CEFET/RJ
DIRETORIA DE PESQUISA E PÓS-GRADUAÇÃO

INTELIGÊNCIA COMPUTACIONAL APLICADA À DETECÇÃO INTRÍNSECA DE PLÁGIO EM DOCUMENTOS TEXTUAIS

Ivair Nobrega Luques

Dissertação de Mestrado apresentada ao Programa de Pós-graduação em Ciência da Computação, Centro Federal de Educação Tecnológica Celso Suckow da Fonseca CEFET/RJ, como parte dos requisitos necessários à obtenção do título de mestre.

Orientador:
Eduardo Bezerra da Silva
Pedro Henrique González Silva

Rio de Janeiro,

24 de Março de 2020

Inteligência Computacional Aplicada à Detecção Intrínseca de Plágio em Documentos Textuais

Dissertação de Mestrado em Ciência da Computação, Centro Federal de Educação Tecnológica Celso Suckow da Fonseca, CEFET/RJ.

Ivair Nobrega Luques

Aprovada por:

Presidente, Prof. Eduardo Bezerra da Silva, D.Sc. (orientador)

Prof. Pedro Henrique González Silva, D.Sc. (coorientador)

Prof. Jorge de Abreu Soares, D.Sc. (CEFET/RJ)

Prof. Igor Machado Coelho, D.Sc. (UFF)

Rio de Janeiro,

24 de Março de 2020

DEDICATÓRIA

Dedico este trabalho a meus amores: Fátima,
Victor e Caio, que são minha inspiração,
motivação, apoio e alegria.

AGRADECIMENTOS

Agradeço a Deus, pela vida. A meus pais, pelos valores que me transmitiram e por me darem a oportunidade da Educação. Aos meus orientadores, pelo guiar atencioso, pelo conhecimento compartilhado e principalmente por não terem desistido de mim, ao se depararem com minhas inúmeras dificuldades. Aos demais professores e colegas de mestrado, sem os quais esta trajetória teria sido impossível. Aos amigos e familiares pelo suporte, encorajamento e por me proporcionarem momentos de alegria e diversão neste período de tanto trabalho.

RESUMO

Inteligência Computacional Aplicada à Detecção Intrínseca de Plágio em Documentos Textuais

Ivair Nobrega Luques

Orientadores:
Eduardo Bezerra da Silva
Pedro Henrique González Silva

Resumo da Dissertação submetida ao Programa de Pós-graduação em Ciência da Computação do Centro Federal de Educação Tecnológica Celso Suckow da Fonseca CEFET/RJ como parte dos requisitos necessários à obtenção do título de mestre.

O acesso à produção acadêmica na forma de documentos relacionados ao ensino e à pesquisa científica tem sido fomentado por movimentos de divulgação de documentos digitais. No entanto, o uso indevido desses documentos pode contribuir para o crescimento de casos de plágio. Redes neurais artificiais têm obtido os melhores resultados na solução de vários problemas de na área de Processamento de Linguagem Natural. Inspirados por isso, neste trabalho, aplicamos uma combinação simples, porém eficaz, de técnicas de Aprendizagem Profunda à tarefa de detecção intrínseca de plágio. Em particular, usamos Skip-Thoughts, um modelo de incorporação para representar cada frase de um documento como um vetor multidimensional. Depois disso, treinamos uma rede neural siamesa usando como conjunto de treinamento uma coleção de pares de frases (cada frase representada como um vetor Skip-Thoughts) extraída de documentos no corpus PAN11. Em seguida, modelamos cada documento como um grafo ponderado e não-dirigido para viabilizar a aplicação do algoritmo de correlação de clusters, que possibilita identificar passagens potencialmente plagiadas. Nossos experimentos computacionais mostram que o modelo neural de rede siamesa resultante é capaz de reconhecer diferenças estilísticas entre frases em um documento. Além disso, a identificação de passagens potencialmente plagiadas por meio da abordagem de correlação de clusters produz resultados comparáveis aos da literatura.

Palavras-chave:
Aprendizagem Profunda; Redes Neurais Artificiais; Agrupamento de dados, Detecção Intrínseca de Plágio.

Rio de Janeiro,

24 de Março de 2020

ABSTRACT

Inteligência Computacional Aplicada à Detecção Intrínseca de Plágio em Documentos Textuais

Ivair Nobrega Luques

Advisors:
Eduardo Bezerra da Silva
Pedro Henrique González Silva

Abstract of dissertation submitted to Programa de Pós-graduação em Ciência da Computação - Centro Federal de Educação Tecnológica Celso Suckow da Fonseca CEFET/RJ as partial fulfillment of the requirements for the degree of master.

Access to information has been fostered by movements of open access to knowledge through digital libraries, which make available large collections of textual documents. However, the misuse of these available documents is contributing to the growth of cases of plagiarism. Machine Learning has aided in detecting plagiarism in many kinds of textual documents, such as published thesis, dissertations, and scientific articles. One particular technique is intrinsic plagiarism detection, in which potentially plagiarized sentences in a document are highlighted by using only the document content as input (that is, no external information source is used). In such a task, an essential step corresponds to figuring out stylistic differences between plagiarized and original sentences inside a suspicious document. Deep Neural Networks have achieved state-of-art results in the solution of several problems in Natural Language Processing in recent years. Inspired by that, in this work, we apply a simple but effective combination of Deep Learning techniques to the task of intrinsic plagiarism detection. In particular, we use Skip-Thoughts, an embedding model to represent each sentence of a document as a multi-dimensional vector. After that, we train a a Siamese neural network using as training set a collections of sentence pairs (each sentence represented as a Skip-Thoughts vector) extracted from documents in the PAN11 corpus. We then model each document as a weighted, non-directed graph to enable the application of the cluster correlation algorithm, which makes it possible to identify potentially plagiarized passages. Our computational experiments show that the resulting Siamese neural network model is capable of recognizing stylistic differences between sentences in a document. Besides, the identification of potentially plagiarized passages through the cluster correlation approach yields results comparable to those in the literature.

Key-words:
Deep Learning; Artificial Neural Networks; Intrinsic Plagiarism Detection.

Rio de Janeiro,

24 de Março de 2020

Ficha catalográfica elaborada pela Biblioteca Central do CEFET/RJ

L966 Luques, Ivair Nobrega.
Inteligência computacional aplicada à detecção intrínseca de plágio em documentos textuais / Ivair Nobrega Luques – 2020.
59f. : il. (algumas color.)., grafs., tabs. ; enc.

Dissertação (Mestrado). Centro Federal de Educação Tecnológica Celso Suckow da Fonseca, 2020.
Bibliografia: f. 55-59.
Orientador: Eduardo Bezerra da Silva.
Coorientador: Pedro Henrique González Silva.

1. Inteligência computacional. 2. Redes Neurais (computação). 3. Plágio. I. Silva, Eduardo Bezerra da (Orient.). II. Silva, Pedro Henrique González (Coorient.). III. Título.

CDD 006.3

Elaborada pela bibliotecária Vanessa Suane de Souza CRB-7/6753

Sumário

Lista de Figuras

Lista de Tabelas

Lista de Abreviações

AM	Aprendizado De Máquina	1, 13, 14
AP	Aprendizagem Profunda	2, 6, 7, 14, 30, 31, 48
IA	Inteligência Artificial	1, 7, 13
IF	Incorporação De Frases	6, 16, 31
LSTM	Long Short-Term Memory	6, 14, 15, 29, 30
PCC	Problema Da Correlação De Clusters	39
PLN	Processamento De Linguagem Natural	2, 6, 37
PP	Páginas Impressas	43, 45
RNA	Redes Neurais Artificiais	2, 14
RNC	Redes Neurais Convolucionais	14
RNR	Redes Neurais Recorrentes	14, 15
RS	Redes Siamesas	6, 15, 16

Capítulo I Introdução

I.1 Contextualização

O noticiário internacional tem apresentado casos de Ministros de Estado, e até Presidentes, perdendo seus títulos acadêmicos por terem praticado plágio em suas dissertações e teses. Um aumento no envolvimento de um maior número de alunos em situações de plágio nas universidades tem sido identificado em pesquisas do *International Center for Academic Integrity* (ICAI) da Universidade de Clemson nos Estados Unidos [Maurer et al., 2006].

Com a proliferação dos repositórios institucionais de documentos digitais, o crescimento dos movimentos de acesso aberto ao conhecimento, e o olhar cada vez mais atento da comunidade científica internacional em relação ao combate ao plágio, tem se buscado soluções que apoiem a identificação automática destes casos.

O plágio pode ser realizado ao se apropriar do trabalho de alguém como se fosse seu; copiar palavras ou ideias de alguém sem dar-lhe o devido crédito; dar informações incorretas sobre a fonte de uma citação; mudar palavras, mas copiar a estrutura da frase de uma fonte sem dar-lhe o crédito ou copiar tantas ideias e palavras de uma fonte que acaba se tornando a maioria do texto de seu próprio trabalho, dando-se crédito ou não [Maurer et al., 2006]. A ocorrência de casos de plágio nas produções acadêmicas e científicas é um problema real, com impactos financeiros e na imagem das instituições de ensino e pesquisa. Há, portanto, uma natural demanda em obter-se progressos na tarefa de detecção automática de plágio.

A tarefa de detecção de plágio pode ser definida como extrínseca ou intrínseca, dependendo do uso, ou não, de uma base de documentos conhecida $\mathcal{D}$, classificada e validada. A detecção extrínseca é aquela em que documentos de uma coleção externa $\mathcal{D}$ são consultados como apoio na identificação trechos que podem ter sido utilizados para realizar o plágio em um dado documento suspeito d_q. Por outro lado, a detecção intrínseca tem por princípio considerar apenas o conteúdo do próprio documento suspeito d_q, buscando identificar o estilo do autor e eventuais trechos do documento que não são compatíveis com este estilo [Alzahrani et al., 2012].

O Aprendizado de Máquina (AM) é uma área de estudo da Inteligência Artificial (IA), iniciada no século XX [Abramson et al., 1963], na qual são desenvolvidos algoritmos utilizados na identificação de padrões complexos a partir de um conjunto de dados. Uma subárea do AM é o

Processamento de Linguagem Natural. O Processamento de Linguagem Natural (PLN) é a área de pesquisa que aborda como modelos computacionais podem compreender e manipular dados na forma de texto ou fala em linguagem natural, e os utilizar para realizar tarefas úteis. Tarefas comuns do PLN são tradução de texto entre línguas, correção ortográfica, transformação de texto em voz e vice-versa, geração automática de frases, classificação de documentos, dentre outras. A detecção de plágio em documentos textuais, tarefa abordada nesta dissertação, pode ser interpretada como uma tarefa de PLN.

Nos últimos anos o estado da arte em PLN tem sido dominado pela utilização de Redes Neurais Artificiais (RNA) com muitas camadas de processamento, ou aprofundada no tempo, é definida como Aprendizagem Profunda (AP) [Otter et al., 2018; Young et al., 2018]. Esta dissertação apresenta uma abordagem para detecção intrínseca de plágio baseada em técnicas de AP e de otimização combinatória.

I.2 Justificativa

Avaliando as publicações resultantes de diversas competições que trataram da questão de plágio, verifica-se que 43 trabalhos são relativos à tarefa extrínseca enquanto apenas 18 incluíram em seu escopo a tarefa intrínseca. Um levantamento recente sobre ferramentas para a detecção de plágio indicou que apenas 13% delas realizam a tarefa intrínseca [Chowdhury and Bhattacharyya, 2018].

O único trabalho encontrado que utilizava redes neurais para detecção de plágio foi na competição de avaliação de similaridade semântica e inferência textual do PROPOR 2016. Neste trabalho Barbosa et al. [2016] apresentam duas propostas: uma que usa uma rede neural e outra apenas com a avaliação de características estilométricas. De acordo com os autores, como os resultados obtidos foram melhores na segunda proposta, esta foi a inscrita para participar da competição. Desta forma não foi publicado no artigo o detalhamento dos resultados do experimento utilizando a rede neural.

Existem diversas outras tarefas de PLN, como por exemplo reconhecimento do contexto, geração de resumos, extração de entidades nomeadas, classificação de sentimentos. Para estes problemas as soluções do estado da arte utilizam as RNA apontando para a possibilidade de utilizá-las também com sucesso na detecção de plágio.

A maioria dos trabalhos que propõem soluções para detecção de plágio em documentos procura identificar características estilométricas do autor de um documento suspeito d_q para, em seguida, verificar se essas características são usadas de forma consistente em todo o conteúdo de d_q. Esse procedimento se baseia na premissa de que eventuais frases plagiadas não apresentariam a mesma consistência de uso das características estilométricas identificadas para o autor. Uma desvantagem desta abordagem é que essas características estilométricas são definidas manualmente [Stamatatos

et al., 2016], o que corresponde a um procedimento laborioso.

O contexto apresentado nesta seção justifica a pesquisa na tarefa intrínseca da detecção de plágio por meio da utilização de redes neurais profundas.

I.3 Objetivos

O objetivo geral desta dissertação é propor uma abordagem para a detecção intrínseca de plágio em documentos por meio da combinação de técnicas de redes neurais artificiais e de técnicas de otimização combinatória. Mais especificamente, os objetivos são:

1. Propor uma abordagem baseada em redes neurais para detectar o quão similar do ponto de vista estilométrico são duas frases quaisquer. Em particular, propomos a utilização de uma arquitetura particular de rede neural artificial (conhecida como rede neural siamesa) para implementar essa abordagem.

2. Propor uma abordagem de agrupamento de frases componentes de um documento d_q que permite formar grupos de frases relacionadas do ponto de vista estilométrico. Em particular, investigamos a aplicação de um procedimento de agrupamento baseado em otimização combinatória conhecido como algoritmo de correlação de clusters.

I.4 Metodologia

Para o desenvolvimento e validação da solução para detecção de plágio em documentos, utilizamos uma base de artigos disponibilizada para a tarefa de detecção intrínseca de plágio na competição PAN2011[1]. Nesta base, cada documento d_q está anotado para indicar eventuais passagens de plágio em seu conteúdo. Cada passagem corresponde a uma ou mais frases.

No **pré-processamento** cada documento da coleção $\mathcal{D}$ é dividido em suas frases componentes (por meio da biblioteca NLTK). Cada frase recebe um código (número inteiro) de identificação. Em seguida, criamos um banco de dados relacional para armazenar os documentos, suas frases componentes e seus autores, além das anotações relativas às passagens de texto correspondentes a plágio.

No próximo passo, utilizamos a abordagem de incorporação de frases (*sentence embbeding*) denominada Skip-Thought [Kiros et al., 2015]. Essa abordagem corresponde a uma rede neural artificial treinada para mapear uma frase de entrada para um vetor multidimensional. Neste passo utilizamos um modelo de mapeamento pré-treinado com o framework PyTorch[2].

[1] https://pan.webis.de/clef11/pan11-web/index.html

[2] Esse modelo pré-treinado e o código para sua manipulação podem ser obtidos em `https://pypi.org/project/skipthoughts/`

A seguir, geramos um novo conjunto de dados já apropriado para o treinamento da rede neural siamesa. Cada item nesse conjunto de dados é uma tripla que relaciona um determinado par de frases componentes de um documento d_q à informação de identificação de similaridade de estilo entre as frases de cada par. Consideramos que um par de frases (s_i, s_j) é similar estilometricamente falando se tanto s_i quanto s_j foram efetivamente escritas pelo autor do documento d_q. Por outro lado, consideramos que (s_i, s_j) não é similar do ponto de vista estilométrico se s_i é uma frase do autor e s_j não é (ou vice-versa).

O conjunto de dados composto por triplas é usado no estágio de **treinamento do modelo** da rede neural siamesa. Os dois primeiros componentes de cada tripla são os vetores Skip-Thought correspondentes a cada uma das duas frases. O terceiro componente da tripla é a marca de seleção (*tag*) da similaridade estilométrica entre as frases. A partir destas informações, a rede neural é treinada para ser capaz de produzir um valor de similaridade estilométrica entre duas frases quaisquer fornecidas como entrada. Para treinamento da rede neural siamesa, utilizamos o framework Keras[3].

A **identificação de eventuais passagens plagiadas** em um documento d_q é realizada em duas etapas. A primeira corresponde a representar d_q como um grafo ponderado e não-dirigido $G(V, E)$. Cada vértice em V corresponde a uma frase de d_q. Cada aresta (v_i, v_j) em E é rotulada com o valor de similaridade entre os vetores Skip-Thoughts (valor esse produzido pela rede neural siamesa previamente treinada) correspondente aos vértices v_i e v_j.

Na segunda etapa, o grafo G gerado para o documento d_q é então dado como entrada para o algoritmo de correlação de *clusters*, um algoritmo de agrupamento que usa uma abordagem de otimização combinatória. Esse algoritmo é aplicado para separar os vértices de G (cada um dos quais corresponde a uma frase do documento d_q) em grupos de mesma característica estilométrica.

Para validação da abordagem proposta nesta dissertação para detecção de plágio, utilizamos o *script* de validação fornecido pela própria competição PAN11[4].

I.5 Organização dos Capítulos

O restante desta dissertação está organizado da seguinte forma. No Capítulo II é apresentado o referencial teórico sobre a tarefa de detecção de plágio, aprendizagem profunda e as arquiteturas de redes neurais artificiais utilizadas. A representação vetorial de palavras e frases também é abordada neste capítulo. O Capítulo III avalia o conteúdo dos trabalhos relacionados encontrados, enfatizando as similaridades e diferenças em relação à proposta desta dissertação. A seguir, o Capítulo IV apresenta as informações sobre o conjunto de dados utilizados, e o detalhamento das

[3] https://keras.io/

[4] Esse script pode ser obtido em `https://pan.webis.de/sepln09/pan09-code/pan09-plagiarism-detection-performance-measures.py`

etapas de pré-processamento, processamento, treinamento do modelo de similaridade estilométrica e a identificação de plágio propriamente dita. Toda a etapa dos experimentos está explicitada no Capítulo V, ficando as conclusões e trabalhos futuros para o Capítulo VI.

Capítulo II Fundamentação Teórica

Este capítulo apresenta a fundamentação teórica necessária para este trabalho e está dividido da seguinte forma: a Seção II.1 detalha conceitualmente a tarefa de detecção de plágio e está organizada em quatro seções. As Seções II.1.1 e II.1.2 discorrem sobre as diferenças entre as abordagens intrínseca e extrínseca para a tarefa de detecção de plágio. A Seção II.1.3 refere-se ao processo de identificação de autores e na Seção II.1.4 são traçadas considerações sobre o uso das características estilométricas de um texto. A existência de um evento de competição internacional dedicada à tarefa de detecção de plágio, e outras tarefas de identificação de autoria, o PAN, é tradada na Seção II.2, pois a base de dados deste evento é utilizada em nossos experimentos. O capítulo contempla também a conceituação da AP e o uso de Redes Siamesas (RS) e Long Short-Term Memory (LSTM) em PLN na Seção II.3. A representação de palavras e frases de forma vetorial e a apresentação de algoritmos de Incorporação de Frases (IF) estão na parte final na Seção II.3.3.

