soma das distâncias absolutas de cada dimensão (DEZA; DEZA, 2009):

$$\text{Manhattan}(a, b) = \sum_i |a_i - b_i| \quad (4)$$

- *Bray-Curtis* é a distância de Manhattan normalizada para o intervalo $[0, 0, 1, 0]$ (DEZA; DEZA, 2009):

$$\text{Bray} - \text{Curtis}(a, b) = \frac{\sum |a_i - b_i|}{\sum |a_i + b_i|} \quad (5)$$

- *Euclidiana*, também conhecida como $L2$, mede a reta entre dois pontos (DEZA; DEZA, 2009). Quando utilizada para comparação, a distância euclidiana penaliza cada desvio quadraticamente:

$$\text{Euclidiana}(a, b) = \|a - b\|_2 = \sqrt{\sum_i |a_i - b_i|^2} \quad (6)$$

- *Chebyshev* utiliza a dimensão com maior desvio absoluto (DEZA; DEZA, 2009):

$$\text{ChebyshevD}(a, b) = \max_i |a_i - b_i| \quad (7)$$

- *Divergência de Kullback-Leibler* (KULLBACK; LEIBLER, 1951) mede a divergência entre duas distribuições de probabilidades, e é definida pela equação 8. Na equação 9 é apresentada uma versão simétrica da distância de entre as distribuições A e B :

$$\text{KLD}(a, b) = - \sum_i^n a_i * \log_2 \left(\frac{a_i}{b_i} \right) \quad (8)$$

$$KLD_{Simétrica}(a, b) = KLD(a, b) + KLD(b, a) \quad (9)$$

- *Jaccard*: é uma medida binária que avalia o percentual interseção entre dois conjuntos (MANNING; SCHÜTZE, 1999):

$$Jaccard(A, B) = \frac{|a \cap b|}{|a \cup b|} \quad (10)$$

- *MinMax, ou Jaccard generalizado*: é definido pela soma dos mínimos dos elementos i de a e b dividido pela soma dos máximos. O resultado está delimitado entre $[0, 0, 1, 0]$:

$$MinMax(a, b) = \frac{\sum_i^n \min(a_i, b_i)}{\sum_i^n \max(a_i, b_i)} \quad (11)$$

2.4.2 Medidas de distância e técnicas específicas da AA

Entre as medidas de distâncias concebidas no contexto da atribuição autoral destacam-se quatro: A regra delta de Burrows (BURROWS, 2002), a distância de Keselj (KEŠELJ et al., 2003), Stamatatos (STAMATATOS, 2009) e SCAP (FRANTZESKOU et al., 2006). Cada uma dessas medidas é descrita a seguir.

Regra Δ de Burrows

A regra Δ de Burrows é um método proposto em Burrows (2002) e possui grande destaque entre os métodos de AA por similaridade (STAMATATOS, 2009; EVERT et al., 2015). A definição original pode ser vista na equação 12, e consiste de uma sequência de passos. São contadas as ocorrências das 150 palavras mais frequentes, e os vetores A e B são padronizados por Z-score usando a média μ e o desvio-padrão σ do corpus. Após a padronização, é calculada a distância de Manhattan, e dividida pelo número de dimensões.

$$\Delta Burrow(a, b) = \frac{1}{N} \sum_{i=1}^N |Zscore(a_i, c_i) - Zscore(b_i, c_i)| \quad (12)$$

$$Zscore(x_i, c_i) = \frac{X_i - \mu(c_i)}{\sigma(c_i)}$$

A regra delta pode ser separada em componentes como tipo de conhecimento, técnica de normalização do documento, técnica de normalização do corpus e função de

distância. Alguns estudos foram dedicados a entender cada componente, como Evert et al. (2015) e Savoy (2016).

Distância de Kešelj e distância de Stamatatos

O estudo em Kešelj et al. (2003) propôs o método de n-gramas mais comuns (do inglês, *Common N-grams* ou CNG) para atribuição autoral. Nesse método, é feita a contagem dos n-gramas no cópua, são selecionados os mais frequentes e é gerado um vocabulário. Esse vocabulário é utilizado para construção dos vetores de frequência relativa para os documentos A e B , sumarizados pela distância dada na equação 13.

$$Keshelj(a, b) = \sum_i^n \left(\frac{2 * (a_i - b_i)}{a_i + b_i} \right)^2 \quad (13)$$

Stamatatos (2009) propõe a utilização de um termo adicional que visa ponderar a distância de cada termo usando a distância de A versus o cópua C .

$$Stamatatos(a, b, c) = \sum_i^n \left(\frac{2 * (a_i - b_i)}{a_i + b_i} \right)^2 * \left(\frac{2 * (a_i - c_i)}{a_i + c_i} \right)^2 \quad (14)$$

SCAP - Source Code Authorship Profiles

SCAP é um método proposto em Frantzeskou et al. (2006) para atribuição autoral de códigos-fontes, sendo também utilizado em outros tipos de textos, como mensagens curtas (ROCHA et al., 2017) e textos longos (POTHA; STAMATATOS, 2014). O método consiste da extração dos n-gramas mais frequentes dos perfis dos autores A e B e do cálculo da interseção entre os vetores. A definição formal é dada na equação 15.

$$SCAP(A, B) = \|A \cap B\| \quad (15)$$

2.5 Seleção de variáveis com Kolmogorov-Smirnov

O problema de aprendizado de máquina frequentemente envolve análise de variáveis contínuas versus variáveis binárias. Exemplo desse cenário é a frequência de uma palavra “arroz” versus pertencer, ou não, à classe alimento. Esse problema aparece na utilização das

medidas de similaridade onde temos a variável contínua dada pela distância e a variável binária dada por confirmação de ser ou não do mesmo autor.

A distância de Kolmogorov-Smirnov (KS) permite avaliar a relação entre uma variável contínua e uma binária. A construção desse método pode ser vista na figura 3, onde uma variável hipotética X associada a uma variável binária Y foi dividida em duas populações. A população para os casos onde $Y = 0$ está representada pelo histograma em fundo liso e distribuição de probabilidade acumulada com linha contínua. A população para os casos onde $Y = 1$ está representada em fundo tracejado e linha tracejada. A distância de KS é definida pela maior distância entre as curvas de probabilidades e é formalmente definida pela equação 16, sendo CDF a função de probabilidade acumulada.

Essa métrica foi utilizada para análise exploratória no experimento sobre verificação autoral descrito na seção 4.1. Mais informações sobre essa medida e suas aplicações podem ser encontradas em Thomas, Crook e Edelman (2017).

Figura 3 – Representação distância Komogorov-Smirnov

Fonte: José Eleandro Custódio, 2020

$$KS(x, y) = \arg\max_x \text{abs}(CDF(x|y = 0) - CDF(x|y = 1)) \quad (16)$$

2.6 Método de aprendizado de máquina

Nessa seção são discutidos alguns métodos de aprendizado de máquina para classificação de uso comum em estudos de AA. Foram considerados os modelos de regressão logística *softmax*, rede neural de convolução e rede neural recorrente. A regressão *softmax* foi escolhida por sua relativa simplicidade. Os modelos conexionistas de convolução e recorrente foram escolhidos devido aos resultados em diversas tarefas. É assumido que o leitor possui familiaridade com a rede neural perceptron com multi-camadas (MLP) (do inglês, *multi-layer perceptron*) (ROSENBLATT, 1962).

2.6.1 Regressão logística *softmax* ou classificador de máxima entropia

Essa seção faz uma breve discussão sobre a regressão *softmax*. Por ter sido descoberto por mais de uma linha de pesquisa, esse algoritmo ganha nomes, algoritmos de otimização e premissas de aplicabilidade diferentes. Nas áreas de estatística e economia, por exemplo, o termo mais utilizado é “regressão logística multinomial” (CHRISTENSEN, 2006; JR; LEMESHOW; STURDIVANT, 2013). Já em processamento de língua natural, é comum encontrar o termo classificador de máxima entropia (MANNING; SCHÜTZE, 1999).

De modo geral, esse algoritmo é um classificador multi-classes com interpretação probabilística. Em comparação com demais métodos de aprendizado de máquina, traz as vantagens de ser inerentemente multi-classes, que é um ponto fraco do algoritmo SVM. Além disso, o método não assume a independência das variáveis de entrada, que é um ponto fraco do algoritmo Naïve Bayes, e apresenta uma saída de probabilidade contínua, em oposição aos degraus dos métodos de árvore. Além da aplicação como classificador, esse método é utilizado como normalizador da camada de saída em redes neurais artificiais e pode ser visto nos trabalhos apresentados na seção 3.4. A logística *softmax* foi escolhido como classificador nos experimentos apresentados na seção 4.

Formalmente, seja $f_c(X)$ uma função que transforma uma matriz de entrada X em um vetor $P(Y = c|X)$, referente à probabilidade de X ser da classe c . Assumindo que $f_c(X)$ é uma combinação linear dos pesos β com X , temos que $f_c(X) = \beta_{0,c} + \beta_c X^T$, onde $\beta_{0,c}$ representa o intercepto. Para cada $x \in X$ temos um vetor de saída $f_c(x)$ com tamanho $|c|$, e é possível ver que as saídas não estão normalizadas. Para produzir a

normalização, é introduzido um fator $\ln Z$ de tal forma que $f_c(X) - \ln Z = \ln P(Y = c|X)$. O compartilhamento de $\ln Z$ entre as equações das classes produz o efeito do balanceamento das probabilidades. Assim temos as seguintes equações (CHRISTENSEN, 2006):

$$\begin{aligned} f_1(X) - \ln Z &= \ln P(Y = 1|X) \\ \dots &\dots \\ f_c(X) - \ln Z &= \ln P(Y = c|X) \end{aligned} \quad (17)$$

Aplicando-se a função exponencial em ambos os lados, temos que

$$P(Y = c|X) = \frac{1}{Z} e^{f_c(X)} \quad (18)$$

onde $P(Y = c|X)$ é a probabilidade de Y ser a classe c para um dado X . Z pode ser encontrado utilizando-se a definição:

$$\sum_{k=1}^C P(Y = k|X) = \frac{1}{Z} \sum_{k=1}^K e^{f_c(X_i)} = 1 \Rightarrow Z = \sum_{k=1}^C e^{f_k(X)} \quad (19)$$

Substituindo Z na equação 18, obtemos a função *softmax* que define a probabilidade de Y ser da classe c dado X .

$$P(Y = c|X) = \text{softmax}(X, c) = \frac{e^{f_c(X)}}{\sum_{k=1}^K e^{f_k(X)}} \quad (20)$$

Os parâmetros, ou pesos, β que otimizam a função *softmax* podem ser encontrados por vários métodos, como gradiente descendente, Newton, LFBG e outros. A figura 4 traz uma representação gráfica da regressão logística *softmax* usando a notação de redes neurais.

Figura 4 – Regressão Logística Multinomial representada como neurônios

Fonte: José Eleandro Custódio, 2020

2.6.2 Redes neurais de convolução

As redes neurais de convolução (CNN)(do inglês, *convolutional neural network*), ou redes de convolução, é um método proposto em LeCun et al. (1989) para identificação de caracteres numéricos e é inspirado no córtex cerebral de animais, em especial, na região responsável pelo processamento dos sinais gerados pela retina (GOODFELLOW; BENGIO; COURVILLE, 2016). Modelos do tipo CNN são redes neurais artificiais especializados que processam dados dispostos em grade e lidam com entradas em 1D, como ondas sonoras, e em 2D, como imagens. A arquitetura da rede CNN é geralmente composta por camadas de convolução, por camadas totalmente conectadas e pela camada de saída.

A camada de convolução é composta de filtros de convolução e da operação de agregação (do inglês, *pooling*). Os filtros são unidades, ou neurônios, que varrem o dado de entrada buscando extrair informações locais, e são compostos por um conjunto de pesos menor que o dado de entrada. Esses pesos são compartilhados em toda a varredura, ou convolução. O número reduzido de pesos e seu compartilhamento são as vantagens da camada de convolução quando comparado com uma rede tradicional, pois requerem menos armazenamento e possuem treinamento mais rápido (GOODFELLOW; BENGIO; COURVILLE, 2016). A operação de agregação, por sua vez, reduz o número de entrada por meio da aplicação de funções como máximo e média.

As camadas totalmente conectadas são similares às camadas da MLP e a camada de saída é geralmente feita pela função *softmax*. O funcionamento de um filtro é ilustrado na figura 5,

Figura 5 – Operação de convolução em 2D

Fonte: Adaptado de Nielsen (2015)

O treinamento da rede de convolução pode utilizar diversos algoritmos de otimização, sendo comum a utilização de gradiente descendente (LECUN et al., 1998) combinadas com técnicas de regularização como *dropout* (SRIVASTAVA et al., 2014), onde a saída de alguns neurônios são ignoradas de forma aleatória.

A utilização das redes de convolução em tarefas relacionadas ao PLN é estudado em Collobert et al. (2011) e em Kim (2014), e a arquitetura típica CNN pode ser visto na figura 6, onde o texto é decomposto em partes como n-gramas de palavras, caracteres e sentenças, e cada parte é representado por *embeddings* ou *one-hot*. Diferentemente das imagens, o dado do tipo texto possui tamanho variável, e para contornar esse problema Collobert et al. (2011) propôs a operação de agregação *max-over-time*.

Figura 6 – Arquitetura de convolução para sentenças e documentos

Fonte: Adaptado de Kim (2014)

2.6.3 Redes neurais recorrentes

Redes neurais recorrentes (RNN) (do inglês, recurrent neural networks) (ELMAN, 1990), são uma família de redes neurais que fazem o processamento de dados sequenciais, onde a dependência temporal é importante para o resultado (GOODFELLOW; BENGIO; COURVILLE, 2016). Esse é o caso do reconhecimento de fala e PLN em geral. Um estudo amplo dessas redes para a modelagem estatística de língua pode ser encontrado em Mikolov (2012).

A forma mais básica de uma rede recorrente incorpora o cálculo feito para a entrada x_{t-1} como parte da entrada de x_t e, conseqüentemente, capturando dependências temporais presentes na sequência 1 até t . Esse neurônio clássico, também conhecido como *vanilla*-

RNN, está representado na figura 7, onde é possível ver também a expansão temporal para uma entrada x_t .

Figura 7 – Expansão de um neurônio de uma rede recorrente

Na prática, as *Vanilla-RNNs* não conseguem capturar dependências longas pois o gradiente da função de erro explode ou desaparece (PASCANU; MIKOLOV; BENGIO, 2013). Esse problema limita a aplicação desse tipo de rede em tarefas relacionadas ao PLN. Visando superar essa deficiência, foi introduzida a rede de memória de curto e longo prazo (do inglês, *Long Short Time Memory*) (LSTM) (HOCHREITER; SCHMIDHUBER, 1997). A LSTM utiliza portões (do inglês, *gates*) que controlam a retenção e o esquecimento das informações de iterações anteriores evitando o problema do gradiente. Mais recentemente, redes do tipo *Gated recurrent unit* (GRU) foram introduzidas em Cho et al. (2014) como simplificação da rede LSTM.

Mais detalhes sobre as redes recorrentes *Vanilla-RNN*, LSTM e GRU podem ser encontrada em Goodfellow, Bengio e Courville (2016) e Nielsen (2015), e mais detalhes sobre a utilização em tarefas relacionadas ao PLN podem ser encontradas em Goldberg (2017).

2.7 Considerações

Nesse capítulo foram analisados os conceitos relacionados à AA, as técnicas de PLN para representação de textos e métodos computacionais empregados para classificação. Vimos que a complexidade do problema de AA fez com que diversas subtarefas fossem propostas, sendo que as grandes vertentes são por vizinho mais próximo e aprendizado de máquina.

Foram analisados os métodos de representação de textos e de aprendizados de máquina. No entanto, não há um consenso na literatura de que existe um método mais efetivo. Sendo necessário um estudo para cada tipo de problema.

3 Revisão bibliográfica

Este capítulo apresenta uma revisão bibliográfica exploratória sobre estudos de AA a partir de textos digitais. Os estudos aqui considerados foram selecionados a partir dos repositórios *ACL Anthology*¹, Scopus², IEEE³ e PAN-CLEF⁴, e foram privilegiados artigos mais recentes, de maior fator de impacto ou pioneiros. Foram excluídos trabalhos de línguas não europeias ou que utilizam técnicas baseadas em grafos, cadeias de Markov e redes complexas.

Para facilitar a compreensão, os trabalhos foram divididos em seções. Na seção 3.1 é visto um histórico da competição PAN. Na seção 3.2 são analisados os trabalhos que utilizaram a representação tradicional de *Bag of N-grams* e os trabalhos que utilizam representação distribuída. Na seção 3.3 são examinados os trabalhos que utilizam abordagem por (dis)similaridade. Na seção 3.4 detalham trabalhos que utilizaram técnicas conexionistas. Por fim, na seção 3.5 são detalhados os trabalhos voltados para a língua portuguesa.

3.1 Competições PAN para atribuição autoral

A competição PAN é um conjunto de eventos científicos e de tarefas compartilhadas, que tem como objetivo o avanço nos estudos das técnicas forenses para textos digitais. Sua sigla deriva das iniciais de sua primeira edição que se chamou *Plagiarism Analysis, Authorship Identification, and Near-Duplicate Detection*.

Na tabela 3 é apresentado um resumo das edições da competição PAN. A coluna *trabalho* traz a referência para os estudos feitos pelos organizadores e que sumarizam as abordagens usadas pelos participantes. A coluna *tarefa* especifica o tipo de problema explorado e contempla detecção de plágio (P), atribuição (A), verificação (V), clusterização (C), ligação autoral (L) e quebra de estilo (Q). Os *idiomas* considerados foram inglês (EN), espanhol (SP), alemão (GE), grego (GR), holandes (DU), italiano (IT) e polonês (PL). Na coluna *Gênero* do texto são considerados romances (NV), contos (CT), e-mails (EM), Ensaaios (ES), artigos de jornais (AR), críticas (RV) e livros-texto de computação (TB). A

¹ <https://aclanthology.coli.uni-saarland.de/>

² <http://www.scopus.com/>

³ <https://ieeexplore.ieee.org/>

⁴ <http://pan.webis.de/>

coluna *Avaliação* especifica as medidas de avaliação, sendo F1 a mais comum, seguida de área sob a curva ROC (AUC) e média das precisões (MAP)⁵. As edições de 2013 e 2014 utilizaram o fator penalizador $C@1$, definido pela equação $C@1 = \text{acurácia} * \text{indecidíveis}$, onde indecidíveis refere-se ao percentual dos casos onde a probabilidade foi exatamente 0,5.

Tabela 3 – Histórico das edições da competição PAN

Trabalho	Tarefa	Idiomas	Gênero	Avaliação
Potthast et al. (2009)	P	EN, SP, GE	NV	F1
Potthast et al. (2010)	P	EN, SP, GE	-	F1
Argamon e Juola (2011)	A, V	EN	EM	F1
Juola (2012)	A, C	EN	NV	F1
Juola e Stamatatos (2013)	V	EN, SP e GR	AR, CT, TB	AUC
Stamatatos et al. (2014)	V	EN, SP, DU e GR	NV, AR, RV, ES	AUC * C@1
Stamatatos et al. (2015)	V	EN, SP, DU e GR	RV, AR, CT	AUC * C@1
Rosso et al. (2016)	C, P	EN, DU e GR	RV, AR	F1, MAP
Tschuggnall et al. (2017)	C, L, Q	EN, SP, DU e GR	NV, AR	F1, MAP
Kestemont et al. (2018)	A, Q	EN, SP, FR, IT e PL	NV	F1
Kestemont et al. (2019)	A, C	EN, SP, FR e IT	NV	F1

Fonte: José Eleandro Custódio, 2020

A série de competições PAN é relevante para a AA pois explora fronteiras de pesquisa e tem atraído um crescente número de pesquisadores e profissionais. Além disso, os conjuntos de dados disponibilizados permitem a replicação de trabalhos de outros pesquisadores e os resultados reportados podem ser utilizados como base para avaliação de novas abordagens. Os corpúscos das edições de 2014 e 2018 foram utilizados para os experimentos apresentados no capítulo 4.

3.2 Modelos de AA usando bag of n-grams, embeddings e variações

O modelo *bag of n-grams* representa de um *baseline* importante para o processamento de língua natural. Esse modelo compreende os modelos de n-gramas de caracteres (*Char-NG*), n-gramas de palavras (*Word-NG*) e n-gramas de POS (*POS-NG*), os quais serão geralmente normalizados pela frequência dos termos, também conhecida como TF. Os trabalhos recentes deste tipo são examinados nessa seção e foram utilizados como inspiração para os experimentos apresentados no capítulo 4.

⁵ Do inglês, mean average precision (MANNING; RAGHAVAN; SCHÜTZE, 2008).

3.2.1 Uso de n-gramas de caracteres (*Char-NG*)

O trabalho em Sapkota et al. (2015) analisou a relevância dos n-gramas de caracteres para atribuição autoral. As sequências de caracteres podem ser extraídas das palavras, sendo delimitados ao início e fim, ou expandir-se entre palavras, abrangendo espaços e pontuação. Este estudo teve como objetivo o entendimento da capacidade preditiva dos n-gramas de caracteres dado suas posições no texto e seus papéis morfológicos. O trabalho restringiu-se a análise de trigramas.

Os trigramas foram organizados em famílias contendo afixos, meio das palavras e pontuação. A família dos afixos foi dividida em *Prefixos*, *Sufixos*, espaços seguidos de prefixos (que será nomeado *EspaçoPrefixo*) e sufixos seguidos de espaço (*SufixoEspaço*). A família das palavras foi organizada em palavras inteiras (*PalavrasInteiras*), onde o tamanho da palavra era igual a três caracteres, meio de palavra (*MeioPalavra*), que não estavam na borda, e multi-palavra (*EntrePalavras*), onde os trigramas podem começar em uma palavra e terminar em outra. A família das pontuações foi dividida em n-gramas nos quais a pontuação pode estar no início (*PontInicio*), no meio do n-grama (*PontMeio*) e no final (*PontFinal*).

Foram considerados os cenários de mesmo domínio e multi-domínio. Os subconjuntos CCAT_10, contendo notícias corporativas, e CCAT_50, contendo notícias industriais, do corpus Reuters RCV1 (LEWIS et al., 2004) foram utilizados para o cenário de mesmo domínio. O corpus *The Guardian*⁶ contendo notícias das categorias mundo, Reino Unido, sociedade e política foram utilizadas para simulação de múltiplos domínios, e divididos em TG1 com textos longos e TG2 com textos curtos. Os conjuntos de dados foram balanceados e ficaram com a seguinte configuração: CCAT_10 contendo 10 autores e 100 textos por autor, CCAT_50 contendo 50 autores e 100 textos por autor, TG1 com 13 autores e 13 textos por autor, e TG2 com 13 autores e 63 textos por autor.

Foram aplicados os classificadores SVM e Naïve Bayes. Os n-gramas *PontMeio*, *PontFinal* e *Palavra inteira* apresentaram o melhor desempenho entre os n-gramas avaliados no cenário de mesmo domínio, atingindo respectivamente 10%, 8% e 9% de acurácia. Os n-gramas *EntrePalavras* e *PontFim* apresentaram o melhor desempenho relativo no cenário multi-domínio com 9,3% e 8,3% de acurácia respectivamente. Apesar da língua inglesa ser

⁶ Esse corpus foi extraído por Stammatatos (2013) usando a API pública disponibilizada em <<https://open-platform.theguardian.com/documentation/>>

morfologicamente pobre, foi possível obter desempenhos distintos por família de n-gramas. O classificador SVM obteve o melhor desempenho quando utilizado com a implementação do pacote LibSVM.

