IDENTIFICANDO PLÁGIO EXTERNO COM LOCALITY-SENSITIVE HASHING

Fellipe Ribeiro Duarte

Tese de Doutorado apresentada ao Programa de Pós-graduação em Engenharia de Sistemas e Computação, COPPE, da Universidade Federal do Rio de Janeiro, como parte dos requisitos necessários à obtenção do título de Doutor em Engenharia de Sistemas e Computação.

Orientador: Geraldo Bonorino Xexéo

Rio de Janeiro
Junho de 2017

IDENTIFICANDO PLÁGIO EXTERNO COM LOCALITY-SENSITIVE HASHING

Fellipe Ribeiro Duarte

TESE SUBMETIDA AO CORPO DOCENTE DO INSTITUTO ALBERTO LUIZ COIMBRA DE PÓS-GRADUAÇÃO E PESQUISA DE ENGENHARIA (COPPE) DA UNIVERSIDADE FEDERAL DO RIO DE JANEIRO COMO PARTE DOS REQUISITOS NECESSÁRIOS PARA A OBTENÇÃO DO GRAU DE DOUTOR EM CIÊNCIAS EM ENGENHARIA DE SISTEMAS E COMPUTAÇÃO.

Examinada por:

Prof. Geraldo Bonorino Xexéo, D.Sc.

Prof. Alexandre de Assis Bento Lima, D.Sc.

Prof. Eduardo Soares Ogasawara, D.Sc.

Prof. Jano Moreira de Souza, Ph.D.

Prof. Ruy Luiz Milidiu, Ph.D.

RIO DE JANEIRO, RJ – BRASIL
JUNHO DE 2017

Duarte, Fellipe Ribeiro

Identificando plágio externo com Locality-sensitive hashing/Fellipe Ribeiro Duarte. – Rio de Janeiro: UFRJ/COPPE, 2017.

XIV, 107 p.: il.; 29, 7cm.

Orientador: Geraldo Bonorino Xexéo

Tese (doutorado) – UFRJ/COPPE/Programa de Engenharia de Sistemas e Computação, 2017.

Referências Bibliográficas: p. 97 – 107.

1. Recuperação de informação. 2. Plágio. I. Xexéo, Geraldo Bonorino. II. Universidade Federal do Rio de Janeiro, COPPE, Programa de Engenharia de Sistemas e Computação. III. Título.

À minha família e amigos,
pois sem vocês
nada disso faria sentido.

Agradecimentos

À Deus, por capacitar e prover;

À minha família, pelo carinho, confiança e incentivo;

Ao Geraldo Xexéo, pela valiosa orientação, pelo tempo, pela paciência, pelos almoços, pelos conselhos e pelo exemplo como pessoa que admiro;

À Danielle Caled, pela parceria de pesquisa e pelas contribuições que foram fundamentais para a conclusão da minha tese;

Aos professores Jano Moreira e Alexandre Assis, por participarem da minha banca de doutorado;

Ao professor Ruy Milidiu, por participar da minha banca de doutorado e pelas contribuições para com o formalismo e as demonstrações da minha tese;

À Ana Rabello, Cláudia Prata, Eliah, Gutierrez , Maria Mercedes, Patrícia Leal, Solange Santos, Sônia Galliano, por sempre me ajudarem com as questões administrativas;

Aos colegas que fiz no CEFET/RJ, pelo apoio e incentivo durante o tempo que estive por lá;

Ao Departamento de Ciência da Computação da UFRRJ, por tudo o que me foi concedido para a conclusão do meu doutorado;

Aos colegas do DCC/UFRRJ, pelo apoio e confiança;

À todos os colegas de pesquisa do LINE/LUDES, pelos papos, dicas, almoços e colaborações;

Ao Eduardo Ogasawara, por aceitar estar presente na minha banca, por toda a contribuição feita durante a minha tese, pela prestatividade e pelos conselhos que me ajudaram a me tornar um profissional e pesquisador melhor;

Ao Eduardo Bezerra, que respeito e admiro muito, pela disponibilidade, pelos conselhos, pelas sugestões e pelas contribuições que me ajudaram desde a qualificação até o processo de escrita da tese;

Ao Filipe Braida, Gustavo Guedes, Leandro Alvim, Bruno Dembogurski e Luís Fernando Orleans, pelo suporte emocional e incentivo que me ajudaram a concluir a tese.

meus sinceros agradecimentos.

Resumo da Tese apresentada à COPPE/UFRJ como parte dos requisitos necessários para a obtenção do grau de Doutor em Ciências (D.Sc.)

IDENTIFICANDO PLÁGIO EXTERNO COM LOCALITY-SENSITIVE HASHING

Fellipe Ribeiro Duarte

Junho/2017

Orientador: Geraldo Bonorino Xexéo

Programa: Engenharia de Sistemas e Computação

A tarefa de recuperação heurística tem como objetivo resgatar um conjunto de documentos dos quais a identificação de plágio externo identifica de pedaços de texto plagiado. Neste contexto, o presente trabalho apresenta os algoritmos *Minmax Circular Sector Arcs* que lidam com a tarefa de recuperação heurística como um problema de busca aproximada dos vizinhos mais próximos. Ademais, os algoritmos *Minmax Circular Sector Arcs* têm como objetivo recuperar documentos com grande quantidade de fragmentos plagiados enquanto reduz a quantidade de tempo para realizar a tarefa recuperação heurística. O ferramental teórico proposto é baseado em dois aspectos: (i) uma propriedade triangular que codifica um conjunto de esboços em um valor único; e (ii) a propriedade baseada em Arcos de Setores Circulares que melhoram a precisão de (i). Ambas as propriedades foram propostas para lidar com espaços de alta dimensionalidade, representando-os em um número pequendo de valores de *hash*. Os dois métodos *Minmax Circular Sector Arcs* aqui propostos, alcunhados de *Minmax Circular Sector Arcs Lower Bound* e *Minmax Circular Sector Arcs Full Bound* alcançaram níveis de recall singelamente mais imprecisos que o método *Minmaxwise* em troca de uma aceleração durante a indexação de documentos e da redução do tempo de extração e busca de consultas em coleções de dados de plágio de alta dimensionalidade.

Abstract of Thesis presented to COPPE/UFRJ as a partial fulfillment of the requirements for the degree of Doctor of Science (D.Sc.)

IDENTIFYING EXTERNAL PLAGIARISM WITH LOCALITY-SENSITIVE HASHING

Fellipe Ribeiro Duarte

June/2017

Advisor: Geraldo Bonorino Xexéo

Department: Systems Engineering and Computer Science

Heuristic Retrieval task aims to retrieve a set of documents from which the external plagiarism detection identifies plagiarized pieces of text. In this context, we present Minmax Circular Sector Arcs algorithms that treats HR task as an approximate k-nearest neighbor search problem. Moreover, Minmax Circular Sector Arcs algorithms aim to retrieve the set of documents with greater amounts of plagiarized fragments, while reducing the amount of time to accomplish the HR task. Our theoretical framework is based on two aspects: (i) a triangular property to encode a range of sketches on a unique value; and (ii) a Circular Sector Arc property which enables (i) to be more accurate. Both properties were proposed for handling high-dimensional spaces, hashing them to a lower number of hash values. Our two Minmax Circular Sector Arcs methods, Minmax Circular Sector Arcs Lower Bound and Minmax Circular Sector Arcs Full Bound, achieved Recall levels slightly more imprecise than Minmaxwise hashing in exchange for a better Speedup in document indexing and query extraction and retrieval time in high-dimensional plagiarism-related datasets.

Sumário

Lista de Figuras **xi**

Lista de Tabelas **xiii**

1 Introdução **1**
1.1 Contextualização 1
1.2 Definição do problema 5
1.3 Objetivos da tese 8
1.4 Contribuições 9
1.5 Organização do trabalho 10

2 Plágio **11**
2.1 Classificando plágio 12
2.1.1 De acordo com o grupo de interesse afetado 13
2.1.2 De acordo com a intenção 13
2.1.3 De acordo com a prática 14
2.2 Estratégia de análise do documento 17
2.2.1 Taxonomias de plágio 17
2.3 Áreas correlacionadas 19
2.3.1 Reúso de texto 19
2.3.2 Duplicação de documentos 20
2.3.3 Documentos Co-derivativos 21
2.3.4 Atribuição de autoria 21
2.4 Cenários de identificação de plágio 23
2.4.1 Identificando plágio intrínseco 23
2.4.2 Identificando plágio externo 24

3 Reduzindo o espaço de comparação no plágio externo **25**
3.1 Recuperação Heurística 26
3.1.1 Pré-processamento de documentos e extração de consultas . . 28
3.1.2 Extração de elementos representativos e expansão de consultas 28
3.1.3 Representação, busca e recuperação de documentos fonte . . . 30

3.1.4 Pós-processamento 31
3.2 Modelos de Recuperação de Informação em plágio 31

4 Indexação e Busca Usando *Locality-Sensitive Hashing* 35
4.1 *Semilattices* e LSH 38
4.2 Tipos LSH 41
4.3 Pipeline para Recuperação Heurística com LSH 42
4.3.1 Operadores 42
4.3.2 (*i*) Tokenização 43
4.3.3 (*ii*) Geração de *fingerprint* 44
4.3.4 (*iii*) Permutação de Características 45
4.3.5 (*iv*) Execução de Função de Seleção 46
4.3.6 (*v*) Avaliação de similaridade 47
4.4 Métodos *Minwise, Maxwise*, and *Minmaxwise hashing* 47
4.4.1 Melhorias ao se combinar os operadores de máximo e mínimo 49
4.5 Algoritmos para Indexação e Busca 51
4.6 Trabalhos relacionados na Recuperação Heurística 52

5 *Locality-Sensitive Hashing* Usando Arcos de Setores Circulares 55
5.1 Propriedade triangular: Codificando o intervalo de um *lattice* 56
5.2 Aumentando o alcance através de Arcos de Setores Circulares 58
5.3 Estimando a similaridade de Jaccard 60
5.4 Algoritmos e seu custo computacional 67

6 Avaliação Experimental 69
6.1 Coleções de dados 70
6.2 Métricas de avaliação 71
6.3 Estimando a Similaridade de Jaccard Par-a-par ($ESJP$) 73
6.4 Recuperação Heurística do Plágio Externo ($RHPE$) 77
6.4.1 Criação do índice e extração consultas 78
6.4.2 Eficiência da busca 80
6.4.3 Eficácia da busca 82
6.4.4 Considerações sobre o experimento 88

7 Conclusões 90
7.1 Sumário das contribuições 92
7.2 Trabalhos futuros 92
7.2.1 (*a*) exploração de abordagens periféricas ao motor de busca . 92
7.2.2 (*b*) paralelismo 93
7.2.3 (*c*) aplicabilidade em outros problemas 93

A Exemplo de plágio **95**

Referências Bibliográficas **97**

Lista de Figuras

1.1 Arcos de Setores Circulares representando 6 conjuntos com a mesma assinatura. 8

2.1 Taxonomia de delitos de plágio adapatada de CESKA *et al.* (2008) . 19
2.2 Taxonomia de delitos de plágio adapatada de ALZAHRANI *et al.* (2012) 20
2.3 Hierarquia dos cenários de identificação de plágio 23

3.1 Sequências de passos de um método de Recuperação Heurística. Adaptado de (EHSAN e SHAKERY, 2016) 27
3.2 grupos de características textuais para plágio externo, adaptado de ALZAHRANI *et al.* (2012) 29

4.1 Método lsh gerando três valores de *hash* para casos de plágio (estrelas), de um documento (círculo azul), e outros documentos (círculos) selecionados da coleção *Plagiarised Short Answers* (CLOUGH e STEVENSON, 2011) 36
4.2 Variando o número de *hashes* para representar 1000 documentos da coleção PAN_{10} (POTTHAST *et al.*, 2010b). Os documentos foram particionados em palavras gerando um vocabulário de 111382 palavras distintas. 38
4.3 Sequência de passos para métodos baseados em operadores 43

5.1 Interpretação geométrica da Propriedade triangular 56
5.2 Equação (5.7a): Ilustrando os Arcos de Setores Circulares para $\lfloor K_{S_i,3} \rfloor$ do exemplo (4.17) 60

6.1 Localizando passos do experimento na Figura 4.3 69
6.2 Pairwise Jaccard similarity MAE results for PSA corpus 76
6.3 Pairwise Jaccard similarity MAE results for METER corpus 76
6.4 Pairwise Jaccard similarity MAE results for PAN plagiarism corpus . 77