II.1 Tarefa de Detecção de Plágio

A detecção de plágio pode ser dividida em duas tarefas distintas: extrínseca e intrínseca. A primeira utiliza vários documentos de apoio, enquanto a segunda utiliza apenas o documento para objeto da avaliação. Com base apenas no documento em análise, identificar se o texto contido no documento possui mais de um estilo de escrita é um caminho de apoio à detecção de plágio. Isto pode ser feito por meio da tarefa de diarização de autores, umas das possíveis abordagens para a identificação de autores. Outra maneira de identificar estilos distintos em um texto é pelas características estilométricas, método este muito utilizado, mas que não é aplicado neste trabalho.

II.1.1 Detecção Extrínseca de Plágio

Na tarefa de detecção extrínseca de plágios, existe um documento original a avaliar e um conjunto de outros documentos, identificados, conhecidos e validados que compõem, junto com o documento original, as entradas do processo. O primeiro passo seleciona, por meio de algum modelo de recuperação de documentos candidatos, aqueles que podem ter sido utilizados para realizar o plágio. Estes modelos podem ser baseados em recuperação de informações, técnicas de

agrupamento ou cruzamento de idioma. O passo seguinte realiza a aplicação de algum método de análise de detecção de plágio, como os baseados em caractere, ou vetores, ou sintaxe, ou semântica, ou *Fuzzy*, ou estrutura, ou estilometria, ou cruzamento de idioma. O último passo, por meio de um pós-processamento, considerando o conhecimento adquirido nos passos anteriores, verifica se os trechos de documentos selecionados possuem citação no documento original. Dentre estes, os que não possuem citação são as saídas do processo, e consideradas as seções suspeitas de plágio do documento [Alzahrani et al., 2012]. Uma visão do processo desta tarefa está representada na Figura II.1.

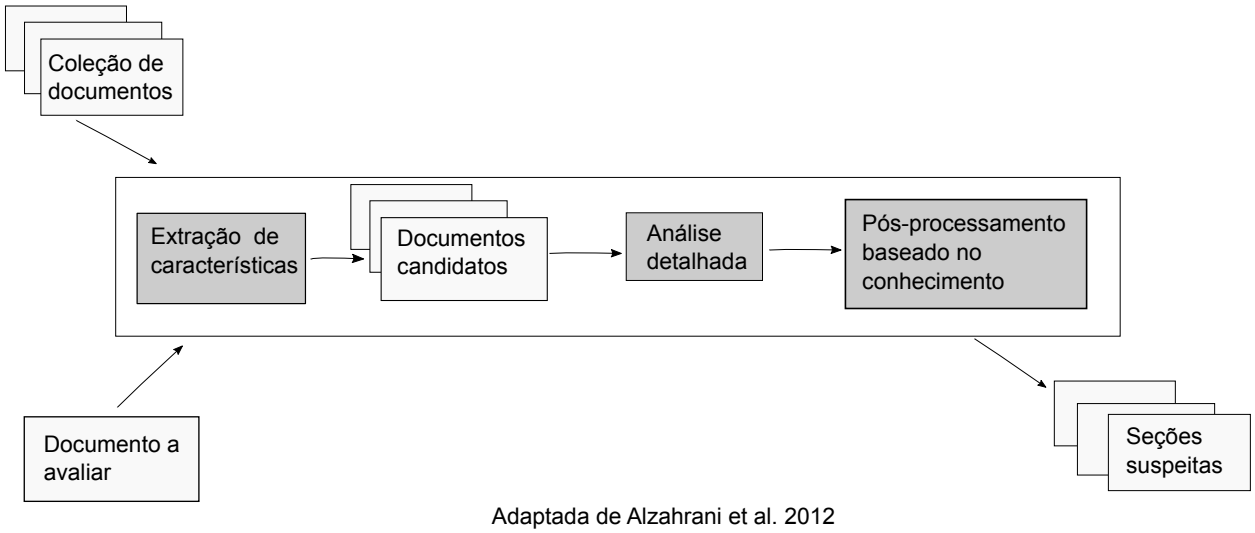

Figura II.1: Detecção Extrínseca de Plágio

II.1.2 Detecção Intrínseca de Plágio

Na tarefa de detecção intrínseca de plágio apenas o próprio documento original a ser avaliado é entrada do processo. Não é utilizado mais nenhum documento adicional como apoio. O primeiro passo do processo realiza uma segmentação no texto, separando-o em diversas seções, parágrafos, frases ou até palavras. Esta segmentação pode ser realizada baseando-se na organização dos parágrafos, estrutura da frase, por número fixo de palavras (n-gramas), número fixo de caracteres, pela presença do caractere '.', ou outro critério definido.

O segundo passo "aprende" qual o "estilo" de escrita do autor do documento apoiando a aprendizagem e a identificação. Uma abordagem para a identificação do estilo são as características estilométricas, que podem ser léxicas, semânticas, sintáticas ou estruturais. Outra possibilidade é obter a aprendizagem do estilo por meio do uso de IA, que é a utilizada neste trabalho com apoio da AP.

O último passo realiza uma análise do documento original e seleciona os trechos do documento que são incompatíveis com o estilo de escrita do autor. Representando vetorialmente frases e apli-

cando algoritmos em grafos, torna-se possível avaliar a incompatibilidade com o estilo predominante no documento.

Os trechos indicados como incompatíveis são as saídas do processo e considerados, portanto, seções suspeitas de plágio do documento original [Eissen and Stein, 2006]. Uma visão do processo desta tarefa está representada na Figura II.2.

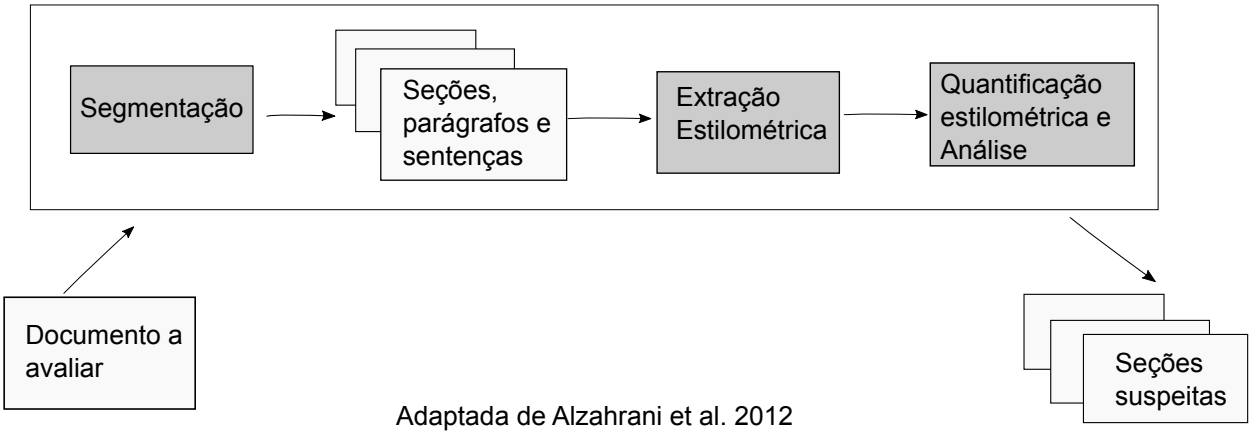

Figura II.2: Detecção Intrínseca de Plágio

II.1.3 Identificação de Autores

Cada pessoa, de forma consciente, ou não, tem sua própria maneira de escrever. O uso do vocabulário, a repetição de palavras, o posicionamento de advérbios, as frases serem mais curtas ou longas, a estruturação do parágrafo, são diferentes de um indivíduo para outro. A identificação de autores trata de ao se avaliar um documento e conseguir identificar quantos autores escreveram o texto, bem como sinalizar qual parte pertence a cada um deles. Há tarefas distintas para a realização desta identificação: a atribuição de autoria, a verificação de autoria, o agrupamento por autor e a diarização de autor. Este trabalho dedica-se a um caso específico da diarização do autor que é a detecção intrínseca, que é quando garantimos que ao menos 70% do texto é de um autor único.

- **Atribuição de Autoria**.

  Neste caso, estão previamente selecionados alguns autores candidatos a ter relação com o documento avaliado. A tarefa é determinar, a partir do estilo de escrita de cada candidato, qual deles realmente escreveu o documento.

- **Verificação de Autoria**.

  Tem-se dois documentos e a tarefa consiste em saber se eles foram escritos pelo mesmo autor. A verificação de semelhança entre os estilos de escrita dos documentos determinará se o resultado é positivo ou negativo.

- **Agrupamento por autor**.

  Dado um conjunto de documentos, que a princípio foram escritos por autores diversos, o objetivo é agrupá-los por autoria. Deve-se identificar o estilo de escrita de cada documento e, por similaridade, reuni-los em um grupo.

- **Diarização de Autor**.

  Dado um documento, deve-se identificar as mudanças de estilo e agrupar as partes do texto que possuem estilo similar. O conjunto de frases agrupadas indicam que foram escritas pelo mesmo autor. A diarização de autor, pela característica do corpus utilizado, pode ser categorizada entre uma de três tarefas distintas:

  Detecção intrínseca de plágio tradicional: Assumindo um autor principal que escreveu pelo menos 70% de um documento, a tarefa é encontrar as porções de texto restantes escritas por um ou vários outros autores.

  Diarização com um determinado número de autores: A base para esta tarefa é um documento que foi composto por um número conhecido de autores. Uma vez criados tantos grupos quanto o número de autores, cada grupo deve conter os fragmentos de texto individuais de cada autor.

  Diarização não restrita: Neste caso, sabe-se que no texto há mais de um autor mas desconhece-se quantos são. Assim, durante a análise e atribuição do texto, também deve ser encontrado o número correto de autores, por meio do agrupamento dos trechos por similaridade de estilos.

II.1.4 Características Estilométricas

Existem diversas características estilométricas que podem ser utilizadas na avaliação dos estilos de escrita presentes em um documento, ou comparativamente entre vários documentos. Quando utilizadas na tarefa de detecção intrínseca devem ser utilizadas preferencialmente estatísticas do texto via características léxicas, características sintáticas para quantificar as palavras por classes e particionar frases, características semânticas que quantifica sinônimos e identifica dependências semânticas, além de características específicas para identificação de palavras-chave [Alzahrani et al., 2012]. A Tabela II.1 elenca uma série de exemplos de características estilométricas.

II.2 Conjunto de Dados - PAN

PAN - *Excellence Network on Digital Text Forensics* é uma série de eventos científicos e tarefas compartilhadas de análise forense e estilometria em textos digitais, promovidos pelo *Webis*

Tabela II.1: Características estilométricas

Léxicas baseadas em caracter
Frequência de caracteres
Tipos de caracteres (letras, dígitos, pontuações)
Frequência de caracteres especiais (e.g. !, &, etc.)
Frequência de n-gramas de comprimento fixo
Frequência de n-gramas de comprimento variável
Métodos de Compressão
Léxicas baseadas em palavras
Comprimento médio das palavras
Comprimento médio das frases
Média de sílabas por palavra
Riqueza de vocabulário
Frequência de palavras
Frequência de palavras de função
Frequência de n-gramas palavras
Erros de ortografia
Erros de formatação
Sintáticas
Classes gramaticais
Frequência de n-gramas partes do discurso
Trechos
Frases e estruturas de frases
Regras de Frequência de reescrita
Erros de sintaxe
Semânticas
Sinônimos
Antônimos
Dependências semânticas
Aplicações Específicas
Estruturais
Palavras-chave
Linguagem específica

Adapted from Alzahrani et al. 2012

Group [1]. Estes eventos ocorrem anualmente junto à conferência CLEF - *Conference and Labs of the Evaluation Forum* [2], e neles são propostas tarefas variadas para as quais pesquisadores submetem seus experimentos no ambiente TIRA [Potthast et al., 2019], disponibilizado pela organização permitindo a avaliação dos trabalhos. Algumas das tarefas propostas nestes eventos são: atribuição de autoria, verificação de autoria, agrupamento de autor, diarização de autor, detecção de alteração de estilo, detecção extrínseca de plágio e detecção intrínseca de plágio. Estes eventos disponibilizam aos participantes bases de dados compostas de conjuntos de documentos, apropriados para as etapas de desenvolvimento e testes dos experimentos de cada pesquisa.

[1] https://webis.de/
[2] https://webis.de/events.html

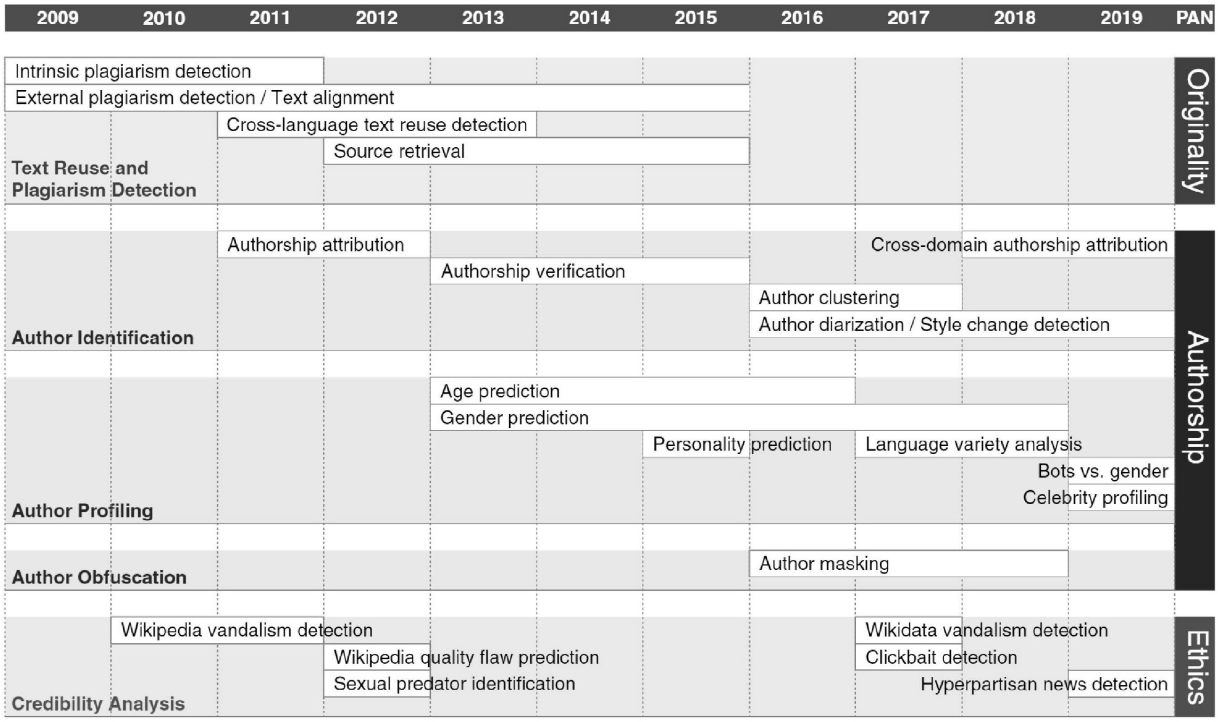

Figura II.3: O Histórico de Tarefas do PAN

Um histórico das tarefas do PAN , a cada ano, é apresentado na figura II.3. A tarefa de detecção intrínseca de plágio esteve presente no intervalo de 2009 a 2011, sendo posteriormente substituída por tarefas de Atribuição de Autoria e Verificação de Autoria [3].

Como esta pesquisa trata da tarefa de detecção intrínseca de plágio, que esteve presente nos PANs de 2009 [Webis at Bauhaus-Universit"at Weimar and NLEL at Universidad Polytécnica de Valencia, 2009] a 2011, se optou pela utilização nestes experimentos o corpus disponibilizado para estas competições. Além de ter documentação disponível e detalhada sobre a composição do corpus e dos detalhes técnicos, isto possibilitaria a comparação com os resultados de vários outros trabalhos que também o utilizaram, desde que aplicadas as mesmas métricas de avaliação.

A tarefa de diarização de autores do PAN de 2016 dá continuidade e estende a tarefa de detecção intrínseca de plágio [Potthast et al., 2011]. O problema original está relacionado com a questão de quando um autor incluiu texto de outros autores sem as referências apropriadas e, em caso afirmativo, identificar quais são as partes afetadas. A identificação das palavras que indicam suspeita de plágio deve ser realizada com base apenas no próprio documento. Quaisquer consultas a outras fontes de informação são proibidas. Por conseguinte, os autores devem ser identificados por meio da análise do estilo de escrita de alguma forma. Esta não é uma restrição artificial, mas tem relevância prática em sistemas de detecção de plágio, por exemplo, para limitar ou pré-ordenar o espaço de busca ou para investigar documentos antigos onde fontes potenciais não estão disponíveis

[3]https://github.com/pan-webis-de/downloads/blob/master/publications/slides/potthast2019a.pdf

digitalmente.

II.2.1 PAN 2009

O corpus de plágio do PAN2009 compreende 41 223 documentos de texto, em que 94 202 casos de plágio artificial foram inseridos automaticamente. O corpus é baseado em 22 874 documentos de comprimento de livro do Projeto Gutenberg. A tarefa de detecção intrínseca de plágio reuniu menos atenção do que a detecção de plágio externo nesta competição, onde quatro sistemas participaram da tarefa.

Ao contrário da detecção extrínseca de plágio, nesta tarefa a linha de base do desempenho não é zero. A razão para isso é que a detecção de intrínseca de plágio é um problema de classificação de uma classe em que tem a ser decidido para cada seção de um documento se é plagiado ou não. A linha de base de desempenho em tais problemas é comumente computada como a suposição ingênua de que tudo pertence à classe alvo, ocasionando distorções no resultado. Apenas a abordagem de Stamatatos [2009a] apresenta melhor desempenho que a linha de base com 0,4607 de recall, 0,2321 precisão e com F de 0,3086.