3.2.2 Uso de métodos de distorção textual

O estilo do autor pode variar com a mudança de tópico, domínio e gênero do texto. Os trabalhos analisados a seguir utilizaram estratégias para mascarar conteúdos específicos, distorcendo o texto de forma a evitar o sobreajuste. Um modelo inspirado nestas técnicas foi avaliado no experimento descrito na seção 4.2.

O trabalho em Markov, Stamatatos e Sidorov (2017) analisou os mesmos cenários e conjuntos de dados e extração de características de Sapkota et al. (2015). O diferencial consistiu da inclusão de uma etapa de pré-processamento para tratamento de dígitos, entidades nomeadas e palavras com frequência alta. Os dígitos de “0” a “9” foram substituídos por “0”s, preservando-se os separadores de milhares e decimais. As entidades nomeadas foram substituídas pelo caractere “#”. Por fim, as palavras com alta frequência foram substituídas pelo caractere “\$” porque possuíam correlação com n-gramas de caracteres. A identificação das entidades nomeadas foi feita com a utilização do Reconhecedor de Entidades Nomeadas de Stanford (FINKEL; GRENAGER; MANNING, 2005).

Essas características foram utilizadas como entrada para um classificador SVM. A utilização dos n-gramas de afixos juntos com os de pontuação apresentou melhor desempenho quando comparado com o *baseline* BOW. A remoção das palavras com alta frequência não foi efetiva no desempenho.

O trabalho em Stamatatos (2017) aplica pré-processamento para mascarar, ou distorcer, conteúdos de características textuais que podem estar associadas a um tópico específico, buscando obter uma atribuição autoral multitópico. Os métodos propostos consistiram da seleção das k palavras mais frequentes e da distorção das demais. A variação *DV-MA*, sigla em inglês para visão distorcida com múltiplos asteriscos, dividiu o texto em tokens e os caracteres dos tokens menos frequentes foram transformados em “*” e “#”, sendo “*” aplicado para letras e “#” para dígitos. Outra variação analisada foi *DV-SA*, visão distorcida com único asterisco, onde os tokens menos frequentes foram substituídos

por um único símbolo, “*” ou “#”. Após a distorção, foram construídos modelos de n-gramas de caracteres e palavras.

A análise foi aplicada no cenário de atribuição autoral de conjunto fechado usando os corpúscos CCAT_10 e The Guardian, detalhados anteriormente em Sapkota et al. (2013). Os resultados obtidos para o corpúscos CCAT_10 foram equivalentes ao *baseline* sem distorção, o que reforça a hipótese de que as palavras menos frequentes possuem papel menos relevantes para AA. Já os resultados obtidos para o corpúscos The Guardian foram de 75% de acurácia para o modelo *DV-MA* contra 65% do *baseline* usando n-gramas de caracteres. A técnica foi aplicada no cenário de verificação autoral utilizando os corpúscos PAN2014 e PAN2015 das respectivas edições da competição PAN. Os resultados obtidos foram medidos em AUC e alcançaram patamares similares aos resultados obtidos pelos vencedores das edições da competição.

3.2.3 AA aplicada a mídia sociais

A AA em mídia sociais é desafiadora porque esse canal apresenta um número de autores alto, uma quantidade limitada de caracteres por texto e o estilo de escrita informal. Os trabalhos a seguir usaram dados de mensagens curtas do Twitter. O primeiro, Schwartz et al. (2013), foi destacado por ser pioneiro, e o segundo (ROCHA et al., 2017) por apresentar uma revisão de literatura.

O trabalho em Schwartz et al. (2013) estudou AA de mensagens curtas do Twitter, avaliando o impacto do número de autores e do tamanho do texto no desempenho do sistema. Foram extraído n-gramas de caracteres e de palavras, e utilizado o classificador SVM. O experimento considerou os cenários de 50 a 1.000 autores e 50 a 1.000 mensagens por autor. Os resultados obtidos variaram entre 30,3% de acurácia para 1000 autores e 200 mensagens por autor, e 50,7% de acurácia para 50 autores com 50 mensagens por autor. Adicionalmente, esse trabalho definiu o conceito de k-assinaturas no qual as k% características mais frequentes foram utilizadas como base para futuras análises não supervisionadas.

O trabalho em Rocha et al. (2017) estudou a atribuição autoral sob a ótica da análise forense de mídias sociais, apresentando uma revisão de literatura e um estudo com dados do Twitter. O trabalho explorou as tarefas de atribuição de conjunto fechado e

de conjunto aberto. A captura dos dados iniciou a busca por usuários cujas mensagens contivessem uma ou mais palavras de função em inglês. O trabalho capturou dados 10 mil usuários durante seis meses no ano de 2014. Cada documento contém no máximo 140 caracteres e incluem URLs, *hashtags* e referências a outros usuários. Foram removidos mensagens com marcação RT, o que indica que a mensagem é um compartilhamento da mensagem de outra pessoa, e mensagens com menos de 4 palavras. Os dados foram pré-processados para substituição de URLs, *hashtags* e números pois considerado que essa informação tornavam a AA artificialmente mais fácil.

A tarefa de AA de conjunto fechado utilizou n-gramas de caracteres, de palavras e de etiquetas POS. Foram avaliados cinco métodos de classificação, sendo eles, os classificadores SVM *power mean* e florestas aleatórias, a medida de distância SCAP e o método de compressão de textos *prediction by partial matching* (PPM-5)(TEAHAN; HARPER, 2003). Os tipos de conhecimento e os métodos foram avaliados em cenários onde variaram o número de autores e o a quantidade de mensagens por autor.

A primeira avaliação usou um conjunto fixo com 50 autores, variou os tipos de conhecimento e a quantidade de mensagens, com conjuntos com 1 até 1.000 mensagens por autor. O treinamento foi feito com SVM e os melhores resultados foram obtidos usando a combinação de todos os tipos de conhecimentos como entrada. A acurácia foi de 5%, 35% e 60% para 1, 50 e 60 mensagens por autor, respectivamente.

A segunda avaliação explorou os métodos de classificação. O uso de Florestas aleatórias obteve resultados inferiores ao uso de SVM. O método de compressão PPM-5 obteve melhores acurácias para conjuntos com poucas mensagens por autor. O método SCAP obteve o pior resultado relativo.

A terceira avaliação analisou o impacto do aumento do número de autores no desempenho. Foram analisados os cenários com 50, 500 e 1.000 autores usando classificador SVM e um número fixo de 500 mensagens. As acurácias reportadas foram de 50%, 40% e 32%, respectivamente. Dada a dificuldade do problema, o trabalho investigou a capacidade do sistema em ordenar os possíveis autores, verificando se o autor verdadeiro estaria entre os k com maior probabilidade. Foi construída uma curva de acurácia, de forma a demonstrar a utilidade do sistema em reduzir o espaço de busca. Foi obtida acurácia de 80% para vizinhança 150, superior aos 50% com vizinhança 1.

A tarefa da AA de conjunto aberto considerou os mesmos tipos de conhecimentos da AA fechada e adicionou 71 medidas estilométricas específicas. Os classificadores utilizados

foram SVM com núcleo de base radial⁷ e regressão logística. Foram utilizadas 600 mensagens de autores conhecidos e 600 mensagens de autores de fora do conjunto de treinamento. O número de autores considerado foi de 50 usuários e duas mensagens cada. A acurácia obtida foi de 65% para regressão logística e 75% para SVM, sendo que o SVM teve queda de 5 pontos percentuais com o aumento do número de autores desconhecidos.

3.2.4 Modelos de AA usando representação distribuída

A quantidade de trabalhos que utilizam representação distribuída para AA ainda é muito menor que os trabalhos que utilizam técnicas tradicionais. A seguir, são examinadas algumas iniciativas encontradas até o momento.

O trabalho em Sari e Stevenson (2016) aplicou o método *k-means* para o problema de agrupamento autoral apresentado na edição de 2016 da competição PAN-CLEF. No agrupamento autoral, os documentos devem ser organizados de tal forma que cada grupo contenha documentos de apenas um autor. A extração de características utilizou a base de dados de *embeddings* pré-treinados usando a arquitetura *skip-gram*. Essa base está disponível para as línguas inglesa e alemã⁸. As demais línguas utilizaram modelos de n-gramas de caracteres com peso TF-IDF. O parâmetro *k* do algoritmo K-means foi otimizado utilizando a medida silhueta e F1. Os resultados utilizando-se *embeddings* foram inferiores, ou iguais, ao TF-IDF, e por esse motivo o trabalho utilizou apenas TF-IDF no modelo final.

O trabalho em Posadas-Durán et al. (2017) usou o método Doc2vec no cenário de atribuição autoral com poucos textos. Foram utilizados os corpúscos PAN2012, Reuters RCV1 e The Guardian. Cada documento foi dividido em sequências de N-gramas de palavras e transformado nas formas distribuídas Doc2vec-DM e Doc2vec-DBOW. Ambas as representações foram concatenadas. O estudo analisou cenários com n-gramas variando de 1 a 5. A classificação foi feita com regressão logística e SVM.

Foram reportados 100% de acurácia para o problema PAN2012. Os conjuntos RCV1_CCAT_10 e RCV1_CCAT_50 apresentaram 84,6% e 75,2% de acurácia utilizando Doc2Vec de 1 a 3-gramas. O corpúscos The Guardian apresentou acurácia de 77% versus 59,3% do *baseline* de trigramas de caracteres.

⁷ Do inglês, RBF ou Radial basis function

⁸ Diversos trabalho utilizaram essa base e partir desse ponto ela será referenciada como *Skip-gram-Google*.

Um ponto relevante levantado sobre o modelo Doc2vec, implementado no pacote Gensim⁹, é que esse algoritmo possui diversos parâmetros inicializados com valores aleatórios, ou que são extremamente sensíveis às condições iniciais. O estudo fixou, assim, diversos parâmetros de forma a viabilizar a reprodutibilidade do experimento. Foram estabelecidos valores fixos para a semente utilizada no gerador de números aleatórios, a quantidade de iterações sobre o corpus, a sequência na qual os documentos foram treinados, e outros.

O trabalho em Adorno et al. (2018) apresentou um estudo utilizando Doc2vec para atribuição autoral fechada usando o corpus The Guardian. Foram utilizadas 10 amostras por autor com média de 1500 caracteres e 1000 palavras por documento. O modelo Doc2vec, implementado no pacote Gensim, foi treinado com n-gramas de palavras, de caracteres e de POS. O tamanho dos n-gramas variou de 1 e 5. Os vetores de *embeddings* de tamanho 100 resultantes do treinamento foram utilizados como entrada para regressão logística *softmax*. Parâmetros adicionais foram avaliados em *grid search*. O número de dimensões do vetor de *embeddings* variou entre 50 e 350, o tamanho da janela dos n-gramas variou entre 3 e 19 e a frequência mínima entre 3 e 4.

A aplicação dos *embeddings* na classificação deu-se utilizando inferência e retreinamento. Na inferência, o modelo treinamento foi utilizado para predição. No retreinamento, os documentos sob avaliação foram adicionados ao modelo Doc2vec e executado novas iterações. Os melhores resultados foram reportados por meio do uso da inferência e extração de característica usando unigramas e bigramas de palavras. O uso de n-gramas de POS trouxe resultados comparáveis ao uso de n-gramas de palavras ou n-gramas de caracteres.

Finalmente, o trabalho em Sari, Vlachos e Stevenson (2017) apresentou um estudo utilizando fastText¹⁰ (BOJANOWSKI et al., 2017b) onde foram extraídos *embedding* de caracteres e palavras. Os *embeddings* foram utilizados na classificação dos documentos. Os corpus Reuters RCV1, IMDB62 e um corpus com textos de sentenças jurídicas foram utilizados na análise. Os resultados foram comparados com outros estudos e a proposta atingiu patamares similares aos modelos tradicionais.

⁹ Disponível em <<https://radimrehurek.com/gensim/>>

¹⁰ Disponível em <<https://fasttext.cc/>>

3.3 Modelos de AA baseados similaridade textual

Essa seção detalha trabalhos que usaram modelos similaridade textual, ou vizinho mais próximo. Trabalhos desse tipo utilizam análise estatística multivariada e serão usados como inspiração para o experimento descrito na seção 4.1.

3.3.1 Uso de estatística multivariada

O trabalho em Savoy (2016) estudou a atribuição autoral utilizando medidas de distância. O trabalho usou a regra Delta de Burrows (BURROWS, 2002), a divergência de Kullback-Leibler (KLD), a distância intertextual de Labbé (LABBÉ, 2007) e o *baseline* Naïve Bayes. Este trabalho traz uma análise recente da utilização de medidas de similaridade, fazendo uma cuidadosa análise exploratória, tratamento do cópús, estabelecimento de metodologia de validação para conjuntos e definição de um limite de confiança estatístico para as medidas de distância.

Foram utilizados dois cópús, *Federalistas* (MOSTELLER; WALLACE, 1963), nomeado FED, e 224 discursos proferidos por presidentes norte-americanos diante do congresso, nomeado DPA. Os textos de ambos os cópús foram pré-processados removendo-se caracteres não pertencentes ao UTF-8 e palavras raras, e posteriormente lematizados. O estudo utilizou as 300 palavras mais frequentes no cópús, a partir das quais foram geradas uma série de estatísticas. A probabilidade de um texto ser de um autor A foi definida pela utilização da distribuição beta, usada para dados com valores limitados entre 0,0 e 1,0. Usando essa inferência, foi calibrado o ponto de corte para considerar um texto como pertencente, ou não, a um autor.

O método descrito foi aplicado ao cópús PDA e obteve 88% de acurácia para a distância KLD e 85,7% para o *baseline* Naïve Bayes. A utilização do método no cópús FED obteve 100% de acurácia para Labbé e 96,9% para *baseline*. Esses resultados consistiram da média das execuções da validação cruzada usando *leave-one-out*. Alguns documentos foram marcados com indecidíveis devido à análise da distribuição de probabilidades das médias e variâncias, obtidas da validação cruzada.

3.3.2 Modelos usando Regra Delta de Burrows

O trabalho em Evert et al. (2017) estudou vários aspectos matemáticos da regra Delta de Burrows. Foi examinado a normalização no documento com as normas L1 e L2, a normalização no cópua pelo *z-score* e a escolha da medida de distância. Entre as medidas foram analisadas as distâncias de Manhattan, proposta originalmente por Burrows, distância euclidiana, Minkowski, cossenos, Bray-Curtis, Canberra, Chebyshev e outras.

O trabalho utilizou um modelo de n-gramas de palavras mais frequentes de um cópua contendo livros de domínio público de literatura em inglês. O método foi avaliado usando a tarefa de agrupamento autoral. O estudo concluiu que a normalização norma L1 foi essencial, e que os melhores resultados foram obtidos pela distância de cossenos. Um ponto levantado no trabalho foi que o aumento no número de dimensões acarretou o aumento da quantidade de pontos extremos, e que a limitação do *z-score* entre -2 e 2 manteve o desempenho satisfatório.

3.3.3 O uso método impostor

O método do impostor se destaca entre as abordagens computacionais para AA devido aos resultados reportados em diversos trabalhos. Uma versão alterada por Seidman (2013) obteve o melhor resultado na edição de 2013 da competição PAN (JUOLA; STAMATATOS, 2013), e a versão de Khonji e Iraqi (2014) obteve o melhor resultado na edição de 2014 (STAMATATOS et al., 2014). O método também foi utilizado em Potha e Stamatatos (2017) e Halvani, Winter e Graner (2017). A seguir, é feita a definição do método e é analisado um trabalho mais recente que utilizou o método.

Este método foi proposto em Koppel e Winter (2013) e traz um método de verificação autoral utilizando textos impostores. Os detalhes são descritos no algoritmo 1. Intuitivamente, esse método utiliza um cópua auxiliar I de tal forma que a distância do documento A versus B é comparada com a distância de A versus S e B versus S , onde $S \in I$. São parâmetros do modelo o percentual T de características que serão utilizadas e o número de repetições R .

Algoritmo 1 Atribuição autoral usando o método impostor

```

1: procedimento IMPOSTOR( $A, B, I$  Impostores,  $T$  Taxa,  $R$  Repetições)
2:    $Score = 0$ 
3:   para todo  $R$  faça
4:     Selecionar  $T\%$  das características.
5:     Selecionar  $SI$  impostores do conjunto de impostores  $I$ .
6:     para todo  $S$  no conjunto  $SI$  faça
7:       se  $Sim(A, B) * Sim(B, A) > Sim(A, S) * Sim(B, S)$  então
8:          $Score = Score + 1/R$ 
9:       fim se
10:    fim para
11:  fim para
12:  Retornar Mesmo se  $score > \Delta$  senão diferentes
13: fim procedimento

```

Fonte: Adaptadas de Koppel e Seidman (2018), Koppel e Winter (2013)

Os pontos centrais desse método são reamostragem e aumento do número de textos do *cópus*. A reamostragem intra-*cópus* considerando T partes de A e B evita o sobreajuste e reduz o efeito de pontos anômalos. O aumento da amostra reduz os efeitos da maldição da dimensionalidade, também reduzindo o sobreajuste.

O trabalho em Koppel e Seidman (2018) estudou o problema da pseudoepigrafia, onde há dúvida se todos os textos atribuídos a um autor foram, de fato, escritos pelo mesmo. Foi avaliado um conjunto de dados contendo 3.540 textos escritos em inglês e pertencentes a 156 *bloggers*. Cada texto foi transformado em 4-gramas de caracteres e calculado a matriz de distâncias entre textos. O trabalho separou os métodos em primeira ordem e segunda ordem. Primeira ordem refere-se as distâncias aplicadas isoladamente, e segunda ordem refere-se ao métodos impostor e *unmasking*. As distâncias analisadas foram MinMax¹¹, Manhattan, euclidiana e cossenos. Essas foram aplicadas aos textos produzindo uma matriz distância. Os autores propuseram usar a matriz com desvio da média e desvio da mediana, ao invés da distância original. Os resultados foram avaliados com a medida AUC e o desvio da mediana trouxe o melhor resultado relativamente para todas as distâncias. Os métodos *unmasking* e impostor apresentaram resultado superior as medidas analisadas individualmente. O estudo trouxe, também, a avaliação dessa técnica utilizando os livros de Shakespeare e as epístolas de Paulo por serem textos comumente estudados pela pseudografia.

¹¹ Definida na equação 11

3.3.4 Método *unmasking*

O método *unmasking* foi proposto em Koppel e Schler (2004) e é detalhado no algoritmo 2. A ideia consiste em remover iterativamente k características que foram mais úteis para separar o autor A do autor B em v validações cruzadas. Esse processo gera curva de degradação da acurácia que é utilizada como entrada para um novo classificador. A premissa é a de que, se os textos são dos mesmos autores, então poucas características são necessárias para distingui-los. Alguns trabalhos aplicaram esse método, e entre os de maior destaque pode-se citar Kocher e Savoy (2017).

Algoritmo 2 Atribuição autoral usando o método *unmasking*

```

1: procedimento UNMASKING(A, B, P partições, I iterações, k remoções)
2:   Construa a CurvaDeGradacao(A,B,P,I, k);
3:   Aplique um classificador na curva
4:   Retorne o resultado do classificador
5: fim procedimento
6: procedimento CURVADEGRADAÇÃO(A, B, V validações, I iterações, k remoções)
7:   para  $v \leftarrow 1$  até V faça
8:     para  $i \leftarrow 1$  até I faça
9:       Construa um modelo para A e B
10:      Adicione a acurácia na curva de degradação
11:      Remova  $k$  do conjunto de características
12:     fim para
13:   fim para
14: fim procedimento

```

Fonte: Adaptado de Koppel e Seidman (2018), Koppel e Winter (2013)

3.4 Modelos de AA usando redes neurais artificiais

As redes neurais artificiais têm sido utilizadas com sucesso em diversas tarefas de processamento de língua natural (COLLOBERT et al., 2011). Em especial, as redes neurais de convolução têm sido aplicadas na análise de sentimento (KIM, 2014) e as redes recorrentes são utilizadas tradução (MIKOLOV et al., 2010). A seguir são analisados usando esses tipos de redes.

3.4.1 Modelos de AA usando CNN e *embeddings* pré-treinados

O trabalho em Rhodes (2015) é um dos primeiros estudos encontrados que apresentou redes de convolução para atribuição autoral. O método proposto foi construído usando um *corpus* de teste composto por textos canadenses do século XIV extraídos do projeto Gutenberg, nomeado Canada, e avaliado usando o *corpus* da competição PAN2012.

Os textos foram transformados em sequências de *embeddings* de palavras, concatenados e completados com zeros de tal modo que a representação final tivesse tamanho fixo. Para tal, foram utilizados *embeddings Skip-gram-Google* contendo 300 dimensões. A arquitetura da rede consistiu-se de filtros com tamanho 3, 4 e 5, *pooling* utilizando *max-over-time* (COLLOBERT et al., 2011), função de ativação usando *ReLU* e a camada de saída usando softmax. O treinamento foi feito usando *backpropagation* e otimização *AdaGrad*. A regularização do modelo foi feita usando *dropout*.

O resultado do método para o conjunto Canadá foi de 76,38% de acurácia. Já a aplicação no conjunto PAN2012 exigiu alguns ajustes. A regularização *droupout* foi aumentada de 0,5 para 0,75, de forma a evitar sobreajuste. Outra dificuldade encontrada no estudo foi a tendência da rede em retornar a classe com maior volume de dados, o que foi solucionado aplicado a regra de Bayes para ajuste das probabilidades. O resultado final para o conjunto PAN2012 foi 12 de 14 documentos classificados corretamente, ou seja, 85% de acurácia. Esse foi um resultado próximo ao obtido pelo melhor colocado da competição que foi de 13 documentos corretamente classificados.

3.4.2 Modelo de AA usando CNN aplicada a dados de mídias sociais

O trabalho em Shrestha et al. (2017) aplicou a rede CNN para AA em dados de mídia social utilizando n-gramas de caracteres, fez um comparativo com outras abordagens e utilizou uma forma de visualização para melhorar a compreensão da saída da rede. Foi utilizado o conjunto de dados construído em Schwartz et al. (2013), descrito na seção 3.2.3. Os dados foram pré-processados para remoção de nomes de usuários, *URLs* e dígitos.

A arquitetura proposta consistiu da camada de entrada usando *embeddings* de caracteres, da camada de convolução e da camada de saída com *softmax*. O trabalho informou que foram utilizados *embeddings* com 300 dimensões, mas não detalhou o processo

de construção. A camada de convolução consistiu-se de filtros de tamanhos 3, 4 e 5 n-gramas, e *pooling* com função *max-over-time* com janela de tamanho 500. Foi utilizado o algoritmo de otimização Adam (KINGMA; BA, 2015) e regularização com 0,25 de *dropout*.

O trabalho avaliou os modelos CNN com *embeddings* de unigramas de caracteres (CNN-1), CNN com *embeddings* de bigramas de caracteres (CNN-2), CNN com *embeddings* de palavras *Skip-gram-Google* (CNN-W), o modelo tradicional de 2-3-4 n-gramas de caracteres com regressão logística (CHAR), e por fim, LSTM com bigramas de caracteres.