6.5 CRT entre o $Minmax$, o $MinmaxCSA$ e o $MinmaxCSA_L$ e o $Minwise$ para extrair uma consulta e o tempo médio, em segundos, para o $Minwise$ extrair uma consulta 79
6.6 Vazão (em documentos por hora) para indexar os documentos fontes da coleção PAN-PC-11 80
6.7 Tempo médio, na coleção PAN plagiarism corpus, de indexação (da coleção) e de extração e busca de consultas. 81
6.8 Calculando a CRT do entre o $Minmax$, o $MinmaxCSA$ e o $MinmaxCSA_L$ e o $Minwise$ para retornar: 10% (recall@3991), 25% (recall@1597), 50% (recall@7983) e 75% (recall@11975) dos documentos do índice. 82
6.9 recall@pos do resultado da Recuperação Heurística na coleção *PAN plagiarism corpus* 83
6.10 DRP para os métodos $Minmax$, CSA e CS_L calculadas a partir dos valores de recall@1597 86
6.11 DRP para os métodos $Minmax$, CSA e CS_L calculadas a partir dos valores de recall@3991 86
6.12 DRP para os métodos $Minmax$, CSA e CS_L calculadas a partir dos valores de recall@7983 87
6.13 DRP para os métodos $Minmax$, CSA e CS_L calculadas a partir dos valores de recall@11975 87

A.1 Documento fonte extraído da coleção *Plagiarised Short Answers (PSA)* (CLOUGH e STEVENSON, 2011) 95
A.2 Plágio de A.1, com poucas alterações, extraído da coleção *Plagiarised Short Answers (PSA)* (CLOUGH e STEVENSON, 2011) 96
A.3 Plágio de A.1, com muitas alterações, extraído da coleção *Plagiarised Short Answers (PSA)* (CLOUGH e STEVENSON, 2011) 96

Lista de Tabelas

1.1 Categorias dos produtos do esforço intelectual de (IFLA, 1998) 2

3.1 10 palavras do Índice Invertido Básico dos exemplos A.1, A.2 e A.3 em anexo 33
3.2 10 palavras do Índice Invertido Completo dos exemplos A.1, A.2 e A.3 em anexo 33

4.1 Examples of $O_k \subset O$. 44
4.2 Ocorrências de d_1 e d_2 no índice invertido baseado em assinaturas . . 47
4.3 Lista de métricas de similaridade entre vetores adaptada de ALZAH-RANI *et al.* (2012) 54

5.1 Comparando os estimadores: Limiar mínimos das esperanças. 66
5.2 Comparando o custo computacional para produzir valores de P hash. 68

6.1 nomeclatura dos métodos utilizada nos experimentos 70
6.2 Estatísticas das coleções de dados utilizadas nos experimentos 71
6.3 Exemplo de número de permutações necessário para selecionar 300 assinaturas. A coluna k apresenta um número de assinaturas selecionados e a coluna N apresenta o número de permutações necessárias para cada método selecionar k assinaturas. 74
6.4 RMSE da similaridade de Jaccard, assim como o tempo de seleção de assinaturas (em segundos), para a coleção PSA. 75
6.5 RMSE da similaridade de Jaccard, assim como o tempo de seleção de assinaturas (em segundos), para a coleção METER. 75
6.6 RMSE da similaridade de Jaccard, assim como o tempo de seleção de assinaturas (em segundos), para a coleção *PAN plagiarism corpus 2011.* 76
6.7 Vazão (em documentos por hora) para indexar os documentos fontes da coleção PAN-PC-11 79
6.8 CRT para 1597 82
6.9 CRT para 3991 82
6.10 CRT para 7983 82

6.11 CRT para 11975 82
6.12 Erro médio para recall@1597, seu desvio padrão (D.P.) e a porcentagem de valores de erro negativos (% neg.) 84
6.13 Erro médio para recall@3991, seu desvio padrão (D.P.) e a porcentagem de valores de erro negativos (% neg.) 84
6.14 Erro médio para recall@7983, seu desvio padrão (D.P.) e a porcentagem de valores de erro negativos (% neg.) 84
6.15 Erro médio para recall@11975, seu desvio padrão (D.P.) e a porcentagem de valores de erro negativos (% neg.) 85

Capítulo 1

Introdução

Plagiarism is sometimes a moral and ethical offense rather than a legal one since some instances of plagiarism fall outside the scope of copyright infringement, a legal offense.
— Joseph Gibaldi, (GIBALDI, 1999)

1.1 Contextualização

Estudos vem sendo feitos para descobrir o crescimento do plágio dentro das instituições de ensino e pesquisa como, por exemplo, MAURER *et al.* (2006) que compara dois *surveys* do projeto do *Center of Academic Integrity (CAI)* e afirma que, em 2005, 40% dos alunos admitiram envolvimento com plágio enquanto o mesmo estudo em 2009 apresentava uma taxa de 10% de alunos envolvidos com plágio. MAURER *et al.* (2006) também analisa o relatório, de 2003, feito na universidade de Rutgers (MUHA, 2003) onde 38% dos alunos estavam envolvidos em plágio pela internet.

O aumento das incidências de plágio nas universidades influenciou no aumento de casos de plágio em artigos submetidos e aceitos em veículos de divulgação de conteúdo científico. Fato que, quando identificado, gera um transtorno que deve ser retratado publicamente como em (BAKER *et al.*, 2008, BERRY *et al.*, 2007). Ademais, outros exemplos de danos causados pelo plágio são:

- A demissão do ministro da defesa alemão, Karl-Theodor zu Guttenberg, em 2011 após acusações de plágio em sua tese de doutorado (PIDD, 2011).
- A demissão do presidente da Hugría, Pal Schmitt, em 2012 após acusações de plágio em sua tese de doutorado (FACSAR, 2012).
- A demissão da ministra de educação alemã, Annette Schavan, em 2013 após a

universidade de Düsseldorf encontrar indícios de plágio em sua tese e destituí-la de seu título de PhD (STAFF e AGENCIES, 2013).

- A demissão do ministro da defesa de Taiwan, Andrew Yang, em 2013 por acusações de plágio encontrado em um artigo (AFP, 2013).

Portanto, o plágio não é apenas um problema de cunho acadêmico visto que as suas consequências podem gerar impacto na política e prejuízos para sociedade. Logo, o ato de plagiar deve ser definido e avaliado de várias perspectivas com o intuito de melhor entendê-lo e, assim, determinar medidas para combatê-lo. Do ponto de vista do esforço intelectual e artístico pode-se definir que:

> O plágio é a reprodução indevida[1] de um dos produtos gerados pelo esforço intelectual e artístico de outros indivíduos.

O modelo FRBR(IFLA, 1998) divide em quatro grupos, vide tabela 1.1, os possíveis produtos do esforço intelectual e artístico. Já o Exemplo 1 foi criado com o intuito de elucidar o processo de criação de um novo Trabalho. Logo, é possível utilizá-los com o propósito de discriminar de quais grupos uma reprodução indevida pode ocorrer.

Tabela 1.1: Categorias dos produtos do esforço intelectual de (IFLA, 1998)

Conceituais	Trabalho : "Criação intelectual ou artística distinta."	Expressão : "A forma intelectual ou artística especifica que um Trabalho toma quando é realizado."
Físicos	Manifestação : "Encarnação física de uma expressão de um trabalho."	Item : "Um simples exemplar de uma manifestação."

Exemplo 1. *Ao se criar um novo livro:*

> *O produto resultante do processo intelectual de se criar um novo livro é considerado um novo* ***Trabalho****. Este novo Trabalho pode ser realizado através de uma ou mais* ***Expressões*** *como, por exemplo, a escrita*[2] *ou a*

[1] sem a autorização ou atribuição de créditos ao(s) autor(es)
[2] a atividade de escrever

gravação da voz. A expressão anterior pode ser encarnada em uma ***Manifestação*** *do tipo livro eletrônico*[3] *ou em uma* ***Manifestação*** *em papel. Por último, a manifestação em papel pode apresentar vários* ***Items*** *que serão vendidos*[4]*.*

O plágio por si só não é considerado como um novo Trabalho. Visto que um grau significante de independência intelectual e esforço artístico devem ser desempenhados para se criar um novo Trabalho (IFLA, 1998). Ademais, dependendo da prática de plágio realizada, é possível reproduzir, de maneiras diferentes, um Trabalho alterando a Expressão, Manifestação ou o Item. Por exemplo, ao se parafrasear ou traduzir uma sentença de um livro para um artigo[5] o Trabalho e a Expressão não estão sendo alterados. Em contrapartida, o texto agora pertence a outra Manifestação da Expressão e, como consequência, também pertence a um Item diferente.

O comportamento do plagiador envolve manipular informação, ideias ou expressões, de outrem, para conseguir algum tipo de vantagem como, por exemplo, melhores notas (GIBALDI, 1999). Portanto, outra perspectiva interessante é analisar o ato de cometer plágio de acordo com a prática, que será discutido em 2.1.3, de onde destacamos cinco práticas comuns e indagações que devemos responder para identificá-las: (i) Existe um texto idêntico ao do texto suspeito? Se a resposta for sim, ocorreu um ato de **Copiar e colar**; (ii) Existe um texto com o mesmo significado do texto suspeito escrito de outra forma? Em caso de resposta positiva ocorreu uma **Paráfrase**; (iii) Existe um texto maior de onde o texto suspeito foi extraído? Se sim, ocorreu um **Resumo**; (iv) Existe um texto em outro idioma igual ao do texto suspeito? Se sim, ocorreu uma **Tradução**; (v) Existem vários textos (da mesma ou de várias fontes) que foram copiados no texto suspeito combinando as abordagens anteriores? Logo, **Coleções misturadas e coladas** foram geradas.

Contudo, é difícil contemplar em uma definição única o que é plágio. Isto se deve à imprecisão dos limites e a ambiguidade conceitual inerente ao plágio (POSNER, 2007). De fato, MAURER *et al.* (2006) apresenta uma variedade de definições de plágio enquanto que a taxonomia de ALZAHRANI *et al.* (2012) organiza em diferentes tipos de plágio e os divide em plágio literal ou plágio inteligente. Na prática de plágio literal, e.g. cópia de verbete ou reordenação de frase, pouco esforço é realizado para ofuscar que uma cópia foi realizada. Já no plágio inteligente a cópia é ofuscada no uso de técnicas mais complexas como, por exemplo, paráfrase, sumarização, tradução ou transformação da voz passiva para voz ativa e vice-versa (ALZAHRANI *et al.*, 2012, 2015).

[3] *e-book*

[4] é importante observar que a Manifestação em livro eletrônico também pode apresentar vários Itens que serão vendidos contudo, são Itens diferentes dos da Manifestação em papel.

[5] sem referenciar a fonte do mesmo e, portanto, cometendo plágio

A consequência desse cenário desafiador é que a identificação de plágio, assim como outras tarefas de recuperação de informação, necessita de informações de complexidades variadas para ser eficaz na tarefa de satisfazer a necessidade de informação do usuários (BAEZA-YATES e RIBEIRO-NETO, 1999, MANNING *et al.*, 2008). De fato, identificar plágio inteligente é difícil, visto que as palavras e a estrutura textual podem ser bem diferentes da composição original (ALZAHRANI *et al.*, 2015), e, portanto, dificilmente um sistema de identificação de plágio afirma que determinado documento é plágio. BARRÓN-CEDEÑO (2012) vai um pouco mais longe ao afirmar que: determinar se um fragmento de texto foi plagiado ou se uma pessoa é culpada de plágio é uma decisão de responsabilidade do especialista, isto é requer julgo humano. Todavia, é viável que o sistema em questão consiga selecionar de uma coleção extensa de documentos, ou até mesmo da internet, um subconjunto de prováveis documentos que foram plagiados e associar graus de similaridade entre o documento suspeito e os documentos apresentados. De fato, quanto maior for a similaridade, ou menor for a diferença, mais provável será que um texto seja derivado do outro (CLOUGH e STUDIES, 2003). Portanto, é de responsabilidade dos sistemas de "detecção de plágio"[6] dar suporte ao especialista na tarefa de decisão e a meta desses sistemas pode ser definida como :

> "auxiliar a detecção manual através da redução da quantidade de tempo comparando documentos[7], possibilitando a comparação de grandes quantidades de documentos e encontrando documentos fontes em recursos eletrônicos disponíveis ao sistema."(CLOUGH e STUDIES, 2003, tradução nossa)

Logo, o desafio para os Sistemas de Detecção de Plágio (SDP) está em reduzir o tempo de representar, buscar e recuperar documentos que podem ter sido copiados. Visto que, o especialista estará aguardando a lista resultante para avaliar se o delito realmente ocorreu. Nesse caso, seria interessante examinar soluções alternativas para reduzir o tempo de execução dos SDP. Isso motiva o presente trabalho, cujo o objetivo é explorar o espaço de busca representando a similaridade entre dois documentos como um problema de intersecção de conjuntos. Problema que, segundo BRODER (1997), pode ser resolvido através de um processo de amostragem de atributos conduzido de forma independente para cada documento.

[6] uma definição melhor para o termo seria "identificação de indícios de plágio". Todavia, preferi atribuir aspas ao termo alcunhado pela bibliografia para ressaltar que não tem significado literal.

[7] apresento o termo como documento, apesar da definição apresentá-lo como "texto", visto que a definição como documento abrangerá texto, código fonte ou qualquer outro elemento que seja comparável computacionalmente sem qualquer ambiguidade.