II.2.2 PAN 2011

A detecção intrínseca de plágio atraiu um interesse renovado no PAN2011. Uma análise dos cadernos submetidos revela um conjunto genérico de blocos de construção, que empregam uma estratégia de fragmentação, um modelo de recuperação de estilo de escrita e um algoritmo de detecção de *outliers*; no entanto, as especificidades diferem significativamente. Em todos os casos, os blocos de construção mencionados são organizados dentro de um processo com as seguintes etapas para um dado documento suspeito:

(1) Fragmentação: o documento é fragmentado sendo que todos os detectores enviados empregam *chunking* de janela deslizante com tamanhos de pedaços variando de 200 a 1000 palavras. O deslizamento da janela varia de 40 a 500 palavras.

(2) Modelo de Recuperação: os pedaços são representados sob o modelo de recuperação de estilo. Os modelos de recuperação para detecção intrínseca de plágio são compostos de uma função de modelo que mapeia textos para representações de recursos, juntamente com uma medida de similaridade para comparar representações. Os detectores enviados usam recursos baseados ou em palavras ou em caracteres.

(3) Detecção de valores discrepantes: com base no modelo de recuperação de estilo, a detecção de valores discrepantes tenta identificar partes do documento suspeito que são visivelmente diferentes das outras. As duas estratégias a seguir foram aplicadas este ano: (1) medindo o desvio do estilo médio do documento e (2) *clustering* do bloco.

(4) Pós-processamento: com relação ao pós-processamento, a maioria dos detectores mescla trechos sobrepostos e consecutivos que foram identificados como *outliers* para diminuir a granularidade de detecção. Após o pós-processamento, os pedaços identificados são devolvidos como passagens potencialmente plagiadas.

II.2.3 PAN 2016

A tarefa compartilhada no PAN2016 se concentra na identificação de autorizações dentro de um único documento. Por isso não é apenas procurado pelo plágio, mas também para as contribuições de diferentes escritores em um documento de autoria múltipla. Entre os exemplos para estes últimos são teses de estudantes escritas em colaboração ou artigos científicos compostos por um conhecido número de pesquisadores que colaboraram.

Para todas as subtarefas, foram fornecidos conjuntos de dados de treinamento e teste distintos. O corpus original contém documentos em 150 tópicos usados nas *TREC Web Tracks* de 2009-2011. A partir desses documentos, os respectivos conjuntos de dados para todas as tarefas foram gerados variando as configurações, como o número e as proporções dos autores em um documento, a decisão, se eles são distribuídos uniformemente, os permutadores nas autorizações podem ocorrer dentro de uma única frase, no final de uma frase ou apenas entre parágrafos. Em geral, o número de documentos de treinamento / teste para as respectivas subtarefas foram: (a) 71/29, (b) 55/31 e (c) 54/29.

O desempenho dos algoritmos submetidos foi medido com métricas diferentes. Para a subtarefa de detecção intrínseca de plágio, as métricas Recall, Precisão e F-score foram utilizadas. Os resultados finais das duas equipes participantes são apresentados na Tabela III.2. Por outro lado, as subtarefas de diarização foram medidas com as métricas de *cluster* agrupadas, pois refletem muito bem a natureza de agrupamento de documentos internos dessas tarefas. Sub resultados finos, dependendo da configuração do conjunto de dados, por exemplo, o número de autores em um documento e sua taxa de contribuição, foram apresentados no documento de síntese desta tarefa [Rosso et al., 2016].

II.3 Redes Neurais

A IA é um termo genérico que implica no uso do computador para simular o comportamento inteligente com a mínima intervenção humana [Schmidhuber, 2015]. Neste trabalho é necessária a utilização da AM, no campo da IA.

A AM não é um conceito recente [Abramson et al., 1963]. Mas no campo da IA lida com a concepção e desenvolvimento de algoritmos para identificar padrões complexos a partir de dados experimentais, sem assumir uma equação pré-estabelecida como modelo e tomar decisões de forma

inteligente [Bischl et al., 2016]. A AM tem sido utilizada na investigação de técnicas para simular o comportamento do cérebro humano em tarefas como reconhecimento facial, reconhecimento de fala e processamento de linguagem natural [Deng and Yu, 2014].

A AP utiliza-se de métodos analíticos para representar conceitos, implementando uma hierarquia de conhecimento em múltiplas camadas. À medida que caminhamos pelas camadas ocorre a aprendizagem baseada nos dados representados nas camadas. Com uma maior disponibilidade de grandes volumes de dados, a partir do início do século XXI, houve um avanço nas pesquisas utilizando a AP [Bengio, 2009]. A AP é utilizada neste trabalho para a criação de um modelo capaz de identificar o estilo de escrita do autor expresso em um texto.

II.3.1 Redes Long Short-Term Memory

Uma das motivações para a investigação de redes neurais artificiais compostas de muitas camadas veio da neurociência, mais especificamente do estudo da parte do cérebro denominada córtex visual. Dessa forma, cada região de neurônios (que é análoga ao conceito de camada em redes neurais artificiais) combina padrões detectados pela região imediatamente anterior para formar características mais complexas.

As Redes Neurais Artificiais (RNA) têm como objetivo simular o comportamento de uma rede biológica de neurônios. Uma RNA possui uma rede de neurônios artificiais onde cada unidade realiza uma computação baseada nas demais unidades em que está conectada [Goodfellow et al., 2016]. Há diferentes arquiteturas de RNA e as formas como os nós destas redes são percorridos definem classes distintas. Uma dessas classes é as das Redes Neurais Recorrentes (RNR) e outras a das Redes Neurais Convolucionais (RNC).

As RNC são redes neurais artificias em que uma determinada camada, em vez de ligar cada entrada a cada neurônio, elas restringem as conexões intencionalmente para que qualquer neurônio aceite as entradas apenas a partir de uma pequena subseção da camada anterior. Nelas o padrão de conectividade entre os neurônios é inspirado na organização do córtex visual dos animais. Neurônios corticais individuais respondem a estímulos apenas em regiões restritas do campo de visão conhecidas como campos receptivos. Os campos receptivos de diferentes neurônios se sobrepõem parcialmente de forma a cobrir todo o campo de visão [Gehring et al., 2017].

RNR constituem uma ampla classe de redes cuja evolução do estado depende tanto da entrada corrente quanto do estado atual. Estas redes tem a possibilidade de realizar computação dependente do contexto e aprender dependências de longo prazo [Visin et al., 2016]. Devido a sua flexibilidade no que concerne ao processamento de entradas de tamanho variável, RNR têm sido aplicadas recentemente em uma variedade de problemas. Dentre as RNR há as redes ELman, Jordan e as LSTM.

As redes LSTM são um tipo mais complexo de RNR proposto em Hochreiter and Schmidhuber [1997]. Este tipo de rede é formada por células (ou blocos) que, por possuírem uma estrutura interna apropriada, podem armazenar um determinado valor por uma quantidade arbitrária de passos de tempo, permitindo assim uma forma de aprendizagem que favorece a sua utilização na metodologia que é aplicada nesta pesquisa [Bezerra, 2016].

II.3.2 Redes Siamesas

RS são redes de ramificação dupla com pesos vinculados, isto é, consistem na mesma rede copiada e mesclada com uma função de energia [Hoffer and Ailon, 2015].

A Figura II.4 mostra uma visão geral dessa arquitetura de rede. O conjunto de treinamento para uma rede siamesa consiste em triplas (x_1, x_2, y) onde x_1 e x_2 são sequências de caracteres (ou frases) e $y \in 0,1$ indica if (x_1 e x_2) que quanto mais próximo de 1, maiores as chances de serem similares e quanto mais próximo de zero, menores as chances. O objetivo do treinamento é minimizar a distância em um espaço de incorporação entre pares semelhantes e maximizar a distância entre os pares não similares.

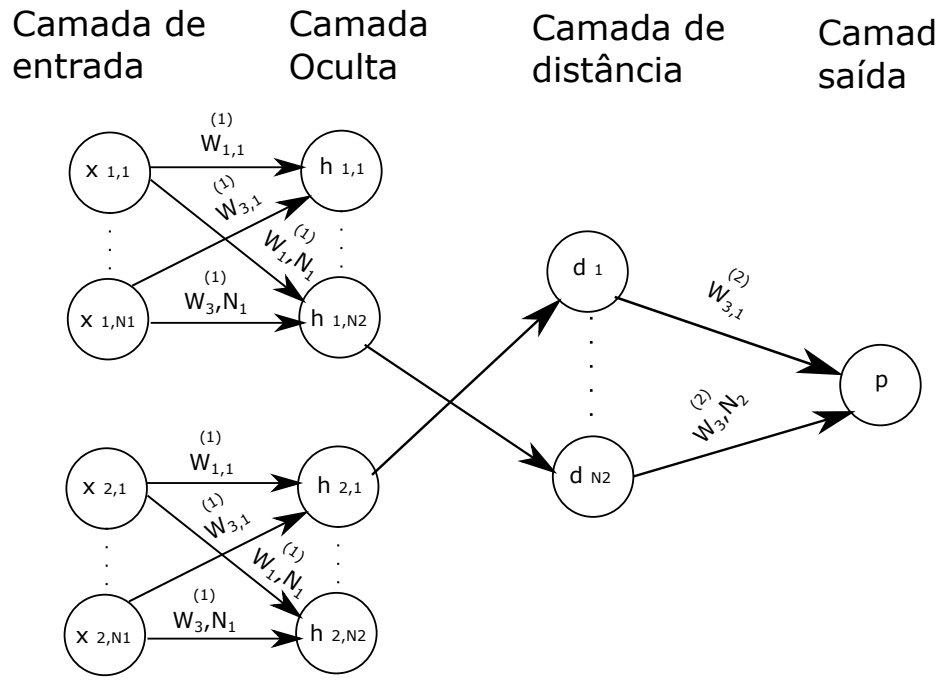

Figura II.4: Redes Siamesas

Uma rede neural siamesa consiste em redes gêmeas que aceitam entradas distintas, mas são unidas por uma função de energia no topo. Esta função computa alguma métrica entre o nível mais alto de representação característica em cada lado. Os parâmetros entre as redes gêmeas estão vinculados. Quando da aplicação no reconhecimento de imagens, a amarração do peso garante que duas imagens extremamente semelhantes possivelmente não poderão ser mapeadas por suas respectivas redes para locais muito diferentes no espaço de características, porque cada rede com-

puta a mesma função. Além disso, a rede é simétrica, de modo que sempre que apresentamos duas imagens distintas para as redes gêmeas, a camada superior conjunta calculará a mesma métrica como se fôssemos apresentados as mesmas duas imagens mas para os gêmeos opostos [Koch et al., 2015].

A Figura II.4 apresenta o esquema de uma rede siamesa simples, de apenas uma camada oculta, para classificação binária. A estrutura da rede é replicada em suas seções superior e inferior, formando assim as redes gêmeas. É importante notar que a matriz de pesos é compartilhada em cada camada da rede. Para a formação da camada de distância existem várias alternativas: sendo a distância euclidiana e a semelhança de cosseno mais utilizada [Koch et al., 2015].

Exemplificaremos, de forma simplificada o funcionamento de uma RS, considerando como entradas da rede duas imagens para as quais queremos avaliar o grau de similaridade.

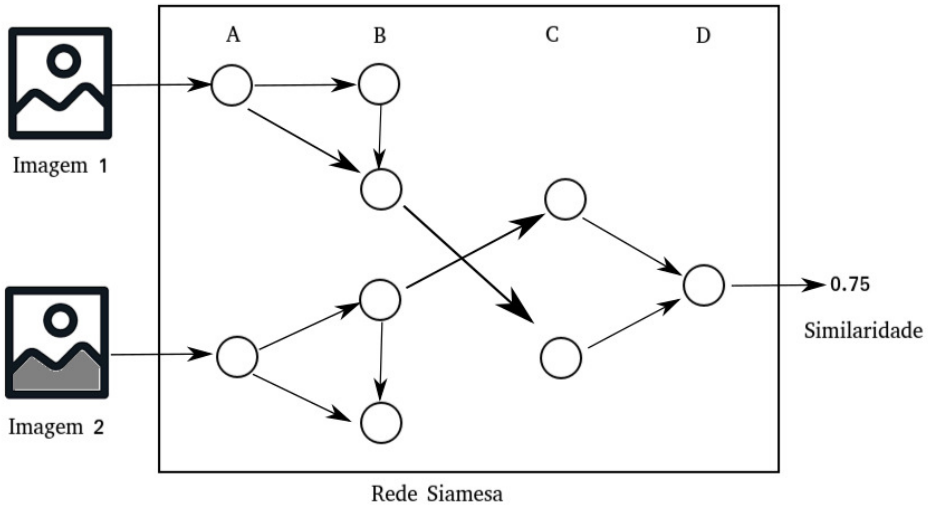

Figura II.5: Exemplo Rede Siamesa

Conforme a Figura II.5 em cada ramo da RS entra uma das imagens[A]. Com os pesos compartilhados a rede gera um valor referente a cada imagem[B]. Então é aplicada uma função de distância que resultará na representação da similaridade entre as imagens[C]. A camada de saída final é gerada por alguma função que garanta representação dos resultados em valores entre 0 e 1[D].

Em nosso trabalho as entradas da RS serão as frases, em sua representação vetorial.

II.3.3 Incorporação de Frases

IF (*sentence embeddings*) é um conjunto de técnicas de modelagem e aprendizado em processamento de linguagem natural em que palavras ou frases são mapeadas para vetores de números reais. É comum que este processo use como base um vocabulário específico de referência. Trata-se de gerar uma forma de representar vetorialmente, com uma grande redução de dimensionalidade,

a relação de distância entre as palavras ou frases. Existem inúmeros métodos para gerar esta representação vetorial, mas em nosso caso utilizamos o Skip-Thoughts [Kiros et al., 2015].

- Skip-Thoughts

  Skip-Thoughts é uma rede neural artificial para a criação de incorporações (ou seja, representações vetoriais) para frases em texto. Ele usa aprendizado não supervisionado e sua arquitetura consiste em um codificador e dois decodificadores. Os Skip-Thoughts são ilustrados na Figura II.6, onde S_{i-1}, S_i e S_{i+1} são frases e $Z(S_i)$ é uma representação vetorial da frase S_i. Os decodificadores e o codificador são redes neurais recorrentes [Kiros et al., 2015]. Durante a fase de treinamento, a rede Skip-Thoughts recebe três frases consecutivas como entrada. A frase do meio é codificada e usada como entrada para os decodificadores, que tentam reconstruir as frases anteriores e subsequentes. No exemplo acima, a frase S_i é codificada em um vetor $Z(S_i)$, então o decodificador anterior tenta criar a frase S_{i-1} recebendo $Z(S_i)$ como entrada, enquanto *Next Decoder* tenta criar S_{i+1} recebendo $Z(S_i)$. Depois que o modelo é treinado, o *Encoder* pode ser usado para gerar incorporações de frases [Kiros et al., 2015].

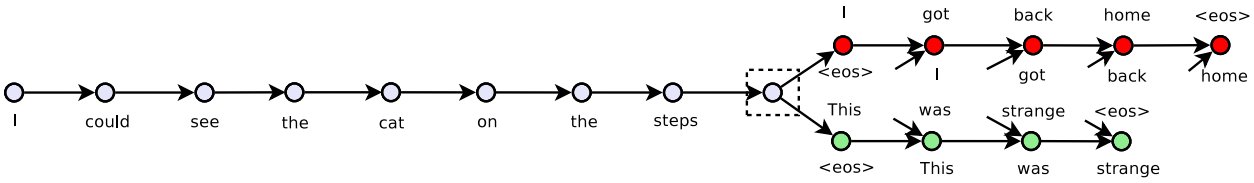

Figura II.6: O Modelo Skip-Thoughts

II.4 Modelos que usam características estilométricas

O uso das características estilométricas no apoio à tarefa de detecção de plágio baseia-se no fato de que cada autor desenvolve uma maneira individual de escrever, caracterizando-se assim um estilo próprio [Alsallal et al., 2013].

Segundo Meyer Zu Eissen and Stein [2006], basicamente o poder de uma abordagem de detecção de plágio depende da qualidade da identificação de características linguísticas. As características estilométricas quantificam aspectos do estilo de escrita e alguns deles foram utilizados com sucesso para discriminar entre livros com respeito à autoria.

A detecção de plágio intrínseco usa a estilometria (análise do estilo literário) para identificar segmentos estilisticamente diferentes em um texto. Com características cuidadosamente escolhidas, a estilometria é robusta mesmo contra tentativas de imitação automatizadas, já que alterar automaticamente a estrutura gramatical de uma frase sem alterar o significado do texto é um desafio [Krause, 2015].

Na estilometria são considerados aspectos léxicos, semânticos, sintáticos e alguns de aplicação específica nesta caracterização de estilo. As Tabela II.2 a II.6 apresentam alguns exemplos desta diversidade de características que podem ser consideradas [Alzahrani et al., 2012].

As características léxicas podem operar no nível de caracteres, ou de palavras. A representação baseada em caracteres considera um documento(d) formado por uma sequência de caracteres ($c_1, c_2...c_n$) onde n é o tamanho do documento em número de caracteres. Já a representação baseada em palavras considera um documento(d) formado por uma sequência de palavras ($p_1, p_2...p_n$) onde n é o tamanho do documento em número de palavras. Em ambas as representações é desconsiderada a estrutura do documento em relação a parágrafos e frases. As Tabelas II.2 a II.3 apresentam algumas características léxicas.

Tabela II.2: Características estilométricas léxicas baseadas em caracteres

Léxicas baseadas em caracteres
Frequência de caracteres
Tipos de caracteres (letras, dígitos, pontuações)
Frequência de caracteres especiais (p.ex !, &, etc.)
Frequência de n-gramas de comprimento fixo
Frequência de n-gramas de comprimento variável
Métodos de Compressão

Na Tabela II.2 são apresentadas algumas características estilométricas léxicas baseadas em caracteres:

- Frequência de caracteres - p.ex. quantas vezes cada letra aparece no texto: A(35), B(18), C(30), ... Z(10)
- Tipos de caracteres - p.ex. quantas letras, dígitos, caracteres especiais e pontuação aparecem no texto.
- Frequência de caracteres especiais - p.ex. quantas vezes aparece no texto os caracteres , #, %, $, &.
- Frequência de n-gramas de comprimento fixo - p.ex. define-se o critério de 2-gram e verifica-se quantas vezes cada 2-gram aparece no texto: aa(0), ab(22), ac(19), ad(23), ... zz(0).
- Frequência de n-gramas de comprimento variável - p.ex. define-se o critério de trabalhar com 2-gram e 4-gram e por amostragem verifica-se no texto a quantidade de ocorrências: br(20), de(89), acro(3), vers(32), tras(39).
- Métodos de Compressão - A frequência de caracteres em um texto pode servir de referência para modelos estatísticos e adaptativos de métodos de compressão. O resultado da compressão

do texto, ou de blocos de texto, auxilia a identificação das características estilométricas [Stamatatos, 2009b].