Foram avaliados três cenários. O primeiro cenário usou um conjunto fixo com 50 autores e 1.000 mensagens por autor, e o melhor resultado relativo foi obtido pela CNN-2 com 76% de acurácia versus 70% do *baseline* CHAR. O segundo cenário fixou a quantidade de mensagens em 200 por autor e variou o número de autores entre 100 e 1.000. O resultado obtido para 100 autores foi de 50% de acurácia para CNN-2 versus 41% para o *baseline*, e para 1000 autores foi de 36,5% para CNN-2 versus 29% para o *baseline*. O terceiro cenário fixou o número de autores em 50 e variou a quantidade de mensagens, neste case a CNN-2 manteve um resultado superior aos demais métodos. De modo geral, as redes CNN-2 e CNN-1 obtiveram os melhores resultados, as redes LSTM-2 e a rede CNN-W, que é similar ao modelo em Rhodes (2015), obtiveram resultados inferiores ao *baseline*. Outras contribuições feitas por esse trabalho foram a detecção de mensagens robóticas e uma proposta de visualização da rede CNN, indicando quais partes do texto foram responsáveis pela maior ativação dos neurônios.

3.4.3 Modelo de AA usando CNN e informação *POS*

O trabalho em Hitschler, Berg e Rehbein (2017) estudou a atribuição autoral onde os documentos apresentam estilo muito similar. Para tal, foi construído um *corpus* contendo artigos científicos escritos por um único autor extraídos do banco de dados da ACL Anthology¹² e limitados ao período de 1967 a 2007. Foram selecionados 808 autores com no mínimo 2 artigos. As 10 primeiras linhas foram descartadas para remover referências ao próprio autor.

Os documentos foram particionados em segmentos com 1.500 palavras, anotado com etiquetas *POS* usando o Anotador de POS de Stanford (FINKEL; GRENNAGER; MANNING, 2005), e filtrando-se as palavras mais frequentes. Essas etiquetas foram

¹² <https://aclanthology.coli.uni-saarland.de/>

representadas por vetores *one-hot*, concatenados com *embeddings Skip-gram-Google* das palavras correspondentes. Cada *embeddings* de sentença foi utilizado como entrada para o método de classificação.

A arquitetura utilizada foi a apresentada em Kim (2014), sendo composta de uma camada de filtro, uma camada de *pooling* com *max-over-time* e a camada de saída com *softmax*. O experimento foi conduzido com 100 filtros de convolução com tamanhos 1, 2 e 3, lote de 20 sentenças por iteração e treinados em 50 épocas. A regularização foi feita com *dropout* de 0.5.

Os melhores resultados obtidos atingiram 13% de acurácia, reportados para o conjunto de testes. Similarmente aos trabalhos em Varela et al. (2016) e Rocha et al. (2017), foi utilizada a k-atribuição, sendo considerados os primeiros 1, 5, 10, 20 e 50 vizinhos. Foi reportado 13% e 50% de acurácia para vizinhança igual à 1 e 50, respectivamente. A técnica apresentada também foi avaliada com o uso do conjunto de dados PAN2012 e obtendo acurácia de 72%, ou seja, 11 de 14 documentos classificados corretamente.

3.4.4 Modelo de AA usando redes recorrentes

Redes recorrentes têm papel central na modelagem computacional de língua (MIKOLOV et al., 2010). Os principais estudos aplicados à tarefa de AA são analisados a seguir.

Os trabalhos em Bagnall (2015) e Bagnall (2016) aplicaram redes recorrente para análise autoral e obtiveram os melhores desempenhos nas edições de 2015 e 2016 da competição PAN. O método apresentado em Bagnall (2015) e, em linhas gerais mantido em Bagnall (2016), consiste de um modelo de língua para prever o próximo caractere do texto. A premissa para essa abordagem é de que o modelo gerado com os textos de um autor produz mais erros quando aplicado aos textos de outros autores.

O sistema consistiu da seguinte arquitetura. Cada documento foi transformado em uma sequência de caracteres representados como *one-hot* e utilizados como camada de entrada. A segunda camada consistiu dos neurônios ocultos H e dos estados de H_{t-1} com função de ativação ReSQRT, sendo t o número de recorrências. A camada de saída, $S_{|A||V|}$, consistiu de um modelo de língua de tamanho $|V|$ para os $|A|$ autores. Como cada autor possui uma saída com múltiplas probabilidades, a arquitetura foi nomeada

como rede recorrente de múltiplas cabeças. O algoritmo de otimização utilizado foi o *backpropagation-through-time* (BPTT). A figura 8 ilustra a arquitetura utilizada.

O processo de atribuição foi feito calculando-se as probabilidades para as sequências de caracteres para cada texto, e escolhendo os autores que gerasse a menor cross-entropia. Os conjuntos de dados de edições anteriores da competição PAN foram utilizados para pré-treinamento. Vários parâmetros foram avaliados, bem como formas de rearranjo dos dados. O sistema final utilizado na competição PAN2015 consistiu de um comitê contendo várias combinações dos parâmetros. Para a edição de 2016, foi adicionada uma etapa de pré-processamento que consistiu da remoção de diacríticos e de palavras raras.

Figura 8 – Arquitetura da rede neural multicabeças

3.4.5 Outras abordagens

A literatura em AA traz ainda várias abordagens recentes que não puderam ser aprofundadas na presente revisão. Dentre estas, destacamos.

O trabalho em Qian, He e Zhang (2016) descreveu um experimento utilizando *Gated Recurrent Unit* (GRU), LSTM para atribuição autoral, e redes siamesas para verificação. As redes foram aplicadas nos conjuntos de dados Reuters RCV1 e extrações do projeto Gutenberg. As redes foram treinadas utilizando os níveis sentença e artigo.

O trabalho em Mohsen, El-Makky e Ghanem (2016) aplicou redes *autoencoders* para fazer a redução da dimensionalidade, e o trabalho em Brocardo et al. (2017) aplicou redes *deep belief* com uma camada de Gauss-Bernulli Restricted Boltzman Machine (GBRBM)

seguidas de n camadas de RBM empilhadas no cenário de verificação autoral com textos curtos. Em ambos os casos, foi utilizado regressão logística como classificador.

3.5 Modelos de AA para o português

O número de trabalhos de atribuição autoral computacional para a língua portuguesa ainda é limitado. A seguir, são examinados dois trabalhos recentes encontrados deste tipo.

O trabalho em Varela et al. (2016) aplicou a AA em obras literárias de domínio público¹³, de onde foram extraídas 20 amostras com tamanhos entre 500 e 2000 palavras. Os textos foram processados e anotados com etiquetas *POS* e dependência semântica. Foram avaliados os cenários de verificação e atribuição autoral.

O modelo de verificação autoral separou o conjunto de dados em instâncias contendo 50, 75 e 100 autores. Cada instância continha dois textos do autor conhecido e uma do autor sob análise. Os pares foram avaliados pela distância euclidiana, e as distâncias foram combinadas por média, mediana e voto majoritário. Esse último obteve o melhor desempenho relativo, atingindo as acurácias de 89% e 95%, no melhor e pior cenário.

O modelo de atribuição utilizou o processo criado para verificação e AA por meio de técnicas de similaridade. O cópús foi dividido em conjuntos com 8, 10, 12 e 15 autores. Foram calculadas as distâncias e listados os dez vizinhos mais próximos. A acurácia foi calculada para cada nível de vizinhança usando a k -atribuição, definida na seção 2.2. Foi reportado 70% para vizinhança 1 e 100% para vizinhança 10.

O trabalho em Markov, Baptista e Lagunas (2017) baseou-se em Sapkota et al. (2015) e analisou os impactos dos diferentes tipos de n -gramas de caracteres na atribuição autoral multi-tópicos com textos em Português. Foram analisados 5.167 artigos do jornal português Público¹⁴ extraídos de vários tópicos. Foi construído um cópús com 50 autores e 50 textos por autor, seguindo a metodologia da competição PAN para construção de conjuntos de dados.

Um modelo de n -gramas de palavras com classificador SVM foi utilizado como *baseline*. A frequência mínima foi otimizada e o melhor resultado para o *baseline* foi 62% de acurácia medida no conjunto de teste. Os modelos de n -gramas de caracteres propostos utilizaram pesos TF e binários, e SVM como classificador. Cada tipo de caractere foi

¹³ Não foram especificados os autores, gênero, domínio, nem o período da literatura

¹⁴ <https://www.publico.pt/>

avaliado em separado e em conjunto com os demais. Foram reportadas acurácias entre 50% e 75%. Os melhores resultados foram obtidos com o uso de prefixos combinados com meio de palavras (*MeioPalavras*).

3.6 Considerações

Esse capítulo trouxe uma revisão bibliográfica acerca dos estudos de AA de destaque na literatura em 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, o idioma, a tarefa de AA, o tipo de conhecimento e o método de aprendizado de máquina utilizado no estudo.

Os conjuntos de dados são, em sua maioria, da competição PAN-CLEF entre as edições de 2012 e 2016, do corpus Reuters RCV1 e do jornal *The Guardian*. Alguns trabalhos utilizaram extrações próprias feitas de redes sociais, literatura, projeto Gutenberg e outros. Os idiomas considerados nos estudos são inglês (EN), português brasileiro (PT-Br), português europeu (PT-Pt), holandês (DU), espanhol (ES) e grego (GR). Os estudos são, em sua maioria, dedicados ao idioma inglês.

As tarefas consideradas pelos estudos foram atribuição (A), verificação (V) e agrupamento (C). Observa-se que três estudos aplicaram seus métodos na atribuição e verificação no mesmo trabalho.

Os tipos de conhecimento considerados foram, em sua maioria, n-gramas de caracteres (C), palavras (W) e POS (P). De modo geral, esses conhecimentos foram pré-processados e foram retidos os símbolos mais frequentes no corpus, seguindo a prática amplamente estabelecida de utilização dos caracteres mais comuns (EVERT et al., 2017; KEŠELJ et al., 2003) e das palavras de função (POTHA; STAMATATOS, 2017; MOSTELLER; WALLACE, 1963).

Os modelos que utilizaram representação distribuídas foram encontrados em número menor. Em sua grande maioria, estes estudos utilizaram dados Google pré-treinados usando o algoritmo Word2Vec(W2V) com arquitetura *skip-gram*.

Os métodos de aprendizado de máquina considerados incluíram SVM, florestas aleatórias (RF), regressão logística *softmax*, redes de convolução, redes recorrentes e o Naïve Bayes. Os métodos de vizinhos mais próximos analisados utilizaram o método do impostor e a regra delta de Burrows.

Considerando-se os trabalhos analisados, podem ser feitas observações. Os modelos de n-gramas de caracteres configuram um *baseline* importante para AA e diversos trabalhos se dedicaram a estimar sua eficácia (SAPKOTA et al., 2015; MARKOV; STAMATATOS; SIDOROV, 2017; SHRESTHA et al., 2017). As técnicas de reamostragem e conjuntos de dados externos melhoram o desempenho de sistemas baseados em distância. Os modelos de distorção textual ajudaram extrair conhecimentos específicos dentro do texto (STAMATATOS, 2017; MARKOV; STAMATATOS; SIDOROV, 2017). As representações distribuídas pré-treinadas não apresentaram resultados significativos, ou seja, o significado semântico parece ser menos relevante que a estrutura morfológica e sintática. As redes recorrentes contribuíram para o avanço do estado da arte, indicando que a informação de sequência é possivelmente relevante para AA. Finalmente, as redes de convolução apresentaram resultados no geral similares aos métodos tradicionais, porém com sobrecarga extra de processamento.

Tabela 4 – Resumo dos trabalhos correlatos

Estudo	Dados	Idioma	Tarefa	Conhecimento	Método
Sapkota et al. (2015)	RCV1, TGuardian	EN	A	C	SVM
Markov, Stamatatos e Sidorov (2017)	RCV1	EN	A	C	SVM
Stamatatos (2017)	RCV1, TGuardian, PAN2013	EN	A,V	C, W	SVM
Schwartz et al. (2013)	Twitter	EN	A	C, W	SVM
Rocha et al. (2017)	Twitter	EN	A,V	C, W, P	SVM, RF e SCAP
Savoy (2016)	Federalistas	EN	A	W	Δ de Burrow, KLD
Evert et al. (2017)	Literatura	EN	C	W	Clusterização
Koppel e Seidman (2018)	Blogs, Shakespeare	EN	A	C	Impostor, unmasking
Varela et al. (2016)	Literatura	PT-BR	A,V	P	SVM
Markov, Baptista e Lagunas (2017)	Artigos de jornais	PT-PT	A	C	SVM e Naïve Bayes
Sari e Stevenson (2016)	PAN2016	EN, DU, GR	C	W2V, W	K-means
Posadas-Durán et al. (2017)	PAN2012, TGuardian	EN	A	D2V de W	Softmax e SVM
Adorno et al. (2018)	TGuardian	EN	A	D2V de C, W, P	Softmax
Sari, Vlachos e Stevenson (2017)	RCV1	EN	A	C, W	FastText
Rhodes (2015)	PAN2012	EN	A	W2V	CNN-Softmax
Shrestha et al. (2017)	Twitter	EN	A	C One-hot	Softmax
Hitschler, Berg e Rehbein (2017)	PAN2012, ACL	EN	A	W2V, P one-hot	CNN-Softmax
Bagnall (2015)	PAN2015	EN, DU, ES, GR	V	C One-hot	RNN-Softmax
Bagnall (2016)	PAN2015	EN, DU, ES, GR	C	C One-hot	RNN-Softmax

Fonte: José Eleandro Custódio, 2020

4 Estudos exploratórios

Esse capítulo traz dois estudos exploratórios com o tema de análise autoral. O primeiro trabalho explorou as técnicas de distância e similaridade aplicadas ao problema de verificação autoral discutidos na seção 2.4. No segundo trabalho é apresentado o estudo de atribuição autoral de conjunto fechado utilizando aprendizado de máquina supervisionado.

4.1 *Experimento 1: Verificação autoral*

As técnicas de similaridade possuem um papel importante para AA por serem aplicáveis em várias frentes, conforme visto na seção 2.4.1. Por esse motivo, foi conduzido um estudo exploratório sobre medidas de similaridade aplicadas à verificação textual, seguido do diagnóstico da capacidade de discriminação, e da proposta de um método para combinar suas saídas. O objetivo desse estudo é aumentar a compreensão da AA por similaridade. O resultado desse trabalho foi apresentado em Custódio e Paraboni (2018b).

4.1.1 Conjunto de dados

A edição de 2014 da competição PAN (STAMATATOS et al., 2014) abordou o tema de verificação autoral. Três corpúscos disponibilizados desta edição foram usados neste trabalho: artigos de jornal em espanhol (PAN14-SP), ensaios em inglês (PAN14-EE) e romances em inglês (PAN14-EN). Cada corpúscos é constituído por um conjunto de problemas a serem resolvidos. Cada problema possui um ou mais textos de um mesmo autor e um texto desconhecido, o qual deseja-se verificar se foi ou não escrito pelo mesmo autor.

O conjunto PAN14-SP é composto por 100 problemas contendo cinco textos para o autor conhecido e média de 35,2 mil caracteres. Seu conteúdo foi extraído do jornal El País. O conjunto PAN14-EE é composto por 200 problemas com um número de documentos por autores variando entre um e cinco, e quantidade de caracteres variando entre 4.300 a 25.000. O conteúdo é composto por ensaios produzidos por estudantes da língua inglesa. Finalmente, o conjunto PAN14-EM é composto por 100 problemas com um texto por

autor e 25,4 mil caracteres por autor. Os textos foram extraídos de romances com direitos autorais livres e disponibilizados nos sites do projeto *Guttenberg*¹ e *FanFiction*².

4.1.2 Análise das medidas de similaridade

Os problemas presentes no conjunto de dados foram transformados em pares de documentos: um documento contendo os textos concatenados do autor conhecido, e outro contendo o texto cuja autoria deseja-se avaliar. A extração de características utilizou n-gramas de caracteres com tamanho entre três e cinco, dos quais foram selecionados os 10% mais frequentes no córpus. Os vetores resultantes foram normalizados pela norma $L1$ e a partir destes calculou-se a similaridade textual utilizando diferentes funções de distância.

Este experimento utilizou quatro medidas de similaridade entre as apresentadas na seção 2.4.1. Foram analisadas Kešelj, Stamatatos, Jaccard e cossenos. As distâncias de Kešelj e Stamatatos, definidas nas equações 13 e 14 respectivamente, foram escolhidas por terem origem na AA computacional. Jaccard, definida na equação 10, foi escolhida por representar a família de medidas binárias. A distância de cossenos, definida na equação 3, foi escolhida por ser amplamente utilizada na literatura de PLN. Por fim, foi avaliado uso da distância de cossenos combinada com o pré-processamento utilizando a análise de componentes principais (PCA) mantendo 95% da variância explicada.

A efetividade de cada medida foi avaliada por meio da análise das distribuições de probabilidades, da métrica KS, da correlação, da métrica AUC e da acurácia. A figura 9 traz o diagnóstico de cada medida aplicada no córpus PAN14-SP. Esse diagnóstico é composto pela análise da distribuição de probabilidade para os casos onde se confirmaram a autoria, no histograma com linhas horizontais, versus os casos onde não se confirmaram a autoria, no histograma com fundo liso. As curvas de probabilidades acumuladas estão em tracejado para mesmo autor e contínua para autores diferentes. A métrica KS encontra-se junto ao nome das medidas.

Algumas observações podem ser feitas sobre as medidas de similaridade usando inspeção visual. As métricas Kešelj e Stamatatos possuem separação similar, sendo que Stamatatos possui concentração maior nos valores mais baixos. Jaccard possui menor

¹ Disponível em <<https://www.gutenberg.org/>>

² Disponível em <<https://www.fanfiction.net/>>

separação entre os modelos, o que é esperado dado que é um modelo binário. Cossenos é o método relativamente mais robusto e a utilização do conjunta com PCA (*Cos_PCA*) obteve 10 pontos percentuais acima do obtido usando medida KS.

A tabela 5 demonstra também que algumas medidas apresentam correlação elevada, e podem em princípio ser desconsideradas na tarefa de verificação autoral.

Figura 9 – Análise da capacidade de separação da medidas de similaridade

Fonte: Custódio e Paraboni (2018b)

Foi criado um modelo para cada medida, de modo que fosse possível calcular as métricas AUC e acurácia. Esses modelos serviram também para comparação com os métodos propostos. A construção seguiu passos idênticos e consistiu das seguintes etapas: as medidas foram convertidas para o mesmo intervalo com o uso de *minmax*, definida por $Mimax(A) = \frac{A - \min(A)}{\max(A) - \min(A)}$, e aplicando-se a regressão logística. Os vetores resultantes tem valores no intervalo $(0, 0, 1, 0)$ e interpretação probabilística. Essa abordagem difere de

Tabela 5 – Matriz de correlação das medidas de similaridade textual.

	Cosseno	Cos_PCA	Jaccard	Kešelj	Stamatatos
Cosseno	1,00				
Cos_PCA	0,56	1,00			
Jaccard	0,53	-0,07	1,00		
Kešelj	0,62	0,06	0,98	1,00	
Stamatatos	0,50	-0,01	0,60	0,61	1,00

Fonte: Adaptado de Custódio e Paraboni (2018b)

estudos como Seidman (2013) onde é comum a construção de um ponto de corte manual. Os resultados são apresentados na tabela 6.

4.1.3 Métodos propostos

Com base na análise discutida na seção anterior, propomos dois métodos para verificação autoral independente de idioma. No primeiro método, denominado MP1, foi realizada a combinação das métricas de distâncias consideradas. A hipótese neste caso é a de que cada tipo de distância avalia um aspecto diferente do estilo de escrita, e a combinação destas pode gerar um método mais robusto. Esse método consistiu da seleção das medidas menos correlacionadas entre si, da concatenação dos vetores coluna, da normalização por *minmax* e classificação pela regressão logística *softmax*. A distância de Kešelj foi descartada do método MP1 porque apresentou correlação maior que 90% com a medida Jaccard.

O segundo método, proposto denominado MP2, utiliza um modelo BOW unificado, onde a matriz dos textos dos autores conhecidos C e a matriz dos textos dos autores desconhecidos D foram consolidadas em uma única representação. Essa unificação se deu-se pela utilização da equação 21, onde cada n-grama j da instância i da matriz C foi comparada com o elemento correspondente em D . A equação baseia-se na distância de χ^2 suavizada pela função *log*. Parâmetros adicionais foram empregados de modo a garantir que a distância existisse mesmo quando a frequência esperada fosse zero. A matriz dada em MP2 passou pela normalização por *minmax*, transformação por PCA para remover multicolinearidade, e regressão logística *softmax* como classificador.

$$MP2(C_{ij}, D_{ij}) = \log \left(1 + \frac{(C_{ij} - D_{ij})^2}{C_{ij} + 1} \right) \quad (21)$$

4.1.4 Resultados

Cada medida de similaridade foi avaliada separadamente e seus resultados comparados com os métodos propostos MP1 e MP2. Os resultados dos dois métodos aplicados aos corpúscos em inglês e espanhol são sumarizados na Tabela 6. Os melhores resultados para cada conjunto de dados aparecem em destaque.

Tabela 6 – Verificação autoral - AUC e acurácia médias em 5-partições.

Modelo	PAN2014 (EE e EM)		PAN2014-SP	
	AUC	Acurácia	AUC	Acurácia
Jaccard	0,60	0,56	0,57	0,52
Cossenos	0,63	0,50	0,88	0,77
Cossenos_PCA	0,63	0,55	0,92	0,83
Kešelj	0,61	0,54	0,71	0,60
Stamatatos	0,60	0,55	0,59	0,54
MP1 – Mix	0,75	0,67	0,72	0,62
MP2 – BOW	0,62	0,53	0,93	0,85

Fonte: Adaptado de Custódio e Paraboni (2018b)

Pode-se notar que os resultados obtidos para textos em língua inglesa utilizando mesma representação e medidas foram inferiores aos obtidos para a língua espanhola. O conjunto de dados PAN2014-EE e EM tiveram melhor resultado com a utilização do método MP2. O conjunto de dados PAN2014-SP teve melhor resultado com o método MP1.

4.1.5 Considerações

Os experimentos realizados sugerem que, apesar da estrutura diferente, as medidas consideradas apresentam alta correlação, e possuem assim poder de discriminação equivalente. Além disso, a combinação das medidas menos correlacionadas permitiu a criação de um método relativamente robusto (MP1), e a equação MP2 mostrou que seria em princípio possível usar um modelo simples, do tipo BOW, para verificação autoral com desempenho satisfatório.

4.2 *Experimento 2: Atribuição autoral de conjunto fechado e multidomínio no desafio PAN-CLEF 2018*

Como visto anteriormente, a abordagem por atribuição é um ponto chave para a análise autoral. Dado isso, foi conduzido um experimento utilizando o corpus PAN2018³ com o objetivo de explorar o problema. O modelo resultante desse experimento foi submetido para participação na competição PAN-CLEF-2018 (KESTEMONT et al., 2018) de AA em múltiplos domínios.