1.2 Definição do problema

Em 2012, uma pesquisa com mais de 23.000 estudantes dos E.U.A. revelou que 74% dos estudantes já copiaram tarefas de outros estudantes, enquanto que 32% já cometeram plágio da Internet (CENTER FOR YOUTH ETHICS, 2012). De fato, a Internet é uma fonte interminável de plágio, visto que ela contém uma imensa quantidade de dados disponíveis com uma agressiva taxa de crescimento. Por exemplo, em setembro de 2016, a Wikipedia apresentava mais de 550 mil novos artigos por mês (WIKIMEDIA, 2016), o que reforça que identificar plágio manualmente na Internet é impraticável.

A pesquisa de identificação de plágio em linguagem natural cresce tirando vantagem da evolução das áreas correlatas como Processamento de Linguagem Natural (PLN), Recuperação de Informação (RI) e Recuperação de Informação Multilíngue (RIM) (ALZAHRANI *et al.*, 2012). Contudo, boa parte das abordagens atuais não satisfazem as necessidades de execução em coleções de dados de alta dimensionalidade e, portanto não têm aplicação direta em SDP visto que, não são abordagens eficientes, em termos de tempo, com dados desta natureza (PAN e MANOCHA, 2012) que permeiam as coleções de documentos dos SDP. Na prática, os SDP realizam uma etapa inicial, conhecida como Recuperação Heurística (POTTHAST *et al.*, 2010a), que seleciona um subconjunto de documentos e, assim, viabiliza o uso das abordagens discutidas acima.

O problema anterior pode ser especificado como uma tarefa de identificar padrões[8] de plágio, em um documento suspeito de plágio d_q, a partir de uma coleção de documentos (POTTHAST *et al.*, 2010a). Tarefa conhecida como identificação de plágio extrínseco ou externo (PE) onde, todos os documentos fontes, i.e. os documentos que foram plagiados, estão disponíveis e são alcançáveis pelos SDP (ALZAHRANI *et al.*, 2012, POTTHAST *et al.*, 2010a). Isto é, se d_q tem passagens de texto plagiadas então todos os seus documentos fontes são acessíveis, pelos SDP, por meio de uma coleção de documentos ou de fontes online, como a internet (VANI e GUPTA, 2014).

O processo de busca e recuperação da tarefa de PE é organizado em três etapas: na etapa de Recuperação Heurística, amostras de d_q são extraídas e um conjunto de documentos candidatos, contendo passagens possivelmente plagiadas, é retornado; na etapa de Comparação Detalhada, as passagens dos documentos selecionados são comparadas com as de d_q para identificar um conjunto de passagens plagiadas; na última etapa, todas as passagens identificadas são analisadas para se remover citações e unificar as passagens contíguas de texto (STEIN *et al.*, 2007).

Portanto, na Recuperação Heurística, são extraídas amostras de um documento,

[8] e.g. pedaços, sentenças ou parágrafos plagiados

suspeito de plágio, d_q e seleciona-se um conjunto de documentos candidatos que contêm passagens possivelmente plagiadas por d_q (STEIN *et al.*, 2007). Logo, a Recuperação Heurística é caracterizada como um problema de Recuperação de Informação em que os modelos apresentam um *Framework* de representação de documentos e uma medida de similaridade (ALZAHRANI *et al.*, 2012, BAEZA-YATES e RIBEIRO-NETO, 1999, POTTHAST *et al.*, 2010a, THOMPSON *et al.*, 2015). O *Framework* modela as representações dos documentos da coleção, dos documentos suspeitos e das relações entre eles enquanto que, a medida de similaridade quantifica as relações de um documento suspeito d_q com os documentos da coleção, garantindo uma ordem para selecionar os documentos com maior chance de terem sidos plagiados por d_q.

Os métodos *Locality-Sensitive Hashing*(LSH) são modelos de Recuperação Heurística que representam documentos como conjuntos de números, que identificam o documento, conhecidos como assinaturas (ALZAHRANI *et al.*, 2015). As assinaturas são geradas a partir de um processo de amostragem de características léxicas, normalmente n-gramas de palavras, tais que o relacionamento entre dois documentos está associado com a interseção dos seus conjuntos de assinaturas, i.e. quanto maior a interseção maior é a relação entre os dois. Logo, as medidas de similaridade dos métodos LSH são as que têm relação com a interseção da representação dos documentos como, por exemplo, Jaccard (BRODER, 1997, JI *et al.*, 2013).

As medidas de similaridade baseadas em características léxicas, como as dos métodos LSH, são as mais eficientes [9] em dados de alta dimensionalidade (THOMPSON *et al.*, 2015). Contudo, como concluiu THOMPSON *et al.* (2015), não são as mais efetivas em todos os casos de plágio pois, quanto mais reescrito for o texto, mais difícil é medir similaridade de forma efetiva nessas abordagens. Por conseguinte, a identificação de casos de plágio inteligente é um dos desafios dos SDP baseados em métodos LSH. Por outro lado, características sintáticas, semânticas e estruturais vêm sendo aplicadas no problema de identificação de plágio (ALZAHRANI *et al.*, 2012, 2015, PAN e MANOCHA, 2012, POTTHAST *et al.*, 2011) e, portanto, são combinadas com métodos LSH para alcançar este objetivo.

Minwise hashing (BRODER, 1997) é o método LSH mais difundido. Nele o problema da interseção é resolvido através de um processo de amostragem, baseado em permutações aleatórias, que é conduzido de forma independente para cada documento (BRODER, 1997). Ademais, *Minwise hashing* e a similaridade de Jaccard são altamente correlacionados visto que a probabilidade do *Minwise hashing* produzir o mesmo valor para dois conjuntos é igual ao valor da sua similaridade de Jaccard (LESKOVEC *et al.*, 2014). Contudo, existem outras propriedades, além do

[9] no presente trabalho usa o termo eficiência em referência a tempo, para realizar a tarefa, enquanto que efetividade está relacionado com quantidade de acertos

valor mínimo, que podem ser utilizadas como, por exemplo, o método *Minmaxwise hashing* que combina os valores de mínimo e máximo para obter representações de forma mais rápida e com resultados ligeiramente melhores no problema de encontrar itens similares (JI *et al.*, 2013) e reúso jornalístico (VIEIRA, 2016).

Nesse contexto, permutar diversas vezes o vocabulário aumenta o tempo para representar a coleção de documentos e o tempo de busca. Portanto, selecionar mais valores de assinaturas, por permutação, é uma forma de se gerar o mesmo número de assinaturas utilizando um número menor de permutações e, consequentemente, reduzir o tempo de representação e busca. O enfoque desse trabalho se insere em reduzir o tempo de permutação representando o intervalo de valores como um número, além do mínimo e do máximo, gerados a partir de uma interpretação geométrica baseada em arcos de setores circulares, como a da figura 1.1 que ilustra 5 conjuntos representados por pontos que pertencem a setores circulares com arcos delimitados por uma região cinza. Logo, um valor numérico de assinatura representa a região cinza levando os pontos 1 e 3 que apresentam o mesmo valor de máximo e, portanto, estão mesmo Arco de Setor Circular a terem a mesma assinatura assim como os pontos 1 e 2 que apresentam o mesmo mínimo mas não o mesmo máximo a terem a mesma assinatura. Logo, as assinaturas geradas a partir dos Arcos de Setores Circulares podem ser utilizadas junto com os valores de mínimo e máximo reduzindo assim o número de permutações necessários para selecionar determinado número de assinaturas.

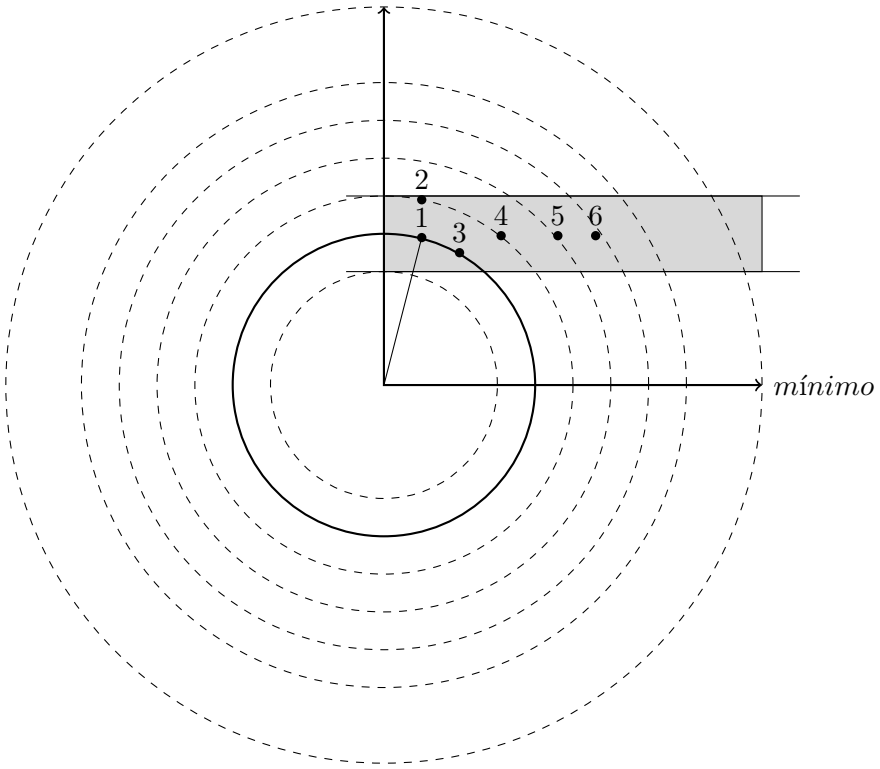


Figura 1.1: Arcos de Setores Circulares representando 6 conjuntos com a mesma assinatura.

1.3 Objetivos da tese

O Objetivo geral deste trabalho é explorar o espaço de busca da Recuperação Heurística representando a similaridade entre dois documentos como um problema de intersecção de conjuntos. Para tanto, os métodos LSH foram utilizados para gerar valores de assinaturas que compõem os conjuntos gerados a partir de cada documento. De maneira mais específica, o objetivo da tese é demonstrar que gerar mais valores de assinatura, por permutação, aumenta a eficiência da execução da Recuperação Heurística.

O Objetivo geral da tese foi dividido nos objetivos a seguir:

- **Avaliar a eficácia e a eficiência dos métodos *Minwise e Minmaxwise hashing* na Recuperação Heurística:** Avaliar a eficácia i.e., avaliar se identificam plágio, e a eficiência (tempo de indexação, de extração consultas e de busca) ao se executar os dois métodos LSH mais conhecidos na etapa de Recuperação Heurística visto que este tipo de avaliação não foi encontrado na literatura vigente.

- **Propor novos valores de assinaturas que podem ser gerados a partir de uma permutação:** Propor um arcabouço teórico que dê suporte à

geração de valores de assinaturas que representam documentos e respeitam a similaridade entre dois documentos.

- **Avaliar a eficácia e a eficiência dos métodos baseados em Arcos de Setores Circulares na Recuperação Heurística:** Avaliar a eficácia e a eficiência, na Recuperação Heurística, dos métodos gerados a partir do arcabouço anterior e comparar seus resultados com os métodos *Minwise e Minmaxwise hashing*.

As contribuições centrais desta tese são os métodos baseados em Arcos de Setores Circulares. Logo, o último objetivo listado avalia se os métodos baseados em Arcos de Setores Circulares apresentam benefícios ao serem utilizados na etapa de Recuperação Heurística. Para tanto, a seguinte hipótese foi proposta:

Hipótese 1. *A partir do conjunto de características de um documento suspeito de plágio e o conjunto de características de seu documento fonte, a combinação da propriedade geométrica baseada em arcos de setores circulares com os valores mínimo e máximo do conjunto de características possibilita a produção da mesma quantidade de valores de assinatura por documento, utilizando um número menor de permutações que o método Minmaxwise hashing para identificar o plágio.*

1.4 Contribuições

Esta tese está contextualizada na área de identificação de Plágio Externo, contribuindo nessa área por apresentar:

i) Adaptação dos métodos *Minwise e Minmaxwise hashing* na etapa da Recuperação Heurística combinando-os com índices invertidos e avaliando a sua eficácia e eficiência;

ii) A propriedade geométrica baseada em arcos de setores circulares, em inglês *Circular Sector Arcs*(CSA) para o problema de *Locality-Sensitive hashing* e todo o seu arcabouço teórico;

iii) Um algoritmo para quantificar a propriedade geométrica chamado de *Minmax Circular Sector Arcs Lower Bound* ($MinmaxCSA_L$);

iv) Um algoritmo para quantificar a propriedade geométrica chamado de *Minmax Circular Sector Arcs Full Bound* ($MinmaxCSA$);

v) Avaliação dos métodos $MinmaxCSA$ e $MinmaxCSA_L$ no problema da Recuperação Heurística.