Tabela II.3: Características estilométricas baseadas em palavras

Léxicas baseadas em palavras
Comprimento médio das palavras
Comprimento médio das frases
Média de sílabas por palavra
Riqueza de vocabulário
Frequência de palavras
Frequência de palavras de função
Frequência de n-gramas palavras
Erros de ortografia
Erros de formatação

Na Tabela II.3 são apresentadas algumas características estilométricas léxicas baseadas em palavras:

- Comprimento médio das palavras - avalia qual o tamanho médio das palavras utilizadas no texto, p.ex. palavras de um caracter (21%), de dois caracteres (24%), de três caracteres (16%), de dez caracteres (1%), gerando um tamanho médio de palavras com $3,45$ caracteres.

- Comprimento médio das frases - avalia qual o tamanho médio das frases utilizadas no texto, p.ex. frases de cinco palavras (1%), de seis palavras (11%), de sete palavras (22%), de oito palavras (34%), de quinze palavras (2%), gerando um tamanho médio de frases com $8,76$ palavras.

- Média de sílabas por palavra - avalia qual o tamanho médio das palavras em sílabas utilizadas no texto, p.ex. palavras de uma sílaba (23%), de 2 sílabas (26%), de 3 sílabas (33%), de 4 palavras (12%), de 6 sílabas (1%), gerando um tamanho médio de palavras com $3,12$ sílabas.

- Riqueza de vocabulário - utiliza critérios como a verificação do número de palavras distintas que aparecem no texto e avalia a quantidade de palavras com apenas uma ou duas ocorrências.

- Frequência de palavras - quantifica, para cada uma das palavras que aparece no texto, a sua ocorrência. Auxilia, de forma complementar, na identificação de repetição de vocabulário, falta do uso de sinônimos, entre outras características de estilo de escrita.

- Frequência de palavras de função - quantifica as ocorrências de palavras de função no texto. Palavras de função são aquelas que auxiliam a que as frases estejam gramaticalmente corretas, apesar de não terem peso semântico. Normalmente são os pronomes, preposições e verbos

auxiliares. Na frase "*Meu* pai foi *para o* hospital" as palavras em destaque são palavras de função.

- Frequência de n-gramas palavras - quantifica a ocorrência no texto do conjunto de palavras definidas pelo n-gram. Com o parâmetro de 2-gram de palavras, a verificação das ocorrências será de pares de palavras do texto.

- Erros de ortografia - verifica a ocorrência de erros de ortografia no texto, bem como a quantidade de incidência dos mesmos e sua repetição. p.ex. omissão de letras, letras repetidas.

- Erros de formatação - verifica a ocorrência de erros de formatação do texto, bem como a quantidade de incidência dos mesmos e sua repetição. p.ex. uso de letras maiúsculas no início das frases, falta de pontuação final nas frases.

As características estilométricas sintáticas se manifestam no uso das classes gramaticais de frases e de palavras em diferentes declarações. O uso de marcadores das classes gramaticais das palavras de um texto sinalizando se são verbos, substantivos, pronomes, adjetivos, advérbios, preposições, conjunções ou interjeições é utilizado no apoio à identificação das características estilométricas. A representação baseada em frases divide o texto em afirmações utilizando como referência os delimitadores de fim de frases: pontos finais, pontos de exclamação e pontos de interrogação.

Tabela II.4: Características estilométricas sintáticas

Sintáticas
Classes gramaticais
Frequência das classes gramaticais em n-gramas
Trechos
frases e estruturas das frases
Regras de Frequência de reescrita
Erros de sintaxe

Na Tabela II.4 são apresentadas algumas características estilométricas sintáticas:

- Classes gramaticais - quantifica a ocorrência no texto de verbos, advérbios, adjetivos, substantivos, pronomes, preposições, conjunções e interjeições.

- Frequência das classes gramaticais em n-gramas - quantifica a ocorrência no texto das classes gramaticais definidas pelo n-gram. Com o parâmetro de 2-gram de palavras, a verificação das ocorrências das classes gramaticais não será realizada por palavra, mas sim por pares de palavras.

- Trechos - Neste caso define-se um tamanho fixo de quantidade de caracteres, delimitando

assim um trecho. p.ex. 100 caracteres. Deste modo a verificação de ocorrência de palavras não se dá no âmbito da frase, mas sim dentro do trecho pré-definido.

- Frases e estruturas das frases - A estrutura do texto em sua organização em frases é considerada aqui. p.ex. a quantidade de frases no texto é uma das características estilométricas.

- Regras de Frequência de reescrita - A identificação de recorrência de escritas no texto, e as regras que caracterizam este uso pelo autor, ajudam a identificação de estilo.

- Erros de sintaxe - p.ex. busca-se identificar aqui a ocorrência de fragmentos de frases, frases de execução e tempo sem correspondência.

As características estilométricas semânticas quantificam o uso de classes, sinônimos, antônimos, hiperônimos e hipônimos em um texto, frase ou declaração. O uso combinado das características semânticas e da marcação das características das classes gramaticais auxiliam a identificação do significado semântico do texto e são muito utilizadas na tarefa de detecção de plágio.

Tabela II.5: Características estilométricas semânticas

Semânticas
Sinônimos
Antônimos
Dependências semânticas

Na Tabela II.5 são apresentadas algumas características estilométricas semânticas:

- Sinônimos - identifica o uso de palavras sinônimas no texto e quantifica a ocorrência das mesmas. O uso de sinônimos, evitando a repetição de palavras, é uma característica estilométrica relevante.

- Antônimos - identifica o uso de palavras antônimas no texto e quantifica a ocorrência das mesmas.

- Dependências semânticas - A dependência semântica entre palavras indica que se as mesmas forem separadas, ou uma delas retirada, haverá prejuízo na compreensão do sentido do texto. Identificar e quantificar a ocorrência destas dependências no texto deve ser incluída em uma avaliação de estilometria.

Há ainda características estilométricas de aplicação específica. As características estruturais capturam a organização do texto que pode ser considerado, por exemplo, como um conjunto de parágrafos. Cabeçalhos, seções, subseções e frases apoiam a verificação da estrutura de um texto.

Tabela II.6: Características estilométricas de aplicações específicas

Aplicações Específicas
Estruturais
Palavras-chave
Linguagem específica

A busca de palavras-chaves no texto, bem como da ocorrência de uso de linguagem específica, apoia a visão da estrutura do texto baseada nas ideias e conceitos que nele estão contidos.

Na Tabela II.6 são apresentadas algumas características estilométricas de aplicações específicas:

- Estruturais - Identificar as características de tamanho médio dos parágrafos em número de frases, como é feita a tabulação dos parágrafos, o uso de saudações, despedidas e assinaturas no texto são aspectos estruturais que devem ser considerados.

- Palavras-chave - A identificação de conteúdos específicos, através da localização de palavras-chaves no texto, apoia a visão da estrutura do texto baseada nas ideias que nele estão contidas.

- Linguagem específica - Em algumas ocasiões espera-se encontrar em um documento a ocorrência de determinada linguagem específica. A verificação da ocorrência, ou não, desta linguagem apoia a identificação do estilo do autor.

II.5 Medidas de avaliação

Considere um detector de plágio em documentos. A qualidade desse detector pode ser medida de forma objetiva usando diferentes medidas de avaliação.

Para uma melhor compreensão dessas medidas utilizadas considere as seguintes quantidades: `VP` - Verdadeiros Positivos, `FP` - Falsos Positivos e `P` - Total de Positivos. Todas as frases identificadas como plágio pelo detector seriam contabilizadas no *total de positivos* (`P`). As frases apontadas como plagiadas que realmente o fossem, seriam consideradas casos *verdadeiros positivos* (`VP`). Já as apontadas indevidamente como plágio representam casos de *falso positivos* (`FP`).

As medidas de avaliação utilizadas nos experimentos desta dissertação para avaliar a qualidade da detecção de passagens plagiadas consideram os contadores apresentados acima. Essas medidas são descritas a seguir.

Recall

Corresponde à razão entre a quantidade de exemplos classificados como pertencentes a uma classe que realmente são daquela classe dividido pela quantidade total de exemplos que pertencem a esta classe, mesmo que sejam classificados em outra. No caso binário, o recall corresponde à

quantidade de verdadeiros positivos divididos pelo total de positivos Potthast et al. [2010], de acordo com a Equação II.1.

$$\text{Recall} = \frac{VP}{P} \tag{II.1}$$

Precisão

Número de exemplos classificados como pertencentes a uma classe, que realmente são daquela classe (verdadeiros positivos), dividido pela soma entre este número, e o número de exemplos classificados nesta classe, mas que pertencem a outras (falsos positivos) [Potthast et al., 2010]. A precisão é computada conforme a Equação II.2.

$$\text{Precisão} = \frac{VP}{VP + FP} \tag{II.2}$$

F1-score

O F1 Score é uma média harmônica entre precisão e recall, apresentada na Equação II.3. Em geral, quanto maior for o valor da F1 score, indica um desempenho melhor [Potthast et al., 2010].

$$\text{F1} = \frac{2 * \text{Precisão} * \text{Recall}}{\text{Precisão} + \text{Recall}} \tag{II.3}$$

Granularidade

Cada caso de plágio deve ser considerado apenas uma vez pelo detector. Todavia existem situações em que um único caso real possa ser reportado múltiplas vezes, gerando uma distorção na avaliação. A métrica da Granularidade, citada em Oberreuter et al. [2012] aumenta à medida que os casos de plágio são computados mais de uma vez. O valor ideal que pode ser obtido para a Granularidade é 1,0, pois significa que cada caso de plágio foi considerado apenas uma vez. A granularidade é definida como um número médio de plágios relatados por uma passagem de texto de plágio, conforme a Equação II.4, onde S_R correspondem aos casos detectados corretamente e C_S corresponde ao número de vezes que o caso S foi detectado.

$$\text{Granularidade} = \frac{1}{|S_R|} \sum_{S \in SR} |C_S| \tag{II.4}$$

Plagdet

Recall, precisão e granularidade foram calculadas para todos os detectores de plágio que participaram do PAN11; no entanto, eles não permitem uma classificação absoluta entre eles. Para tal, as três medidas podem ser combinadas em uma única pontuação geral chamada Plagdet Potthast et al. [2011]. A forma de cálculo de Plagdet é apresentada na Equação II.5.

$$\text{Plagdet} = \frac{F1}{\log_2(\ 1 + \text{Granularity})} \tag{II.5}$$

Capítulo III Trabalhos Relacionados

Este capítulo apresenta a metodologia de pesquisa e os trabalhos relacionados aos métodos e aos problemas. A metodologia de pesquisa científica adotada, e os resultados obtidos de sua busca, estão na Seção III.1. A Seção III.2 trata dos trabalhos relacionados ao problema da tarefa de detecção intrínseca de plágio, contendo diversas propostas de solução. Na Seção III.3 são apresentados os trabalhos referentes aos métodos que estão contidos na metodologia proposta: redes LSTM, redes siamesas, incorporação de palavras (*word embeddings)*, similaridade de frases e agrupamento de correlação (*correlation-clustering*). Ao final é indicado qual a área ainda a ser melhor explorada e onde esta dissertação contribui com as suas conclusões.

III.1 Sobre a Metodologia de pesquisa

A metodologia de pesquisa adotada foi a do mapa sistemático. Foram consultadas para a busca dos trabalhos relacionados as bases: *Scopus*, *Science Direct*, *SCielo*, *IEEE xplore* e *ACM Digital Library*. A plataforma do Google Acadêmico também foi utilizada como ferramenta de busca complementar.

Inicialmente foi utilizada a *string* de busca *plagiarism detection*. As bases *Science Direct* e *IEEE Xplore* totalizaram 18 artigos. A base Scopus retornou 75 artigos. No entanto, não houve retorno para a base *ACM Digital Library* nem para a base *Scielo*. Foi realizada então a busca utilizando a *string plagiarism*, mais genérica, que retornou 66 artigos, mas sem relevância com esta pesquisa. O mesmo ocorreu com a base *ACM Digital Library*. Neste caso a *string* mais genérica retornou 151 artigos.

Esta mesma consulta realizada no Google Acadêmico retornou 1170 resultados. Refinando a busca, filtrando-a artigos a partir de 2009, o resultado foi de 1080 artigos. Considerando um número ainda excessivo, a *string Intrinsic Plagiarism Detection* foi utilizada retornando 573 artigos.

Para complementar, as mesmas bases foram consultadas com a *string Plagiarism Detection* e retornaram 507 artigos na *Science Direct* e 330 artigos na *ACM Digital Library*.

Neste cenário foram considerados novos critérios para inclusão na pesquisa: a prioridade para artigos que utilizavam na pesquisa as bases dos eventos do PAN, pois esta é a base utilizada nesta pesquisa. Um segundo critério adotado foi o de número de citações do artigo, sendo os mais

citados selecionados. Artigos que tratavam de detecção de plágio mas apenas usando a metodologia extrínseca também foram desconsiderados.

Ao final foram selecionados 236 artigos para a leitura dos resumos. Destes, mediante a avaliação de sua relação ao projeto desta pesquisa, 14 artigos foram lidos integralmente, resumidos e apresentados neste capítulo.

III.2 Trabalhos relacionados ao problema

Os trabalhos relacionados ao problema são apresentados categorizados em os vencedores das edições do PAN, relativos à tarefa de detecção intrínseca de plágio, os que tratam da diarização do autor e as pesquisas dedicadas à detecção intrínseca de plágio fora da competição do PAN.

III.2.1 Vencedores do PAN

As referências, na busca de valores de *baseline* para avaliação dos resultados obtidos pela metodologia proposta neste trabalho, são os trabalhos vencedores na tarefa de identificação intrínseca de plágio na competição do PAN, nas edições 2009, 2011 e 2016.

Tabela III.1: Resultados da Detecção de Plágio Intrínseco - PAN11

Plagdet	Precisão	Recall	Granularidade	Referencia
0.19	**0.23**	**0.46**	**1.21**	**Stamatos**
0.33	0.34	0.31	1.00	Oberreuter et al
0.17	0.43	0.11	1.03	Kestemont et al.
0.08	0.13	0.07	1.05	Akiva et al.
0.07	0.11	0.08	1.48	Rao et al.

A Tabela III.1 mostra o desempenho dos detectores intrínsecos de plágio no PAN-PC-11. Como linha de base, as colunas do trabalho de Stamatatos [2009a] mostram o desempenho do detector com melhor desempenho do PAN2009 [Potthast et al., 2011].

1. O melhor resultado alcançado no PAN2009 foi o de Stamatatos [2009a]. A ideia principal da abordagem proposta foi definir uma janela deslizante sobre o comprimento do texto e comparar o texto na janela com o documento inteiro. Assim, é criada uma função que quantifica as mudanças de estilo dentro do documento. As anomalias dessa função apoiam a detecção das seções plagiadas. O estudo utilizou um perfil de agrupamento dos caracteres de entrada de três em três. Na Figura III.1 a linha tracejada indica o limite do critério de passagem plagiado. A função binária acima indica passagens reais plagiadas (valores altos).

2. Em Kestemont et al. [2011] que também trabalha com perfis de caracteres agrupados em três, incluindo uma matriz de distâncias que apoia a identificação dos trechos que não mantém a similaridade entre si.

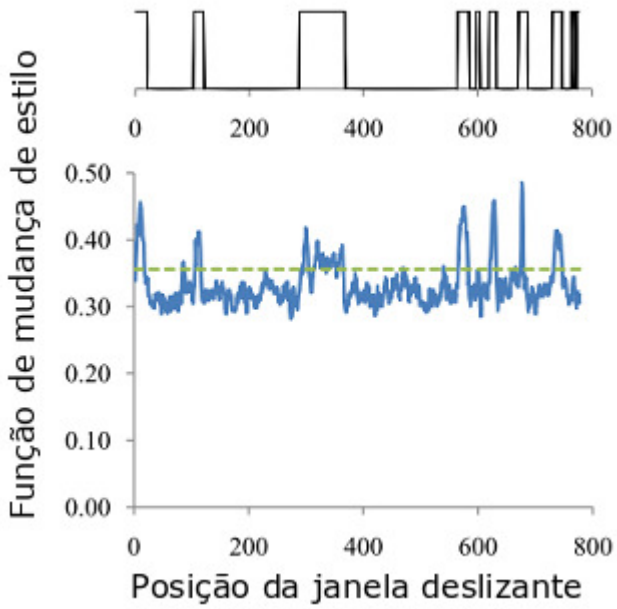

Figura III.1: Detecção Intrínseca - Stamatos 2009

3. O trabalho de detecção intrínseca que se destacou no PAN2011 foi o de Oberreuter et al. [2012], cuja principal contribuição está associada à abordagem de detecção de *outliers* para identificação de mudanças no estilo do autor.

4. Os trabalhos de Akiva [2011] e Rao et al. [2011] tiveram resultados bem inferiores na competição na tarefa de detecção intrínseca de plágio.

III.2.2 Diarização de Autor

Na competição de 2016 foi inserido na competição o conceito de diarização de autor, sendo a detecção intrínseca de plágio uma tarefa específica onde se garante que ao menos 70% do texto é de um autor principal. Para esta tarefa específica destacaram-se:

1. Kuznetsov et al. [2016] onde o trabalho investiga métodos para detecção intrínseca de plágio e diarização do autor. O método de detecção intrínseca de plágio proposto divide um documento de texto em sessões, vetoriza as frases, treina um modelo de classificação e encontra valores aberrantes na saída do classificador. Para adaptar a estrutura para o problema de diarização do autor, ele separa adicionalmente as estatísticas de saída em um conjunto de *clusters* correspondente aos diferentes autores. Se o número de autores for desconhecido, o método o estimará pela maximização da medida de discrepância do *cluster*.

2. A abordagem proposta baseou-se em uma técnica independente de linguagem para identificar de maneira exclusiva um autor baseado em seu texto escrito: Recursos Léxicos. Quinze características léxicas foram usadas em combinação com o método *ClustDist* para a detecção

de texto anômalo com o documento [Sittar et al., 2016].

A Tabela III.2 indica os resultados dos melhores trabalhos de detecção intrínseca no PAN2016.