4.2.1 Conjunto de dados

A atribuição autoral em múltiplos domínios é um desafio computacional considerável porque os autores mudam a forma de escrita dependendo do estilo do texto. A edição de 2018 da competição PAN utilizou textos de literatura de fantasia para simular essa mudança de domínio. O corpus PAN2018 foi construído com textos de *fanfiction*⁴ onde os fãs de livros famosos da literatura utilizam os mundos dessas histórias como base para suas obras. Alguns exemplos desses mundos, ou domínios, são *Star Wars*, *Harry Potter*, *Sherlock Holmes* e *Senhor dos Anéis*. Assim, a tarefa foi denominada *atribuição autoral cross-fandom*.

Podemos ver na tabela 7 os dez problemas, ou tarefas considerados na competição, e suas características. Observa-se que foram disponibilizados dados em cinco línguas, gerando a necessidade de um sistema independente de língua. Para cada língua, foram criadas instâncias com cinco e vinte autores, com média de sete documentos por autor e média de 4.500 caracteres por documento.

4.2.2 O sistema *baseline* e o sistema de avaliação PAN-CLEF 2018

Além do corpus, foram disponibilizados pela organização da competição um sistema de classificação *baseline* independente de língua e um sistema de avaliação que utiliza a saída geradas no formato especificado pela organização para gerar estatísticas de desempenho do modelo computacional.

³ Disponível em <<https://pan.webis.de/clef18/pan18-web/author-identification.html>>

⁴ Disponível em <<https://www.fanfiction.net/>>

Tabela 7 – Características do problemas do PAN 2018

Problema	Língua	Autores	Treinamento		Teste	
			Docs.	Carac./doc.	Docs.	Carac./doc.
01	EN	20	140	4.327	105	4.370
02	EN	5	35	4.342	21	4.296
03	FR	20	140	4.492	49	4.508
04	FR	5	35	4.522	21	4.532
05	IT	20	140	4.720	80	4.787
06	IT	5	35	4.847	46	4.765
07	PL	20	140	5.145	103	5.200
08	PL	5	35	5.049	15	5.214
09	ES	20	140	4.794	117	4.788
10	ES	5	35	4.955	64	4.827

Fonte: Adaptado de Custódio e Paraboni (2018a)

O sistema de atribuição *baseline*, aqui denominado *Bas.PAN*, utiliza n-gramas de caracteres como extração de características. Os n-gramas possuem tamanho fixo e valor inicial sugerido. A matriz de contagem é filtrada pela frequência mínima no córpus com valor sugerido de cinco ocorrências e normalizada utilizando a frequência relativa (TF). A classificação é feita pelo classificador SVM com núcleo linear. Dado que esse algoritmo não trabalha com multi-classes, foram disponibilizadas as estratégias um-contra-todos e um-contra-um, implementados com a biblioteca *scikit-learn* (PEDREGOSA et al., 2011).

O sistema de avaliação, por sua vez, utiliza o resultado gerado pelo sistema de classificação e gera as estatísticas medida-F1, acurácia, precisão e revocação. Essas medidas foram sumarizadas por média de cada classe e problema. O sistema também gera a média final para todos os problemas.

4.2.3 Método proposto

A hipótese central para o método proposto foi de que a tarefa de AA se baseia na combinação de diferentes fontes de conhecimento, como preferência lexical, inflexões morfológicas, uso de maiúsculas, espaços e pontuação. Essas preferências podem ser obtidas por meio de modelos baseados em caracteres ou palavras.

Baseado nessas observações, a abordagem proposta para a AA multidomínio consiste de uma série melhorias em relação ao sistema *Bas.PAN*. Em linhas gerais, o sistema com n-gramas fixos foi substituído por n-gramas de tamanho variável, foi adicionada uma

camada de pré-processamento, e o classificador SVM linear foi substituído pela regressão logística multinomial. Esse sistema foi aplicado em três tipos de entradas e as previsões desses três sistemas independentes, denominados *Std.charN*, *Dist.charN* e *Std.wordN*, foram combinadas gerando a classificação final na forma de um comitê de máquina. A figura 10 ilustra os três classificadores e o *comitê*.

Figura 10 – Arquitetura do comitê de máquina para AA multidomínio

Fonte: Adaptado de Custódio e Paraboni (2018a)

O modelo *Std.wordN* consistiu de n-gramas de palavras com n de tamanho variável. A justificativa para utilizar mais de um tamanho de n-gramas vem da necessidade de capturar expressões como "*above all*" e "*and so on*", que são bigramas e trigramas respectivamente.

O modelo *Std.charN* utiliza modelo de caracteres e têm como objetivo capturar preferências sintáticas e morfológicas em um cenário independente de língua. O uso de n-gramas com comprimento variável tem objetivo similar ao modelo *Std.wordN*.

O modelo *Dist.charN* utiliza distorção textual e foi inspirado em Stamatatos (2017). O objetivo da distorção é mascarar conteúdos que estariam se comportando como ruído para o modelo. Em Stamatatos (2017) utiliza-se distorções para mascarar palavras menos frequentes, que estariam relacionadas ao tópico. O modelo *Dist.charN* atua sobre caracteres, filtrando a utilização de espaços, pontuação e caracteres com diacríticos, de forma a mascarar caracteres comuns, ou seja, letras minúsculas e maiúsculas. A tabela 8 ilustra a aplicação desse método.

Tabela 8 – Exemplo de distorção de texto aplicada ao primeiro documento do problema 09 da base de treinamento.

Texto original	Texto transformado
-¿Y cómo sabes que no lo ama?	-¿* *ó** ***** *** ** ** ***?
-Inglaterra se preguntó a su vez si habría un muñeco del esposo también.	-***** ** *****ó * ** *** ** ****í* ** **ñ*** *** ***** *****é*.

Fonte: Adaptado de Custódio e Paraboni (2018a)

A premissa para utilização deste tipo de distorção é a de que a pontuação e espaços representam a estrutura do texto e não carregam conteúdo semântico, sendo assim independentes de domínios. Os caracteres com diacríticos, por outro lado, representam estruturas morfológicas complexas, como no caso do tempo verbal futuro no português, onde os verbos terminam com o sufixo *ão*, e no caso de preferência por estrangeirismos como *naïve* e *résumé*. Outra motivação para este tipo de distorção é capturar palavras escritas incorretamente, como *e/é*, no caso verbo estar do português. Espera-se que a aplicação da distorção atue como filtro, e capture as características mencionadas sem a distração dos morfemas de conteúdo.

Cada modelo produziu um vetor com a probabilidade de cada candidato ser o autor da instância analisada. Os três vetores foram concatenados e utilizados como entrada para um quarto modelo, que é o comitê propriamente dito. Por utilizar entradas não binárias, esse comitê é do tipo *fuzzy*.

4.2.4 Experimento

Os modelos propostos e o sistema fornecido pelos organizadores da competição PAN foram calibrados utilizando-se o corpus de desenvolvimento no ambiente Python e com a biblioteca *Scikit Learn* (PEDREGOSA et al., 2011).

O modelo *Bas.PAN* foi explorado por *grid search* e validação cruzada de 5 partições. Os parâmetros com melhor desempenho foram 4-gramas, 10 de frequência mínima e estratégia um-contratodos. O classificador SVM foi utilizado com parâmetros padrões fornecidos pela biblioteca.

Os modelos *Std.charN*, *Dist.charN* e *Std.wordN* foram construídos seguindo-se um procedimento uniforme, variando-se apenas a extração de características. Este procedimento consistiu da vetorização, aplicação do TF-IDF, normalização no documento e no corpus,

redução de dimensionalidade com PCA e, por fim, aplicação do classificador. O PCA foi utilizado para remover a multicolinearidade gerada pelos n-gramas de tamanho variável. Cada algoritmo consiste de vários parâmetros com vários valores de teste possíveis. A tabela 9 detalha o plano de otimização do modelo.

Tabela 9 – Parâmetros analisados pelo processo de classificação do PAN2018

Módulo	Parâmetros	Valores analisados
Extração de características	Faixa n-gram	Início=(1 to 3) - Fim=(1 to 5)
	Freq. min. doc.	[0,01, 0,05, 0,1, 0,5]
	Freq. max. doc.	[0,25, 0,50, 0,90, 1,0]
	TF	Normal, sublinear
	IDF	Normal, suavizado
Transformações	Normalização no documento	L1, L2
	Normalização no corpus	MaxAbsScaler
	PCA % da variância explicada	[0,10, 0,25, 0,50, 0,75, 0,90, 0,99]
Classificador	Regressão logística	Multinomial-Softmax

Fonte: Adaptado de Custódio e Paraboni (2018a)

Os valores ótimos para os três modelos foram determinados pelo uso de *grid search* com validação cruzada com 5 partições. Os valores encontrados estão ilustrados na tabela 10, no qual o início/fim denotam o comprimento da subsequência que foi concatenada. Por exemplo, início= 2 e fim= 5 representa a concatenação das subsequências [(2, 2), (2, 3), ..., (4, 3), (4, 5)].

Tabela 10 – Valores ótimos encontrados para PAN2018

Módulo	Parâmetros	Valores ótimos
Extração de características	Faixa n-gram	Std.charN - Início=2 Fim=5
		Dist.charN - Início=2 Fim=5
		Word.charN - Início=1 Fim=3
	Freq. min. doc.	0,05
	Freq. max. doc.	1,0
	TF	Sublinear
Transformação	IDF	Suavizado
	Normalização no documento	L2
	PCA	0,99

Fonte: Adaptado de Custódio e Paraboni (2018a)

4.2.5 Resultados

Os modelos apresentados podem ser avaliados pela inspeção das características mais relevantes. As tabelas 31, 32 e 33 da seção A.1 em apêndice contém as 15 características mais importantes para cada modelo, respectivamente, e foram geradas considerando-se o peso na regressão logística.

Por ser uma abordagem independente de língua, as palavras de função não foram utilizadas explicitamente como entrada para os modelos. No entanto, nota-se que as palavras de função foram produzidas explicitamente pelo modelo *Std.wordN*. Além disso, nota-se que todos os modelos trazem explicitamente preferências no uso de palavras, pontuação e espaços, e que *Std.distN* evidencia o papel dos aspectos estruturais.

Todos os modelos foram treinados a partir do conjunto de desenvolvimento. Os modelos foram calibrados individualmente usando os parâmetros descritos na tabela 9, e os resultados da medida F1 macro calculada pelo sistema de avaliação para o sistema *baseline* fornecido pelo PAN-CLEF 2018, os três modelos propostos e o *comitê* estão apresentados na tabela 11.

Tabela 11 – PAN-CLEF 2018 - F1 para AA no corpus de desenvolvimento

Problema	Língua	Autores	Bas.PAN	Std.charN	Dist.charN	Std.wordN	Comitê
01	EN	20	0,514	0,609	0,479	0,444	0,625
02	EN	5	0,626	0,535	0,333	0,577	0,673
03	FR	20	0,631	0,681	0,568	0,418	0,776
04	FR	5	0,747	0,719	0,586	0,572	0,820
05	IT	20	0,529	0,597	0,491	0,497	0,578
06	IT	5	0,614	0,623	0,595	0,520	0,663
07	PL	20	0,455	0,470	0,496	0,475	0,554
08	PL	5	0,703	0,948	0,570	0,922	0,922
09	ES	20	0,709	0,774	0,589	0,616	0,701
10	ES	5	0,593	0,778	0,802	0,588	0,830
Média			0,612	0,673	0,551	0,563	0,714

Fonte: Adaptado de Custódio e Paraboni (2018a)

A partir desses resultados, pode-se fazer algumas observações. Primeiro, nota-se que o modelo de caracteres *Std.charN* obteve o melhor resultado entre os modelos quando analisado individualmente. Nota-se também que *Dist.charN* tem desempenho inferior a *Std.charN*. Esse resultado é esperado dado que *Dist.charN* contém menos informações, podendo ser considerado um subconjunto de *Std.charN*. De modo mais expressivo, observa-

se também que o modelo *comité* apresenta um resultado consistentemente melhor que os demais modelos. Esse resultado deve-se às melhorias propostas como utilização de n-gramas de tamanho variável, combinação de mais de um modelo. Do ponto de vista linguístico, observa-se que o estilo do autor pode ser capturado de forma mais eficiente por meio de modelos que considerem a estrutura do texto, a utilização de afixos e a preferência por algumas palavras.

4.2.6 Resultados publicados pela PAN-CLEF 2018

Nessa subseção é apresentado o resultado reportado pelos organizadores da PAN-CLEF 2018 para AA multidomínio (KESTEMONT et al., 2018). Na qual, o método proposto apresentou um desempenho superior aos demais competidores, e obteve o primeiro lugar geral do evento. Na tabela 12 é possível ver o *ranking* geral⁵ ordenado pela medida F1 macro. Trabalhos com resultados inferiores ao *baseline* foram omitidos.

Tabela 12 – PAN2018 - Visão geral

Equipe	F1	Precisão	Revocação	Acurácia
Custódio e Paraboni (2018a)	0,685	0,672	0,784	0,779
Murauer, Tschuggnall e Specht (2018)	0,643	0,646	0,741	0,752
Halvani e Graner (2018)	0,629	0,649	0,729	0,715
<i>Mosavat</i>	0,613	0,615	0,725	0,721
HaCohen-Kerner et al. (2018)	0,598	0,605	0,701	0,732
Rodríguez et al. (2018)	0,588	0,580	0,706	0,707
PAN18-BASELINE	0,584	0,588	0,692	0,719

Fonte: Adaptado de Kestemont et al. (2018)

Na tabela 13 é apresentado o resultado da competição por língua, no qual pode-se notar que os melhores resultados foram obtidos para línguas francesa e espanhola, sendo que a diferença entre o resultado da presente proposta em relação ao segundo colocado foi de mais de 10 pontos percentuais.

Finalmente, na tabela 14 é possível ver que o desempenho do trabalho permaneceu estável mesmo com o aumento no número de autores.

⁵ *Mosavat* não divulgou o método utilizado.

Tabela 13 – PAN2018 - F1 reportada pelos organizadores - Por língua

Equipe	Geral	EN	FR	IT	PL	ES
Custódio e Paraboni (2018a)	0,685	0,744	0,668	0,676	0,482	0,856
Murauer, Tschuggnall e Specht (2018)	0,643	0,762	0,607	0,663	0,450	0,734
Halvani e Graner (2018)	0,629	0,679	0,536	0,752	0,426	0,751
<i>Mosavat</i>	0,613	0,685	0,615	0,601	0,435	0,731
HaCohen-Kerner et al. (2018)	0,598	0,672	0,609	0,642	0,431	0,636
Rodríguez et al. (2018)	0,588	0,601	0,510	0,571	0,556	0,705
PAN18-BASELINE	0,584	0,697	0,585	0,605	0,419	0,615

Fonte: Adaptado de Kestemont et al. (2018)

Tabela 14 – PAN2018 - F1 reportada pelos organizadores - Por número de autores

Equipe	Quantidade de autores			
	20	15	10	5
Custódio e Paraboni (2018a)	0,648	0,676	0,739	0,677
Murauer, Tschuggnall e Specht (2018)	0,609	0,642	0,680	0,642
Halvani e Graner (2018)	0,609	0,605	0,665	0,636
<i>Mosavat</i>	0,569	0,575	0,653	0,656
HaCohen-Kerner et al. (2018)	0,570	0,566	0,649	0,607
Rodríguez et al. (2018)	0,556	0,556	0,660	0,582
PAN18-BASELINE	0,546	0,532	0,595	0,663

Fonte: Adaptado de Kestemont et al. (2018)

4.3 Experimento 2: Atribuição autoral de conjunto aberto e multidomínio no desafio PAN-CLEF 2019

A atribuição autoral de conjunto aberto é uma extensão do problema de atribuição. Nessa abordagem, os textos sobre análise não necessariamente foram escritos pelos autores conhecidos do conjunto de treinamento. Esse problema foi o tema do desafio PAN-CLEF 2019 e, de forma similar ao PAN-CLEF 2018, foram disponibilizados um corpus, três sistemas *baselines* e um sistema de avaliação.

Essa seção apresenta um experimento usando o corpus PAN2019 e o modelo submetido para participação na competição PAN-CLEF-2019 (KESTEMONT et al., 2019).

4.3.1 Conjunto de dados

A primeira etapa do trabalho consistiu da análise do cópús PAN-CLEF-2019⁶, e comparando os resultados obtidos pelos sistemas *baseline*. Essa análise é descrita a seguir.

O cópús de desenvolvimento PAN-CLEF-2019 contempla vinte problemas escrito em cinco línguas (inglês, Frances, Italiano e Espanhol), com nove autores candidatos por problema, sete documentos por candidato e uma média de 4500 caracteres por documentos. Os organizadores disponibilizaram também três sistemas *baselines*, sendo eles, um modelo de compressão (TEAHAN; HARPER, 2003; POTTHAST et al., 2016), o método do impostor (KOPPEL; SEIDMAN, 2018), e um classificador SVM baseado em trigramas de caracteres. Mais detalhes dessas técnicas pode ser encontrado nos capítulos 2 e 3, e no trabalho em (ROCHA et al., 2017). A figura 11 apresenta uma comparação entre resultados da medida macro F1 obtidas pelos três sistemas *baselines* para cada língua.

Figura 11 – Comparação dos métodos baselines em AA de conjunto aberto

Fonte: Adaptado de Custódio e Paraboni (2019)

Da figura 11 podemos notar que o classificador SVM obteve o melhor desempenho entre os métodos apresentados. Sendo que o resultado obtido para os problemas língua inglesa (EN) foi similar para os métodos apresentados.

⁶ Disponível em <<https://pan.webis.de/clef19/pan19-web/author-identification.html>>

A figura 12 apresenta uma comparação entre os mesmos sistemas baselines de acordo com o percentual dos documentos cujo autor não é conhecido. Podemos notar que a proporção de documentos de autores desconhecidos, ou a abertura da tarefa de AA, tem um impacto considerável no desempenho de todos os sistemas. Esse resultado confirma a intuição geral que o problema de AA de conjunto aberto é mais desafiador que o problema de conjunto fechado.

Figura 12 – Medida macro-F1 para o corpus PAN-CLEF 2019 de acordo com o número de documentos desconhecidos.

Fonte: Adaptado de Custódio e Paraboni (2019)

4.3.2 Método proposto para AA de conjunto aberto

Assim como no experimento apresentado para o problema de conjunto fechado, e exposto em (CUSTÓDIO; PARABONI, 2018a), esse experimento assume que evidência da identidade autoral pode ser encontrada em elementos morfológicos, sintáticos e semânticos. Esses elementos podem ser modelados com o uso de n-gramas de caracteres e palavras (GOLDBERG, 2017). Como esses métodos são relativamente independentes entre si, as informações capturadas por um método de extração de características não necessariamente é capturado por outro.

Baseado nessas observações, esse experimento segue o apresentado em (CUSTÓDIO; PARABONI, 2018a) e lida com a AA de conjunto aberto usando um comitê de classifica-

dores. Mais especificamente, esse trabalho estende o trabalho (CUSTÓDIO; PARABONI, 2018a) adicionando um quarto método de extração de características e fazendo os ajustes para lidar com os conjuntos abertos.

Ranking de caracteres para AA

Os modelos de linguagem são centrais em um grande número de tarefas na área de processamento de língua natural. Diversos estudos tentaram estimar a probabilidade de uma palavra (ou caractere) aparecer após um determinado símbolo (GOLDBERG, 2017). Os n-gramas e as redes recorrentes (BAGNALL, 2015; SHRESTHA et al., 2017) são os métodos deste tipo mais conhecidos.

Este trabalho possui especial interesse nos modelos de linguagem baseados em grafos de adjacência de caracteres. Assumimos que o nível de influência de um nó pode ser utilizado como extração de característica e ser fonte de conhecimento para AA. A influência pode ser medida, por exemplo, usando o algoritmo PageRank (PAGE et al., 1999; SCHULT, 2008), no qual a influência de um nó p_i é definida pela equação 22, onde N é o número de nós, α é o fator ponderador, M é conjunto de nós para o qual p_i aponta, e L é o número de elementos para o qual p_j aponta.

$$PR(d, p_i) = \frac{1 - \alpha}{N} + \alpha \sum_{p_j \in M(p_i)} \frac{PR(d, p_j)}{L(d, p_j)} \quad (22)$$

Usando esse método como base, esse trabalho propõe um modelo de ranking de caracteres para a AA, chamado por *Rank.char*. Este modelo consiste da extração do grafo de adjacência dos caracteres para cada documento, removendo símbolos com baixa frequência, aplicado o grafo gerado no algoritmo PageRank e, por fim, alimentando um fluxo de aprendizado de máquina que será discutido na próxima seção.

4.3.3 Método proposto para AA de conjunto aberto

Existem diversas estratégias possíveis para combinar o resultado de um conjunto de classificadores. Nesse trabalho, foi utilizado um comitê composto de um classificador de classificadores, de forma similar a Custódio e Paraboni (2018a). A arquitetura proposta está ilustrada na figura 13. Os três primeiros classificadores são similares ao apresentado

em Custódio e Paraboni (2018a), e o último método (*Rank.char*) representa a presente extensão.

Figura 13 – Arquitetura do comitê para AA multidomínio

Fonte: Adaptado de Custódio e Paraboni (2019)

Dado que o resultado do comitê é um vetor de probabilidades, o problema do conjunto aberto é tratado por uma etapa adicional que atribui o documento à classe desconhecida ($<UNK>$) quando o desvio padrão do vetor de probabilidades for menor que o ponto de corte de 0.05.

4.3.4 Avaliação

Os modelos foram encontrados usando o conjunto de treinamento do PAN-CLEF 2019 usando o processo a seguir. A matriz de características foram normalizadas usando MaxAbsScaler do Python e passaram por redução de dimensionalidade com PCA. O PCA ajudou reduzir o impacto a correlação entre os n-gramas de tamanho variável que foram concatenados. O resultado final foi submetido a regressão logística multinomial. A lista dos hiper-parâmetros avaliados estão sumarizados na tabela 15.

Tabela 15 – Parâmetros do modelo

Módulo	Parâmetros	Valores analisados
Rare symbols	Freq. min. doc.	[0.01, 0.05, 0.1, 0.5]
	Freq. max. doc.	[0.25, 0.50, 0.90, 1.0]
TF-IDF	Extrator	Std.charN, Dist.charN, Std.wordN, Rank.char
	N-gram	(1,1) to (2,5)
	IDF	Normal, smoothed, none
PageRank	Alpha	[0.1, 0.5, 0.85, 0.90, 0.99]
Transformação	Normalização do doc.	L1, L2
	Scaling	MaxAbsScaler, StandardScaler
	PCA % da variância	0.99 (Fixo)
Classificador	Regressão logística	Softmax com Newton-cg

Fonte: Adaptado de Custódio e Paraboni (2019)

Os valores ótimos foram determinados por validação cruzada em busca em grid com *3-folds*. Os valores ótimos podem ser vistos na tabela 16.

Tabela 16 – Valores ótimos encontrados para PAN2019

Módulo	Parâmetros	Valores ótimos
Extração de características	N-gram	Std.charN (2,5) Dist.charN (2,5) Std.wordN (1,3) Rank.char (2,2)
	Alpha	0.9
	Freq. min. doc.	0.01
	Freq. min. doc.	1.0
	IDF	Suavisado
	Normalização do doc.	L2
Transformação	Scaler	Standard Scaler

Fonte: Adaptado de Custódio e Paraboni (2019)

Complementarmente aos resultados reportados pelo experimento, um grande número de alternativas foram consideradas. Estas incluem o uso do BM25 e representações *one-hot*, e uso de *bagging*, *boosting*, perceptrons multicamadas, árvores de decisão e outro. Os resultados obtidos foram próximos ao encontrado com o método descrito, e portanto descartados.