1.5 Organização do trabalho

O presente trabalho está organizado em 7 capítulos: no capítulo 2 são descritos o conceitos relacionados com plágio; o capítulo 3 apresenta a etapa de Recuperação Heurística do plágio externo e apresenta os modelos de Recuperação de Informação para a etapa; o capítulo 4 apresenta a família de métodos LSH, seus algoritmos para indexar e buscar e os trabalhos relacionados com os modelos LSH na Recuperação Heurística; no capítulo 5 as definições, formalizações e os métodos propostos, pela presente tese, são formalizados e explicados; no capítulo 6 os experimentos conduzidos e avaliados são apresentados; enquanto que o capítulo 7 apresenta as conclusões sobre os experimentos e os trabalhos futuros identificados durante a tese.

Capítulo 2

Plágio

> *The wrong in plagiarism lies in misrepresenting that a text originated from the person claiming to be its author when that person knows very well that it was derived from another source, knows also that the reader is unlikely to know this, and hopes to benefit from the reader's ignorance.*
>
> — Pamela Samuelson, (SAMUELSON, 1994)

Plágio é a prática de tomar o trabalho ou as ideias de outrem e fazê-las passar como próprias (OXFORD, 1999). Ainda de acordo com OXFORD (1999) e HARPER (2001) a palavra teve origem no início do século XVII do latim *plagiarius* "sequestrador, sedutor, saqueador, aquele que sequestra a criança ou o escravo de outro" que foi utilizada pelo poeta Martial[1] no sentido de "roubo literário"(HARPER, 2001) ou de forma mais abrangente por GIBALDI (1999) como roubo intelectual. De fato, utilizar de ideias, informações ou expressões alheias em benefício próprio, com o objetivo de obter vantagens ou melhorar uma nota, constitui fraude (GIBALDI, 1999). O que leva a discussão do problema além do âmbito acadêmico, sendo discutido, inclusive, na esfera judicial como no livro (POSNER, 2007).

Copiar disfarçando o ato da cópia (AURÉLIO, 2001); usar de palavras ou ideias sem dar o devido crédito ao autor (MERRIAM-WEBSTER, 2017); "Cometer furto literário, apresentando como sua uma idéia ou obra, literária ou científica, de outrem." (MERRIAM-WEBSTER, 2017); "Usar obra de outrem como fonte sem mencioná-la." (MERRIAM-WEBSTER, 2017); e "Imitar, servil ou fraudulentamente." (MERRIAM-WEBSTER, 2017) são outras formas de se alcunhar o termo plágio. Uma lista de sinônimos para plágio também é apresentada em (DICTIONARY.COM, 2017), eles são: apropriação, violação, pirataria, falsificação, roubo,

[1] http://en.wikipedia.org/wiki/Martial

empréstimos, surrupiar e uso indevido.

Por último destacamos a definição 1 que entendemos como a que apresenta a questão do plágio de forma mais adequada para o propósito deste trabalho:

Definição 1. *Segundo DICTIONARY.COM (2017), plágio é :*

1. *Um ato ou instância de utilização, ou de imitação próxima, da linguagem e pensamentos de outro autor, sem autorização, e a representação do trabalho de outrem como seu próprio sem o devido crédito ao real autor.*

2. *Um pedaço de escrita, ou de outro trabalho, que reflete uso ou imitação não autorizada.*

Com o intuito de melhor compreender as facetas inerentes às práticas do plágio este capítulo é dividido em 3 seções. A seção 2.1 apresenta perspectivas diferentes de se analisar o plágio, a seção 2.3 apresenta áreas de pesquisa próximas a de identificação de plágio e a seção 2.4 discute os cenários em que plágio pode ser identificado.

2.1 Classificando plágio

O plágio é um assunto de cunho multidisciplinar e, apesar dele ser bem caracterizado em algumas situações, é importante considerar que existem várias formas de analisar e compreender o mesmo. Por exemplo, MAURER *et al.* (2006) elenca que em áreas como a jurídica e a literária um artigo consiste de uma conjectura seguida de citações, de outras fontes, que validem ou refutem a tese proposta. Enquanto que em outras áreas como as engenharias e a ciência da computação a contribuição real está mais focada no equipamento ou no algoritmo proposto ao invés de apenas na descrição do porquê o problema é importante.

Um tipo peculiar de prática de plágio é o plágio de código fonte de *software* que é mais simples de detectar do que plágio em linguagem natural (CLOUGH, 2000) e, portanto, está fora do escopo deste trabalho [2]. Outras práticas associadas ao plágio são polêmicas como, por exemplo, o *Ghostwriting* e o auto plágio. A prática de *Ghostwriting* é comum na escrita de discursos políticos e nas rotinas de apresentação de comediantes conhecidos onde os reais autores raramente recebem os créditos pela obra (MARTIN, 1994). CLOUGH (2000) cita um exemplo nocivo de auto plágio onde um autor reaproveitou vários artigos anteriores e submeteu para uma conferência um artigo sem uma citação do que foi reaproveitado. Logo, o auto plágio é uma das razões pelas quais várias revistas profissionais têm procedimentos

[2] uma discussão um pouco mais detalhada sobre o plágio de código fonte pode ser encontrada em (CLOUGH, 2000, CLOUGH e STUDIES, 2003, PARKER e HAMBLEN, 1989).

de revisões por pares (SAMUELSON, 1994). Durante o processo, a revisão tem como objetivo não permitir que ocorram múltiplas publicações do mesmo autor com o mesmo material (SAMUELSON, 1994). Outro instrumento utilizado o editor tomar posse de parte dos direitos autorais[3](CLOUGH, 2000) quando um autor publica um texto, e, portanto, se um autor reutilizar o texto do artigo, o editor tem o direito de cobrar os encargos dos direitos autorais do autor (CLOUGH, 2000).

A partir dos exemplos anteriores é possivel notar que a fronteira do que é plágio é surpreendentemente turva (MAURER *et al.*, 2006). Portanto, nas próximas seções analisaremos as diferentes perspectivas do plágio com o intuito de entender qual abordagem pode ser utilizada em cada caso.

2.1.1 De acordo com o grupo de interesse afetado

De acordo com CAVANILLAS (2008) o ato de plagiar afeta dois grupos de interesses : os interesses do autor e os interesses do público alvo do trabalho.

Nos interesses do autor, plagiar afeta diretamente as questões de posse e exploração dos direitos do trabalho (CAVANILLAS, 2008). Já para os interesses do público alvo, o ato é fraudulento, pois leva o público alvo a crer que o trabalho pertence a quem plagia e não ao real autor (CAVANILLAS, 2008).

2.1.2 De acordo com a intenção

Plágio nem sempre é intencional, em alguns casos ele pode ser feito de forma não intencional, por falta de conhecimento sobre trabalhos existentes, ou acidental (MAURER *et al.*, 2006). Um exemplo a destacar é a *cryptomnesia* que é definida por TAYLOR (1965) como :"A existência de memórias escondidas da consciência". Por conseguinte, as memórias envolvidas na *cryptomnesia* não são mais reconhecidas como memórias mas são vivenciadas como ideias recentemente criadas, o que pode ser caracterizado como plágio não intencional (TAYLOR, 1965). Portanto, MAURER *et al.* (2006) caracteriza o plágio de acordo a intenção do autor em quatro categorias: O **plágio acidental** é gerado pela falta de conhecimento sobre o que é plágio ou de não compreender as regras de referências adotadas em um instituto; o **plágio não intencional** é gerado pela vasta gama de conhecimento disponível, que pode influenciar nos pensamentos e levar a mesma ideia a ser gerada em indivíduos diferentes; O **plágio intencional** é um ato deliberado de copiar parte ou o trabalho completo de outrem sem atribuir o crédito ao criador original; e o **auto plágio** que usa um ou mais trabalhos previamente publicado sem referenciá-los.

Um fato importante a destacar é que o ato do plágio, assim como outras características de autoria, é uma construção cultural e expressão ideológica de um modelo

[3]em inglês *copyrights*

da sociedade, em determinado momento da história. Logo, o plágio pode ser influenciado por fatores sociais, econômicos ou políticos (SUREDA e COMAS, 2008). Além do mais, em qualquer expressão criativa existem dois pares dicotômicos de termos e concepções: a imitação em contrapartida com a realidade e a colaboração em contrapartida com a autonomia (SUREDA e COMAS, 2008). Portanto, dependendo do contexto social em que esteja o autor da nova obra, a imitação pode se tornar uma prática comum. E, consequentemente, levar ao desconhecimento do autor de que a prática de plágio não é aceitável ou, até mesmo, ao desconhecimento do que é plágio. Um exemplo de prática gerada pela construção cultural são as sátiras onde a qualidade está relacionada com quão parecida a imitação é de forma que seja possível identificar o alvo da cópia.

Existem fatores externos ao sistema educacional que influenciam na prática do plágio (SUREDA e COMAS, 2008) como, por exemplo: 1. a ideia amplamente adotada pelos jovens de que tudo que está disponível na internet pertence a todos e pode ser emprestado, usado, empossado e disseminado à vontade; 2. modelos e esquemas sociais baseados na cultura da simples reprodução ao invés da cultura de reprodução com criação; 3. exemplos negativos, disponíveis diariamente, geralmente baseados na falta de ética em várias áreas como a corrupção política, a fraude acadêmica, especulação nas finanças, guerras justificadas por falsas evidências entre outros (SUREDA e COMAS, 2008).

Contudo, também existem fatores intrinsecamente relacionados ao sistema educacional que influenciam na prática de plágio, pelos alunos, e que devem ser combatidos (SUREDA e COMAS, 2008). Estes fatores são agrupados de forma interessante em BARRÓN-CEDEÑO (2012) como : *a*) **orientados ao professor**, em que o problema reside nas estratégias de ensino e no modelo de atribuição de tarefas (normalmente reflete a falta de comprometimento nas atribuições anteriores); *b*) **orientados ao aluno**, onde o problema se encontra nas atitudes do aluno com relação à escola e ao processo de aprendizagem; *c*) **orientados ao sistema educacional**, em que o problema está na falta de regras, políticas e instruções claras por parte da instituição educacional. BARRÓN-CEDEÑO (2012), assim como CAVANILLAS (2008), SUREDA e COMAS (2008), discute de forma mais minuciosa a relação da postura dos professores em relação ao plágio.

2.1.3 De acordo com a prática

Antes de iniciar qualquer discussão sobre as práticas comuns de plágio, deve-se analisar em quais condições o plágio se encontra. LEUNG e CHAN (2007) classificou as condições em questão em três grupos diferentes, eles são:

Condição 1. ***Cópia de uma fonte que não apresenta versão eletrônica***[4]***:***

A cópia é realizada, de forma manual, de uma fonte que não tem representação digital (algo impresso como um livro) o que leva ao avaliador, na maioria dos casos o professor, a identificar o plágio caso ele tenha lido a fonte e lembre da mesma.

Condição 2. ***Cópia direta de uma fonte com versão eletrônica :*** *A cópia é realizada de uma fonte com representação digital e existe a possibilidade do plágio ser detectado caso o conteúdo copiado seja acessível ao sistema detecção de plágio.*

Condição 3. ***Cópia de uma fonte com versão eletrônica e alteração de conteúdo intencional :*** *O conteúdo copiado é intencionalmente modificado para evitar que o plágio seja detectado. As modificações mais comuns são na estrutura da sentença, como na voz e no tempo, e substituições por sinônimos. Por exemplo:*

> *[Sentença original] Peter eats many apples. [Mudança no tempo] Peter ate many apples. [Mudança por sinônimo] Peter consumes many apples.*

A partir das condições 2 e 3 o ato de plagiar pode ser classificado como[5]:

a) **Copiar e colar :** É a prática de plágio mais comum, fácil de fazer (WEBER-WULFF, 2010), e provável de acontecer(MARTIN, 1994) e é definida por MAURER *et al.* (2006) como:

> "Copiar palavra por palavra do conteúdo textual."

MARTIN (1994) a define como "plágio de palavra-por-palavra" que:

> "... ocorre quando alguém copia frases ou passagens de um trabalho publicado sem o uso de aspas, sem reconhecer a fonte ou ambos."

b) **Plágio de fontes secundárias :** O autor cita a fonte original do trabalho sem informar a fonte secundária de onde obteve a informação sobre a citação(MARTIN, 1994).

c) **Plágio da forma da fonte :** O autor usa a estrutura de argumentação da fonte secundária, olhando e citando a fonte primária do texto, mas não indica que existe uma dependência das citações da segunda fonte (MARTIN, 1994).

d) **Plágio artístico :** Apresentar o trabalho de outrem usando mídias diferentes como texto, imagens, voz ou vídeos (MAURER *et al.*, 2006).

[4]esta condição não pode ser solucionada por sistemas computacionais e portante está fora do escopo deste trabalho.