Tabela III.2: PAN 2016 - Resultados Tarefa A

Resultados da Detecção Intrínseca de Plágio

Colocação	**Time**	**Micro-Média**			**Macro-Média**		
		Recall	Precisão	F	Recall	Precisão	F
1	Kusnetzov et al.	**0,19**	**0,29**	**0,22**	**0,15**	**0,28**	**0,17**
-	Stamatos	**0,16**	**0,25**	**0,18**	**0,15**	**0,24**	**0,16**
2	Sittar et al.	0,07	0,14	0,08	0,10	0,14	0,10

Tabela III.3: PAN 2016 - Resultados Tarefas B e C

Resultados da Diarização de Autor

Nº Autores	**Rank**	**Time**	**Bcubed**		
			Recall	Precisão	F
Desconhecido	-	Stamatos	**0.67**	**0.52**	**0.58**
(Tarefa B)	1	Kusnetzov et al.	0.46	0.64	0.52
	2	Sittar et al.	0.47	0.28	0.32
Conhecido	-	Stamatos	**0.62**	**0.56**	**0.52**
(Tarefa C)	1	Kusnetzov et al.	0.42	0.64	0.48
	2	Sittar et al.	0.47	0.31	0.35

III.2.3 Detecção intrínseca fora do PAN

Fora da competição do PAN existem trabalhos relevantes para a tarefa de detecção intrínseca de plágio.

1. Um trabalho de referência na detecção intrínseca é o de Eissen and Stein [2006]. Neste experimento, cada documento foi dividido em partes que variaram de 40 a 200 palavras. As características estilométricas tradicionais foram selecionadas:

   - estatísticas de texto, que operam no nível do personagem,
   - características sintáticas, que medem o estilo de escrita no nível da frase,
   - recursos de fala para quantificar o uso de classes de palavras,
   - conjuntos de palavras de classe fechada para contar palavras especiais e
   - características estruturais, que refletem a organização do texto.

   O principal mecanismo para definir o estilo de um autor foi baseado em uma função estatística da classe de frequência média de palavras. A classe média de frequência de palavras de um

documento nos diz algo sobre a complexidade do estilo e o tamanho do vocabulário de um autor pois ambos são características altamente individuais.

2. Em Bensalem et al. [2014] há uma forma diferente de apresentação de texto considerando agrupamentos com diferentes números de caracteres (*n-gramas*). Há uma descrição dos fragmentos de texto para que a distância entre eles seja identificada como plágio. Aqui os fragmentos de texto são representados por um vetor reduzido, onde cada característica passa por uma série de frequências de uma classe de (*n-gramas*). Portanto, a dimensão de cada vetor é igual ao número de classes e não ao número de (*n-gramas*). O algoritmo *Naïve Bayes*, implementado no software *WEKA*,[1] foi utilizado como classificador.

3. Uma proposta mais recente para a detecção intrínseca é a de AlSallal et al. [2017]. A abordagem proposta, denominada *Perceptron Multi-Layer*, refere-se à utilização de propriedades estatísticas das palavras mais comuns e através delas identificar padrões de uso que permitam definir estilos de autoria. O procedimento de geração de modelos é focado apenas em um autor, de cada vez. O conjunto de recursos do modelo intrínseco foi baseado na frequência das palavras mais comuns, suas frequências relativas na série de livros e o desvio dessas frequências em todos os livros para um autor em particular. Os resultados superaram estatisticamente os modelos Redes Bayseanas, *Support Vector Machine* e Floresta Aleatória, com uma precisão total de 97% [AlSallal et al., 2017].

III.3 Trabalhos relacionados aos métodos

Há alguns trabalhos cujo objetivo é aprender a similaridade entre frases utilizando redes siamesas, mas não com o objetivo de detecção de plágio.

1. Em Neculoiu et al. [2016] é proposta uma arquitetura profunda para aprender as semelhanças entre sequências de caracteres de tamanho variável. Redes LSTM foram implementadas nos ramos de uma rede siamesa para realizar o aprendizado. O modelo foi usado na tarefa de normalizar cargas com base em uma taxonomia anotada manualmente. Partindo de um pequeno conjunto de dados, novas fontes de variação foram incorporadas gradualmente. O modelo deve identificar semelhanças semânticas e não semânticas e ser invariante a incorreções de ortografia, substituições por sinônimos, inclusão de palavras extras supérfluas e também auto corrigível. O melhor resultado do modelo foi na sua invariância à inclusão de palavras extras [Neculoiu et al., 2016].

2. Já em Mueller [2016] é apresentada uma adaptação siamesa de uma rede LSTM para dados rotulados compostos por pares de sequências de tamanho variável. O modelo é aplicado para

[1] https://www.cs.waikato.ac.nz/ml/weka/

avaliar a semelhança semântica entre frases e excede o estado da arte, superando os modelos de características estilométricas e dos sistemas de redes neurais de maior complexidade. Para essas aplicações, são fornecidos vetores de incorporação de palavras suplementados com informações de sinônimos às LSTMs que usam um vetor de tamanho fixo para codificar o significado subjacente expresso em uma frase. Ao restringir as operações subsequentes a depender de uma simples métrica de Manhattan, força-se as representações de frases aprendidas pelo modelo a formar um espaço altamente estruturado cuja geometria reflete relações semânticas complexas [Mueller, 2016].

3. Um estudo para a identificação de similaridade semântica na língua portuguesa foi realizado pela equipe *Blue Man Group*, na competição de avaliação de similaridade semântica e inferência textual do PROPOR 2016. Considerando vetores semânticos de palavras criados com toda a Wikipédia em língua portuguesa, são seguidas duas frentes distintas. Na primeira, é implementado um conjunto de características da literatura para treinar os modelos de regressão e classificação baseados em vetores de suporte. Na segunda frente são explorados métodos de AP, tais quais redes neurais siamesas. As avaliações preliminares com os conjuntos de dados de treinamento e experimentação demonstrou que a primeira direção era mais promissora, fazendo com que a pesquisa não continuasse com a proposta usando as redes neurais siamesas [Barbosa et al., 2016].

4. A pesquisa de Kiros et al. [2015] propõe um codificador para a representação de frases em vetores: o *Skip-Thoughts*. É descrita uma abordagem para o aprendizado não supervisionado de um codificador de frases distribuído genérico. Usando a continuidade do texto, treina-se um modelo codificador-decodificador que tenta reconstruir as frases circundantes de uma passagem codificada. Frases que compartilham propriedades semânticas e sintáticas são mapeadas para representações vetoriais semelhantes. A seguir, apresenta-se um método simples de expansão de vocabulário para codificar palavras que não eram vistas como parte do treinamento, permitindo expandir o vocabulário para um milhão de palavras. Após o modelo treinado, são extraídos e avaliados os vetores com modelos lineares em tarefas distintas como: relacionamento semântico, classificação de frases de imagens, classificação de tipo de pergunta e conjuntos de dados de sentimento e subjetividade. O resultado final é um codificador pronto para uso que pode produzir representações de frases altamente genéricas, robustas e com bom desempenho na prática [Kiros et al., 2015] Importante destacar que o *skip-thoughts* é um modelo pré-treinado e que não possui relação de dependência com o corpus do PAN utilizado nesta pesquisa.

5. A aprendizagem *sequência a sequência* usando redes convolucionais está na pesquisa de Geh-

ring et al. [2017] para contribuir na tarefa de traduções automáticas do inglês para o francês, alemão e romeno. A abordagem consiste em mapear uma sequência de entrada para uma sequência de saída de comprimento variável por meio de redes neurais recorrentes, neste caso utilizando uma rede neural convolucional multi-camada. Primeiro, os elementos de entrada são incorporados em um espaço vetorial. Em seguida, a estrutura de blocos convolucional é responsável pelo processo de codificação e decodificação. Os gradientes são escalados como uma estratégia de normalização. Uma atenção em várias etapas combina a etapa de decodificação atual com a anterior.

6. Um trabalho recente encontrado relativo à similaridade entre frases usando redes siamesas é o de Ichida et al. [2018]. Nele é utiliza uma arquitetura de rede neural siamesa usando duas GRU *(Gating Recurrent Units)* que receberão como entrada IF (*word embeddings*) obtidas através do modelo *Word2vec Skip-Gram*. O dataset do experimento é o SICK dataset *(Sentences Involving Compositional Knowledge)* utilizado nos exercícios do Workshop Internacional de Avaliação Semântica de 2014(SemEVal-2014)[2], que contém 5000 pares de frases para treinamento, 500 pares para validação e 4500 pares para teste. Os resultados deste trabalho superam o dos demais apresentados no SemEval-2014. Os testes baseiam-se na identificação do grau de semelhança semântica entre frases, como por exemplo: *"a woman is slicing potatoe"* e *"a woman is cutting potatoes"* e utiliza as métricas *Pearson*, *Spearman* e *MSE*.

III.4 Discussão

A relevância do estudo apresentado nesta dissertação é justificada considerando que nos trabalhos relacionados pode-se observar que na tarefa específica de detecção intrínseca de plágio há pouca utilização de redes neurais siamesas no aprendizado do modelo, a rara aplicação de incorporação de frases, e a não utilização de técnicas de otimização combinatória, como a solução para a correlação de *clusters*, na etapa de agrupamento das frases plagiadas. Os trabalhos de Ichida et al. [2018], Gehring et al. [2017], Neculoiu et al. [2016] e Mueller [2016] apesar de utilizarem redes neurais não são aplicados na tarefa de identificação de plágio. Já os diversos trabalhos das competições do PAN 2009 e PAN 2011, na busca de soluções na tarefa de identificação intrínseca de plágio, não incorporam em sua solução modelos de AP. Em AlSallal et al. [2017] temos uma proposta de solução para a identificação intrínseca de plágio, mas fora das competições do PAN e utilizando o *CEN(Corpus of English Novel)* dataset.

A Figura III.2 apresenta um resumo das semelhanças e diferenças dos trabalhos relacionados apresentados no capítulo III.

[2]http://alt.qcri.org/semeval2014/

Trabalhos Relacionados - Semelhanças e Diferenças

Trabalho	Plágio Intínseco	Redes Neurais	Incorporação Frases	Clusterização	Dataset PAN 11	Características Estilométricas
Abordagem proposta nesta dissertação	X	X	X	X	X	
Stamatos [2009]	X				X	X
Oberreuter et al. [2011]	X				X	X
Kestemont et al. [2011]	X				X	X
Kuznetsov et al. [2016]	X					X
Sittar et al. [2016]	X			X		
Stein [2006]	X					X
Bensalem [2014]	X				X	X
AlSallal [2017]	X	X				X
Neculoiu [2016]		X				
Mueller [2014]		X				
Barbosa [2016]			X			
Kiros [2015]			X			
Gehring [2017]		X	X			
Ichida [2018]		X				

Figura III.2: Trabalhos Relacionados

Capítulo IV Redes Neurais Profundas para Detecção de Plágio

Este capítulo apresenta detalhes da abordagem para detecção intrínseca de plágio proposta nesta dissertação. A Seção IV.1 apresenta uma formalização do problema de detecção de passagens plagiadas, objeto principal de estudo desta dissertação. Seção IV.2 apresenta uma visão geral da abordagem proposta e em seguida detalha cada um de seus passos. A discussão sobre restrições de escopo deste trabalho encontra-se na Seção IV.3 deste capítulo.

IV.1 Formalização do Problema

Considere que d_q é um documento suspeito, i.e., um documento para o qual é necessário verificar se contém passagens plagiadas. Consideramos que documento d_q é composto por um conjunto de frases. Além disso, consideramos que essas frases estão reunidas em sequências que chamamos de passagens. Cada k-ésima passagem p_k corresponde a uma sequência de pelo menos uma frase em d_q e é identificada por um par ordenado de números inteiros positivos (p_k^d, p_k^c), onde p_k^d é a posição em d_q na qual o conteúdo de p_k inicia, e p_k^c é o comprimento da passagem p_k. Dizemos que duas passagens p_k e p_l são não-sobrepostas se elas não possuem frases em comum.

O problema de detecção intrínseca de passagens plagiadas corresponde a identificar $\{(p_k^d, p_k^c)\}$, o conjunto (possivelmente vazio) de passagens não-sobrepostas em d_q constituídas exclusivamente por frases plagiadas. Nas próximas seções deste capítulo apresentamos o detalhamento da abordagem que propomos para resolver o problema aqui formalizado.

IV.2 Passos da abordagem de detecção

Uma visão geral dos diferentes passos componentes da abordagem proposta nesta dissertação é apresentada na Figura IV.1. As próximas seções apresentam detalhes acerca de cada um desses passos. As primeiras etapas, relativas ao pré-processamento, a geração de um novo dataset apropriado ao nosso experimento e o processamento destes dados são detalhadas na Seção IV.2.1. O mapeamento de frases para vetores multidimensionais é apresentado na Seção IV.2.2. Na Seção IV.2.3 é explicado como se realiza treinamento do modelo para a aprendizagem de um métrica para computar a similaridade estilométrica entre frases, enquanto que na Seção IV.2.4 descrevemos como é realizada a representação de um documento suspeito d_q como um grafo. Aspectos relativos à

aplicação do algoritmo de correlação de *clusters* sobre grafo resultante da etapa anterior são apresentados na Seção IV.2.5.

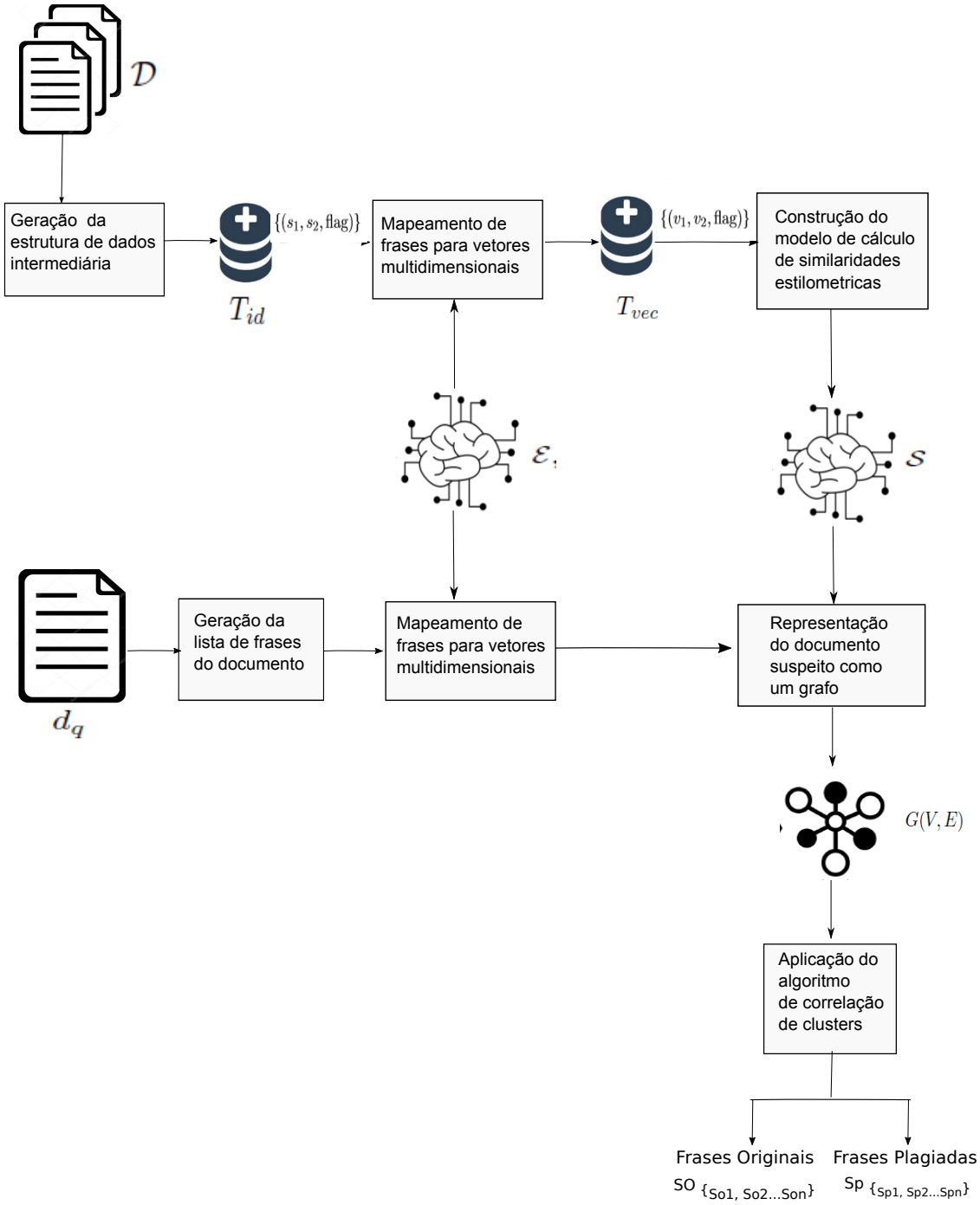

Figura IV.1: Visão geral da abordagem proposta para detecção de plágio.

IV.2.1 Geração da estrutura de triplas

Para geração do modelo de detecção, consideramos a existência de um corpus $\mathcal{D}$ (i.e., uma coleção de documentos) contendo passagens plagiadas. Consideramos também que é possível consultar o deslocamento e comprimento de cada uma dessas passagens. Além disso, para cada do-

cumento d_q, é possível identificar suas frases componentes, seu autor, e uma lista (possivelmente vazia) de passagens plagiadas. Cada passagem plagiada em um documento d_q é identificada por meio de um par de números (`deslocamento`, `comprimento`). Neste par, `deslocamento` é a posição do documento d_q na qual inicia a passagem plagiada, e `comprimento` corresponde à quantidade de caracteres da passagem. Cada passagem é composta por uma ou mais frases.

Considerando a estrutura de informações descrita acima, é possível identificar quais frases de um documento d_q são originais (i.e., são do autor do documento), e quais frases são correspondentes a plágio. Mais especificamente, é possível gerar um conjunto de dados em que cada elemento é uma tripla da forma (s_i, s_j, flag), em que s_i e s_j são os identificadores de duas frases contidas em um mesmo documento d_q, e flag $\in \{0, 1\}$ indica se s_i e s_j são ambas do autor, ou se uma delas é uma frase plagiada.

O conjunto de treinamento necessário para ajustar o modelo de similaridade estilométrica é construído a partir de um corpus $\mathcal{D} = \{d_i\}_{i=1}^{i=N}$. Estes documentos são por sua vez compostos de frases, sendo que algumas delas podem corresponder a plágio.

Inicialmente, cada documento d_i é pré-processado para gerar uma estrutura de dados intermediária que é posteriormente transformada para finalmente gerar o conjunto de treinamento necessário para o ajuste do modelo de rede neural siamesa.