4.3.5 Resultados

A tabela 17 apresenta a medida F1 macro baseado no corpus de desenvolvimento do PAN-CLEF 2019 e software de avaliação (GOLLUB; BURROWS; STEIN, 2012), obtida para os sistemas *baselines*, os quatro classificadores individuais, o comitê EACH-USP

Tabela 17 – F1 macro baseado do corpus de desenvolvimento do PAN-CLEF 2019 AA

#	Baselines			Canais				Comitês	
	Comp	Imp	SVM	Std.charN	Std.wordN	Dist.charN.	Rank.char	EACH-USP	atual
01	0.66	0.66	0.67	0.69	0.61	0.65	0.57	0.86	0.83
02	0.34	0.40	0.44	0.47	0.45	0.39	0.19	0.51	0.49
03	0.49	0.42	0.50	0.62	0.47	0.46	0.37	0.63	0.67
04	0.51	0.49	0.35	0.40	0.31	0.24	0.20	0.48	0.46
05	0.36	0.27	0.49	0.42	0.46	0.39	0.23	0.61	0.53
06	0.67	0.56	0.67	0.65	0.45	0.61	0.47	0.74	0.74
07	0.52	0.43	0.50	0.57	0.46	0.41	0.52	0.56	0.60
08	0.49	0.36	0.51	0.56	0.37	0.46	0.25	0.66	0.61
09	0.61	0.26	0.61	0.67	0.44	0.52	0.13	0.70	0.71
10	0.52	0.35	0.46	0.53	0.28	0.44	0.41	0.59	0.61
11	0.56	0.42	0.62	0.62	0.37	0.64	0.48	0.75	0.73
12	0.50	0.51	0.58	0.62	0.37	0.47	0.46	0.61	0.65
13	0.72	0.49	0.67	0.73	0.44	0.53	0.43	0.76	0.76
14	0.78	0.53	0.60	0.81	0.42	0.57	0.57	0.70	0.85
15	0.72	0.33	0.74	0.71	0.38	0.42	0.34	0.88	0.77
16	0.67	0.60	0.74	0.81	0.66	0.53	0.48	0.72	0.74
17	0.62	0.32	0.57	0.62	0.53	0.45	0.18	0.68	0.65
18	0.65	0.53	0.69	0.80	0.61	0.54	0.28	0.82	0.79
19	0.41	0.28	0.55	0.52	0.52	0.41	0.28	0.62	0.60
20	0.24	0.29	0.52	0.44	0.31	0.22	0.12	0.47	0.44
Geral	0.55	0.42	0.57	0.61	0.44	0.47	0.35	0.67	0.67

Fonte: Adaptado de Custódio e Paraboni (2019)

extraído de Custódio e Paraboni (2018a), e o método proposto para AA de conjunto aberto.

A partir desse resultados observamos que o presente método mantém um bom desempenho relativo. A figura 14 apresenta uma comparação entre as medidas F1 macro obtidas para os métodos *baselines*, n-gramas de caracteres de tamanho variável, o método EACH-USP e o presente método para cada língua. Desses resultados notamos que o uso do *Rank.char* foi mais efetivo para o conjunto em Italiano. Além disso, a tarefa pareceu ser mais desafiadora para a língua inglesas do que as demais línguas.

Por fim, na figura 15 apresenta a comparação dos mesmos métodos de acordo com percentual de documentos de autoria desconhecida. Notamos que, todos os métodos apresentam uma queda no desempenho a medida que aumenta os percentual de desconhecidos.

Figura 14 – F1 macro por idioma.

Fonte: Adaptado de Custódio e Paraboni (2019)

Figura 15 – F1 macro de acordo com o percentual de documentos de autoria desconhecida.

Fonte: Adaptado de Custódio e Paraboni (2019)

4.3.6 Considerações sobre a AA de conjunto aberto

Esse trabalho apresentou uma extensão ao trabalho em Custódio e Paraboni (2018a) apresentando uma abordagem para a atribuição autoral de conjunto em cenário multido-mínio que se baseia em modelos completamente otimizados de n-gramas de caracteres, n-gramas de palavras e ranking de caracteres. Para esse fim, os resultados obtidos pelos classificadores individuais na forma de probabilidades foram combinados em comitê com

votação suavizada, e os autores desconhecidos foram detectados usando desvio padrão do vetor final de probabilidades.

O presente método é geralmente superior ao obtido pelos sistemas *baselines* mas não foram superiores ao trabalho Custódio e Paraboni (2018a). Embora a representação compacta de textos gerada pelo modelo *Rank.char* gere ganhos para os resultados, o modelo *Dist.CharN* se mantém como a melhor fonte de informação para a atribuição autoral de conjunto aberto.

5 Método proposto: Atribuição autoral dinâmica

Os estudos exploratórios demonstraram que métodos de extração de características diferentes capturam diferentes aspectos do estilo de escrita de um texto. Os resultados obtidos foram robustos para os conjuntos de dados propostos, e encorajaram a exploração de novos métodos de extração de características.

Do ponto de vista linguístico, os modelos apresentados buscaram capturar elementos sintáticos, como pontuação, elementos morfológicos, como preferências por certas desinências, e palavras e expressões de função, como "above all". Esses resultados indicam que a atribuição autoral baseia-se mais em elementos independentes de conteúdo.

O modelo apresentado para a atribuição autoral de conjunto aberto demonstrou que adicionar novos canais nem sempre irá melhorar a performance e, portanto, é necessário introduzir um mecanismo de seleção dos canais para cada conjunto ou domínio. Por fim, espera-se que o método proposto seja capaz de se adaptar aos diversos domínios onde a atribuição autoral venha a ser aplicada.

Espera-se que os modelos de n-gramas de elementos anotações linguísticas, sejam eles *tags*, *pos* ou *dep*, consigam capturar estruturas repetitivas de mais alto nível. A utilização dos *word embeddings* não foi efetiva na literatura, no entanto, espera-se que devam contribuir para o desempenho quando combinado aos demais métodos.

O método proposto para AA é uma extensão do comitê de máquina apresentado nos estudos preliminares, adicionando um mecanismo dinâmico para seleção dos modelos individuais mais relevantes e novos modelos individuais com outros tipos de extração de características.

A lista completas dos tipos de características e métodos individuais considerados pode ser vista a seguir. Os modelos *std.char*, *dist.char* e *std.word* usam n-gramas de tamanho variável como extração de característica e foram definidos no experimento preliminar, na seção 4.2.3. Os modelos *tag*, *pos* e *dep* são consistem de n-gramas de *tokens* e estão definidos na seção 5.2. O modelo *w2v* está definido na seção 5.3. Cada extração de características foi utilizado como entrada para o fluxo de classificação apresentado na seção 5.1.

1. **std.char**: é um método que extrai n-gramas de tamanho variável de caracteres.
2. **dist.char**: é um método que pré-processa o texto e extrai n-gramas de caracteres.
3. **std.word**: é um método que extrai n-gramas de palavras.

4. **tag**: é um método que pré-processa o texto transformando palavras em *tokens* de classes gramaticais e extrai n-gramas desses *tokens*.
5. **pos**: é um método similar a tag, no entanto, os *tokens* são menos granulares.
6. **dep**: é um método que pré-processa o texto transformando-o em *tokens* que representam dependência sintática e extrai n-gramas.
7. **w2v**: é um método que utiliza a representação distribuída de palavras.

A escolha do comitê com métodos tradicionais em detrimento aos métodos de aprendizado profundo (AP) se deve principalmente a dois fatores. O AP requer uma grande quantidade de dados e a AA geralmente possui conjunto de dados pequenos. Além disso, os métodos de AP encontrados na literatura não apresentaram um desempenho superior aos métodos tradicionais (KESTEMONT et al., 2018; KESTEMONT et al., 2019).

Nas seções seguintes é detalhado a construção do modelo de comitê, o novo método para seleção dinâmica e os novos canais de extração de informações.

5.1 *Modelo de atribuição autoral geral*

Os métodos de extração de informações de texto apresentados nessa proposta foram organizados usando um fluxo de aprendizado de máquina com mesma estrutura. A figura 16 apresenta o fluxo geral utilizado pelos métodos. Os documentos t são vetorizados por uma função de extração de características $V(t)$, o resultado é normalizado pela função $N(X)$ em seguida é aplicado a redução de dimensionalidade com PCA e por fim é aplicado um classificador multinomial que gera a probabilidade, $P(Y = k)$, para cada classe k .

5.2 *Modelo baseado em anotações POS e dependência sintática*

A organização sintática dos textos contém informações sobre como um autor estrutura as frases e sentenças. Esse tipo de informação pode ser extraído através de anotadores pré-treinados. Um exemplo desse tipo de anotador é o Spacy¹. Esse anotador extrai diversas informações do texto, e este trabalho irá considerar três delas: *TAG*, *POS* e *DEP*. As *TAG* referem-se as classes gramaticais de baixo nível, podendo ser específicas a

¹ Disponível em <<https://spacy.io>>

Figura 16 – Modelo de atribuição autoral geral

Fonte: José Eleandro Custódio, 2020

uma língua. As POS referem-se a versão padronizadas das *TAG* para várias línguas. Por fim, o modelo *DEP* extrai relações como sujeito e predicado. Na tabela 18 podem ser visto as anotações para frase "*Apple is looking at buying U.K. startup for \$1 billion*".

Tabela 18 – Exemplo de anotações sintáticas

	Apple	is	looking	at	buying	U.K.	startup	for	\$	1	billion
tag	nnp	vbz	vbg	in	vbg	nnp	nn	in	cd	cd	cd
pos	propn	verb	verb	adp	verb	propn	noun	adp	sym	num	num
dep	nsubj	aux	root	prep	pcomp	compound	dobj	prep	quantmod	compound	pobj

Fonte: Adaptado de <<https://spacy.io/usage/linguistic-features>>

Após o pré-processamento, cada texto torna-se uma sequência de tokens de anotações. Por exemplo, as *tags* para frase acima gerarão a sequência "nnp vbz vbg in vbg nnp nn in cd cd cd"², as quais podem ser contadas de forma similar ao modelo *bag of words*.

5.3 Modelo baseado em word embeddings

Os modelos de *word embeddings*, vistos na seção 2.3.2, são representações vetoriais que buscam capturar informações semânticas das palavras. Do ponto de vista da atribuição autoral, espera-se que esse tipo de modelo extraia os contextos semânticos utilizados com frequência por um autor.

O treinamento dos *embedding* é feito com um alto volume de dados, geralmente extraídos de fontes públicas como Wikipedia. Esse trabalho utilizou os *embeddings* treinados

² Uma lista completa das anotações e suas descrições podem ser encontradas em <<https://spacy.io/api/annotation>>

com o corpus CoNLL17³, exceto para língua inglesa que utilizou os *embeddings* Google News⁴.

Do ponto de vista prático, cada documento é representado pela média dos vetores de embeddings de cada palavra. Por exemplo, um corpus com 70 documentos e vocabulário de 40 mil palavras irá gerar uma matriz densa de 70 por 300, caso utilize *embedding* de 300 dimensões. Esse método é frequentemente chamado de *bag-of-means* (ZHANG; ZHAO; LECUN, 2015). É comum também a ponderação de cada palavra pelo peso TF-IDF correspondente, ou seja, combinando o modelo tradicional com a representação distribuída. Esse trabalho utilizou essa abordagem e ponderou o *embedding* com TF-IDF.

5.4 Modelo de atribuição autoral dinâmica - DynAA

A AA autoral dinâmica assume que cada método de extração de características captura um aspecto do estilo de escrita. Essa premissa conduz às questões: Como combinar os métodos? E como escolher os métodos que mais contribuem para a AA em um determinado domínio?

Combinar os canais nos remete aos modelos de comitê de máquina, sendo escolhido o modelo de voto ponderado (no inglês *soft voting*) de forma a preservar as probabilidades geradas por cada método. Os pesos foram otimizados utilizando um novo classificador que possui a forma $\sigma(\sum_c \sum_i (w_{ci} * c_i) + k)$, onde c_i é a probabilidade do autor i segundo o método c ser o autor do documento, w_{ci} é o peso c_i , k é uma constante e σ é a função *sigmoid*. O resultado do processo é um novo vetor s com i probabilidades. A figura 19 ilustra o caso de um documento com três métodos e cinco autores candidatos possíveis.

Tabela 19 – Estrutura do comitê de máquina para AA para um documento e cinco autores.

³ Disponível em <http://vectors.nlpl.eu/repository/>.

⁴ Disponível em <http://code.google.com/archive/p/word2vec/>.

Esse tipo de comitê pode ser considerado também como um classificador de classificadores, sendo que a principal vantagem é a extração dos pesos dinamicamente. Por outro lado, esse método pode apresentar sobre-ajuste e tornar-se complexo à medida que adicionamos mais fontes de informação. Para evitar ambos os problemas, será introduzido um método de seleção baseado no algoritmo *forward* comum na literatura estatística (MCCORNACK, 1970; JR; LEMESHOW; STURDIVANT, 2013). Nesse método, é escolhido o canal com melhor desempenho e, em seguida, é adicionado novos canais até que não seja observado melhorias.

O algoritmo 3 detalha o método proposto de atribuição autoral dinâmica (DynAA). Intuitivamente, esse algoritmo recebe um vetor c contendo as matrizes de d documentos por a autores geradas por cada método. É calculado a medida F1-macro para cada método, ordenado de forma decrescente e escolhido o método com melhor desempenho. Em seguida, é escolhido o próximo método da lista métodos e concatenado com o canal selecionado, é calculado o F1 e avaliado se houve uma melhora no desempenho. Caso afirmativo, o melhor modelo será a combinação dos métodos. Esse processo é executado iterativamente enquanto houver ganhos de desempenho.

Algoritmo 3 Comitê dinâmico para atribuição autoral

```

1: procedimento DYNAA( $C[]$  canais)
2:    $f1\_canais[]$  = calcular F1 para cada canal em  $C$ .
3:   Ordenar  $f1\_canais$  em ordem decrescente
4:    $melhorModelo = [C[0]]$ 
5:    $melhorF1 = f1\_canais[0]$ 
6:    $melhorou = Verdadeiro$ 
7:   enquanto  $melhorou$  faça
8:      $melhorou = Falso$ 
9:     para todo  $canal \in C$  faça
10:      Calcule o F1 para comitê de  $melhorModelo + m$ 
11:      se  $F1 > melhorF1$  então
12:         $melhorou = Verdadeiro$ 
13:         $melhorF1 = F1$ 
14:        Adicionar  $m$  a  $melhorModelo$ 
15:      fim se
16:    fim para
17:  fim enquanto
18: fim procedimento

```

Fonte: José Eleandro Custódio, 2020

5.5 *Considerações*

Este capítulo apresentou um modelo de comitê de máquina com seleção dinâmica de modelos. O método procura explorar diferentes estratégias para extração de características e escolher aquelas que são mais eficientes para cada domínio, idioma e outros. Este modelo será avaliado com base em seus componentes individuais e de forma global no próximo capítulo.

6 Avaliação

A avaliação do método proposto compreende a construção e otimização dos sete modelos listados na seção 5, e a otimização do comitê de atribuição autoral dinâmica (DynAA). Os documentos foram submetidos a cada método e geradas as matrizes de d documentos por a autores. Essas matrizes foram utilizadas como entrada para o DynAA.

Os corpúscos considerados foram separados em desenvolvimento e teste. Os dados de desenvolvimento foram separados em treinamento e amostra para validação do método comitê. Os métodos foram treinados com validação cruzada e foi utilizada a amostra de validação para seleção dos canais. Após a otimização, os métodos foram aplicados aos dados de testes e os resultados de teste foram reportados.

6.1 Conjunto de dados

Os corpúscos considerados são conjunto de dados de avaliação do PAN-CLEF 2018 composto por textos literários, um corpus de letras de músicas contendo textos em português e inglês, e um corpúscos de redes sociais extraídos de microblogs.

6.1.1 Corpus de avaliação PAN-CLEF-2018 - *PAN-EVAL-2018*

A competição PAN-CLEF disponibiliza dois corpúscos a cada edição. O primeiro é o corpúscos de desenvolvimento do sistema e que foi analisado nos estudos exploratórios. O segundo foi disponibilizado após a competição, e consiste de problemas idênticos ao corpúscos de desenvolvimento. Esse segundo corpúscos será utilizado para avaliação do sistema proposto. A tabela 20 apresenta as estatísticas para cada instância.

Analisando as estatísticas, podemos notar que os problemas em Polonês utilizam um maior número de caracteres por palavras, utilizam menos espaços e mais símbolos (ou caracteres diferentes das letras latinas). A língua inglesa apresenta menos caracteres por palavras, e conseqüentemente, utiliza mais espaços.

Tabela 20 – Estatística descritiva do corpus PAN-EVAL-2018

#	Língua	Autores	Palavras	Caract.	Caract/Palavra	% Espaços	% Símbolos
01	EN	20	816	4.444	5,4	18,7	3,9
02	EN	15	814	4.431	5,4	18,7	3,9
03	EN	10	816	4.476	5,5	18,8	3,8
04	EN	5	848	4.665	5,5	18,5	2,9
05	FR	20	786	4.413	5,6	18,0	6,7
06	FR	15	790	4.429	5,6	18,1	6,6
07	FR	10	784	4.393	5,6	18,3	7,0
08	FR	5	781	4.311	5,5	18,0	7,3
09	IT	20	806	4.823	6,0	17,2	4,5
10	IT	15	792	4.742	6,0	17,3	4,8
11	IT	10	820	4.961	6,0	17,5	4,5
12	IT	5	799	4.709	5,9	17,1	4,6
13	PL	20	778	5.188	6,7	16,4	11,2
14	PL	15	792	5.269	6,7	16,3	11,0
15	PL	10	763	5.086	6,7	16,7	11,3
16	PL	5	783	5.249	6,7	16,3	10,7
17	SP	20	814	4.686	5,8	18,3	5,6
18	SP	15	822	4.743	5,8	18,4	5,5
19	SP	10	825	4.755	5,8	18,6	5,6
20	SP	5	839	4.880	5,8	17,8	5,6

Fonte: Adaptado de Custódio e Paraboni (2019)

6.1.2 Corpus de letras de músicas - *Lyrics*

A atribuição autoral de letras de músicas foram investigadas usando dados em língua turca (Kırmacı; Oğul, 2015) e língua bengali (Hossain; Al Marouf, 2018). Para o presente trabalho, foi coletado um corpus de letras de músicas em inglês e português em um cenário multidiomático cobrindo 49 gêneros musicais, incluindo sertanejo, pop, rock, jazz, bossa nova e outros. O conjunto foi extraído de um repositório de contribuição pública¹, onde os usuários podem inserir letras escritas pelos seus artistas favoritos, ou por si mesmos.

O conjunto inicial continha 150 mil canções compostas pelos 100 artistas mais cantados no ano de 2018. Foram selecionadas músicas que tivesse a identificação do compositor, e descartado músicas com autoria múltipla. Composições com menos de 750 caracteres foram removidas. Como o idioma de cada música não estava disponível, foi elaborado um procedimento detectá-lo. Esse procedimento consistiu do cálculo da

¹ <<https://www.letras.mus.br/>>

similaridade de cossenos entre cada documento e um *cópus* de *stopwords* de 50 idiomas². O documento foi associado ao idioma que apresenta a maior similaridade.

Diversas canções possuem mais de uma versão, como, por exemplo, versão original e versão estendida. Essas duplicações foram removidas caso tivessem similaridade de cossenos maior ou igual a 0.6. Esse procedimento removeu menos de 1% dos documentos.

Após as limpezas, foram construídos dez cenários usando a mesma estrutura do *cópus* PAN-CLEF. Para criar uma configuração multidomínio, foram selecionados autores que tivessem composições em pelo menos três gêneros musicais e foi garantido que os conjuntos de treinamento e teste contivessem os mesmos estilos. Cada problema combina a língua (EN-Inglês ou PT-Português) e o número de autores (de 5 a 25 com intervalo de 5). A tabela 21 traz as estatísticas descritivas para esse *cópus*.

Tabela 21 – Estatística descritiva do *cópus* Lyrics

#	Língua	Autores	Gêneros	Palavras	Caract.	Caract/Palavra	% Espaços	% Símbolos
01	PT	5	9	205	1,067	5,2	19,4	5,0
02	PT	10	12	212	1,086	5,1	20,0	5,2
03	PT	15	13	207	1,050	5,1	19,8	5,0
04	PT	20	13	218	1,126	5,8	19,6	5,1
05	PT	25	14	227	1,177	5,2	19,8	5,1
06	EN	5	8	274	1,289	4,7	20,6	3,0
07	EN	10	11	303	1,437	4,8	20,2	3,1
08	EN	15	20	309	1,458	4,7	20,3	3,4
09	EN	20	21	305	1,434	4,7	20,4	3,3
10	EN	25	21	283	1,342	4,8	20,3	3,1

Fonte: Adaptado de Custódio e Paraboni (2019)

Analisando o *cópus* podemos ver que a utilização de caracteres de espaço é praticamente idêntico entre os documentos em PT e EN. Os símbolos e caracteres especiais está mais presente em PT que EN. Esses fatores podem ter sido influenciados pelo fato que o conteúdo semântico é possivelmente mais relevante para esse domínio que a estrutura do texto. A quantidade média de caracteres é de 1000 caracteres por documento, ou seja, menos de um quarto do *cópus* PAN-EVAL-2018.

6.1.3 Corpus de redes sociais - Social

Redes sociais são um campo fértil para AA. A seção 3.2.3 descreve um conjunto de dados de mídia sociais extraídos do Twitter e analisado em Rocha et al. (2017). Esses

² <<https://github.com/6/stopwords-json>>

dados foram gentilmente cedidos para esse trabalho e serão analisados com os métodos propostos.

O foco do trabalho em Rocha et al. (2017) foi a avaliação de métodos de aprendizado de máquina, e analisou cenários variando o número de autores entre 1 e 1000 e a quantidade de tweet entre 1 e 1000. Para o presente trabalho, foram extraídas amostras do conjunto fornecido, e será analisado o número de autores entre 5, 10, 20, 50, ou seja, focando em número menor de autores. O número de documentos no conjunto de treinamento foi analisado usando duas abordagens. As 8 primeiras instâncias de problemas utilizaram 1 tweet por documento, variando o número de autores e o número de tweets. As demais instâncias utilizaram mais de um tweet por documento, concatenando tweets consecutivos até atingir o número mínimo de 100, 250 e 500 caracteres por documento. O objetivo do segundo cenário é investigar a estabilidade do método a medida que aumentamos a quantidade de informação. Na tabela 22 podem ser vistos os 32 problemas e suas estatísticas.

Analisando o *cópus* Social vemos que a utilização de caracteres especiais e símbolos é o dobro do *cópus* de Lyrics. Isso é esperado dado que as redes sociais usam *emoticons* e pontuações.