[5]todas as práticas aqui listadas podem ser realizadas na condição 1.

e) **Plágio de idéia :** É um pouco mais abstrato que o plágio artístico, pois um pensamento original de outra pessoa é usado sem qualquer dependência ou referência a fonte (MARTIN, 1994). Para MAURER *et al.* (2006) é caracterizado ao se usar conceitos ou opiniões similares que não são conhecimento comum.

f) **Plágio de tradução :** Utilizar conteúdo traduzido sem referenciar o trabalho original (MAURER *et al.*, 2006). Para tanto, o ator pode utilizar ferramentas de tradução, como o Google Translator [6], para produzir um rascunho grosseiro de tradução e, em seguida, corrigir as seleções de palavras incorretas e os erros gramaticais (WEBER-WULFF, 2010). Em alguns casos, o trabalho de tradução é tão grande que o ator acredita que tem méritos pelo que está fazendo (WEBER-WULFF, 2010). Contudo, o trabalho gerado não é original já que ainda está baseado no trabalho de outro alguém WEBER-WULFF (2010).

g) **Paráfrase :** É quando algumas palavras são modificas, mas não muito, sem que a fonte original seja citada (MARTIN, 1994). Ademais, segundo MAURER *et al.* (2006), a paráfrase é caracterizada quando :

> "Há mudanças na gramática, usando palavras de significado similar, reordenando as sentenças do trabalho original ou reafirmando o mesmo conteúdo em diferentes palavras."

h) **Coleções misturadas e coladas**[7] **:** Outra técnica em que copia-se os parágrafos de várias fontes de forma que, para quem lê, aparenta que os parágrafos foram colocados em uma sacola, misturados e em seguida colados de forma aleatórias (WEBER-WULFF, 2010).

i) **Colcha de retalho ou mosaicos de orações**[8] **:** Aparentemente, existe uma crença entre os alunos de que mudando um específico número de palavras da fonte pode-se, de alguma forma, descaracterizar que o artigo apresenta plágio (WEBER-WULFF, 2010).

j) **Plágio estrutural :** O ator parafraseará outro autor sem atribuir ao mesmo o respectivo crédito (WEBER-WULFF, 2010). Esta atividade pode incluir o uso da estrutura argumentativa, as fontes, as configurações experimentais e até mesmo a meta de pesquisa (WEBER-WULFF, 2010). Este tipo de plágio é difícil de descobrir e de provar (WEBER-WULFF, 2010) visto que pode ser visto como a combinação de outra atividades como, por exemplo a paráfrase, o plágio de tradução e o plágio da forma da fonte.

[6] https://translate.google.com
[7] em inglês Shake & Paste Collections
[8] em inglês Clause Quilts or Mosaics

k) **Plágio de conspiração**[9] **:** Este tipo de plágio ocorre tanto no plágio de linguagem natural quanto no de código fonte (WEBER-WULFF, 2010) onde um aluno resolve um problema e os outros realizam sutis mudanças antes de entregar os resultados (WEBER-WULFF, 2010).

2.2 Estratégia de análise do documento

A identificação de indícios de plágio requer que um método, a partir de um documento suspeito d_s, indique se existem indícios de que ocorreu plágio. Com o intuito de compreender e facilitar a comparação dos métodos MAURER *et al.* (2006) os agrupa em três categorias: 1. **Análise de estilo -** Também conhecida como estilometria, é baseada no estilo de escrita único e individual de cada pessoa (MAURER *et al.*, 2006). Isto é, aplica-se a análise estilométrica para determinar se d_s realmente foi escrito pelo autor alegado (BARRÓN-CEDEÑO, 2012). Para EISSEN e STEIN (2006) os atributos estilométricos quantificam o estilo de escrita e podem ser organizados em cinco categorias: *a*) Estatísticas do texto, que operam no nível léxico de caractere; *b*) Atributos sintáticos, que medem o estilo de escrita no nível de sentença; *c*) Atributos *part-of-speech* para quantificar o uso das classes gramaticais; *d*) Conjunto de classes de palavras próximas, para contar palavras especiais; *e*) Atributos estruturais, que refletem a organização do texto; 2. **Comparação de documento fonte -** Comparando d_s com uma coleção de possíveis documentos fonte (BARRÓN-CEDEÑO, 2012). Para MAURER *et al.* (2006) este grupo pode ser dividido em duas categorias: *a*) As que operam localmente e fazem a análise em bases de dados locais ou buscas na internet. *b*) As baseadas em servidores em que o usuário envia d_s e o processo de identificação é executado remotamente; 3. **Busca de fragmentos -** Seleciona de d_s fragmentos de textos que o caracterizam e os submete a um ou vários motores de busca (BARRÓN-CEDEÑO, 2012, MAURER *et al.*, 2006). Normalmente é realizada por um instrutor ou examinador de forma manual (MAURER *et al.*, 2006).

2.2.1 Taxonomias de plágio

Em vista das diferentes formas de se analisar e definir plágio, a figura 2.1 apresenta a taxonomia, proposta por CESKA *et al.* (2008), que define alguns tipos de plágio. O tipo de plágio "Grande parte do documento" envolve os tipos de plágio em que são aplicáveis os modelos de comparação de documentos clássicos como, por exemplo, o Modelo Vetorial (ALZAHRANI *et al.*, 2012, BAEZA-YATES e RIBEIRO-NETO, 1999, EISSEN e STEIN, 2006); O tipo "Tradução" envolve mo-

[9] em inglês Collusion

delos de comparação de documentos em diferentes línguas; Já os tipos "pequena parte do documento" e "Transformação" envolvem modelos que dependem da disponibilidade de um *corpus* de referência.

Os tipos de plágio que apresentam *corpus* de referência podem ser analisados com as estratégias, definidas anteriormente, de "comparação de documento fonte" ou de "busca de fragmentos" onde são extraídos atributos do *corpus* que auxiliarão na identificação dos documentos, ou trechos deles, que foram plagiados. Atributos estes conhecidos como externos que são definidos em BARRÓN-CEDEÑO (2012) como:

> " Os **atributos externos** são aqueles usados para comparar um texto com o documento suspeito d_s, com o intuito de encontrar conteúdo emprestado. Isto é, são usados para comparar d_s com documentos do *corpus*, geralmente escrito por outro autor, buscando por similaridade em fragmentos, em erros ou em estruturas. "

Já os tipos que não apresentam o *corpus* utilizam da estratégia de "análise de estilo" para identificar os trechos que possivelmente foram plagiados. Esta análise está relacionada com a exploração de atributos intrínsecos que BARRÓN-CEDEÑO (2012) define como:

> "Os **atributos intrínsecos** estão relacionados com a evolução do texto contido em um documento suspeito d_s. Diferenças não esperadas entre fragmentos em d_s podem ser causadas pela adição de texto escrito, originalmente, de uma fonte externa. Portanto, modelos que exploram os atributos intrínsecos detectam casos de plágio sem considerar outro documento além de d_s."

Uma taxonomia um pouco mais complexa é apresentada na figura 2.2. Nela, ALZAHRANI *et al.* (2012) relacionam padrões linguísticos, características textuais e técnicas automatizadas para identificar os tipos de plágio. Em seguida, os elementos da taxonomia são divididos, de acordo com o comportamento do plagiador, em plágio literal e plágio inteligente. O plágio literal é o tipo mais comum de plágio, onde pouco esforço é realizado para esconder o delito de plágio enquanto que o plágio inteligente esconde, ofusca e modifica o trabalho original para enganar o leitor assumindo, assim, a autoria do trabalho (ALZAHRANI *et al.*, 2012).

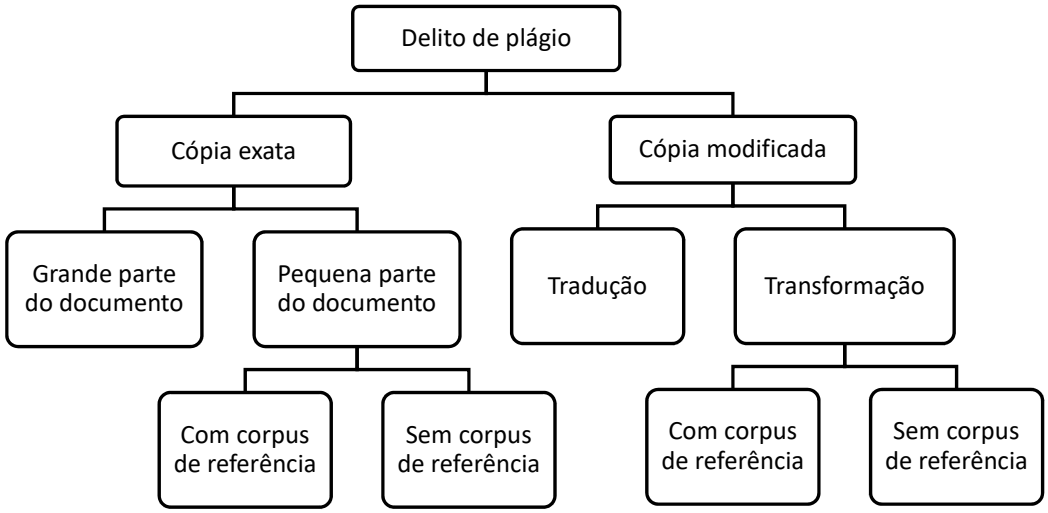


Figura 2.1: Taxonomia de delitos de plágio adapatada de CESKA *et al.* (2008)

2.3 Áreas correlacionadas

2.3.1 Reúso de texto

O reúso de texto é comum em vários cenários onde, ao menos em parte, documentos são baseados em outros documentos que já existem (ADEEL NAWAB *et al.*, 2012). No meio acadêmico, sem o devido reconhecimento, é visto como um pecado básico, enquanto no jornalismo é aceitável e, de fato, uma prática do negócio[10] (CLOUGH *et al.*, 2002). Contudo, apesar de várias iniciativas de pesquisa para lidar com o problema (SEO e CROFT, 2008), detectar o reúso de texto é uma tarefa mais difícil quando o texto original foi alterado (ADEEL NAWAB *et al.*, 2012). Logo, CLOUGH *et al.* (2002) define o desafio de "medir o reúso" da seguinte maneira:

> " Dados dois textos, é possível, com um valor de probabilidade aceitável, determinar quando um texto é derivado de outro?"

Ao tentar responder a pergunta anterior, CLOUGH *et al.* (2002) argumenta sobre a necessidade de especificar níveis de unidades textuais em que o reúso será identificado. O Nível mais alto é o nível do **documento completo** que pode ser

[10]desde que seja paga a taxa para uso da notícia

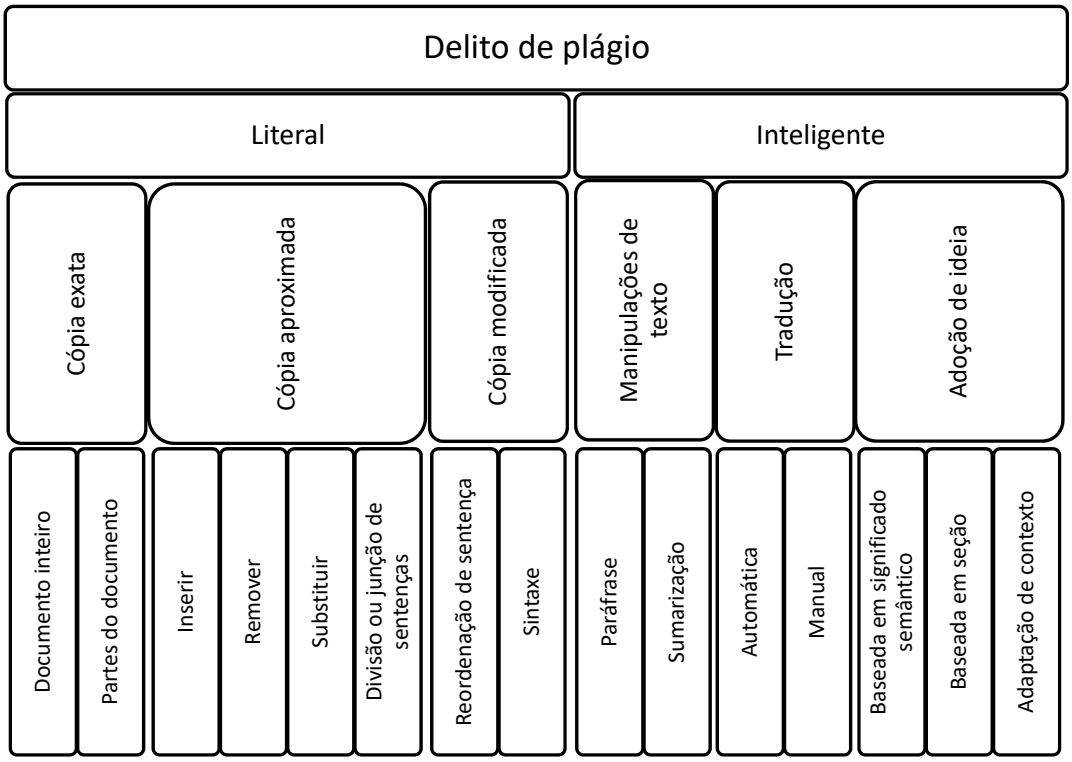


Figura 2.2: Taxonomia de delitos de plágio adapatada de ALZAHRANI *et al.* (2012)

classificado como **Totalmente derivado**, **Parcialmente derivado** e **Não derivado**. O segundo nível **(nível léxico)** identifica reúso na sequência de palavras e é classificado como: **Cópia literal**[11], que expressa a mesma informação copiando palavra por palavra; **Reescrita**, que expressa a mesma informação parafraseando o texto; **Novo** que expressa a informação que não apareceu em outro documento[12].