Essa estrutura intermediária corresponde a um conjunto de triplas: frase 1, frase 2 e um campo indicando se as frases tem o mesmo estilo, ou não.

Para exemplificar a construção dessa estrutura de dados, considere um documento fictício composto pelas frases $\{s_1, s_2, s_3, s_4, s_5\}$. Considere ainda que as frases s_1, s_3 e s_5 sejam originais e que as frases s_2, s_4 sejam plagiadas. A parte da estrutura de dados intermediária correspondente a este documento é apresentada na Tabela IV.1.

Tabela IV.1: Estrutura de dados intermediária

s_i	s_j	***flag***
s_1	s_2	0
s_1	s_3	1
s_1	s_4	0
s_1	s_5	1
s_2	s_3	0
s_2	s_4	1
s_2	s_5	0
s_3	s_4	0
s_3	s_5	1
s_4	s_5	0

.

O procedimento exemplificado acima é executado para todos os documentos da coleção $\mathcal{D}$. O

Algoritmo 1 apresenta o pseudocódigo correspondente à geração de estrutura de dados intermediária para todos os documentos de uma coleção $\mathcal{D}$. Abaixo, são fornecidos detalhes acerca dos principais passos desse algoritmo.

Algorithm 1 Geração da estrutura de dados intermediária T_{id}

```
1: Entrada: D - uma coleção de documentos.
2: Saída: T_id - um conjunto de dados de 3-tuplas.
3: T_id ← ∅
4: for d_i ∈ D do
5:     O(d_i) ← frases originais em d_i
6:     P(d_i) ← frases plagiadas em d_i
7:     for p ∈ P(d_i) do
8:         for o ∈ O(d_i) do
9:             T_id ← T_id ∪ {(p, o, 0)}
10:        end for
11:    end for
12:    for o_1, o_2 ∈ O(d_i) e o_1 ≠ o_2 do
13:        T_id ← T_id ∪ {(o_1, o_2, 1)}
14:    end for
15:    if A(d_i) = A(d_{i+1}) then
16:        for o_1 ∈ O(d_i) do
17:            for o_2 ∈ O(d_{i+1}) do
18:                T_id ← T_id ∪ {(o_1, o_2, 1)}
19:            end for
20:        end for
21:    end if
22: end for
23: return T_id
```

A entrada do algoritmo de geração de triplas é uma coleção de documentos $\mathcal{D}$ com trechos plagiados (linha 1). Esse algoritmo produz como saída (linha 2) o conjunto de dados de 3-tuplas T_{id}.

Para cada documento $d_i \in \mathcal{D}$ (linha 4) cria-se um conjunto de frases originais em d_i (denotado por $O(d_i)$ (linha 5), e o conjunto de frases plagiadas $P(d_i)$ (linha 6).

O algoritmo combina cada frase plagiada $p \in P(d_i)$ (linha 7) com cada uma das frases originais $o \in O(d_i)$ (linha 8). Ao conjunto T_{id} é adicionado um novo elemento $(o, p, 0)$, indicando que o par não possui o mesmo estilo (linha 9). De forma análoga, para o conjunto de frases $O(d_i)$, formam-se pares (linha 12) e T_{id} recebe novo elemento correspondente à tripla $(o_1, o_2, 1)$, indicando que o par possui o mesmo de estilo (linha 13).

Se o autor do documento seguinte $\mathrm{A}(d_{i+1})$ for o mesmo autor do documento atual $\mathrm{A}(d_i)$ (linha 15), para cada frase original o_1 do conjunto $\mathrm{O}(d_i)$ é feita a combinação com cada uma das frases originais o do conjunto $O(d_{i+1})$ (linha 17). O conjunto T_{id} recebe novo elemento correspondente à tripla $(o_1, o_2, 1)$, indicando que o par possui o mesmo estilo (linha 18).

IV.2.2 Mapeamento de frases para vetores multidimensionais

Note que o conjunto de triplas T_{id} resultante da transformação descrita na Seção IV.2.1 contém em cada elemento os identificadores de duas frases. O próximo passo da abordagem é o de mapeamento de frases para vetores multidimensionais. Este passo toma T_{id} como ponto de partida para transformar o conteúdo de cada frase em um vetor por meio da aplicação de um modelo de incorporação de frases (Seção II.3.3). Denotamos esse modelo de incorporação por $\mathcal{E}$. Mais especificamente, o modelo $\mathcal{E}$ é usado para realizar uma transformação sobre cada elemento de T_{id} para gerar outro conjunto, que denotamos por T_{vec}. Cada elemento em T_{vec} é uma tripla da forma (v_i, v_j, flag), em que v_i e v_j são representações vetoriais das frases identificadas por s_i e s_j, respectivamente, e flag é conforme definido anteriormente.

O vetor correspondente a cada frase pode ser obtido por meio de algum modelo neural de incorporação de frases. Nos experimentos computacionais realizados para validação da abordagem de detecção (Capítulo V), instanciamos o modelo $\mathcal{E}$ por meio do framework Skip-Thoughts.

IV.2.3 Construção do modelo de cálculo de similaridades estilométricas

Por meio de um processo de aprendizado supervisionado, o conjunto T_{vec} é usado para ajustar os pesos de uma rede neural siamesa (Seção II.3.2). Esse processo de aprendizado gera um modelo, que denotamos por $\mathcal{S}$, capaz de medir a similaridade estilométrica entre duas frases dadas como entrada.

O processo de treinamento faz com que a rede neural aprenda a comparar as duas frases de entrada de forma estilística e determinar a similaridade entre elas.

Este modelo de identificação de similaridades pode ser aplicado a diversas outras tarefas de PLN, não sendo restrito a utilização em identificação de plágios.

IV.2.4 Representação do documento suspeito como um grafo

A realização do passo de detecção das passagens plagiadas em um documento suspeito d_q ocorre em diversas etapas. Inicialmente d_q é dado como entrada para uma função que determina a lista de suas frases componentes. Suponha que d_q contenha n frases. Cada um dos $\binom{n}{2}$ pares de frases possíveis em d_q é mapeado para sua representação vetorial e em seguida é dado como entrada para o modelo de rede siamesa previamente ajustado. O modelo de rede siamesa treinado irá gerar um valor entre 0 e 1 para cada par de frases representando a similaridade estilométrica entre elas, em que 0 mínima similaridade e 1 representa máxima similaridade. Como resultado, são produzidos $\binom{n}{2}$ valores de similaridade.

O Algoritmo 2 apresenta o pseudocódigo que gera a lista de distâncias entre pares de frases

Algorithm 2 Computação das distâncias entre frases de d_q

1: **Entrada:** d_q, $\mathcal{E}$, $\mathcal{S}$
2: **Saída:** Lista de distâncias L
3: $S(d_q) \leftarrow$ BreakDoc(d_q)
4: **for** $s_1, s_2 \in S(d_q)$ and $s_1 \neq s_2$ **do**
5: $\quad v_1 \leftarrow \mathcal{E}(s_1)$
6: $\quad v_2 \leftarrow \mathcal{E}(s_2)$
7: $\quad \delta \leftarrow \mathcal{S}(v_1, v_2)$
8: $\quad L.append(s_1, s_2, \delta)$
9: **end for**
10: **return** L

componentes de um documento suspeito d_q.

- O Algoritmo 2 recebe como entradas o documento suspeito d_q, assim como os modelos $\mathcal{E}$ e $\mathcal{S}$ de incorporação de frases e de cálculo de similaridades estilométricas (linha 1). A saída do algoritmo é uma lista L com os valores de distância entre cada combinação possível de frases de d_q (linha 2). A função BreakDoc retorna $S(d_q)$, a lista de frases em d_q (linha 3). Cada frase é então combinada com as demais frases do documento d_q gerando pares.

- Cada par (linha 4) é mapeado para seu vetor multidimensional por meio do modelo $\mathcal{E}$ (linhas 5 e 6).

- O par de vetores (v_1, v_2) é então passado como entrada ao modelo $\mathcal{S}$ que produz a distância estilométrica (δ) entre as frases correspondentes s_1 e s_2 (linha 7).

- Finalmente a tripla contendo as frases (s_1) e (s_2) e a distância (δ) obtida para o par (linha 8) é adicionada à lista L.

De posse da lista L com os $\binom{n}{2}$ valores de distância, este passo representa o documento d_q como um grafo ponderado e não-dirigido $G(V, E)$. Cada vértice de G é uma frase de d_q, e as arestas são rotuladas com o valor de similaridade entre os vértices correspondentes. Esse valor de similaridade é computado a partir do valor de distância produzido previamente pelo modelo de rede neural siamesa.

Para mapear a lista de distâncias entre frases de um documento para a lista correspondente de similaridades, inicialmente todos os elementos de L são mapeados para o intervalo $[0, 1]$. Isso é feito pela divisão de todos os valores de distância pelo maior valor de distância encontrado. Em seguida, subtrai-se $0,5$ de cada um desses valores no intervalo $[0, 1]$. O resultado são os valores de similaridades na faixa entre $-0,5$ e $0,5$. Esse mapeamento é necessário para possibilitar a aplicação adequada do algoritmo de correlação de *clusters* ao grafo produzido.

IV.2.5 Aplicação do algoritmo de correlação de clusters

Uma vez construído o grafo G correspondente ao documento d_q conforme descrito acima, G é dado como entrada para o algoritmo de correlação de *clusters*. Esse algoritmo cria uma partição dos vértices em G, que é utilizada para inferir as passagens potencialmente plagiadas em d_q.

No intuito de agrupar as frases de d_q com a mesma característica estilométrica, considera-se o grafo G como uma instância do Problema da Correlação de Clusters (PCC). Ao resolver o PCC usando o grafo correspondente ao documento d_q, são produzidos grupos e então gerado um arquivo de saída contendo as frases candidatas a plágio destacadas.

Considerando um grafo ponderado não-dirigido, $G(V, E)$, no qual um peso positivo na aresta indica nós semelhantes, enquanto pesos negativos indicam nós não semelhantes, o algoritmo de correlação de *clusters* monta *clusters* agrupando nós, maximizando contratos e minimizando discordâncias. Os contratos são obtidos somando-se as ponderações de arestas positivas em cada *cluster* com o somatório dos pesos das arestas negativas entre *clusters*). As discordâncias resultam do somatório de pesos das arestas negativas dentro de um *cluster* com a soma de pesos das arestas positivas entre *clusters*).

A formulação clássica para o problema de Correlação de *clusters* é um modelo de Programação Linear Inteira proposto para problemas de agrupamento [Demaine et al., 2006], nos quais uma variável de decisão binária x_{i_j} é atribuída a cada par de vértices $i, j \in V, i \neq j$, e definido da seguinte forma [Figueiredo and Moura, 2013]:

$$X_{i_j} = \begin{cases} 0, & \text{Se vértices i e j estão no mesmo grupo} \\ 1, & \text{caso contrário} \end{cases} \tag{IV.1}$$

Esta formulação de Programação Linear Inteira minimiza o desequilíbrio total e é descrita a seguir.

Minimize

$$\sum_{(i,j)\in A^-} W_{i_j}(1 - X_{i_j}) + \sum_{(i,j)\in A^+} W_{i_j} X_{i_j} \tag{IV.2}$$

Sujeito a

$$X_{i_p} + X_{p_j} \geq X_{i_j}, \qquad \forall \;\; i, p, j \in V, \tag{IV.3}$$

$$X_{i_j} = X_{j_i}, \qquad \forall \; i, j \in V, \tag{IV.4}$$

$$X_{i_j} \in (0, 1), \qquad \forall \; i, j \in V, \tag{IV.5}$$

A equação (IV.3) diz que: se os vértices i e p estão em um mesmo *cluster*, bem como os vértices p e j, então os vértices i e j também estão no mesmo *cluster*.

A *constraint* (IV.4) escrita para $i, j \in V$ estabelece que as variáveis x_{i_j} e x_{j_i} assumem sempre o mesmo valor nesta formulação. A *constraint* (IV.5) impõem restrições binárias às variáveis, enquanto a função objetivo (IV.2) minimiza o desequilíbrio que pode existir nas partições do grafo.

Ao contrário de outros algoritmos de *clustering*, o algoritmo de correlação de *clusters* não requer a escolha do número de *clusters* k antecipadamente porque o objetivo, para minimizar a soma dos pesos das arestas de corte, é independente do número de *clusters* [Figueiredo and Moura, 2013; Becker, 2005]. Essa característica do algoritmo é adequada para nosso problema porque não há como saber quantos grupos de frases serão formados a priori para o documento d_q.

Ao executar o algoritmo de correlação de *clusters* sobre o grafo G de um documento d_q, nosso objetivo é formar grupos a partir dos vértices de G de tal forma a minimizar a quantidade de arestas de valor negativo dentro de um grupo e minimizar a quantidade de arestas de valor positivo entre um grupo e outro. Visto que os pesos das arestas de G são provenientes do modelo de cálculo de similaridades estilométricas, isso incentiva o algoritmo a encontrar uma solução de agrupamento na qual as frases que são efetivamente do autor de d_q sejam posicionadas em um mesmo grupo. Chamemos esse grupo (de vértices correspondente a frases que são efetivamente do autor) de C_o. Do mesmo modo, o algoritmo é incentivado a posicionar as frases componentes de passagens plagiadas em grupos distintos do grupo C_o.

A Figura IV.2 ilustra um exemplo de aplicação do algoritmo de correlação de *clusters* sobre o grafo correspondente a um documento fictício composto por 10 frases. Neste exemplo, um agrupamento formado por 3 grupos é formado pela execução do algoritmo. Também neste exemplo, as frases s_1, s_3, s_8, s_9 e s_{10} seriam componentes das passagens sinalizadas como plágio em potencial.

IV.3 Discussão sobre restrições de escopo

Neste capítulo, apresentamos detalhamento da abordagem para detecção de passagens plagiadas em documentos textuais. Essa abordagem possui algumas limitações de aplicação, discutidas a seguir.

- Utilizamos na pesquisa um corpus contendo apenas documentos escritos na língua inglesa, para possibilitar avaliações comparativas de desempenho com resultados de competições internacionais que possuem esta restrição.

- É possível que um mesmo autor, ao longo dos anos, seja por experiência, ou por aquisição de conhecimentos, altere seu estilo de escrita. Se considerarmos textos escritos de um mesmo autor com intervalo de décadas, por exemplo, a partir de suas primeiras publicações, a tendência

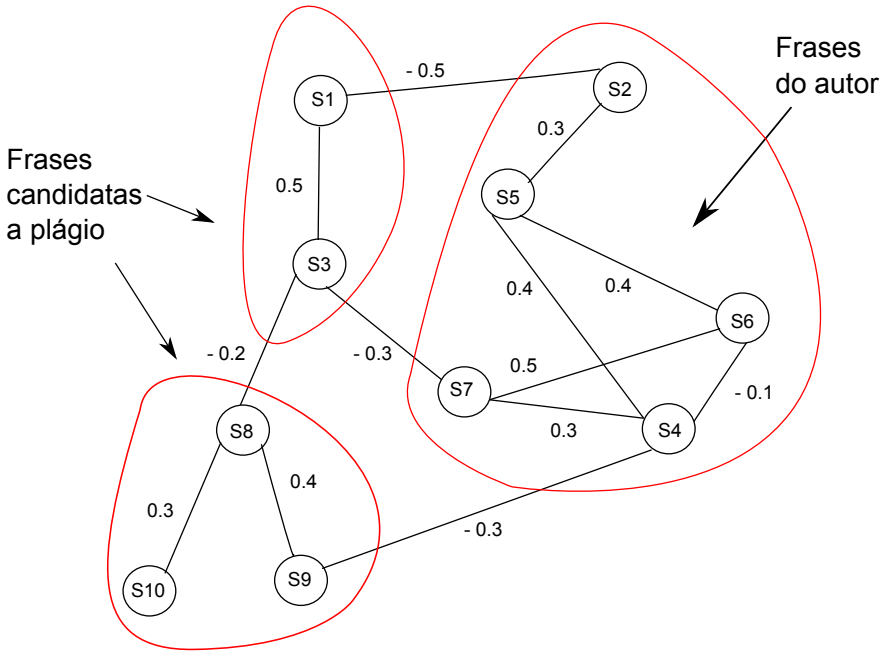

Figura IV.2: Exemplo de aplicação do algoritmo de correlação de clusters

é que essa probabilidade aumente. Na abordagem de detecção proposta nesta dissertação, esta situação geraria uma indicação de falso-positivo para plágio. Na versão atual de nossa abordagem, não há tratamento para esta situação de exceção.

- Esta pesquisa dedica-se exclusivamente à tarefa de detecção intrínseca, em detrimento da detecção extrínseca. Há um número maior de pesquisas dedicadas à tarefa de detecção extrínseca porque as pesquisas relativas a esta tarefa iniciaram-se antes, tem modelos já mais consolidados e são, portanto, hoje de mais fácil aplicação.

- Outra restrição de escopo que adotamos neste trabalho é relativa a considerarmos que cada documento possui um único autor.

- Outra pressuposição de nossa abordagem é que cada documento suspeito é formado majoritariamente por frases efetivamente do autor. Isso parece se verificar na maioria dos casos de plágio. Por exemplo, no corpus PAN11, pelo menos 70% do conteúdo de cada documento é efetivamente do autor.

Capítulo V Experimentos

Neste capítulo, são descritos os experimentos computacionais realizados para validação da abordagem de detecção proposta. Na Seção V.1 são considerados os passos necessários para a realização dos experimentos: o manuseio dos conjuntos de dados, o treinamento das redes neurais e os passos para a execução da tarefa de detecção intrínseca de plágio. Na Seção V.2, são descritos os conjuntos de dados escolhidos para uso nos experimentos e quais as motivações para a escolha desses conjuntos. Por fim na Seção V.5 são mencionados quais os resultados são objetivados com os experimentos.

V.1 Infraestrutura utilizada

Os experimentos desta dissertação foram conduzidos em uma estação de trabalho com sistema operacional Ubuntu. Esta estação possui 66 GB de RAM, 8 CPUs Intel i7 com quatro núcleos cada e uma única GPU Nvidia GeForce GTX 1080 Ti com 11 GB de memória e 3584 núcleos. Os diversos passos da abordagem proposta foram implementados com o uso das linguagens Python e C++.

V.2 Conjunto de Dados

Para validação de nossa abordagem de detecção intrínseca de plágio, utilizamos o corpus disponibilizado no PAN 2011. Esse corpus corresponde a um conjunto de 4.753 documentos, com um total de 6.620.242 frases, das quais 365.541 estão anotadas como plágio Potthast et al. [2011]. Esses documentos estão distribuídos por um total de 1733 autores.