Tabela 22 – Estatística descritiva do corpus Social

#	Autores	# Docs	Palavras	Caract.	Caract/Palavra	% Espaços	% Símbolos
01	5	10	14	70	4.7	18.9	4.6
02	5	50	11	50	4.4	20.4	3.6
03	10	10	14	63	4.4	20.9	3.8
04	10	50	9	48	4.8	18.5	4.6
05	20	10	11	56	5.0	18.2	5.9
06	20	50	15	74	4.8	19.3	3.2
07	50	10	13	68	5.3	17.7	5.2
08	50	50	14	70	5.0	18.4	4.6
09	5	10	30	152	5.0	21.3	3.2
10	5	10	60	292	4.8	22.0	4.4
11	5	10	102	523	5.1	21.2	3.1
12	5	50	34	177	5.2	20.6	5.1
13	5	50	58	310	5.3	20.3	5.4
14	5	50	102	538	5.2	20.0	3.5
15	10	10	29	144	5.0	21.1	2.9
16	10	10	58	294	5.1	20.9	4.7
17	10	10	105	541	5.1	20.6	4.8
18	10	50	25	132	5.2	20.4	2.0
19	10	50	54	283	5.2	20.6	5.5
20	10	50	104	543	5.2	20.4	4.2
21	20	10	28	139	5.0	21.2	4.6
22	20	10	55	276	5.0	21.5	3.9
23	20	10	106	544	5.2	20.7	4.7
24	20	50	25	131	5.2	20.7	3.7
25	20	50	58	299	5.1	20.8	4.8
26	20	50	103	536	5.2	20.6	4.8
27	50	10	27	140	5.2	20.7	4.4
28	50	10	56	284	5.1	20.9	4.4
29	50	10	107	540	5.0	20.9	4.0
30	50	50	27	138	5.0	21.1	3.9
31	50	50	56	286	5.0	20.9	4.1
32	50	50	107	539	5.1	21.1	4.0

Fonte: José Eleandro Custódio, 2020

6.2 Resultados

As tabelas 23, 25 e 27 apresentam a medida F1 macro para os corpú PAN-EVAL-2018, Lyrics e SocialAA, respectivamente. A primeira coluna apresenta o identificador do problema, ou cenário de avaliação, a coluna SVM apresenta o resultado do classificador *baseline* SVM com 4-gramas de caracteres, a coluna DynAA apresenta o resultado do método dinâmico e as demais colunas apresentam o resultado para cada tipo de método. As demais colunas apresentam os resultados para método DynAA e os métodos individuais. Os melhores resultados estão indicados em negrito e os canais que foram selecionados pelo método proposto estão indicados com '*’.

As tabelas 24, 26 e 28 apresentam visões consolidadas dos resultados. São apresentadas as médias por número de autores, língua e quantidade de caracteres presentes nos textos.

Tabela 23 – Medida F1 macro para o corpú PAN-EVAL-2018

#	Língua	SVM	DynAA	Std.Char	Dist.Char	Std.Word	TAG	POS	DEP	W2V
01	EN	0.51	0.65	*0.51	*0.44	*0.34	0.50	0.37	0.35	0.28
02	EN	0.52	0.60	*0.55	*0.46	0.37	*0.57	0.45	0.21	0.14
03	EN	0.77	0.87	*0.87	0.47	0.42	0.50	0.46	0.38	0.14
04	EN	0.82	0.85	*0.80	*0.59	0.56	0.85	0.85	0.72	0.55
05	FR	0.43	0.41	*0.43	*0.38	*0.21	0.33	0.27	0.16	0.09
06	FR	0.59	0.52	*0.42	*0.39	*0.43	0.32	0.27	0.16	0.13
07	FR	0.38	0.68	*0.57	*0.50	0.28	0.54	0.46	0.32	0.06
08	FR	0.64	0.57	*0.64	*0.57	0.36	0.29	0.31	0.22	0.29
09	IT	0.66	0.66	*0.68	*0.48	*0.36	0.39	0.24	0.26	0.09
10	IT	0.38	0.58	0.43	*0.52	*0.36	*0.36	0.34	0.22	0.07
11	IT	0.60	0.61	*0.59	*0.57	0.44	*0.35	0.22	0.20	*0.07
12	IT	0.28	0.47	0.35	0.18	0.46	*0.45	0.43	0.29	0.34
13	PL	0.43	0.49	*0.42	*0.36	0.32	-	-	-	0.09
14	PL	0.45	0.40	*0.50	*0.27	0.22	-	-	-	0.06
15	PL	0.36	0.34	*0.43	*0.30	0.24	-	-	-	0.09
16	PL	0.39	0.56	*0.36	*0.47	0.27	-	-	-	0.19
17	SP	0.47	0.70	*0.52	*0.53	0.36	*0.52	0.46	*0.28	0.08
18	SP	0.54	0.68	*0.55	*0.48	*0.42	0.40	0.40	*0.33	*0.09
19	SP	0.62	0.82	*0.60	*0.64	0.49	0.55	0.45	0.21	0.14
20	SP	0.48	0.54	*0.54	0.64	0.20	0.79	0.71	0.41	*0.07

Fonte: José Eleandro Custódio, 2020

Tabela 24 – Medida F1 macro para o corpus PAN-EVAL-2018 consolidado por número de autores e língua.

		SVM	DynAA	Std.Char	Dist.Char	Std.Word	TAG	POS	DEP	W2V
Autores	5	0.52	0.60	0.54	0.49	0.37	0.60	0.57	0.41	0.29
	10	0.55	0.66	0.61	0.50	0.37	0.48	0.39	0.27	0.10
	15	0.49	0.56	0.49	0.42	0.36	0.41	0.36	0.23	0.10
	20	0.50	0.58	0.51	0.44	0.32	0.43	0.33	0.26	0.13
Língua	EN	0.66	0.74	0.68	0.49	0.42	0.60	0.53	0.42	0.28
	FR	0.51	0.54	0.51	0.46	0.32	0.37	0.33	0.21	0.14
	IT	0.48	0.58	0.51	0.44	0.40	0.39	0.30	0.24	0.14
	PL	0.41	0.45	0.43	0.35	0.26	0.00	0.00	0.00	0.11
	SP	0.53	0.69	0.55	0.57	0.37	0.57	0.50	0.31	0.10

Fonte: José Eleandro Custódio, 2020

Tabela 25 – Medida F1 macro para o corpus Lyrics

#	Língua	Autores	SVM	DynAA	Std.Char	Dist.Char	Std.Word	TAG	POS	DEP	W2V
01	PT	5	0.50	0.62	*0.58	0.40	0.52	0.33	0.40	0.26	*0.54
02	PT	10	0.30	0.36	0.35	0.32	*0.35	*0.26	0.25	0.21	0.29
03	PT	15	0.31	0.30	*0.32	0.15	0.32	*0.13	0.13	0.14	0.18
04	PT	20	0.21	0.26	*0.28	0.13	*0.21	0.14	0.15	0.13	0.18
05	PT	25	0.18	0.24	*0.24	0.12	*0.20	0.18	0.11	0.14	*0.15
06	EN	5	0.54	0.59	*0.64	0.46	0.38	*0.44	0.39	0.28	0.40
07	EN	10	0.39	0.46	*0.43	*0.38	*0.47	*0.46	0.38	0.30	0.33
08	EN	15	0.28	0.42	*0.35	*0.29	*0.34	*0.31	0.19	*0.21	*0.34
09	EN	20	0.33	0.43	*0.46	*0.18	*0.33	*0.31	0.21	0.18	0.24
10	EN	25	0.28	0.30	*0.28	*0.19	*0.26	*0.22	0.17	0.14	*0.18

Fonte: José Eleandro Custódio, 2020

Tabela 26 – Medida F1 macro para o corpus Lyrics consolidado por número de autores e língua.

		SVM	DynAA	Std.Char	Dist.Char	Std.Word	TAG	POS	DEP	W2V
Autores	5	0.52	0.61	0.61	0.43	0.45	0.39	0.39	0.27	0.47
	10	0.35	0.41	0.39	0.35	0.41	0.36	0.32	0.25	0.31
	15	0.30	0.36	0.33	0.22	0.33	0.22	0.16	0.17	0.26
	20	0.27	0.34	0.37	0.16	0.27	0.23	0.18	0.15	0.21
	25	0.23	0.26	0.26	0.15	0.23	0.20	0.14	0.14	0.17
Língua	PT	0.30	0.35	0.35	0.22	0.32	0.21	0.21	0.18	0.27
	EN	0.37	0.44	0.43	0.30	0.36	0.35	0.27	0.22	0.30

Fonte: José Eleandro Custódio, 2020

Tabela 27 – Medida F1 macro para o corpus SocialAA

#	Autores	Caract.	SVM	DynAA	Std.Char	Dist.Char	Std.Word	TAG	POS	DEP	W2V
01	5	70	0.13	0.13	0.40	*0.13	0.07	-	0.20	-	0.10
02	5	50	0.33	0.30	*0.30	*0.73	*0.53	*0.73	0.30	*0.23	0.33
03	10	63	0.10	0.28	0.22	*0.28	-	0.13	0.13	0.09	0.22
04	10	48	-	0.20	0.10	*0.52	*0.10	*0.03	-	-	0.17
05	20	56	0.08	0.18	0.12	*0.13	0.12	0.08	0.05	0.10	0.09
06	20	74	0.23	0.20	0.25	*0.21	0.22	*0.13	0.12	0.12	0.17
07	50	68	0.07	0.09	0.12	*0.12	0.12	0.03	0.03	0.08	0.04
08	50	70	0.07	0.11	0.10	*0.19	*0.02	*0.10	0.09	*0.04	0.05
09	5	152	0.28	1.00	0.50	*1.00	0.47	0.47	0.23	0.27	0.30
10	5	292	0.33	0.73	0.33	*0.73	0.10	0.23	0.30	0.50	0.10
11	5	523	0.30	1.00	0.47	*1.00	0.50	*0.47	0.40	0.30	0.13
12	5	177	1.00	1.00	*1.00	*1.00	*1.00	*0.60	0.53	0.40	0.73
13	5	310	1.00	1.00	*1.00	*1.00	*0.73	1.00	*0.47	1.00	*0.33
14	5	538	1.00	1.00	*1.00	1.00	*1.00	1.00	1.00	1.00	0.73
15	10	144	0.07	0.37	0.32	*0.52	*0.22	0.23	0.15	*0.17	*0.27
16	10	294	0.23	0.73	*0.48	*0.55	0.48	0.53	0.15	0.07	0.30
17	10	541	0.43	1.00	0.80	*0.87	*0.60	0.73	0.50	0.23	*0.53
18	10	132	0.40	0.63	0.40	*0.87	0.42	*0.17	0.25	*0.24	0.32
19	10	283	0.87	1.00	*0.87	*0.87	*0.87	1.00	*0.73	0.62	0.65
20	10	543	0.87	1.00	*0.87	*1.00	*0.73	1.00	0.73	0.87	0.73
21	20	139	0.24	0.30	0.34	*0.27	0.27	*0.32	0.18	0.23	0.30
22	20	276	0.41	0.70	0.51	*0.75	0.42	*0.30	0.20	0.19	0.38
23	20	544	0.53	0.75	*0.63	*0.73	0.59	*0.63	0.48	*0.34	0.44
24	20	131	0.36	0.44	*0.35	*0.40	0.32	*0.37	0.19	*0.17	0.23
25	20	299	0.82	0.88	*0.77	*0.68	0.50	*0.65	0.25	*0.23	0.26
26	20	536	0.93	1.00	*0.93	*1.00	0.53	0.70	0.47	0.28	0.33
27	50	140	0.06	0.15	*0.17	*0.19	0.08	0.02	*0.09	*0.05	0.03
28	50	284	0.37	0.55	*0.41	*0.48	0.29	*0.35	0.17	*0.23	0.18
29	50	540	0.51	0.71	*0.53	*0.60	0.50	0.45	0.30	*0.20	0.25
30	50	138	0.39	0.46	*0.46	*0.32	*0.25	*0.14	0.09	*0.08	0.07
31	50	286	0.56	0.84	*0.75	*0.47	0.36	*0.27	0.15	*0.11	0.34
32	50	539	0.82	0.95	*0.87	*0.73	0.73	*0.58	0.44	0.34	0.53

Fonte: José Eleandro Custódio, 2020

Tabela 28 – Medida F1 macro para o corpus Social consolidado por número de autores e número de caracteres por documento.

		SVM	DynAA	Std.Char	Dist.Char	Std.Word	TAG	POS	DEP	W2V
Autores	5	0.55	0.77	0.62	0.82	0.55	0.56	0.43	0.46	0.35
	10	0.37	0.65	0.51	0.68	0.43	0.48	0.33	0.29	0.40
	20	0.45	0.56	0.49	0.52	0.37	0.40	0.24	0.21	0.28
	50	0.36	0.48	0.43	0.39	0.29	0.24	0.17	0.14	0.19
No. caracteres	50	0.17	0.25	0.20	0.62	0.32	0.38	0.15	0.12	0.25
	60	0.09	0.23	0.17	0.21	0.06	0.10	0.09	0.09	0.15
	70	0.13	0.14	0.22	0.16	0.11	0.06	0.11	0.06	0.09
	130	0.38	0.54	0.38	0.63	0.37	0.27	0.22	0.20	0.28
	140	0.19	0.32	0.32	0.32	0.21	0.18	0.13	0.13	0.17
	150	0.28	1.00	0.50	1.00	0.47	0.47	0.23	0.27	0.30
	180	1.00	1.00	1.00	1.00	1.00	0.60	0.53	0.40	0.73
	280	0.55	0.75	0.60	0.70	0.52	0.55	0.37	0.35	0.40
	290	0.38	0.77	0.52	0.58	0.31	0.34	0.20	0.23	0.25
	300	0.82	0.88	0.77	0.68	0.50	0.65	0.25	0.23	0.26
	310	1.00	1.00	1.00	1.00	0.73	1.00	0.47	1.00	0.33
	520	0.30	1.00	0.47	1.00	0.50	0.47	0.40	0.30	0.13
	540	0.73	0.92	0.81	0.85	0.67	0.73	0.56	0.47	0.51

Fonte: José Eleandro Custódio, 2020

6.3 Discussões sobre os resultados

6.3.1 Discussões sobre os resultados para PAN-EVAL-2018

A partir dos dados da tabela 23 podem ser feitas algumas observações. O resultado do método DynAA foi superior ao *baseline* SVM na maioria dos cenários para o PAN-EVAL-2018. O método de n-gramas de caracteres de tamanho variável foi o principal método para esse domínio. As distorções textuais foram mais relevantes para idioma espanhol. Os *embeddings* apresentaram o pior desempenho entre os métodos, o que pode indicar que o conteúdo semântico tem menos relevância para textos literários em cenário multidomínio. O melhor resultado dos *embeddings* foi encontrado para o inglês, possivelmente porque o corpus Google News foi treinando com um volume superior de dados que CoNLL17. As anotações linguísticas foram importantes em diversos casos, sendo as anotações mais

granulares provenientes das *TAGs* as mais relevantes. O idioma polonês foi o mais desafiador, tendo tirado pouco proveito dos métodos usados.

A tabela 24 ilustra a visão consolidada por número de autores e idioma. Podemos ver que o método proposto apresentou desempenho superior em todos os cenários. Esse resultado é particularmente importante porque o conjunto de teste do PAN contém um número reduzido de documentos e apresenta um desbalanceamento de classes, simulando situações reais de aplicação da AA. Vale notar que, os dados da tabela 24 consolidam um volume maior de informações, sendo assim, espera-se maior estabilidade nos resultados apresentados. Nota-se que o método *DynAA* estabiliza em um patamar superior aos demais.

6.3.2 Discussões sobre os resultados para Lyrics

A tabela 25 traz os resultados dos métodos avaliados no corpus Lyrics. O método n-gramas de caracteres foi, também, a fonte de informação principal para esse gênero literário. As distorções tiveram maior peso para os dados em inglês. Os modelos de n-gramas de palavras foram inferiores aos modelos de caracteres. Os modelos baseados em anotações linguísticas não foram determinantes para esse conjunto de dados, possivelmente porque as letras de músicas contêm refrão, quebras de linhas no meio das sentenças, onomatopeias etc. O modelo de *embeddings* teve um desempenho melhor nesse conjunto de dados que no conjunto PAN-EVAL-2018, no entanto, ainda inferior aos modelos de caracteres.

O método *DynAA* apresentou um F1 médio superior ao *baseline* e aos demais métodos, quando visto na tabela consolidada 26. Apesar dos resultados obtidos serem promissores, é possível ver que esse domínio é mais desafiador que o PAN-EVAL-2018. Isso se deve possivelmente pela perda da informação ou inclusão de ruído quando os usuários contribuem para o repositório. Por outro lado, temos documentos menores que o conjunto de dados literário do PAN.

6.3.3 Discussões sobre os resultados para Social

A tabela 27 traz a medida F1 para o corpus Social. Os problemas de 1 a 8 apresentam casos em que foram utilizados um tweet por documento, variando o número de autores. É possível ver que todos os métodos tiveram um baixo desempenho e, mesmo combinando diversos canais, não foi possível obter um resultado estável. Os problemas de 9 a 32

apresentam os resultados para os problemas onde foram concatenados mais de um tweet por documento, variando o número de autores e a quantidade mínima de caracteres ([150, 250 e 500]). É possível ver que os problemas com 5 autores e maior número de caracteres apresentaram um sobreajuste acertando 100% dos casos. Entre os métodos, destaca-se o modelo de distorção textual. Esse modelo tira proveito da variabilidade linguística presente nos *emoicons* e no estilo livre de escrita. Na tabela 28 é possível ver que o desempenho dos métodos aumenta à medida que temos um maior número de caracteres por documento, e que diferentemente do *Lyrics* corpus, os métodos tiveram um desempenho satisfatório com menos de 1000 caracteres.

6.3.4 Considerações sobre o método proposto

O método proposto explorou diversos níveis linguísticos e ambientes utilizados para comunicação. Os resultados obtidos foram bem diferentes entre si, o que reforça a necessidade de estudo de métodos específicos.

Os resultados obtidos pelos modelos de caracteres foram os mais robustos e flexíveis. Este trabalho explorou diversas otimizações para esses modelos. Entre eles pode-se citar, a utilização de *n*-gramas de tamanho variável e o uso de distorções textuais.

As anotações linguísticas tiveram um desempenho médio, e esperado, sendo inferior aos modelos de caracteres. Sendo que as *TAGs* tiveram um desempenho superior as *POS* e as dependências sintáticas, *DEP*.

Os modelos de *embeddings* pré-treinados tiveram desempenho menor. No geral, a qualidade dessa representação está relacionada com o volume de dados do conjunto onde foram treinadas. Além disso, a semântica de um texto pode ser importante para identificar um autor, porém não é confiável para um cenário multidomínio.

O fator de maior impacto para o desenvolvimento dos modelos é a disponibilidade de dados. Sua escassez inviabilizou a exploração de métodos de aprendizado profundo e transferência de conhecimento (do inglês, *transfer learning*), e também dificultou a exploração mais consistente dos hiper-parâmetros, sendo comum parâmetros diferentes atingirem o mesmo resultado durante testes exploratórios.

6.4 Análise complementar - relevância dos n-gramas de caracteres

Em complemento à avaliação discutida no capítulo anterior, foi conduzida uma análise do modelo de n-grama de caracteres. Esse modelo apresentou um desempenho consistente em diversos estudos, no entanto, é de difícil interpretação devido ao baixo nível linguístico. Alguns estudos examinados no capítulo 3 tentaram explicar porque esse modelo é útil para a AA. Em especial, o trabalho em Sapkota et al. (2015) agrupou os n-gramas em categorias baseadas na posição dos mesmos no texto. No entanto, considerou apenas 3-gramas de caracteres e não avaliou a importância de cada categoria dentro de um sistema com múltiplas categorias. A seguir, é analisado o modelo *std.char*, apresentado na seção 4.2.3 e que não possui essas limitações.

O processo proposto na figura 16, apresentado na seção 5.1 pode ser formalizado pela equação 23. Os documentos t são transformados na matriz de *bag*, $X_{d,g}$, de d documentos por g n-gramas. X é normalizado com o uso da função MaxAbs, N , e a matriz resultante é multiplicada pela matriz de PCAs, $C_{c,g}$, com c componentes. Por fim, as probabilidades são geradas usando a função logística softmax σ com parâmetros $L_{c,k}$ e $L_{1,k}^0$ sendo k o número de autores.

$$\begin{aligned}
 P(k|X_{d,g}) &= Y_{d,k} = \sigma(N(X_{d,g}) \cdot C_{c,g}^T \cdot L_{c,k} + L_{1,k}^0) \\
 X_{d,g} &= V(t) \\
 P(k|d) &= \operatorname{argmax}_k P(k|X_{d,g})
 \end{aligned}
 \tag{23}$$

Partindo da equação 23, podemos extrair os pesos, ou relevância, para cada n-grama dentro do sistema e responder algumas questões de pesquisa. A seguir é detalhada a equação de relevância, elaborado algumas questões e os resultados são comentados.

A equação 23 é composta pelos pesos do PCA ($C_{c,g}$), pelos pesos do classificador ($L_{c,k}$ e $L_{1,k}^0$), pela matriz de n-grama ($X_{d,g}$), pela função de normalização N e pela função softmax σ . Como a função σ é monotônica crescente, a mesma pode ser omitida, assim como o termo constante $L_{1,k}^0$. Já a matriz de n-gramas X pode ser representada pela sua média $T_{1,g}$. Assim, a relevância R de um n-grama g com relação a um autor k pode ser

definida como sendo a multiplicação ponto-a-ponto da média dos n-gramas pelo valor absoluto dos pesos do PCA vezes os pesos do classificador.

$$\begin{aligned} T_{1,g} &= \overline{N(X_{d,g})} \\ R_{g,k} &= T_{1,g} \times |C_{c,g}^T \cdot L_{c,k}| \end{aligned} \quad (24)$$

Por fim, utilizando a média dos n-gramas entre os autores, $\overline{R_{g,k}}$ e normalizando pela norma $L1$, de forma a somar 1, temos a equação 25:

$$R_{1,g} = \frac{R_{g,k}}{|\overline{R_{g,k}}|_1} \quad (25)$$

Este trabalho considera as questões de pesquisa listadas na tabela 29. A coluna categoria contém um acrônimo para questão e a coluna conceito traz a forma de medida criada para avaliar a questão de pesquisa.

Tabela 29 – Categorias de n-gramas de caracteres analisados

#	Questões	Conceito
ngram	Quais os tamanhos dos n-gramas que mais contribuem para AA?	Número de caracteres ou tamanho do n-grama.
nascii	Diacríticos e símbolos não latinos são relevantes para AA?	% do n-grama composto por caracteres que não são a-z,A-Z,dígitos, pontuação, ou espaços.
lower	As letras minúsculas são mais ou menos relevantes ?	% do n-grama composto por caracteres minúsculos.
upper	As letras maiúsculas são mais ou menos relevantes ?	% do n-grama composto por caracteres maiúsculos.
punct	A pontuação pode ser descartada?	% do n-grama composto por caracteres de pontuação.
space	Os espaços podem ser descartados?	% do n-grama composto por caracteres de espaço e nova linha.
WS	A fronteira após a palavra é relevante?	Presença ou não de espaço no início do n-grama.
SW	A fronteira antes da palavra é relevante?	Presença ou não de espaço no fim do n-grama.
WSW	Espaços e pontuações entre palavras são relevantes?	Presença ou não de espaço no meio do n-grama.