SEO e CROFT (2008) divide o problema de reúso em: *a*) Reúso em coleções da Web[13] - coleções que contêm várias versões duplicadas ou parcialmente-duplicadas de documentos; *b*) Reúso de texto local - cópias de sentenças, fatos ou passagens, de várias fontes, sem explicitar que houve cópia. O que pode gerar um texto em que uma pequena parte foi realmente criada enquanto o resto foi resultado de modificações sucessivas nos textos extraídos.

2.3.2 Duplicação de documentos

A duplicação de documentos é um tipo especial de Reúso em coleções da Web em que duplicações de páginas da internet, conhecidas como clones (KUMAR e GOVINDARAJULU, 2009), reduzem o número de respostas válidas para uma busca e, consequentemente, também reduzem a acurácia do conjunto de respostas para

[11]em inglês *verbatim*
[12]ao menos não no mesmo contexto
[13]conjunto de páginas extraídas da internet

o usuário (CHOWDHURY *et al.*, 2002). Contudo, a definição do que constitui duplicação não é clara onde a noção geral é a de que se um documento contém aproximadamente o mesmo conteúdo semântico ele é uma duplicata, tendo ou não a mesma sintaxe (CHOWDHURY *et al.*, 2002). Arquivos que ostentam pequenas "não similaridades", e não são idênticos do ponto de vista de "cópias exatas", mas que são idênticos de uma forma notável, são conhecidos como "parcialmente duplicados" (KUMAR e GOVINDARAJULU, 2009) como, arquivos com poucas palavras diferentes; arquivos com mesmo conteúdo, mas diferentes formatos (e.g. mesmo texto e fontes diferentes); e arquivos com o mesmo conteúdo mas diferentes formatos de arquivos (e.g. mesmo conteúdo em txt ou em pdf) (KUMAR e GOVINDARAJULU, 2009).

2.3.3 Documentos Co-derivativos

Um problema peculiar de reúso de texto é o de Documentos co-derivativo onde documentos digitais mudam e evoluem, de forma contínua, gerando assim várias versões de documentos em diferentes estágios do seu desenvolvimento (HOAD e ZOBEL, 2003). De fato, muitas das coleções de documentos mantêm várias versões de documentos, em diferentes estágios do seu desenvolvimento (BERNSTEIN e ZOBEL, 2004, HOAD e ZOBEL, 2003) como, por exemplo revisões de documentos, sumários ou resumos (BERNSTEIN e ZOBEL, 2004). Logo, dois documentos são co-derivativos :

a) "Se ambos têm origem da mesma fonte"(HOAD e ZOBEL, 2003);

b) " Se compartilham conteúdo isto é, para dois documentos serem co-derivativos alguma parte dos dois deve ser derivada de um terceiro documento"(BERNSTEIN e ZOBEL, 2004).

(BERNSTEIN e ZOBEL, 2004) conceitualiza uma coleção de documentos co-derivativos como um grafo onde:

> "O relacionamento co-derivado de uma coleção é um grafo em que cada nó é um único documento e a presença ou ausência de uma aresta representa a presença ou ausência do relacionamento de co-derivação entre aqueles documentos".

2.3.4 Atribuição de autoria

O problema de determinar o autor de um pedaço de texto vem levantando questões metodológicas por séculos (JUOLA, 2006). Apesar de ser de fundamental importância para as ciências humanas, as questões relacionadas ao problema de

atribuição de autoria não podem ser apenas do interesse dos estudiosos das ciências humanas. Do ponto de vista prático, competem também aos políticos, aos jornalistas e advogados (entre outros interessados)(JUOLA, 2006).

As abordagens tradicionais para resolver o problema de atribuição autoral envolvem a análise, de acadêmicos especializados, nas obras ou no estilo de escrita de determinado autor (JUOLA, 2007). Contudo, JUOLA (2007) afirma que existem problemas com esta abordagem que apresentaremos a seguir: *i*) Nem sempre existem tantos estudiosos especialistas em obras de autores desconhecidos, como para Shakespeare e outros escritores famosos; *ii*) Contudo, caso existam, como é possível medir a precisão entre os especialistas diferentes? Logo, métodos estatísticos testáveis, objetivos e "não tradicionais", de atribuição de autoria, emergiram com o surgimento da estatística moderna (JUOLA, 2007, STAMATATOS, 2009b).

Também conhecida como estilometria, a atribuição de autoria "não tradicional" (JUOLA, 2006, 2007, STAMATATOS, 2009b) apresenta uma variedade de questões de pesquisas como, por exemplo: *i*) Esta pessoa realmente escreveu determinado documento? *ii*) Qual destas pessoas escreveu aquele documento? *iii*) Todos estes documentos foram escritos pela mesma pessoa? *iv*) Quando este documento foi escrito? *v*) Qual o sexo do autor? Portanto, podemos definir a atribuição de autoria, de suporte estatístico ou computacional, como :

i) "métodos que, a partir de alguns atributos textuais, conseguem distinguir textos escritos por diferentes autores"(STAMATATOS, 2009b)

ii) "qualquer tentativa de inferir as características do criador de determinado pedaço de texto" (JUOLA, 2006)

iii) "a tarefa de determinar ou verificar a autoria de um texto baseada somente na evidência interna ao texto"(KOPPEL *et al.*, 2009)

Na forma mais simplista do problema, a partir de exemplos de escrita dos autores candidatos, pede-se para determinar qual deles é o autor de um texto anônimo (KOPPEL *et al.*, 2009). Todavia, existem outras análises que podem ser definidas para o problema, onde STAMATATOS (2009b) destaca a verificação de autor, a detecção de plágio, caracterização ou identificação de perfil de autores e a detecção de inconsistências de estilo. Portanto, os cenários possíveis do problema de atribuição de autoria são **Identificação de perfil**, **Verificação de perfil** e **"Agulha no palheiro"** (JUOLA, 2006, KOPPEL *et al.*, 2009)

Na **Identificação de perfil** não existe um conjunto de autores candidatos e, portanto o desafio é prover a maior quantidade de informação demográfica e psicológica possível do autor (KOPPEL *et al.*, 2009). Isto é, identificar qualquer propriedade

do(s) autor(es) de uma amostra do texto (JUOLA, 2006). Na **Verificação de perfil**, existe apenas um suspeito e o desafio é determinar se o suspeito é ou não o autor do texto (KOPPEL *et al.*, 2009). Por fim, na **"Agulha no palheiro"** existem milhares de candidatos, contudo todo candidato apresenta um conjunto bem limitado de exemplos de escrita (KOPPEL *et al.*, 2009).

2.4 Cenários de identificação de plágio

Na subseção 2.2.1, discutiu-se que os atributos para identificar plágio são divididos em dois grupos: Os atributos relacionados a um *corpus* de referência, conhecido como externo, e os atributos intrínsecos. Logo, a figura 2.3 apresenta os cenários de identificação de plágio intrínseco e externo que estão diretamente relacionados com os grupos de atributos em questão. O cenário de plágio externo ainda se divide em dois problemas: o cenário de plágio externo no nível de documento, em que se tenta identificar quais documentos foram plagiados e o plágio externo no nível de fragmento, que identifica pares de fragmentos de texto que representam o plágio.

$$\text{Plágio} = \begin{cases} \text{Intrínseco} \\ \text{Externo} = \begin{cases} \text{Nível de documento} \\ \text{Nível de fragmento} \end{cases} \end{cases}$$

Figura 2.3: Hierarquia dos cenários de identificação de plágio

2.4.1 Identificando plágio intrínseco

A tarefa de identificação de plágio intrínseco lida com os cenários em que não existe *corpus* de referências disponível e baseia-se, exclusivamente, em mudanças ou inconsistências em dado documento (STAMATATOS, 2009a). Isto é, para lidar com a identificação de plágio intrínseco deve-se detectar passagens plagiadas, de um documento suspeito, baseando-se exclusivamente nas irregularidades ou inconsistências do documento (STAMATATOS, 2009a).

STEIN e EISSEN (2007) caracterizam a identificação de plágio intrínseco como:

> "Dado um documento d, alegadamente escrito por um autor, queremos identificar as seções de d que resultam de outro autor e não estão legendadas pela citação apropriada."

A tarefa da identificação é encontrar evidência suficiente para aceitar, ou refutar, a hipótese de que um determinado documento foi feito por um autor (STEIN e EISSEN, 2007). Para tanto, o conteúdo de d é analisado, buscando inconsistências, em d como, por exemplo, mudanças no vocabulário, complexidade e mal fluxo

(BARRÓN-CEDEÑO, 2012). Segundo POTTHAST *et al.* (2011), um conjunto de passos define a tarefa de identificação de plágio intrínseco, eles são: 1. **estratégia de segmentação** - Onde o documento é segmentado; 2. **modelo de recuperação de estilo de escrita** - Os segmentos são representados no modelo de recuperação de estilo de escrita; 3. **algoritmo de detecção de *outlier*** - As diferenças de estilo são identificadas por detecção de *outliers* entre as representações dos segmentos; 4. **pós-processamento** - Os segmentos suspeitos de plágio, que estão sobrepostos, são identificados e consolidados em apenas um. Isto é, ao final do pós-processamento, os segmentos identificados como potenciais passagens plagiadas são retornados.

2.4.2 Identificando plágio externo

POTTHAST *et al.* (2011) explica o processo de identificação de plágio externo, de um documento suspeito d_q, em uma coleção de potenciais documentos fontes D da seguinte forma: (1) Todos os documentos são pré-processados, usando um *pipeline* de indexação, que normaliza os *tokens* removendo *stop words*, radicaliza o resto e substitui palavras por sinônimos. Caso necessário, documentos são traduzidos para apenas uma linguagem (normalmente inglês) por algum serviço de tradução; (2) Um conjunto de "documentos fonte" candidatos é resgatado da coleção D; (3) Cada documento candidato é comparado com o documento suspeito para extrair as passagens similares; (4) As passagens extraídas são pós-processadas para remoção de falsos positivos, e o resto é apresentado como detecções de plágio em potencial.

O problema de identificação de plágio externo ainda pode ser dividido em dois níveis, o do documento e o do fragmento (BARRÓN-CEDEÑO, 2012), onde:

i) "**Identificação de plágio externo no nível de documento**. Determina quando d_q contém texto emprestado de um documento específico $d \in D$";

ii) "**Identificação de plágio externo no nível de fragmento**. Determina quando o fragmento $f_q \in d_q$ foi emprestado do fragmento $f \in d(d \in D)$".

O capítulo 3 apresenta, de forma mais detalhada, o processo de identificação de plágio externo com o intuito de analisar de forma mais objetiva apenas o problema do plágio externo.

Capítulo 3

Reduzindo o espaço de comparação no plágio externo

> *"Did the author of a document d_q commit a plagiarism offense?"*
>
> — Benno Stein *et al*, (STEIN *et al.*, 2007)

A tarefa de identificação de plágio externo assume que os documentos fonte estão disponíveis e são alcançáveis. Isto é, se d_q tem texto plagiado então todos os seus documentos fontes são acessíveis, ao SDP, por meio de uma coleção de documentos ou recurso online [1] (VANI e GUPTA, 2014). Portanto, o processo de identificação de plágio externo é organiza em três estágios: (i) amostras de d_q são extraídas e um conjunto com passagens possivelmente plagiadas é retornado; (ii) as passagens são comparadas com as amostras de d_q para identificar um conjunto de pares de passagens plagiadas; (iii) todos os pares de passagens selecionados são analisados para se remover as citações corretas e agrupar passagens contíguas do texto (STEIN *et al.*, 2007). Além disso, formalmente falando, (i) busca um conjunto de documentos $D_{src}^{d_q}$, composto de possíveis fontes, para um caso suspeito de plágio representado pelo documento d_q; e (ii) efetua uma comparação par-a-par entre d_q e cada documento fonte $d_{src} \in D_{src}^{d_q}$, e, em seguida, seleciona de $D_{src}^{d_q}$ as fontes mais prováveis de terem sido plagiadas (ALZAHRANI *et al.*, 2012, POTTHAST *et al.*, 2010a). O item (i) é conhecido como Recuperação Heurística (ALZAHRANI *et al.*, 2012), Seleção de candidato (THOMPSON *et al.*, 2015), Recuperação de fonte (POTTHAST *et al.*, 2010a) ou Identificação no nível de documento (K e GUPTA, 2017b) enquanto que itens (ii) e (iii) são conhecidos, respectivamente, como análise detalhada e pós-processamento baseado em conhecimento (ALZAHRANI *et al.*, 2012, POTTHAST *et al.*, 2010a). Além disso, as abordagens de (ii) e (iii) foram avaliadas e classificadas

[1] por exemplo, servidor web ou a própria Internet

por ALZAHRANI *et al.* (2012) como "abordagens de alinhamento de texto" e por POTTHAST *et al.* (2013) como "análise exaustiva de documentos fonte-suspeito".