Não havia como carregar todo o conteúdo do corpus em memória, portanto foi implementada uma solução de processamento em batch.

Cada documento está associado a dois arquivos. O conteúdo propriamente dito é fornecido como um arquivo em formato TXT. Informações relativas aos metadados do documento são fornecidas em outro arquivo, dessa vez em formato XML. Os metadados disponibilizados para cada documento são os seguintes: autor, título, idioma e indicação do deslocamento e comprimento de cada passagem correspondente a plágio. Cada documento possui apenas um autor, embora o mesmo autor possa estar associado a mais de um documento no corpus. Além disso, as passagens plagiadas podem ser

de fontes diferentes.

Um exemplo da estrutura e conteúdo do arquivo XML de metadados é apresentado na Figura V.1.

```
1 <?xml version="1.0" encoding="UTF-8">
2 <document reference="suspicious-document00001.txt">
3   <feature name="about" authors="Sheridan, Philip Henry" title="The
      memoirs of General Philip H. Sheridan, Volume II., Part 5" lang="en"
       />
4   <feature name="md5hash" value="89e25d615559fca16dec63ed8ea8cfdb" />
5   <feature name="plagiarism" type="artificial" obfuscation="none"
      this_language="en" this_offset="6620" this_length="2243" />
6   <feature name="plagiarism" type="artificial" obfuscation="none"
      this_language="en" this_offset="27922" this_length="267" />
7   <feature name="plagiarism" type="artificial" obfuscation="none"
      this_language="en" this_offset="29465" this_length="229" />
8 </document>
```

Figura V.1: Arquivo XML de metadados de um dos documentos do corpus PAN2011

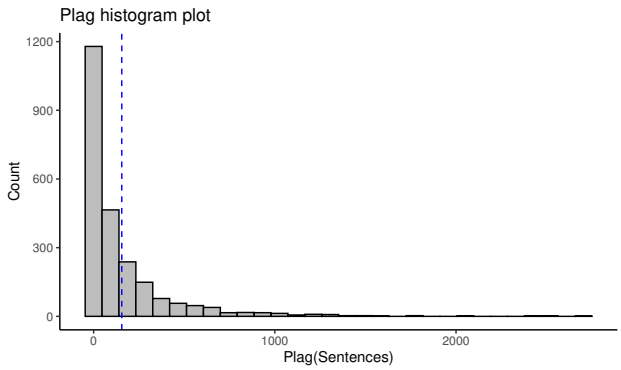

Figura V.2: Número de frases plagiadas por documento

Avaliando a quantidade de frases plagiadas em cada documento, a FiguraV.2 mostra que a maioria dos documentos (63.72%) contém até 100 frases plagiadas, e também que o número médio de frases plagiadas por documento, considerando todo o corpus, é de 77 frases.

A Tabela V.1 apresenta o sumário estatístico do corpus do PAN 2011 original. Considerando os documentos que contenham algum plágio, mais da metade deles, cerca de 57%, são casos leves de plágio pois tem entre 5% e 20% de seu conteúdo não original e os mais comprometidos, com mais de 80% de seu conteúdo plagiado, representam 10% do corpus.

A métrica de Páginas impressas (pp), *"printed pages"*, corresponde a uma página com 1000

Tabela V.1: Sumário Estatístico PAN corpus 2011.

Estatística dos Documentos		
Tipo do Documento		
documentos originais		50%
documentos suspeitos		
-com plágio		25%
-sem plágio		25%
Plagio por Documento		
leve	(5%-20%)	57%
médio	(20%-50%)	15%
alto	(50%-80%)	18%
totalmente	(>80%)	10%
Tamanho do Documento		
pequeno	(1-10 pp.)	50%
médio	(10-100 pp.)	35%
longo	(100-1000 pp.)	15%

Estatística dos Casos de Plágio		
Tamanho do caso		
pequeno	(<150 palavras)	35%
médio	(150-1150 palavras)	38%
longo	(>1150 palavras)	27%

palavras, e é utilizada na verificação do tamanho médio dos documentos no PAN. No corpus, cerca de 85% dos documentos são de até 100 pp, sendo considerados, portanto, documentos de tamanho pequeno ou médio.

Tabela V.2: Percentagem de Plágio

Percentagem de Plágio	
% plag	nºdocs
1 - 10%	580
11 - 20%	166
21 - 33%	45
34 - 50%	19
$>$ 50%	4

A Tabela V.2 apresenta uma visão da distribuição em quantidade de documentos por faixas percentuais de frases plagiadas, detalhando-se mais a faixa de até 50% de plágio, e reforça que a grande maioria dos documentos tem até 10% de conteúdo plagiado.

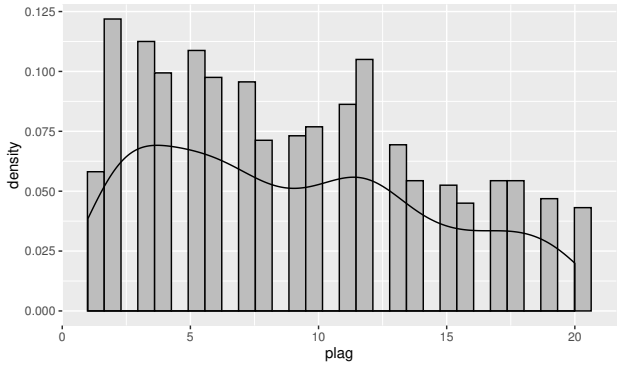

Figura V.3: Número de frases plagiadas

A distribuição de densidade é melhor quando se considera apenas documentos que contenham plágio e os restringe a até 20 frases plagiadas, como apresentado na Figura V.3, o que apoia a decisão de utilizar-se um recorte de dados para o experimento.

Após a restrição do corpus, selecionando apenas documentos que continham plágio, e até 20 frases plagiadas, novas visões puderam ser obtidas.

A Figura V.4 apresenta a nova distribuição de conteúdo das frases originais por documento. Pode-se observar que a maioria dos documentos (92,6 %) possui até 500 frases originais.

Ao inspecionar a Figura V.5, fica claro que a área mais densa na relação entre frases plagiadas e originais está nos documentos de até 200 frases originais. Esta é outra informação relevante a ser considerada na definição do recorte do corpus a ser utilizado no experimento.

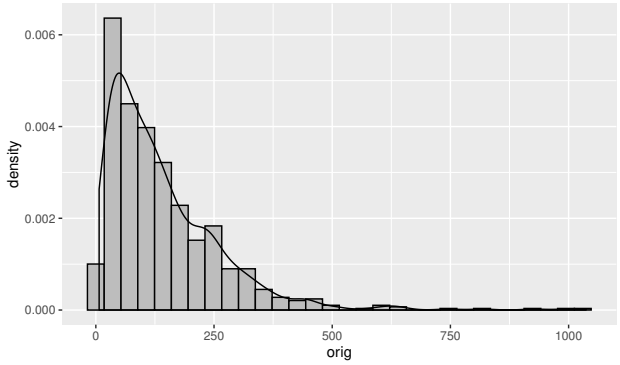

Figura V.4: Número de frases originais em documentos com até 20 frases plagiadas

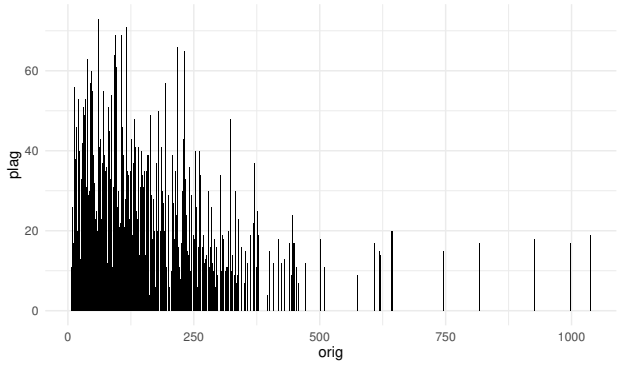

Figura V.5: frases plagiadas versus frases originais

V.3 Configuração do modelo de incorporação de frases

Para obter uma instanciação de $\mathcal{E}$, o modelo de incorporação de frases (Seção IV.2.2), utilizamos um modelo pré-treinado da rede neural Skip-Thoughts (Seção II.3.3).

Optamos por gerar os vetores de frases e armazená-los em arquivos para uso em etapas posteriores da abordagem de detecção. Para isso, primeiramente realizamos o pré-processamento de todos os documentos do corpus PAN 2011. Como atividade inicial desse pré-processamento, usamos a biblioteca NLTK[1] para fragmentar cada documento em suas frases componentes e para obter a lista de todas as frases contidas nesse corpus. Como resultado, foi formada uma lista de 6.620.242 frases. Em seguida, utilizamos a biblioteca gemsim[2] para construir o *vocabulário*, i.e., a lista de *tokens* únicos contidos naquelas frases. O vocabulário resultante contém 220.214 entradas. Configuramos a biblioteca gemsim para considerar frases de no máximo 30 (trinta) *tokens*. Como efeito, o conteúdo excedente das frases que ultrapassaram esse limite foi desconsiderado. Além disso, também configuramos essa biblioteca para considerar apenas os *tokens* com quantidade de ocorrências maior ou igual do que três.

Uma vez que cada frase f_i do corpus estava representada como uma lista l_i dos identificadores

[1]https://www.nltk.org
[2]https://radimrehurek.com/gensim/

de seus *tokens* componentes, aplicamos o modelo pré-treinado do Skip-Thougths a cada l_i para obter o vetor denso correspondente à frase f_i. Utilizamos a configuração *default* do Skip-Thoughts, que gera cada vetor de frase com 2.400 dimensões.

Alguns arquivos não puderam ser processados porque por conta de seu tamanho seu processamento demandava mais memória do que a que estava efetivamente disponível. Via de regra, documentos com mais de 5.500 frases apresentaram problemas para serem processados. Sendo assim, do total de 4.753 documentos do corpus PAN 2011, foram gerados vetores de frases para 4.358 deles.

V.4 Treinamento da rede siamesa

Com o propósito de instanciar o modelo $\mathcal{S}$ para de cálculo de similaridades estilométricas (Seção IV.2.3), utilizamos um subconjunto das triplas contidas na estrutura de dados intermediária T_{id} (previamente construídas a partir do corpus PAN 2011; ver Algoritmo 1). Para determinar esse subconjunto, adotamos dois critérios de seleção, que são descritos a seguir.

1. Em primeiro lugar, para limitar a quantidade de triplas provenientes de um dado documento, no máximo 40 (quarenta) frases plagiadas foram consideradas para cada documento. Sejam n_o e n_p as quantidades de frases originais e plagiadas em um documento. Sendo assim, as quantidades máximas de frases originais ($o_{\max}$) e plagiadas ($p_{\max}$) consideradas por documento foram calculadas conforme as expressões abaixo:

$$o_{\max} = min(n_o, 40)$$
$$p_{\max} = min(n_p, 40)$$

2. Em segundo lugar, com o propósito de gerar um conjunto de dados balanceado (i.e., um conjunto com aproximadamente as mesmas quantidades de triplas positivas e negativas), a quantidade de triplas positivas provenientes de um documento d foi no máximo igual à quantidade de triplas negativas provenientes de d.

Aplicamos o procedimento descrito acima sobre 200 documentos (selecionados aleatoriamente) do corpus PAN. Como resultado, obtivemos um conjunto de dados de 443519 exemplos. A divisão desse conjunto em conjuntos de treinamento, validação e teste foi realizada conforme descrito a seguir:

- conjunto de treinamento: 64% do total (283852 exemplos)

- conjunto de validação: 16% do total (70963 exemplos)
- conjunto de teste: 20% do total (88704 exemplos)

A arquitetura da rede neural siamesa utilizada foi tal que cada uma das duas sub-redes componentes possui a seguinte configuração:

- camada de entrada com 2400 neurônios. Esse valor se deve à dimensionalidade do vetor representativo de frases gerado pelo modelo Skip-Thoughts.
- duas camadas ocultas, cada qual composta por 128 neurônios, com função de ativação Relu, seguida de uma camada de *Dropout* com parâmetro $0,1$.
- camada de saída composta por 128 neurônios, com função de ativação Relu.

Os dois vetores (cada um de 128 dimensões) resultantes da cada sub-rede são passados como entrada para uma camada que computa a distância euclidiana entre eles. Essa distância corresponde à saída produzida pela rede siamesa. Seguem alguns outros detalhes relevantes relativos ao treinamento da rede neural siamesa.

- Como otimizador durante o treinamento, foi utilizado o RMSProp[3].
- Para controlar o superajuste (*overfitting*), foi utilizada a técnica de *Early Stopping* [Caruana et al., 2000] durante o treinamento, com parâmetro *patience* definido como 4.
- A quantidade máxima de épocas de treinamento foi definida como 40.
- O tamanho do lote de exemplos de treinamento (*batch size*) utilizado foi 128.
- A função de custo empregada foi a função de perda contrastiva (*contrastive loss*) [Hadsell et al., 2006]. Essa função de perda permite à rede aprender os parâmetros de uma métrica de distância parametrizada, de maneira que objetos similares são aproximados e não similares são mantidos separados. O conhecimento prévio reunido no conjunto de treinamento (correspondente às triplas positivas e negativas) é usado para sinalizar as proximidades entre objetos.

Os parâmetros de configuração da rede siamesa foram obtidos de forma empírica, a partir dos valores padrão da implementação utilizada, com exceção do já justificado número de neurônios na camada de entrada.

A rede siamesa acima descrita foi implementada e treinada utilizando o ambiente de AP fornecido para Google, o Tensorflow [Abadi et al., 2016]. Nós treinamos a rede neural sobre os exemplos

[3]Esse otimizador foi proposto por Geoffrey Hinton em uma de suas aulas (`https://www.cs.toronto.edu/~tijmen/csc321/slides/lecture_slides_lec6.pdf`)

do conjunto de treinamento. Configuramos o processo de treinamento para um máximo de 40 épocas, mas, devido à parada precoce (*early stopping*), o treinamento convergiu após sete épocas. A Figura V.6 e a Figura V.7 mostram a evolução dos valores do escore de precisão e da função de perda durante o treinamento do modelo, respectivamente, nos conjuntos de dados de treinamento e validação.

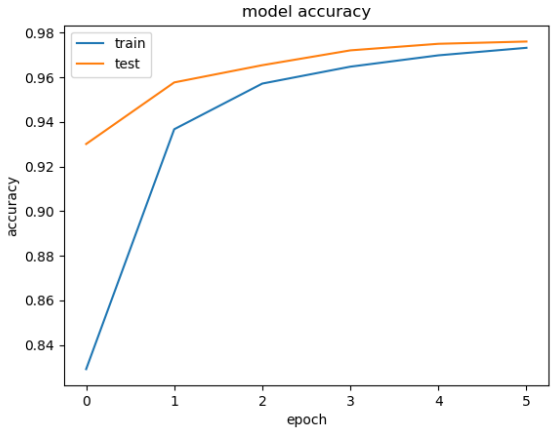

Figura V.6: Evolução da acurácia durante o treinamento.

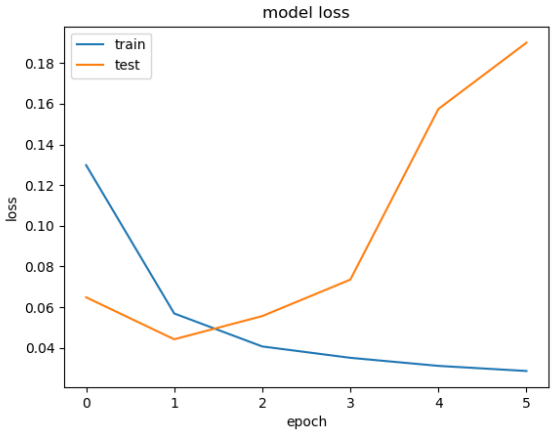

Figura V.7: Função de perda durante o treinamento.

Após o treinamento, avaliamos o modelo ajustado sobre o conjunto de testes para avaliar sua capacidade de generalização. A matriz de confusão resultante é a apresenta na Tabela V.3.

Tabela V.3: Matriz de confusão resultante da aplicação da rede siamesa ao conjunto de teste.

		Classe verdadeira		
		0	1	Total
Classe detectada	0	43489	919	44408
	1	1236	43060	44296
	Total	44725	43979	88704

A Tabela V.4 apresenta um relatório das pontuações obtidas sobre o conjunto de dados de teste

em relação à precisão, recall e escore F1. O escore F1 obtido com valor médio de 98% (considerando as duas classes) indica que o modelo de rede neural siamesa foi treinado com sucesso. Essa rede neural siamesa é a instanciação do modelo abstrato $\mathcal{S}$ descrito na Seção IV.2.3.

Tabela V.4: Relatório de Classificação

Relatório de Classificação				
	precisão	recall	F1	suporte
0	0.97	0.98	0.98	44408
1	0.98	0.97	0.98	44296

V.5 Avaliação da detecção de plágio

Os últimos passos da abordagem para detecção de plágio proposta nesta dissertação são a representação do documento suspeito como um grafo (Seção IV.2.4) e a aplicação do algoritmo de correlação de clusters. A transformação de cada documento em um grafo é realizada por meio da execução do pseudocódigo apresentado no Algoritmo 2, com o posterior mapeamento da lista de distâncias L para valores de similaridades que são usados como pesos das arestas do grafo.

O algoritmo de correlação de clusters foi implementado em linguagem C++ utilizando a biblioteca CPLEX da IBM[4]. Essa implementação recebe o grafo correspondente a um documento (em um arquivo em formato textual que contém a lista de adjacências do grafo) e produz a configuração de agrupamento encontrada. Essa configuração é então processada para identificar as passagens com plágio em potencial, conforme descrito na Seção IV.2.5.

Para validar os últimos dois passos da abordagem, consideramos o subconjunto de documentos do PAN 2011 com até 100 frases componentes. Isso gerou um total de 400 documentos. A razão pela qual essa validação foi feita em um subconjunto do corpus (em vez de ser feita no corpus como um todo) se deve à complexidade computacional do problema de correlação de clusters, que é comprovadamente da classe de complexidade NP-completo [Bansal et al., 2004].

A Tabela V.5 apresenta os resultados dos competidores da edição 2011 da competição de detecção intrínseca de plágio. Esse foi o último ano em que essa competição foi realizada. Nessa tabela, a primeira, segunda, quarta e quinta entradas apresentam o desempenho dos quatro primeiros colocados na competição (conforme medido pela métrica Plagdet, descrita na Seção II.5).[5]

Ao analisar os dados da Tabela V.5, percebe-se que, apesar do valor relativamente baixo obtido por nossa proposta, ele é comparável aos valores obtidos pelos quatro primeiros colocados na tarefa de detecção de plágio do PAN 11.