Fonte: José Eleandro Custódio, 2020

O fluxo apresentado na seção anterior foi executado nos problemas do PAN2018 usando um conjunto de parâmetros único, de tal forma que fosse possível a comparação

entre diferentes idiomas. Os parâmetros considerados são: Faixa de n-gramas entre 1 e 5, TF sublinear e IDF normal. Após o treinamento, foi calculada a matriz de relevância por n-grama, e cada n-grama foi anotado com as categorias presentes na tabela 29 e os pesos sumarizados por língua.

Na tabela 30 pode ser visto que unigramas de caracteres não são relevantes para AA e que 4-gramas e 5-gramas possuem maiores pesos. Os n-gramas que possuem até 25% de caracteres não ASCII (códigos da tabela unicode com valor acima de 192) apresentaram maior peso na classificação. Os n-gramas contendo caracteres minúsculas (*lower*) foram mais relevantes quando contiveram 75% do tamanho do n-grama. As letras maiúsculas (*upper*), pontuações (*punct*) e espaços (*space*) são mais relevantes quando presentes em até 25% do n-grama. Esse resultado também pode ser visto na fronteira entre palavras onde o padrão "palavra-[espaço/pontuação]-palavra" (*WSW*) contém 36% dos pesos.

A análise por idioma do corpus PAN2018 mostra que a língua inglesa se baseia mais nos caracteres de espaços, maiúsculos e minúsculos que os demais idiomas. Os caracteres *non-ascii* foram mais relevantes para o polonês e francês.

Tabela 30 – Relevância dos n-gramas de caracteres avaliados por categoria de conteúdo aplicados ao corpus de desenvolvimento do PAN-2018.

ngram	EN	PL	FR	IT	SP	Geral	lower	EN	PL	FR	IT	SP	Geral
1	0.01	0.00	0.01	0.01	0.01	0.01	0%	0.02	0.03	0.03	0.02	0.02	0.03
2	0.05	0.05	0.05	0.04	0.04	0.05	25%	0.04	0.07	0.05	0.04	0.05	0.05
3	0.19	0.20	0.18	0.17	0.18	0.19	50%	0.16	0.26	0.20	0.16	0.19	0.19
4	0.34	0.35	0.34	0.34	0.34	0.34	75%	0.49	0.37	0.44	0.42	0.43	0.43
5	0.41	0.40	0.42	0.44	0.43	0.42	100%	0.28	0.27	0.28	0.35	0.31	0.30

nascii	EN	PL	FR	IT	SP	Geral	upper	EN	PL	FR	IT	SP	Geral
0%	0.81	0.58	0.70	0.80	0.75	0.73	0%	0.91	0.92	0.92	0.93	0.91	0.92
25%	0.15	0.32	0.25	0.16	0.20	0.22	25%	0.08	0.07	0.07	0.06	0.08	0.07
50%	0.03	0.08	0.05	0.03	0.04	0.04	50%	0.01	0.01	0.01	0.01	0.01	0.01

punct	EN	PL	FR	IT	SP	Geral	space	EN	PL	FR	IT	SP	Geral
0%	0.81	0.83	0.81	0.84	0.85	0.83	0%	0.37	0.45	0.42	0.43	0.42	0.42
25%	0.15	0.14	0.17	0.13	0.12	0.14	25%	0.59	0.49	0.52	0.50	0.52	0.52
50%	0.03	0.02	0.02	0.02	0.02	0.02	50%	0.05	0.06	0.06	0.06	0.06	0.06

Fronteiras	EN	PL	FR	IT	SP	Geral
WS	0.15	0.15	0.13	0.13	0.13	0.14
SW	0.13	0.12	0.13	0.12	0.12	0.12
WSW	0.35	0.27	0.32	0.30	0.31	0.31

Fonte: José Eleandro Custódio, 2020

7 Conclusões

Esta dissertação explorou diversos aspectos da atribuição autoral. Esse tema é cada vez mais importante do ponto de vista forense, devido ao recente aumento das *fake news*, da cópia de conteúdo com autoria protegida e outros mal usos da internet. Por esse motivo, esse trabalho explorou mais de um domínio de forma a ampliar o conhecimento da AA.

A principal contribuição do trabalho foi a utilização de múltiplos modelos com diferentes extrações de características de tal forma a explorar diversas fontes de informação. A utilização do comitê do tipo classificador de classificadores foi inédito na literatura sobre AA encontrada nos mecanismos de buscas conhecidos, e os resultados do comitê foram superiores ao métodos individuais.

7.1 Sobre as hipóteses de pesquisa

Esse trabalho considerou as seguintes hipóteses de pesquisa:

- H1: A combinação de mais de um tipo de extração de característica aumentará o desempenho de sistema.
- H2: O uso de modelos dependentes de idioma do tipo *part-of-speech* aumentará o desempenho de sistemas de AA.
- H3: O uso de modelos dependentes de idioma do tipo representação distribuída (*embeddings*) aumentará o desempenho de sistemas de AA.

A hipótese *H1* foi confirmada, indicando que o estilo de escrita está presente em diversos níveis linguísticos, e que combinar métodos que explorem esses níveis resultou em um desempenho superior a utilizar um método em separado.

A hipótese *H2* se confirmou em apenas alguns casos. Na maioria dos casos, o modelo de n-gramas de caracteres teve desempenho superior e suficiente.

A hipótese *H3* também se confirmou em apenas alguns casos. O corpus Lyrics tirou mais proveito do conteúdo semântico que os demais corpus.

7.2 *As variações do problema de AA*

As abordagens estudadas nesse trabalho foram verificação, atribuição de conjunto fechado e de conjunto aberto. A AA de conjunto fechado foi o objetivo principal do trabalho, e a verificação e AA de conjunto aberto foram estudadas de forma exploratória.

Na verificação autoral, foi possível ver que a escolha da função de similaridade é tão importante quanto da extração de características. Na AA de conjunto aberto, foi possível ver que os classificadores tradicionais tem a tendência de escolher uma classe, e mesmo os métodos puros de conjunto aberto (ex. *one-class svm* e detecção de anomalias), explorados em etapas iniciais de desenvolvimento do modelo, tiveram desempenho pior com o aumento do número documentos de autorias desconhecidas.

Os experimentos de AA de conjunto fechado demonstraram que os principais limites são o número de autores, a quantidade documentos e a quantidade de caracteres.

7.3 *Os diferentes domínios*

O método proposto foi aplicado em cenários multidomínio e com textos de diversas fontes. Foram exploradas apenas as informações contidas nos textos. No entanto, acreditamos que um sistema prático deve utilizar outros conhecimentos como fonética, grafo social, metadados como data e hora, entre outros.

7.4 *Os diferentes idiomas*

As características textuais analisadas são, em geral, independentes de idioma. Ou seja, é possível aplicar o mesmo modelo *corpus* em textos outros idiomas. Apesar do relativo sucesso, é possível notar que mesmo o modelo de caracteres possui um desempenho variável entre os conjuntos de dados.

Foi possível notar que certos idiomas possuem uma maior variabilidade morfológica que outros, e que alguns tiraram proveito de *n*-gramas de tamanho variável indicando que subsequências são importantes.

Devido ao viés computacional deste trabalho, não foram explorados métodos puramente linguísticos e métodos que usam conhecimentos externos da identidade de

cada autor, como por exemplo, dados demográficos. Os métodos estudados se restringiram àqueles que pudessem usar algum método automatizado.

7.5 *As técnicas de aprendizado de máquina*

Durante o processo exploratório, foram analisados diversos métodos como árvore, *boosting*, redes de convolução, LSTMs e outros foram avaliados. Todavia, o melhor resultado veio da utilização do PCA com regressão logística multinomial ou regressão logística com regularização. Outras tentativas que foram abandonadas foram o treinamento dos *embeddings* com o próprio cópua e treinamento de *embeddings* de caracteres.

Dado esse cenário altamente restritivo, o comitê surgiu como uma solução natural para exploração dos métodos base. Em todos os cenários, o fator que mais impactou na escolha dos modelos estudados foi a escassez de dados para exploração confiável dos hiper-parâmetros e para exploração de aprendizado profundo.

7.6 *Trabalhos futuros*

Esse trabalho se restringiu a conjuntos de dados com até 50 autores e as técnicas mencionadas. Ficam como trabalhos futuros as explorações de conjunto maior número de autores e estudos que utilizem outras técnicas de extração de características e aprendizado de máquina.

Entre as extrações de características, imaginamos que os avanços recente obtidos com os *embeddings* contextuais (BERT (DEVLIN et al., 2019), ELMo (PETERS et al., 2018)) possam, enfim, demonstrar que o aprendizado profundo é útil para AA.

Fica como trabalho futuro a exploração de outros tipos de classificadores não lineares, em substituição do comitê construído com regressão logística softmax como classificador de classificadores.

Referências¹

- ADORNO, H. G. et al. Document embeddings learned on various types of n-grams for cross-topic authorship attribution. *Computing*, Springer-Verlag Wien, p. 1–16, 2018. ISSN 0010485X. Citado 2 vezes nas páginas 48 e 60.
- AFROZ, S. et al. Doppelgänger finder: Taking stylometry to the underground. In: *2014 IEEE Symposium on Security and Privacy, SP 2014, Berkeley, CA, USA, May 18-21, 2014*. [S.l.]: IEEE Computer Society, 2014. p. 212–226. Citado na página 23.
- ALSULAMI, B. et al. Source code authorship attribution using long short-term memory based networks. In: *Computer Security - ESORICS 2017 - 22nd European Symposium on Research in Computer Security, Oslo, Norway, September 11-15, 2017, Proceedings, Part I*. [s.n.], 2017. p. 65–82. Disponível em: <https://doi.org/10.1007/978-3-319-66402-6_6>. Citado 2 vezes nas páginas 18 e 24.
- ARGAMON, S.; JUOLA, P. Overview of the international authorship identification competition at PAN-2011. *CEUR Workshop Proceedings*, v. 1177, 2011. ISSN 16130073. Citado na página 42.
- BAGNALL, D. Author identification using multi-headed recurrent neural networks. In: Jones G.J.F. Cappellato L., F. N. S. J. E. (Ed.). *CEUR Workshop Proceedings*. [S.l.]: CEUR-WS, 2015. v. 1391, p. 1–9. ISSN 16130073. Citado 3 vezes nas páginas 55, 60 e 76.
- BAGNALL, D. Authorship clustering using multi-headed recurrent neural networks. In: Cappellato L. Ferro N., M. C. B. K. (Ed.). *CEUR Workshop Proceedings*. [S.l.]: CEUR-WS, 2016. v. 1609, p. 791–804. ISSN 16130073. Citado 4 vezes nas páginas 18, 55, 56 e 60.
- BENGIO, Y. et al. A Neural Probabilistic Language Model. *Journal of Machine Learning Research*, v. 3, p. 1137–1155, 2003. Citado 2 vezes nas páginas 19 e 28.
- BLEI, D. M.; NG, A. Y.; JORDAN, M. I. Latent dirichlet allocation. *Journal of machine Learning research*, v. 3, n. Jan, p. 993–1022, 2003. Citado na página 28.
- BOJANOWSKI, P. et al. Enriching word vectors with subword information. *Transactions of the Association for Computational Linguistics - TACL*, v. 5, p. 135–146, 2017. Disponível em: <<https://transacl.org/ojs/index.php/tacl/article/view/999>>. Citado na página 28.
- BOJANOWSKI, P. et al. Enriching word vectors with subword information. *Transactions of the Association for Computational Linguistics*, v. 5, p. 135–146, 2017. ISSN 2307-387X. Citado 2 vezes nas páginas 28 e 48.
- BROCARD, M. L. et al. Authorship verification using deep belief network systems. *International Journal of Communication Systems*, v. 30, n. 12, 2017. ISSN 10991131. Citado na página 56.
- BURROWS, J. 'Delta': a measure of stylistic difference and a guide to likely authorship. *Literary and Linguistic Computing*, v. 17, n. 3, p. 267–287, 2002. ISSN 0268-1145. Citado 2 vezes nas páginas 33 e 49.

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

- CHAWLA, N. V. et al. SMOTE: synthetic minority over-sampling technique. *J. Artif. Intell. Res.*, v. 16, p. 321–357, 2002. Citado na página 22.
- CHEN, X. et al. Authorship similarity detection from email messages. In: *Machine Learning and Data Mining in Pattern Recognition - 7th International Conference, MLDM 2011, New York, NY, USA, August 30 - September 3, 2011. Proceedings*. [S.l.: s.n.], 2011. p. 375–386. Citado na página 18.
- CHO, K. et al. Learning phrase representations using RNN encoder-decoder for statistical machine translation. In: MOSCHITTI, A.; PANG, B.; DAELEMANS, W. (Ed.). *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, EMNLP 2014, October 25-29, 2014, Doha, Qatar, A meeting of SIGDAT, a Special Interest Group of the ACL*. [S.l.]: ACL, 2014. p. 1724–1734. Citado na página 40.
- CHRISTENSEN, R. *Log-linear models and logistic regression*. [S.l.]: Springer Science & Business Media, 2006. Citado 2 vezes nas páginas 36 e 37.
- COLLOBERT, R. et al. Natural language processing (almost) from scratch. *The Journal of Machine Learning Research*, JMLR.org, v. 12, p. 2493–2537, nov. 2011. ISSN 1532-4435. Citado 3 vezes nas páginas 39, 52 e 53.
- CUSTÓDIO, J.; PARABONI, I. Multi-channel Open-set Cross-domain Authorship Attribution. In: CAPPELLATO, L. et al. (Ed.). *CLEF 2019 Labs and Workshops, Notebook Papers*. CEUR-WS.org, 2019. Disponível em: <<http://ceur-ws.org/Vol-2380/>>. Citado 6 vezes nas páginas 74, 75, 77, 78, 79 e 80.
- CUSTÓDIO, J. E.; PARABONI, I. EACH-USP Ensemble Cross-domain Authorship Attribution: Notebook for PAN at CLEF 2018. In: CAPPELLATO, L. et al. (Ed.). *Working Notes Papers of the CLEF 2018 Evaluation Labs*. [S.l.]: CLEF and CEUR-WS.org, 2018. (CEUR Workshop Proceedings). ISSN 1613-0073. Citado 15 vezes nas páginas 67, 68, 69, 70, 71, 72, 73, 75, 76, 77, 79, 80, 81, 117 e 118.
- CUSTÓDIO, J. E.; PARABONI, I. Similaridade de Textos aplicada à Verificação Autoral. In: *1st International Congress on Digital Humanities in Rio de Janeiro*. [S.l.]: Fundação Getúlio Vargas, 2018. Citado 4 vezes nas páginas 61, 63, 64 e 65.
- CUSTÓDIO, J. E.; PARABONI, I. An ensemble approach to cross-domain authorship attribution. In: CRESTANI, F. et al. (Ed.). *Experimental IR Meets Multilinguality, Multimodality, and Interaction*. Cham: Springer International Publishing, 2019. p. 201–212. ISBN 978-3-030-28577-7. Citado 2 vezes nas páginas 89 e 90.
- DEERWESTER, S. et al. Indexing by latent semantic analysis. *Journal of the American society for information science*, Wiley Online Library, v. 41, n. 6, p. 391–407, 1990. Citado na página 28.
- DEVLIN, J. et al. BERT: pre-training of deep bidirectional transformers for language understanding. In: *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.: s.n.], 2019. p. 4171–4186. Citado na página 105.
- DEZA, M. M.; DEZA, E. *Encyclopedia of distances*. [S.l.]: Springer, 2009. 1–583 p. Citado 2 vezes nas páginas 31 e 32.

ELMAN, J. L. Finding structure in time. *Cognitive science*, Wiley Online Library, v. 14, n. 2, p. 179–211, 1990. Citado na página 39.

EVERT, S. et al. Understanding and explaining delta measures for authorship attribution. *Digital Scholarship in the Humanities*, v. 32, n. suppl_2, p. ii4–ii16, 2017. Citado 4 vezes nas páginas 18, 50, 58 e 60.

EVERT, S. et al. Towards a better understanding of burrows’s delta in literary authorship attribution. In: *Proceedings of the Fourth Workshop on Computational Linguistics for Literature, CLfL@NAACL-HLT 2015, June 4, 2015, Denver, Colorado, USA*. [S.l.: s.n.], 2015. p. 79–88. Citado 2 vezes nas páginas 33 e 34.

FINKEL, J. R.; GRENAGER, T.; MANNING, C. D. Incorporating non-local information into information extraction systems by gibbs sampling. In: *ACL 2005, 43rd Annual Meeting of the Association for Computational Linguistics, Proceedings of the Conference, 25-30 June 2005, University of Michigan, USA*. [S.l.: s.n.], 2005. p. 363–370. Citado 2 vezes nas páginas 44 e 54.

FRANTZESKOU, G. et al. Source code author identification based on N-gram author profiles. *IFIP International Federation for Information Processing*, v. 204, p. 508–515, 2006. ISSN 15715736. Citado 2 vezes nas páginas 33 e 34.

GHAEGINI, M. R. Intrinsic author identification using modified weighted KNN notebook for PAN at CLEF 2013. In: FORNER, P. et al. (Ed.). *Working Notes for CLEF 2013 Conference, Valencia, Spain, September 23-26, 2013*. [S.l.]: CEUR-WS.org, 2013. (CEUR Workshop Proceedings, v. 1179). Citado na página 23.

GOLDBERG, Y. *Neural Network Methods in Natural Language Processing*. [S.l.]: Morgan & Claypool Publishers, 2017. Citado 4 vezes nas páginas 26, 40, 75 e 76.

GOLDBERG, Y.; LEVY, O. word2vec Explained: deriving Mikolov et al.’s negative-sampling word-embedding method. *CoRR*, abs/1402.3722, 2014. Disponível em: <<http://arxiv.org/abs/1402.3722>>. Citado na página 30.

GOLLUB, T.; BURROWS, S.; STEIN, B. First Experiences with TIRA for Reproducible Evaluation in Information Retrieval. *SIGIR 12 Workshop on Open Source Information Retrieval (OSIR 12)*, p. 52–55, 2012. Citado na página 78.

GOODFELLOW, I.; BENGIO, Y.; COURVILLE, A. *Deep Learning*. [S.l.]: MIT Press, 2016. <http://www.deeplearningbook.org>. Citado 3 vezes nas páginas 38, 39 e 40.

GROSSMAN, D. A.; FRIEDER, O. *Information retrieval: Algorithms and heuristics*. [S.l.]: Springer Science & Business Media, 2012. v. 15. Citado na página 32.

HACOHEN-KERNER, Y. et al. Cross-domain Authorship Attribution: Author Identification using Char Sequences, Word Uni-grams, and POS-tags Features: Notebook for PAN at CLEF 2018. In: CAPPELLATO, L. et al. (Ed.). *Working Notes Papers of the CLEF 2018 Evaluation Labs*. [S.l.]: CLEF and CEUR-WS.org, 2018. (CEUR Workshop Proceedings). ISSN 1613-0073. Citado 2 vezes nas páginas 72 e 73.

HALVANI, O.; GRANER, L. Cross-Domain Authorship Attribution Based on Compression: Notebook for PAN at CLEF 2018. In: CAPPELLATO, L. et al. (Ed.). *Working Notes Papers of the CLEF 2018 Evaluation Labs*. [S.l.]: CLEF and CEUR-WS.org,

2018. (CEUR Workshop Proceedings). ISSN 1613-0073. Citado 3 vezes nas páginas 18, 72 e 73.

HALVANI, O.; WINTER, C.; GRANER, L. On the usefulness of compression models for authorship verification. In: *Proceedings of the 12th International Conference on Availability, Reliability and Security, Reggio Calabria, Italy, August 29 - September 01, 2017*. [S.l.: s.n.], 2017. p. 54:1–54:10. Citado na página 50.

HARRIS, Z. S. Distributional structure. *Word*, Taylor & Francis, v. 10, n. 2-3, p. 146–162, 1954. Citado na página 27.

HITSCHLER, J.; BERG, E. V. D.; REHBEIN, I. Authorship attribution with convolutional neural networks and pos-eliding. In: *Proceedings of the Workshop on Stylistic Variation*. Association for Computational Linguistics, 2017. p. 53–58. Disponível em: <<http://aclweb.org/anthology/W17-4907>>. Citado 2 vezes nas páginas 54 e 60.

HOCHREITER, S.; SCHMIDHUBER, J. Long short-term memory. *Neural Computation*, v. 9, n. 8, p. 1735–1780, 1997. Citado na página 40.

Hossain, R.; Al Marouf, A. Banglamusicstylo: A stylometric dataset of bangla music lyrics. In: *2018 International Conference on Bangla Speech and Language Processing (ICBSLP)*. [S.l.: s.n.], 2018. p. 1–5. Citado na página 89.

JOULIN, A. et al. Bag of tricks for efficient text classification. In: *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics, EACL 2017, Valencia, Spain, April 3-7, 2017, Volume 2: Short Papers*. [S.l.: s.n.], 2017. p. 427–431. Citado na página 28.

JR, D. H.; LEMESHOW, S.; STURDIVANT, R. X. *Applied logistic regression*. [S.l.]: John Wiley & Sons, 2013. v. 398. Citado 2 vezes nas páginas 36 e 86.

JUOLA, P. An Overview of the Traditional Authorship Attribution Subtask Notebook for PAN at CLEF 2012. In: *Working Notes for CLEF 2013 Conference*. [S.l.: s.n.], 2012. p. 37–41. ISBN 978-88-904810-3-1. ISSN 2038-4963. Citado na página 42.

JUOLA, P. The rowling case: A proposed standard analytic protocol for authorship questions. *Digital Scholarship in the Humanitie*, v. 30, n. Suppl-1, p. i100–i113, 2015. Citado na página 18.

JUOLA, P.; STAMATATOS, E. Overview of the author identification task at PAN 2013. In: *Working Notes for CLEF 2013 Conference, Valencia, Spain, September 23-26, 2013*. [S.l.: s.n.], 2013. Citado 3 vezes nas páginas 18, 42 e 50.

KEŠELJ, V. et al. N-Gram-Based Author Profiles for Authorship Attribution. In: *Proceedings of the conference pacific association for computational linguistics (PACLING)*. [S.l.: s.n.], 2003. v. 3, p. 255–264. Citado 5 vezes nas páginas 21, 24, 33, 34 e 58.

KESTEMONT, M. Function Words in Authorship Attribution From Black Magic to Theory ? *3rd Workshop on Computational Linguistics for Literature (CLfL 2014)*, n. January 2014, p. 59–66, 2014. Citado na página 24.

KESTEMONT, M. et al. Cross-genre authorship verification using unmasking. *English Studies*, Taylor & Francis, v. 93, n. 3, p. 340–356, 2012. Citado na página 22.

KESTEMONT, M. et al. Overview of the Cross-domain Authorship Attribution Task at PAN 2019. In: CAPPELLATO, L. et al. (Ed.). *CLEF 2019 Labs and Workshops, Notebook Papers*. [S.l.]: CEUR-WS.org, 2019. Citado 5 vezes nas páginas 20, 23, 42, 73 e 83.