Muitos métodos de detecção de plágio são eficientes quando lidam com plágio em textos com poucas modificações, mas falham em identificar plágio de idéia feito por paráfrase, sumarização ou tradução (ALZAHRANI *et al.*, 2012). De fato, métodos recentes lidam com plágio inteligente baseando-se em sintaxe (ABDI *et al.*, 2015a, EKBAL *et al.*, 2012, FRANCO-SALVADOR *et al.*, 2016, K e GUPTA, 2017a, NAWAB *et al.*, 2016, OSMAN *et al.*, 2012, PEREIRA e ZIVIANI, 2003, ZHANG e CHOW, 2011), semântica (ABDI *et al.*, 2015a, ALZAHRANI *et al.*, 2015, FRANCO-SALVADOR *et al.*, 2016, K e GUPTA, 2017a, OSMAN *et al.*, 2012, PAUL e JAMAL, 2015), estrutura (ZHANG e CHOW, 2011) e entre várias línguas (BARRÓN-CEDEÑO *et al.*, 2013, EHSAN e SHAKERY, 2016, FRANCO-SALVADOR *et al.*, 2016, K e GUPTA, 2017a). Por exemplo, ABDI *et al.* (2015b) lida com plágio inteligente, em inglês, usando o *thresaurus* Wordnet para comparar pares de sentenças a partir de termos semanticamente relacionados. Contudo, é impraticável aplicar métodos de identificação de plágio inteligente, em grandes coleções, sem uma considerável redução no espaço de comparação (MEUSCHKE e GIPP, 2014, ZHANG e CHOW, 2011). De fato uma das dificuldades de se identificar plágio, de forma eficiente, é responder rapidamente a uma consulta onde a cópia se encontra entre milhões de documentos. Além disso, cada documento apresenta centenas de palavras (ZHANG e CHOW, 2011) e, portanto, o tamanho dos dados faz com que o custo de armazenamento e busca, em grandes coleções de documentos, seja um desafio para a identificação de plágio externo (WANG *et al.*, 2015).

Este capítulo discute como a Recuperação Heurística (RH) reduz o espaço de comparação, possibilitando assim que técnicas de identificação de plágio inteligente possam ser aplicadas em (*ii*) e (*iii*). Logo a RH é essencial para que um SDP possa atender aos problemas reais de Plágio Externo onde a seção 3.1 discute as etapas que compõe o passo de Recuperação Heurística enquanto que os modelos de RI para o estágio de RH são apresentados na seção 3.2.

3.1 Recuperação Heurística

O estágio de Recuperação Heurística é uma tarefa de recuperação de informação representada como uma tupla $[D, D_{susp}, F_r, R(d_q, d_{src})]$ em que D é um conjunto composto de representações dos documentos da coleção; D_{susp} é um conjunto composto de representações de documentos para as consultas, isto é o conjunto dos documentos suspeitos; F_r é um *framework* utilizado para representar os documentos de D ou D_{susp} (e.g., o modelo vetorial (SALTON *et al.*, 1975)); $R(d_q, d_{src})$ é uma função de rankeamento que avalia a probabilidade de $d_{src} \in D^{d_q}_{src} \mid D^{d_q}_{src} \subset D$ ser um

documento fonte para $d_q \in D_{susp}$ (BAEZA-YATES e RIBEIRO-NETO, 1999).

O objetivo do estágio de RH é remover falsos positivos, construindo uma coleção menor, para reduzir a carga de trabalho nos próximos estágios do processo, em que buscas exaustivas, computacionalmente custosas e demoradas, por regiões sobrepostas, são executadas (MEUSCHKE e GIPP, 2014, THOMPSON *et al.*, 2015). A meta é formalmente definida como: Um conjunto de documentos retornados $D_{src}^{d_q}$ deve maximizar a pontuação da função de rankeamento $R(d_q, d_{src})$ para cada documento $d_{src} \in D_{src}^{d_q}$ se d_{src} realmente foi copiado por um documento suspeito d_q, em que $d_q \in D_{susp}$. Por exemplo, assuma que d_5 e d_{10} são os documentos fonte de $d_q \in D_{susp}$; para uma coleção de documentos $D = \{d_1, d_2, d_3, d_4, \boldsymbol{d_5}, d_6, d_7, d_8, d_9, \boldsymbol{d_{10}}\}$, um *framework* de RH deve recuperar um conjunto de documentos $D_{src}^{d_q}$ contendo ambos d_5 e d_{10}. Ademais, a ordenação gerada por $R(d_q, d_5)$ e $R(d_q, d_{10})$ deve calcular os dois maiores valores entre todos os documentos $d_{src} \in D_{src}^{d_q}$ para d_5 e d_{10}.

As próximas subseções apresentam as peculiaridades de cada passo do estágio de RH conforme a Figura 3.1 ilustra. Nela o estágio de RH apresenta um motor de busca, com todos os documentos fonte representados através de um modelo de Recuperação de Informação (RI). O modelo de RI também é aplicado em cada documento suspeito gerando consultas que devem resgatar o maior número possível de documentos fontes enquanto reduz os custos de buscar e recuperá-los (POTTHAST *et al.*, 2013).

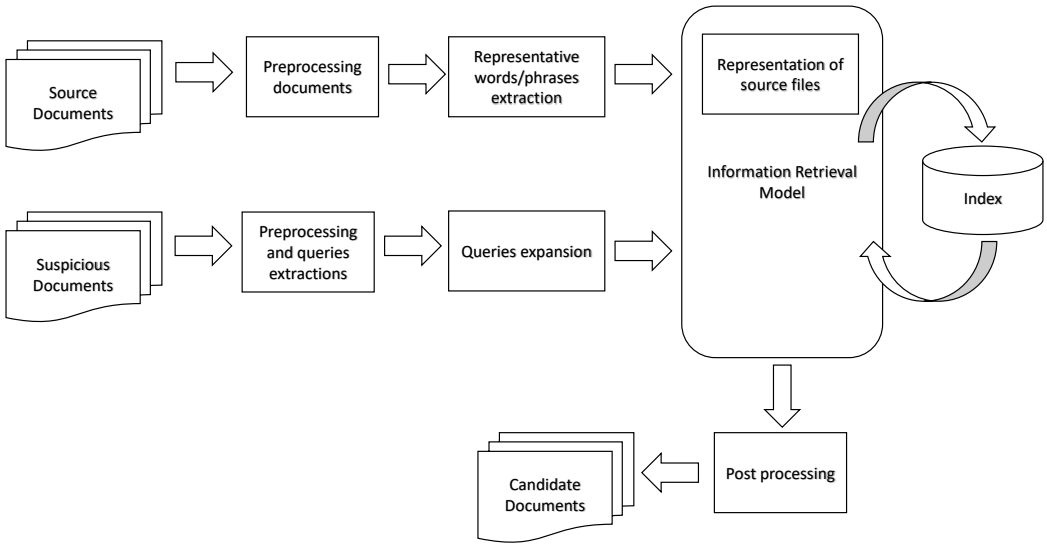


Figura 3.1: Sequências de passos de um método de Recuperação Heurística. Adaptado de (EHSAN e SHAKERY, 2016)

3.1.1 Pré-processamento de documentos e extração de consultas

O pré-processamento converte o texto dos documentos fonte e dos documentos suspeitos para uma representação textual que facilita a identificação de padrões. Nele, o texto pode ser convertido para letras minúsculas e *stopwords*, pontuação, espaços em branco extras e outros caracteres podem ser removidos. Além disso, outras abordagens de pré-processamento podem ser aplicadas como, por exemplo, identificação de classes gramaticais, em inglês *parts-of-speech*, identificação de entidades nomeadas, identificação de radicais de palavras e remoção de afixos (EHSAN e SHAKERY, 2016).

A segmentação dos documentos fontes e a extração de consultas, a partir dos documentos suspeitos, são realizadas em seguida. Em ambos os passos, os documentos são segmentados em pedaços, como n-gramas, sentenças ou parágrafos, que podem ser sobrepostos e também são conhecidos como passagens de textos (EHSAN e SHAKERY, 2016, POTTHAST *et al.*, 2011). POTTHAST *et al.* (2011) enumeram algumas abordagens de segmentação como, por exemplo, sem segmentação, segmentos de 50-linhas, segmentos de 100 palavras, segmentos de 200 palavras e segmentos de 5 sentenças.

Por fim, vale a pena destacar que as consultas são selecionadas a partir das passagens de um documento suspeito onde, a razão para se segmentar o documento suspeito é distribuir a responsabilidade de identificar o plágio em todo o conteúdo do documento (POTTHAST *et al.*, 2011).

3.1.2 Extração de elementos representativos e expansão de consultas

A extração de elementos representativos é realizada, a partir dos segmentos de um texto, com o objetivo de selecionar os elementos que maximizam a chance de recuperar os documentos fontes correspondentes a um documento suspeito (POTTHAST *et al.*, 2011). Logo, após a segmentação de um documento fonte, os segmentos representativos são selecionados, quando necessário traduzidos, e encaminhados ao modelo de recuperação de informação para armazená-los. Já para as consultas existem duas formas de se realizar a extração de elementos representativos: (*i*) selecionar palavras ou frases chave como consultas ou (*ii*) utilizar o documento inteiro como consulta (EHSAN e SHAKERY, 2016, POTTHAST *et al.*, 2011).

Utilizar o documento inteiro melhora a eficácia da busca, mas demanda mais computação em comparação aos métodos baseados em palavras-chave. Contudo, em coleções extensas, o número de palavras que normalmente representam as con-

sultas do tipo (*ii*) é grande, fazendo com que métodos baseados em (*ii*) sejam impraticáveis. Logo, para subterfugiar o problema de (*ii*) em grandes coleções, métodos de redução de dimensionalidade baseados em representações latentes, de baixa dimensionalidade, foram aplicados para capturar a semântica do documento inteiro em uma consulta (ZHANG e CHOW, 2011).

No passo de expansão de consultas, cada consulta é estendida ou melhorada a partir de características textuais. ALZAHRANI *et al.* (2012) categoriza as características textuais para a identificação de plágio externo em léxicas, sintáticas, semânticas e estruturais, conforme são apresentadas na figura 3.2. Vale ressaltar que as mesmas características também podem ser aplicadas nos documentos fonte para melhorar assim as suas representações do ponto de vista sintático, semântico, léxico e estrutural. Por exemplo, a partir de substituições por sinônimos, hiperônimos e hipônimos ou por métodos baseados na semântica do texto como a rotulagem do papel semântico (em inglês *semantic role labelling (SRL)*) é possível identificar que um pedaço de texto foi parafraseado por outro (OSMAN *et al.*, 2012, PAUL e JAMAL, 2015) .

$$
\text{Características} = \begin{cases}
\text{Léxicas} = \begin{cases} \text{n-gramas de caracteres de tamanho fixo} \\ \text{n-gramas de caracteres de tamanho variado} \\ \text{n-gramas de palavras} \end{cases} \\
\text{Sintáticas} = \begin{cases} \text{segmentação usando janelamento} \\ \textit{part-of-speech} \text{ e estrutura de frase} \\ \text{posição/ordem da palavra} \\ \text{sentença} \end{cases} \\
\text{Semânticas} = \begin{cases} \text{sinônimos, hiperônimos e hipônimos} \\ \text{dependências semânticas} \end{cases} \\
\text{Estruturais} = \begin{cases} \text{específicas de bloco} \\ \text{específicas de conteúdo} \end{cases}
\end{cases}
$$

Figura 3.2: grupos de características textuais para plágio externo, adaptado de ALZAHRANI *et al.* (2012)

As características léxicas representam um texto como uma sequência de n-gramas, de caracteres (CNG) ou de palavras (PNG), onde o processo de criação é conhecido como *fingerprinting* ou *shingling*. Portanto, uma *fingerprint* de um documento deve identificar unicamente um documento, assim como a *fingerprint* humana faz. As características sintáticas identificam elementos que possam ser utilizados para medir a similaridade sintática entre dois textos como, por exemplo, as classes gramaticais (substantivos, adjetivos, advérbios, etc), sentenças, segmentação usando janelamento (para extrair passagens maiores que sentenças) e a ordem em que as palavras aparecem. As características semânticas extraem conjuntos de pa-

lavras, de sinônimos, de hiperônimos e de hipônimos de tal forma que seja possível quantificar em que contexto semântico as palavras de um texto foram utilizadas para, em seguida, medir se dois textos apresentam alguma similaridade semântica. As características estruturais capturam a semântica inerente à estrutura para a organização do texto. Por exemplo, documentos podem ser representados como uma coleção de parágrafos ou passagens que, por sua vez, são compostos de elementos de forma hierárquica e, portanto, representam a semântica da estrutura do texto como uma árvore de características (ALZAHRANI *et al.*, 2012).