[4] `https://www.ibm.com/br-pt/analytics/cplex-optimizer`

[5] (Esses resultados estão publicamente disponíveis na tabela intitulada `INTRINSIC PLAGIARISM DETECTION PERFORMANCE` em `https://pan.webis.de/clef11/pan11-web/plagiarism-detection.html`

Tabela V.5: Desempenho de detecção intrínseca de plágio

Plagdet	**Granularidade**	**Participante**
0.3255	1.00	G. Oberreuter Universidad de Chile, Chile
0.1680	1.03	M. Kestemont, K. Luyckx, and W. Daelemans University of Antwerp, Belgium
0.1189	1.00	abordagem proposta nesta dissertação CEFET/RJ
0.0841	1.05	N. Akiva Bar Ilan University, Israel
0.0694	1.48	S. Rao, P. Gupta, K. Singhal, and P. Majumder DA-IICT, India

Capítulo VI Conclusões

Neste capítulo apresentam-se as conclusões, tendo como base os resultados dos experimentos desta pesquisa. As adversidades experimentadas e as adaptações necessárias nos modelos de referência são também relacionadas. A avaliação se os objetivos iniciais foram alcançados integralmente e as observações de situações inesperadas também são indicadas na Seção VI.1. Ao longo desta pesquisa foram identificados possíveis trabalhos futuros, que são apresentados ao final deste capítulo na Seção VI.2.

VI.1 Análise Retrospectiva

A etapa de pré-processamento gerou a necessidade de trabalho maior do que a planejada inicialmente. A geração do banco de dados relacional com a segmentação dos documentos em frases e as combinações dos pares das frases originais e das plagiadas foi refeita algumas vezes para ajustes necessários de desempenho. A quantidade necessária de registros a serem processados pela rede neural siamesa que permitisse a condição do treinamento do modelo para um aprendizado efetivo não tinha como ser carregado de uma vez todo para a memória devido a restrições do ambiente computacional. Foi realizada com sucesso a implementação de uma adaptação para que o processamento fosse realizado em *batch*. Mesmo assim, a quantidade de tuplas geradas para o processamento impedia um desempenho satisfatório. Foi realizado então uma seleção no corpus original para trabalhar com documentos que continham no máximo 20 frases plagiadas. Ao longo do processo optamos para trabalhar com a implementação de representação de *sentence embeddings* do *Skip-thoughts*. A pesquisa utilizaria as redes neurais LSTM profundas de Neculoiu et al. [2016]. Todavia utilizamos outra implementação para a rede siamesa do trabalho pela menor complexidade de adaptação. Outra necessidade de adaptação foi dos arquivos contendo as informações do grafo de saída do modelo treinado para a solução utilizada de implementação da correlação de *clusters*, cuja saída também foi adaptada para gerar os arquivos XML para aplicação nas métricas utilizadas no PAN, a fim de possibilitar as avaliações de desempenho. Consideramos que os objetivos iniciais da pesquisa foram alcançados. Os resultados obtidos são compatíveis com os trabalhos apresentados nas competições do PAN que utilizam o mesmo corpus, e obtendo relevante resultado na granularidade, comprovando assim a aplicabilidade da aprendizagem profunda na detecção intrínseca de

plágio.

VI.2 Trabalhos Futuros

Uma primeira possibilidade seria a realização de experimentos avançando na parametrização deste trabalho. Um *grid search* [Bergstra and Bengio, 2012] para os ajustes dos hiperparâmetros de configuração da rede siamesa atenderia ao propósito de otimização e identificação de melhores valores na avaliação dos resultados.

Uma das opções para trabalho futuro é a realização das mesmas tarefas para documentos em língua portuguesa. Existe uma tradicional competição internacional de detecção de plágio em língua portuguesa, a PROPOR, que em sua edição 2016 propõe tarefas de detecção de plágio. Poderíamos utilizar para a incorporação de palavras o próprio *word2vec* com o pré-treinamento realizado em português. Como corpus poderíamos realizar uma geração de *dataset* a partir do conjunto de teses e dissertações da Escola Nacional de Saúde Pública Sérgio Arouca [1], disponibilizada em seu repositório institucional ARCA [2].

Há possibilidade de utilização do *skip-thoughts* como solução de incorporação de frases em textos em Língua Portuguesa.

Na definição do dicionário de apoio à aplicação de incorporação de frases poderíamos considerar a influência das palavras sinônimas.

A verificação do desempenho do modelo de identificação de similaridades aplicado em um texto, e depois no mesmo texto automaticamente traduzido para outro idioma por alguma ferramenta disponibilizada na internet, é uma possibilidade de experimento adicional.

Existe a possibilidade de um autor cometer o autoplágio. Com o acréscimo de uma etapa de detecção extrínseca de plágio, considerando uma base de documentos conhecidos de determinado autor, trabalhando em conjunto com a detecção intrínseca desta pesquisa, identificar casos de autoplágio.

Para aumentar o volume de dados, os documentos de um mesmo autor poderiam ter suas frases combinadas entre eles para formar mais triplas, com a anotação das que possuem o mesmo estilo.

Poderíamos utilizar outra solução para a identificação das frases plagiadas que não fosse a correlação de *clusters*, a partir do grafo resultante, como por exemplo a identificação do conjunto independente máximo [Das and Chaudhuri, 2012].

Constatamos que para utilizarmos a correlação de *cluster* no documento inteiro o processamento fica muito demorado devido à complexidade do algoritmo. Uma possibilidade seria aplicarmos o algoritmo um parágrafo por vez, em vez de aplicar no documento todo conforme estamos fazendo

[1] http://http://ensp.fiocruz.br/
[2] https://www.arca.fiocruz.br/

atualmente.

Pretendemos também investigar a utilização de outras soluções de incorporação de frases que não a de *Skip-Thoughts*, como a *Sent2Vec*, apresentada em Pagliardini et al. [2018], ou então a solução BERT de [Devlin et al., 2018].

Como alternativa para agrupar as sentenças seguindo padrões identificados pelo modelo, alguma meta-heurística poderia ser aplicada em substituição ao algoritmo de Correlação de *Clusters* desta proposta. As técnicas que constituem algoritmos de meta-heurística variam de procedimentos simples de busca local a processos complexos de aprendizagem Blum [2003].

Um dos ajustes de refinamento dos pesos possíveis seria a incorporação do grau de proximidade entre pares de frases no texto. Pares de frases sem similaridade estilométrica em sequência poderiam ter mais peso na indicação de plágio do que pares de frases distantes. Em [Pang and Lee, 2004] é implementada uma função não crescente que decai com respeito à distância d entre as frases, para detectar esta influência. Os experimentos usam como alternativa f(d) = $e^(1 - d)$ e f(d) = $1/d^2$.

Referências Bibliográficas

Abadi, M., Barham, P., Chen, J., Chen, Z., Davis, A., Dean, J., Devin, M., Ghemawat, S., Irving, G., Isard, M., Kudlur, M., Levenberg, J., Monga, R., Moore, S., Murray, D. G., Steiner, B., Tucker, P., Vasudevan, V., Warden, P., Wicke, M., Yu, Y., and Zheng, X. (2016). TensorFlow: A system for large-scale machine learning. . 48

Abramson, N., Braverman, D., and Sebestyen, G. (1963). Pattern recognition and machine learning. *IEEE Transactions on Information Theory*, 9(4):257–261. 1, 13

Akiva, N. (2011). Using clustering to identify outlier chunks of text notebook for PAN at CLEF 2011. In *CEUR Workshop Proceedings*, volume 1177. 27

Alsallal, M., Iqbal, R., Amin, S., and James, A. (2013). Intrinsic Plagiarism Detection Using Latent Semantic Indexing and Stylometry. In *2013 Sixth International Conference on Developments in eSystems Engineering*, pages 145–150. IEEE. 17

AlSallal, M., Iqbal, R., Palade, V., Amin, S., and Chang, V. (2017). An integrated approach for intrinsic plagiarism detection. *Future Generation Computer Systems*. 29, 31

Alzahrani, S. M., Salim, N., and Abraham, A. (2012). Understanding plagiarism linguistic patterns, textual features, and detection methods. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, 42(2):133–149. 1, 7, 9, 18

Bansal, N., Blum, A., and Chawla, S. (2004). Correlation clustering. *Mach. Learn.*, 56(1–3):89–113. 50

Barbosa, L., Cavalin, P., Guimar, V., and Kormaksson, M. (2016). Blue Man Group no ASSIN : Usando Representações Distribuídas para Similaridade Semântica e Inferência Textual. ., 8(2):15–22. 2, 30

Becker, H. (2005). COMS E6998: Advanced Topics in Computational Learning Theory A Survey of Correlation Clustering. Technical report, .. 40

Bengio, Y. (2009). Learning deep architectures for AI. *Foundations and trends® in Machine Learning*. 14

Bensalem, I., Rosso, P., and Chikhi, S. (2014). Intrinsic Plagiarism Detection using N-gram Classes. *EMNLP*. 29

Bergstra, J. and Bengio, Y. (2012). Random search for hyper-parameter optimization. *J. Mach. Learn. Res.*, 13(null):281–305. 53

Bezerra, E. (2016). Capítulo 3 Introdução à Aprendizagem Profunda. *SBBD 2016*. 15

Bischl, B., Lang, M., Kotthoff, L., Schiffner, J., Richter, J., Studerus, E., Casalicchio, G., and Jones, Z. M. (2016). Mlr: Machine learning in r. *J. Mach. Learn. Res.*, 17(1):5938–5942. 14

Blum, C. (2003). Metaheuristics in Combinatorial Optimization: Overview and Conceptual Comparison Metaheuristics in Combinatorial Optimization. Technical Report 3. 54

Caruana, R., Lawrence, S., and Giles, L. (2000). Overfitting in neural nets: Backpropagation, conjugate gradient, and early stopping. In *Proceedings of the 13th International Conference on Neural Information Processing Systems*, NIPS'00, page 381–387, Cambridge, MA, USA. MIT Press. 48

Chowdhury, H. A. and Bhattacharyya, D. K. (2018). Plagiarism: Taxonomy, tools and detection techniques. 2

Das, K. N. and Chaudhuri, B. (2012). Heuristics to find maximum independent set: An overview. In *Advances in Intelligent and Soft Computing*, volume 130 AISC, pages 881–892. 53

Demaine, E. D., Emanuel, D., Fiat, A., and Immorlica, N. (2006). Correlation clustering in general weighted graphs. *Theoretical Computer Science*, 361(2):172 – 187. Approximation and Online Algorithms. 39

Deng, L. and Yu, D. (2014). Deep Learning: Methods and Applications. *Foundations and Trends® in Signal Processing*, 7(3-4):197–387. 14

Devlin, J., Chang, M., Lee, K., and Toutanova, K. (2018). BERT: pre-training of deep bidirectional transformers for language understanding. *CoRR*, abs/1810.04805. 54

Eissen, S. M. z. and Stein, B. (2006). Intrinsic plagiarism detection. In Lalmas, M., MacFarlane, A., Rüger, S., Tombros, A., Tsikrika, T., and Yavlinsky, A., editors, *Advances in Information Retrieval*, pages 565–569, Berlin, Heidelberg. Springer Berlin Heidelberg. 8, 28

Figueiredo, R. and Moura, G. (2013). Mixed integer programming formulations for clustering problems related to structural balance. *Social Networks*, 35(4):639–651. 39, 40

Gehring, J., Auli, M., Grangier, D., Yarats, D., and Dauphin, Y. N. (2017). Convolutional Sequence to Sequence Learning. .. 14, 30, 31

Goodfellow, I., Bengio, Y., and Courville, A. (2016). *Deep Learning*. MIT Press. `http://www.deeplearningbook.org`. 14

Hadsell, R., Chopra, S., and LeCun, Y. (2006). Dimensionality reduction by learning an invariant mapping. In *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06)*, volume 2, pages 1735–1742. 48

Hochreiter, S. and Schmidhuber, J. (1997). Long short-term memory. *Neural Comput.*, 9(8):1735–1780. 15

Hoffer, E. and Ailon, N. (2015). Deep metric learning using triplet network. In *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, volume 9370, pages 84–92. Springer, Cham. 15

Ichida, A. Y., Meneguzzi, F., and Ruiz, D. D. (2018). Measuring semantic similarity between sentences using a siamese neural network. In *2018 International Joint Conference on Neural Networks (IJCNN)*, pages 1–7. 31

Kestemont, M., Luyckx, K., and Daelemans, W. (2011). Intrinsic plagiarism detection using character trigram distance scores notebook for PAN at CLEF 2011. In *CEUR Workshop Proceedings*, volume 1177. 26

Kiros, R., Zhu, Y., Salakhutdinov, R., Zemel, R. S., Torralba, A., Urtasun, R., and Fidler, S. (2015). Skip-thought vectors. 3, 17, 30

Koch, G., Zemel, R., and Salakhutdinov, R. (2015). Siamese neural networks for one-shot image recognition. In *ICML Deep Learning Workshop*, volume 2. 16

Krause, M. (2015). Stylometry-based fraud and plagiarism detection for learning at scale. . 17

Kuznetsov, M., Motrenko, A., Kuznetsova, R., and Strijov, V. (2016). Methods for Intrinsic Plagiarism Detection and Author Diarization—Notebook for PAN at CLEF 2016. In Balog, K., Cappellato, L., Ferro, N., and Macdonald, C., editors, *CLEF 2016 Evaluation Labs and Workshop – Working Notes Papers, 5-8 September, Évora, Portugal*. CEUR-WS.org. 27

Maurer, H., Kappe, F., and Zaka, B. (2006). Plagiarism – a survey. *Journal of Universal Computer Science*, 12(8):1050–1084. 1

Meyer Zu Eissen, S. and Stein, B. (2006). Intrinsic Plagiarism Detection. ., pages 565–569. 17

Mueller, J. (2016). Siamese Recurrent Architectures for Learning Sentence Similarity. *Proceedings of the 30th Conference on Artificial Intelligence (AAAI 2016)*, pages 2786–2792. 29, 30, 31

Neculoiu, P., Versteegh, M., and Rotaru, M. (2016). Learning Text Similarity with Siamese Recurrent Networks. In *Proceedings of the 1st Workshop on Representation Learning for NLP*, pages 148–157. 29, 31, 52

Oberreuter, G., L 'huillier, G., Ríos, S. A., and Velásquez, J. D. (2012). Approaches for Intrinsic and External Plagiarism Detection Notebook for PAN at CLEF 2011. .. 23, 27

Otter, D. W., Medina, J. R., and Kalita, J. K. (2018). A survey of the usages of deep learning in natural language processing. 2

Pagliardini, M., Gupta, P., and Jaggi, M. (2018). Unsupervised learning of sentence embeddings using compositional n-gram features. *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*. 54

Pang, B. and Lee, L. (2004). A sentimental education: Sentiment analysis using subjectivity summarization based on minimum cuts. In *Proceedings of the 42Nd Annual Meeting on Association for Computational Linguistics*, ACL '04, Stroudsburg, PA, USA. Association for Computational Linguistics. 54

Potthast, M., Eiselt, A., Barrón-Cedeño, A., Stein, B., and Rosso, P. (2011). Overview of the 3rd International Competition on Plagiarism Detection. *Proceedings of the Conference on Multilingual and Multimodal Information Access Evaluation (CLEF2011), Uncovering Plagiarism, Authorship, and Social Software Misuse Worksop (PAN'11)*, pages 1–10. 11, 23, 26, 42

Potthast, M., Gollub, T., Wiegmann, M., and Stein, B. (2019). *TIRA Integrated Research Architecture*, pages 123–160. Springer International Publishing, Cham. 10

Potthast, M., Stein, B., Barrón-Cedeño, A., and Rosso, P. (2010). An Evaluation Framework for Plagiarism Detection. *Proceedings of the 23rd International Conference on Computational Linguistics COLING 2010*, I(August):997–1005. 23

Rao, S., Gupta, P., Singhal, K., and Majumder, P. (2011). External & intrinsic plagiarism detection : VSM & discourse markers based approach notebook for PAN at CLEF 2011. In *CEUR Workshop Proceedings*, volume 1177. 27

Rosso, P., Rangel, F., Potthast, M., Stamatatos, E., Tschuggnall, M., and Stein, B. (2016). Overview of pan'16. In Fuhr, N., Quaresma, P., Gonçalves, T., Larsen, B., Balog, K., Macdonald,

C., Cappellato, L., and Ferro, N., editors, *Experimental IR Meets Multilinguality, Multimodality, and Interaction*, pages 332–350, Cham. Springer International Publishing. 13

Schmidhuber, J. (2015). Deep learning in neural networks: An overview. *Neural Networks*, 61:85 – 117. 13

Sittar, A., Iqbal, H., and Nawab, R. (2016). Author Diarization Using Cluster-Distance Approach—Notebook for PAN at CLEF 2016. In Balog, K., Cappellato, L., Ferro, N., and Macdonald, C., editors, *CLEF 2016 Evaluation Labs and Workshop – Working Notes Papers, 5-8 September, Évora, Portugal*. CEUR-WS.org. 28

Stamatatos, E. (2009a). Intrinsic plagiarism detection using character n-gram profiles. *CEUR Workshop Proceedings*, 502:38–46. 12, 26

Stamatatos, E. (2009b). A survey of modern authorship attribution methods. *Journal of the American Society for Information Science and Technology*, 60(3):538–556. 19

Stamatatos, E., Tschuggnall, M., Verhoeven, B., Daelemans, W., Specht, G., Stein, B., and Potthast, M. (2016). Clustering by authorship within and across documents. In *CEUR Workshop Proceedings*, volume 1609, pages 691–715. 2

Visin, F., Romero, A., Cho, K., Matteucci, M., Ciccone, M., Kastner, K., Bengio, Y., and Courville, A. (2016). ReSeg: A Recurrent Neural Network-Based Model for Semantic Segmentation. In *2016 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 426–433. IEEE. 14

Webis at Bauhaus-Universit"at Weimar and NLEL at Universidad Polytécnica de Valencia (2009). PAN Plagiarism Corpus 2009 (PAN-PC-09). http://www.webis.de/research/corpora. Martin Potthast, Benno Stein, Andreas Eiselt, Alberto Barrón, and Paolo Rosso (editors). 11

Young, T., Hazarika, D., Poria, S., and Cambria, E. (2018). Recent trends in deep learning based natural language processing [review article]. *IEEE Computational Intelligence Magazine*, 13(3):55–75. 2