KESTEMONT, M. et al. Overview of the Author Identification Task at PAN-2018: Cross-domain Authorship Attribution and Style Change Detection. In: CAPPELLATO, L. et al. (Ed.). *Working Notes Papers of the CLEF 2018 Evaluation Labs*. [S.l.]: CLEF and CEUR-WS.org, 2018. (CEUR Workshop Proceedings). ISSN 1613-0073. Citado 9 vezes nas páginas 18, 20, 22, 23, 42, 66, 72, 73 e 83.

KHONJI, M.; IRAQI, Y. A slightly-modified gi-based author-verifier with lots of features (ASGALF). In: *Working Notes for CLEF 2014 Conference, Sheffield, UK, September 15-18, 2014*. [S.l.: s.n.], 2014. p. 977–983. Citado na página 50.

KIM, Y. Convolutional Neural Networks for Sentence Classification. In: Alessandro Moschitti and Bo Pang and Walter Daelemans (Ed.). *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, EMNLP 2014, October 25-29, 2014, Doha, Qatar, A meeting of SIGDAT, a Special Interest Group of the ACL*. [S.l.: s.n.], 2014. p. 1746–1751. Citado 3 vezes nas páginas 39, 52 e 55.

KINGMA, D. P.; BA, J. Adam: A method for stochastic optimization. In: *International Conference on Learning Representations - ICLR 2015*. Ithaca, NY: arXiv.org, 2015. p. 13. Disponível em: <<http://arxiv.org/abs/1412.6980>>. Citado na página 54.

KJELL, B.; WOODS, W. A.; FRIEDER, O. Discrimination of authorship using visualization. *Inf. Process. Manage.*, v. 30, n. 1, p. 141–150, 1994. Citado na página 24.

KLIMT, B.; YANG, Y. The enron corpus: A new dataset for email classification research. In: BOULICAUT, J. et al. (Ed.). *Machine Learning: ECML 2004, 15th European Conference on Machine Learning, Pisa, Italy, September 20-24, 2004, Proceedings*. [S.l.]: Springer, 2004. (Lecture Notes in Computer Science, v. 3201), p. 217–226. Citado na página 18.

KOCHER, M.; SAVOY, J. A simple and efficient algorithm for authorship verification. *Journal of the Association for Information Science and Technology*, v. 68, n. 1, p. 259–269, 2017. Citado 2 vezes nas páginas 18 e 52.

KOPPEL, M.; SCHLER, J. Authorship verification as a one-class classification problem. *Twenty-first international conference on Machine learning - ICML '04*, p. 62, 2004. Citado na página 52.

KOPPEL, M.; SCHLER, J.; ARGAMON, S. Computational Methods in Authorship Attribution. *Journal of the Association for Information Science and Technology*, v. 60, n. 1, p. 9—26, 2009. ISSN 19335954. Citado 2 vezes nas páginas 22 e 25.

KOPPEL, M. et al. The "Fundamental Problem" of Authorship Attribution. *English Studies*, v. 93, n. 3, p. 284–291, 2012. ISSN 0013838X. Citado 2 vezes nas páginas 23 e 30.

KOPPEL, M.; SEIDMAN, S. Detecting pseudepigraphic texts using novel similarity measures. *Digital Scholarship in the Humanities*, v. 33, n. 1, p. 72–81, 2018. Citado 5 vezes nas páginas 18, 51, 52, 60 e 74.

KOPPEL, M.; WINTER, Y. Determining if two documents are written by the same author. *Journal of the American Society for Information Science and Technology*, v. 65, n. 1, p. 178–187, 2013. ISSN 15322882. Citado 3 vezes nas páginas 50, 51 e 52.

KULLBACK, S.; LEIBLER, R. A. On information and sufficiency. *The annals of mathematical statistics*, JSTOR, v. 22, n. 1, p. 79–86, 1951. Citado na página 32.

Kırmacı, B.; Oğul, H. Evaluating text features for lyrics-based songwriter prediction. In: *2015 IEEE 19th International Conference on Intelligent Engineering Systems (INES)*. [S.l.: s.n.], 2015. p. 405–409. Citado na página 89.

LABBÉ, D. Experiments on authorship attribution by intertextual distance in english. *Journal of Quantitative Linguistics*, Taylor & Francis, v. 14, n. 1, p. 33–80, 2007. Citado na página 49.

LE, Q. V.; MIKOLOV, T. Distributed representations of sentences and documents. In: *Proceedings of the 31th International Conference on Machine Learning, ICML 2014, Beijing, China, 21-26 June 2014*. [S.l.: s.n.], 2014. p. 1188–1196. Citado 3 vezes nas páginas 28, 30 e 31.

LECUN, Y. et al. Backpropagation Applied to Handwritten Zip Code Recognition. *Neural Computation*, v. 1, n. 4, p. 541–551, 1989. Disponível em: <<https://doi.org/10.1162/neco.1989.1.4.541>>. Citado na página 38.

LECUN, Y. et al. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, IEEE, v. 86, n. 11, p. 2278–2324, 1998. Citado na página 39.

LEWIS, D. D. et al. RCV1: A new benchmark collection for text categorization research. *Journal of Machine Learning Research*, v. 5, p. 361–397, dez. 2004. ISSN 1532-4435. Citado na página 43.

MANNING, C.; SCHÜTZE, H. *Foundations of statistical natural language processing*. [S.l.]: MIT press, 1999. Citado 2 vezes nas páginas 33 e 36.

MANNING, C. D.; RAGHAVAN, P.; SCHÜTZE, H. *Introduction to information retrieval*. [S.l.]: Cambridge University Press, 2008. ISBN 978-0-521-86571-5. Citado na página 42.

MARKOV, I.; BAPTISTA, J.; LAGUNAS, O. P. Authorship attribution in portuguese using character N-grams. *Acta Polytechnica Hungarica*, Budapest Tech Polytechnical Institution, v. 14, n. 3, p. 59–78, 2017. ISSN 17858860. Citado 2 vezes nas páginas 57 e 60.

MARKOV, I.; STAMATATOS, E.; SIDOROV, G. Improving Cross-Topic Authorship Attribution: The Role of Pre-Processing. In: *Proceedings of the 18th International Conference on Computational Linguistics and Intelligent Text Processing (CICLing 2017)*. [S.l.: s.n.], 2017. Citado 3 vezes nas páginas 44, 59 e 60.

MCCORNACK, R. L. A comparison of three predictor selection techniques in multiple regression. *Psychometrika*, v. 35, n. 2, p. 257–271, Jun 1970. ISSN 1860-0980. Citado na página 86.

MIKOLOV, T. *Statistical language models based on neural networks*. Tese (Doutorado) — Brno University of Technology, 2012. Disponível em: <<http://www.fit.vutbr.cz/~imikolov/rnnlm/thesis.pdf>>. Citado na página 39.

MIKOLOV, T. et al. Recurrent neural network based language model. In: *INTERSPEECH 11th Annual Conference of the International Speech Communication Association, Japan*. [S.l.: s.n.], 2010. p. 1045–1048. Citado 2 vezes nas páginas 52 e 55.

MIKOLOV, T. et al. Distributed Representations of Words and Phrases and their Compositionality. In: *Advances in Neural Information Processing Systems 26: 27th Annual Conference on Neural Information Processing Systems 2013. Proceedings of a meeting held December 5-8, 2013, Lake Tahoe, Nevada, United States*. [S.l.: s.n.], 2013. p. 3111–3119. Citado 2 vezes nas páginas 28 e 29.

MOHSEN, A. M.; EL-MAKKY, N. M.; GHANEM, N. Author Identification using Deep Learning. In: *Proceedings - 2016 15th IEEE International Conference on Machine Learning and Applications, ICMLA 2016*. [S.l.]: Institute of Electrical and Electronics Engineers Inc., 2016. p. 898–903. ISBN 9781509061679. Citado na página 56.

MOSTELLER, F.; WALLACE, D. L. *Inference in an Authorship Problem*. 1963. 275–309 p. Citado 3 vezes nas páginas 25, 49 e 58.

MURAUER, B.; TSCHUGGNALL, M.; SPECHT, G. Dynamic Parameter Search for Cross-Domain Authorship Attribution: Notebook for PAN at CLEF 2018. In: CAPPELLATO, L. et al. (Ed.). *Working Notes Papers of the CLEF 2018 Evaluation Labs*. [S.l.]: CLEF and CEUR-WS.org, 2018. (CEUR Workshop Proceedings). ISSN 1613-0073. Citado 2 vezes nas páginas 72 e 73.

NEAL, T. J. et al. Surveying stylometry techniques and applications. *ACM Comput. Surv.*, v. 50, n. 6, p. 86:1–86:36, 2017. Citado 4 vezes nas páginas 21, 23, 24 e 25.

NIELSEN, M. A. *Neural networks and deep learning*. [S.l.]: Determination Press, 2015. Citado 2 vezes nas páginas 38 e 40.

PAGE, L. et al. *The PageRank Citation Ranking: Bringing Order to the Web*. [S.l.], 1999. Previous number = SIDL-WP-1999-0120. Disponível em: <<http://ilpubs.stanford.edu:8090/422/>>. Citado na página 76.

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 na página 40.

PEDREGOSA, F. et al. Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research*, v. 12, p. 2825–2830, 2011. Citado 3 vezes nas páginas 27, 67 e 69.

PENG, J.; CHOO, K. kwang R.; ASHMAN, H. Astroturfing detection in social media: Using binary n-gram analysis for authorship attribution. In: *2016 IEEE Trustcom/BigDataSE/ISPA*. [S.l.: s.n.], 2016. p. 121–128. ISBN 9781509032051. Citado na página 18.

PENNINGTON, J.; SOCHER, R.; MANNING, C. D. Glove: Global vectors for word representation. In: MOSCHITTI, A.; PANG, B.; DAELEMANS, W. (Ed.). *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, EMNLP 2014, October 25-29, 2014, Doha, Qatar, A meeting of SIGDAT, a Special Interest Group of the ACL*. [S.l.]: ACL, 2014. p. 1532–1543. Citado na página 28.

PETERS, M. E. et al. Deep contextualized word representations. In: *Proc. of NAACL*. [S.l.: s.n.], 2018. Citado na página 105.

POSADAS-DURÁN, J.-P. et al. Applications of the distributed document representation in the authorship attribution task for small corpora. *Soft Computing*, Springer Verlag, v. 21, n. 3, p. 627–639, feb 2017. ISSN 14327643. Citado 2 vezes nas páginas 47 e 60.

POTHA, N.; STAMATATOS, E. A profile-based method for authorship verification. In: *Artificial Intelligence: Methods and Applications - 8th Hellenic Conference on AI, SETN 2014, Ioannina, Greece, 15-17*. [S.l.: s.n.], 2014. p. 313–326. Citado na página 34.

POTHA, N.; STAMATATOS, E. An improved impostors method for authorship verification. In: *Experimental IR Meets Multilinguality, Multimodality, and Interaction - 8th International Conference of the CLEF Association, CLEF 2017, Dublin, Ireland, September 11-14*. [S.l.: s.n.], 2017. p. 138–144. Citado 2 vezes nas páginas 50 e 58.

POTTHAST, M. et al. Overview of the 2nd International Competition on Plagiarism Detection. In: *Working Notes Papers of the CLEF 2010 Evaluation Labs*. [S.l.: s.n.], 2010. ISBN 978-88-904810-2-4. ISSN 2038-4963. Citado na página 42.

POTTHAST, M. et al. Who wrote the web? revisiting influential author identification research applicable to information retrieval. In: FERRO, N. et al. (Ed.). *Advances in Information Retrieval*. Cham: Springer International Publishing, 2016. p. 393–407. ISBN 978-3-319-30671-1. Citado na página 74.

POTTHAST, M. et al. Overview of PAN'17: Author identification, author profiling, and author obfuscation. *Lecture Notes in Computer Science*, v. 10456 LNCS, p. 275–290, 2017. ISSN 16113349. Citado 2 vezes nas páginas 18 e 21.

POTTHAST, M. et al. Overview of the 1st International Competition on Plagiarism Detection. *Proceedings of the SEPLN'09 Workshop on Uncovering Plagiarism, Authorship and Social Software Misuse (PAN 09)*, p. 1–9, 2009. ISSN 1613-0073. Citado na página 42.

QIAN, C.; HE, T.; ZHANG, R. Deep Learning based Authorship Identification. *Stanford University - CS224D Projects*, p. 1–9, Jan 2016. Citado na página 56.

RHODES, D. Author Attribution with CNN's. *Stanford University - CS224D Projects*, p. 1–8, 2015. Citado 3 vezes nas páginas 53, 54 e 60.

ROCHA, A. et al. Authorship Attribution for Social Media Forensics. *IEEE Transactions on Information Forensics and Security*, v. 12, n. 1, p. 5–33, 2017. ISSN 15566013. Citado 10 vezes nas páginas 20, 23, 25, 34, 45, 55, 60, 74, 90 e 91.

RODRÍGUEZ, C. M. D. C. et al. CIC-GIL Approach to Cross-domain Authorship Attribution: Notebook for PAN at CLEF 2018. In: *CEUR Workshop Proceedings*. [S.l.: s.n.], 2018. Citado 2 vezes nas páginas 72 e 73.

ROSENBLATT, F. Principles of neurodynamics. perceptrons and the theory of brain mechanisms. Spartan Book, 1962. Citado na página 36.

ROSSO, P. et al. Overview of PAN'16 - New Challenges for Authorship Analysis: Cross-Genre Profiling, Clustering, Diarization, and Obfuscation. In: *Experimental IR Meets Multilinguality, Multimodality, and Interaction - 7th International Conference of the CLEF Association, CLEF 2016, Évora, Portugal, September 5-8, 2016, Proceedings*. [S.l.: s.n.], 2016. p. 332–350. Citado 2 vezes nas páginas 23 e 42.

RUMELHART, D. E.; HINTON, G.; WILLIAMS, R. J. Learning Representations by Back-propagating Errors. In: ANDERSON, J. A.; ROSENFELD, E. (Ed.). *Neurocomputing: Foundations of Research*. Cambridge, MA, USA: MIT Press, 1988. p. 696–699. ISBN 0-262-01097-6. Citado na página 28.

SALTON, G.; WONG, A.; YANG, C. A vector space model for automatic indexing. *Association for computational machinery*, v. 18, n. 11, p. 613–620, 1975. Citado 2 vezes nas páginas 26 e 27.

SAPKOTA, U. et al. Not all character n-grams are created equal: A study in authorship attribution. In: *NAACL HLT 2015, The 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Denver, Colorado, USA*. [S.l.: s.n.], 2015. p. 93–102. Citado 6 vezes nas páginas 43, 44, 57, 59, 60 e 99.

SAPKOTA, U. et al. The use of orthogonal similarity relations in the prediction of authorship. In: *Computational Linguistics and Intelligent Text Processing - 14th International Conference, CICLing 2013, Samos, Greece, March 24-30, 2013, Proceedings, Part II*. [S.l.: s.n.], 2013. v. 7817 LNCS, n. PART 2, p. 463–475. Citado na página 45.

SAPKOTA, U. et al. Cross-Topic Authorship Attribution: Will Out-Of-Topic Data Help? In: *Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers*. [S.l.: s.n.], 2014. p. 1228–1237. ISBN 9781941643266. Citado 2 vezes nas páginas 22 e 24.

SARI, Y.; STEVENSON, M. Exploring Word Embeddings and Character N -Grams for Author Clustering Notebook for PAN at CLEF 2016. *CEUR Workshop Proceedings*, 2016. ISSN 16130073. Citado 5 vezes nas páginas 18, 24, 28, 47 e 60.

SARI, Y.; VLACHOS, A.; STEVENSON, M. Continuous N-gram Representations for Authorship Attribution. In: *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics, EACL 2017, Valencia, Spain, April 3-7, 2017, Volume 2: Short Papers*. [S.l.]: Association for Computational Linguistics (ACL), 2017. v. 2, p. 267–273. Citado 2 vezes nas páginas 48 e 60.

SAVOY, J. Authorship Attribution Based on Specific Vocabulary. *ACM Transactions on Information Systems*, v. 30, n. 2, p. 1–30, 2012. ISSN 10468188. Citado na página 25.

SAVOY, J. Estimating the probability of an authorship attribution. *Journal of the Association for Information Science and Technology*, v. 67, n. 6, p. 1462–1472, 2016. ISSN 23301643. Citado 4 vezes nas páginas 18, 34, 49 e 60.

SCHULT, D. A. Exploring network structure, dynamics, and function using networkx. In: *In Proceedings of the 7th Python in Science Conference (SciPy)*. [S.l.: s.n.], 2008. p. 11–15. Citado na página 76.

SCHWARTZ, R. et al. Authorship Attribution of Micro-Messages. In: *Empirical Methods in Natural Language Processing*. [S.l.: s.n.], 2013. p. 1880–1891. ISBN 9781937284978. Citado 5 vezes nas páginas 18, 22, 45, 53 e 60.

SEIDMAN, S. Authorship verification using the impostors method: Notebook for PAN at CLEF 2013. *CEUR Workshop Proceedings*, v. 1179, p. 13–16, 2013. ISSN 16130073. Citado 2 vezes nas páginas 50 e 64.

SHRESTHA, P. et al. Convolutional Neural Networks for Authorship Attribution of Short Texts. In: *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics*. [S.l.]: Association for Computational Linguistics (ACL), 2017. v. 2, p. 669–674. ISBN 9781510838604. Citado 8 vezes nas páginas 18, 21, 22, 24, 53, 59, 60 e 76.

SRIVASTAVA, N. et al. Dropout: a simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, v. 15, n. 1, p. 1929–1958, 2014. Citado na página 39.

STAMATATOS, E. A survey of modern authorship attribution methods. *Journal of the American Society for Information Science and Technology*, v. 60, n. 3, p. 538–556, 2009. ISSN 15322882. Citado 5 vezes nas páginas 18, 21, 30, 33 e 34.

STAMATATOS, E. On the robustness of authorship attribution based on character n-gram features. *Journal of Law and Policy*, v. 21, n. 2, p. 421–439, 2013. Citado na página 43.

STAMATATOS, E. Authorship attribution using text distortion. *Conference of the European Chapter of the Association for Computational Linguistics, EACL 2017 - Proceedings of Conference*, v. 1, 2017. Citado 6 vezes nas páginas 18, 24, 44, 59, 60 e 68.

STAMATATOS, E. et al. Overview of the author identification task at PAN 2014. In: *Working Notes for CLEF 2014 Conference, Sheffield, UK, September 15-18, 2014*. [S.l.: s.n.], 2014. p. 877–897. Citado 5 vezes nas páginas 20, 22, 42, 50 e 61.

STAMATATOS, E. et al. Overview of the PAN/CLEF 2015 evaluation lab. *Lecture Notes in Computer Science*, v. 9283, p. 518–538, 2015. ISSN 16113349. Citado na página 42.

TEAHAN, W. J.; HARPER, D. J. Using compression-based language models for text categorization. In: *Language modeling for information retrieval*. [S.l.]: Springer, 2003. p. 141–165. Citado 2 vezes nas páginas 46 e 74.

THOMAS, L.; CROOK, J.; EDELMAN, D. *Credit scoring and its applications*. [S.l.]: Siam, 2017. v. 2. Citado na página 35.

TSCHUGGNALL, M. et al. Overview of the author identification task at PAN-2017: style breach detection and author clustering. In: *Working Notes of CLEF*. [S.l.: s.n.], 2017. Citado 4 vezes nas páginas 18, 21, 23 e 42.

TURIAN, J. P.; RATINOV, L.; BENGIO, Y. Word representations: A simple and general method for semi-supervised learning. In: HAJIC, J.; CARBERRY, S.; CLARK, S. (Ed.). *ACL 2010, Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*. [S.l.]: The Association for Computer Linguistics, 2010. p. 384–394. Citado na página 28.

TURNEY, P. D.; PANTEL, P. From frequency to meaning: Vector space models of semantics. *Journal of artificial intelligence research*, abs/1003.1141, 2010. Citado 3 vezes nas páginas 22, 26 e 27.

VARELA, P. J. et al. A computational approach based on syntactic levels of language in authorship attribution. *IEEE Latin America Transactions*, v. 14, n. 1, p. 259–266, 2016. ISSN 15480992. Citado 6 vezes nas páginas 18, 19, 24, 55, 57 e 60.

VARTAPETIANCE, A.; GILLAM, L. Quite simple approaches for authorship attribution, intrinsic plagiarism detection and sexual predator identification. In: *CLEF 2012 Evaluation Labs and Workshop, Online Working Notes, Rome, Italy, September 17-20, 2012*. [S.l.: s.n.], 2012. Citado na página 18.

ZHANG, X.; ZHAO, J. J.; LECUN, Y. Character-level convolutional networks for text classification. In: *Advances in Neural Information Processing Systems 28: Annual Conference on Neural Information Processing Systems 2015, December 7-12, 2015, Montreal, Quebec, Canada*. [S.l.: s.n.], 2015. p. 649–657. Citado na página 85.

Apêndice A – Apêndice

A.1 Relevância das características para o experimento de AA multidomínio PAN2018

As tabelas 31, 32 e 33 apresentam as 10 características mais relevantes para AA extraídas do *Problem00002* da competição PAN2018, que compreende textos escritos por cinco autores usando língua inglesa. Nessa representação, espaços foram codificados para sobrescrito e a relevância representa os pesos da regressão logística multinomial. Esses pesos foram estimados através da padronização para média zero e desvio padrão 1.

Tabela 31 – Características textuais relevantes para *Std.charN*

Candidatos				
01	02	03	04	05
_as_l	_Sti	_sub	_joi	_day,
_'	_"Can	_suc	_gh	_dev
_prec	_"Ca	_I_fi	_er	_dete
_I'd	_"Be	_succ	_glow	_plac
_"Are	_K	_subs	_Is	_mut
_Re	_but_	_I_f	_sta	_must
_smel	_Ofte	_"T	_gor	_Dro
_leak	_posi	_a_t	_sorr	_day_
_is_s	_For	_"St	_eat_	_she_
_spu	_Ri	_a_sw	_If_t	_chi

Fonte: Adaptadas de Custódio e Paraboni (2018a)

Tabela 32 – Características textuais relevantes para *Dist.charN*

Candidatos				
01	02	03	04	05
* ‘**	__**_	"*’	*_~	*_ _ *
__**_	__**_ (	"* **	*_~	’*_*
*’	__**_ *	!)_*’	*_.*	"_
).	*!	*!!	’**	*_ _
),_	___ ’	*’*_*	’****	*_ _ _
*_ _ *	*!_*	**_*’	"_**’	’*.
*_ _	*_ “**	**_**	_É****	_ “*
_’**	_~	**_*’	_ “*’	_ -
!)_	_~*	_**!	_**..	_ - _

Fonte: Adaptadas de Custódio e Paraboni (2018a)

Tabela 33 – Características textuais relevantes para *Std.wordN*

Candidatos				
01	02	03	04	05
about_what	against_his	an_odd	although	and_pulled_him
and_practically	and_it_was	and_then_he	an_eye	and_pulling
any_of	and_so	acknowledged	and_said	across_his
any_more	and_already	and_he_had	and_takes	across_the
and_nearly	and_steve	are_your	and_just	and_all
and_pulled	and_say	again_to	ancient	against_her
agree	accent	and_tell	amount_of	among
all_tony	and_wet	and_forth	always	about_what_to
ah	apparently	are_just	and_grinned	acting
and_wet_and	after	and_grabbing	about_the	about_their

Fonte: Adaptadas de Custódio e Paraboni (2018a)