3.1.3 Representação, busca e recuperação de documentos fonte

A representação, busca e recuperação dos documentos fonte é realizada por modelos de Recuperação de Informação que são divididos por ALZAHRANI *et al.* (2012) em três grupos: (*i*) modelos de recuperação de informação Monolíngue, (*ii*) técnicas de agrupamento e (*iii*) modelos de recuperação multilíngue. De forma geral, os modelos de (*iii*) dependem de similaridades léxicas entre as línguas, *thesaurus* multilíngues, modelos que podem ser treinados simultaneamente em todas as línguas, usando *parallel corpora*, ou de métodos que transformem a tarefa para uma tarefa de recuperação monolíngue (BARRÓN-CEDEÑO *et al.*, 2013). Além disso, a diferença entre (*i*), (*ii*) e (*iii*) está na forma de selecionar o conjunto de documentos identificados. Visto que (*ii*) gera agrupamentos de documentos similares na esperança de que um documento plagiado esteja no agrupamento com a maior quantidade de documentos que ele plagiou enquanto que (*i*) e (*iii*) selecionam a coleção de documentos mais similares à consulta gerada pelo documento suspeito.

Os modelos de Recuperação de Informação para identificação de plágio são compostos de um *framework* F_r e uma função de ranking $R(d_q, d_{src})$ onde, por exemplo, técnicas de agrupamento podem ser utilizadas como F_r e em seguida a similaridade do cosseno ou de Jaccard podem ser avaliadas, como função de ordenação $R(d_q, d_{src})$, entre o documento suspeito e os documentos do grupo identificado por F_r (ALZAHRANI *et al.*, 2012).

As funções de ordenação são divididas em medidas de similaridade baseadas em *strings*, baseadas no *corpus* e baseadas em conhecimento (GOMAA e FAHMY, 2013, MIHALCEA *et al.*, 2006). As medidas baseadas em *strings* são as que calculam as similaridades baseadas nos caracteres ou nos termos dos textos enquanto que as medidas baseadas em *corpus* e conhecimento treinam diferentes algoritmos que aprendem, a partir do texto de uma coleção de documentos, padrões estatísticos que indicam a correlação entre os documentos da coleção. Esses padrões estatísticos podem ser utilizados para identificar que ocorreu plágio entre documentos suspeitos

e algum documento da coleção. Alguns exemplos de métodos baseados em *corpus* e em conhecimento são os que usam técnicas como *Principal Components Analysis (PCA)* (ZHANG e CHOW, 2011), *Latent Semantic Analysis (LSA)* (SOLEMAN e PURWARIANTI, 2014) ou bases de dados léxicas, como a Wordnet (ABDI *et al.*, 2015b, NAWAB *et al.*, 2016, OSMAN *et al.*, 2012), para avaliar se ocorreu plágio (THOMPSON *et al.*, 2015).

Contudo, a abordagem baseada em *strings* é a única que não é influenciada pela maldição da alta dimensionalidade (BENGIO *et al.*, 2003) visto que, em alta dimensão, é crucial distribuir a função de probabilidade de massa onde realmente importa, ao invés de uniformemente em todas as direções dos elementos de treinamento, o que é difícil em problemas com grandes volumes de dados como o plágio. Isto é, o algoritmo precisa generalizar e encontrar elementos que não se encontram na coleção de treinamento (BENGIO *et al.*, 2003). Além disso, as medidas baseadas em termos são as medidas mais eficientes em dados de alta dimensão, como documentos de texto, e são amplamente utilizadas em problemas para medir similaridade entre textos na área de Recuperação de Informação (THOMPSON *et al.*, 2015).

3.1.4 Pós-processamento

Segundo, EHSAN e SHAKERY (2016) o objetivo do pós-processamento é evitar que documentos não plagiados, com similaridade coincidente, sejam retornados ao se buscar a fonte de um documento suspeito de plágio. A similaridade coincidente pode ocorrer, por exemplo, em documentos escritos sobre o mesmo assunto. Outro exemplo é entre documentos que apresentam uma linha de argumentação parecida mas que foram escritos sem o conhecimento da existência do outro documento.

Do ponto de vista prático, se a similaridade entre um segmento suspeito de plágio e algum documento fonte se diferenciar pouco das similaridades do segmento suspeito e os outros documentos a similaridade pode ser coincidente. Em contrapartida, identificar pares de segmentos com similaridades que se destacam é um indício de plágio onde EHSAN e SHAKERY (2016) determina que um limiar de escolha pode ser calculado pela média da diferença das similaridades entre outros documentos.

3.2 Modelos de Recuperação de Informação em plágio

Segundo MOHAMED e MARCHAND-MAILLET (2015) os cenários de busca para os modelos de Recuperação de Informação são divididos em: Busca Exaustiva (PEREIRA e ZIVIANI, 2003, VANI e GUPTA, 2014); Busca Exata (SOLEMAN e PURWARIANTI, 2014, ZHANG e CHOW, 2011); e Busca Aproximada (BARRÓN-

CEDEÑO *et al.*, 2013, EHSAN e SHAKERY, 2016, EKBAL *et al.*, 2012, MEUSCHKE e GIPP, 2014, NAWAB *et al.*, 2016, OSMAN *et al.*, 2012).

A Busca Exaustiva compara um documento suspeito $d_q \in D_{susp}$ com todos os documentos de uma coleção D mantendo o rastro de quais documentos são similares a d_q. A Busca Exata reduz o número de comparações de d_q, feito pela Busca Exaustiva, a partir de técnicas de particionamento ou decomposição como por exemplo, a kd-tree (MOHAMED e MARCHAND-MAILLET, 2015). A kd-tree particiona os espaço de dimensões, em partições binárias, isolando os documentos que não apresentam dimensões iguais em quadrantes, isto é partições, diferentes (BENTLEY, 1975). Contudo, ambas as abordagens têm problemas com coleções de texto extensas: Técnicas de Busca Exaustiva não escalam quando D é grande e as técnicas de Busca Exata não são eficientes para dados com alta dimensionalidade (MOHAMED e MARCHAND-MAILLET, 2015). Além disso, outro "gargalo", para as técnicas discutidas, são as limitações de armazenamento e custo computacional para carregar os dados originais em memória (WANG *et al.*, 2015).

A Busca Aproximada lida com os problemas da Busca Exaustiva e da Busca Exata aceitando uma pequena imprecisão nos resultados para melhorar o tempo de busca, assim como, reduzir o espaço necessário, em memória, para se realizar a busca (MOHAMED e MARCHAND-MAILLET, 2015, WANG *et al.*, 2015). Os métodos baseados em Busca Aproximada representam os documentos fonte em um mecanismo orientado a termos, por exemplo orientado a palavras, em que cada termo identifica os documentos que o contêm (BAEZA-YATES e RIBEIRO-NETO, 1999). Por exemplo, OSMAN *et al.* (2012) propõe um modelo orientado a tópico onde os tópicos são extraídos combinando *Semantic Role Labeling (SRL)* com a Wordnet[2]; cada tópico é representado como uma lista de documentos que o contém; e a função de rankeamento é calculada pela similaridade de Jaccard entre o documento suspeito e os documentos que têm os mesmos termos do documento suspeito. O modelo de MEUSCHKE e GIPP (2014) apresenta um método baseado em co-citações [3] combinadas com o índice de palavras em comum para busca de plágio em apenas uma língua. Já o modelo de BARRÓN-CEDEÑO *et al.* (2013) lida com a tarefa de plágio multilíngue traduzindo todos os textos e então aplicando um método monolíngue, orientado a palavras, com a similaridade do cosseno sendo usada como função de rankeamento.

De acordo com BAEZA-YATES e RIBEIRO-NETO (1999) um índice invertido, ou arquivo invertido, é um mecanismo orientado a termos, para indexar uma coleção de textos, que pode ser utilizado nos métodos de Busca Aproximada. Um índice invertido é composto de um vocabulário onde cada termo do vocabulário identifica os

[2]https://wordnet.princeton.edu/

[3]i.e. se os dois citam o mesmo documento

documentos que o contêm. Os índices invertidos são divididos em Índices Invertidos Básicos (IIB) e Índices Invertidos Completos (IIC). Um IIB apresenta, para cada termo, uma lista de tuplas (documento, frequência) onde cada tupla representa o documento que contém o termo e quantas vezes o termo aparece no mesmo. Já o IIC apresenta, para cada termo, uma tupla (documento, frequência, lista de posições) onde cada tupla de um documento também indexa em quais posições do documento o termo apareceu permitindo, assim, a reconstrução do texto original. As Tabelas 3.1 e 3.2 apresentam, respectivamente, o IIB e o IIC gerado para 10 palavras dos exemplos, em anexo, A.1, A.2 e A.3.

Tabela 3.1: 10 palavras do Índice Invertido Básico dos exemplos A.1, A.2 e A.3 em anexo

abstraction	[(A.1, 1), (A.2, 1)]
accomplished	[(A.1, 1), (A.2, 1), (A.3, 1)]
allows	[(A.3, 1)]
ancestor	[(A.1, 3), (A.2, 3), (A.3, 1)]
animals	[(A.3, 2)]
attributes	[(A.1, 1), (A.3, 2)]
belongs	[(A.1, 1), (A.2, 1)]
called	[(A.1, 3), (A.2, 3)]
categorization	[(A.1, 2), (A.2, 2)]
object	[(A.1, 1), (A.2, 1), (A.3, 2)]

Tabela 3.2: 10 palavras do Índice Invertido Completo dos exemplos A.1, A.2 e A.3 em anexo

abstraction	[(A.1, 1, [94]), (A.2, 1, [94])]
accomplished	[(A.1, 1, [138]), (A.2, 1, [138]), (A.3, 1, [86])]
allows	[(A.3, 1, [51])]
ancestor	[(A.1, 3, [33, 148, 143]), (A.2, 3, [33, 148, 143]), (A.3, 1, [89])]
animals	[(A.3, 2, [56, 63])]
attributes	[(A.1, 1, [25]), (A.3, 2, [13, 53])]
belongs	[(A.1, 1, [67]), (A.2, 1, [67])]
called	[(A.1, 3, [127, 9, 79]), (A.2, 3, [127, 9, 79])]
categorization	[(A.1, 2, [49, 46]), (A.2, 2, [49, 46])]
object	[(A.1, 1, [0]), (A.2, 1, [0]), (A.3, 2, [1, 19])]

As técnicas mais comuns de Busca Aproximada, com índice invertido, são as baseadas em *Locality-sensitive hashing (LSH)*. De fato, as técnicas baseadas em LSH são escaláveis e garantem tempo de resposta pequeno para os problemas de similaridade entre textos (JI *et al.*, 2013), busca de imagens (DONG *et al.*, 2008, MOHAMED e MARCHAND-MAILLET, 2015) e busca de áudio e formas (DONG *et al.*, 2008). Outra técnica que emprega índice invertido no problema de Busca Aproximada é a *Permutation-based index(PBI)* onde PBI tem como fundamento a distribuição

geométrica do elementos de D enquanto que LSH se baseia na distribuição de probabilidade da distância entre d_q e $d_i \in D$ (MOHAMED e MARCHAND-MAILLET, 2015).

As técnicas baseadas em LSH são as técnicas mais comuns na busca heurística do Plágio Externo monolíngue e multilíngue (ALZAHRANI *et al.*, 2012). Essas abordagens combinam uma lista de valores únicos de *fingerprints*, representadas como valores numéricos gerados por funções de hash, com métricas de similaridade para recuperar os documentos que compartilham um número considerável de valores de hash com o documento suspeito (ALZAHRANI *et al.*, 2012). As abordagens LSH representam os dados, de cada documento, como conjuntos finitos, e.g. conjuntos das palavras ou dos termos, enquanto preservam a similaridade par-a-par o que é uma solução viável em cenários de larga escala como avaliado em (JI *et al.*, 2013, WANG *et al.*, 2015) e em (ALZAHRANI *et al.*, 2012) que é um estudo mais aprofundado em métodos de identificação de plágio.

O capítulo 4 descreve, de forma mais detalhada, a família de métodos LSH assim como apresenta os trabalhos que relacionam LSH na Busca Heurística do Plágio Externo.

Capítulo 4

Indexação e Busca Usando *Locality-Sensitive Hashing*

> *"Most current approaches for KNN computation are unable to satisfy the runtime requirements for high dimensional datasets"*
> — Jia Pan and Dinesh Manocha, (PAN e MANOCHA, 2012)

Na busca dos vizinhos mais próximos em coleções grandes (e.g. coleções de tamanho $n > 1$ milhão), ou com dimensionalidade alta, os resultados para técnicas de Busca Aproximada devem ser próximos aos das técnicas de Busca Exata (PAN e MANOCHA, 2012). Logo, a Busca Aproximada deve justificar que a perda em precisão, em relação a Busca Exata, é compensada, de alguma forma, por uma melhoria nos custos de execução por consulta. De fato, o custo de execução ideal, para a Busca Aproximada, deve se manter entre o tempo constante $\mathcal{O}(1)$ e o tempo sublinear $\mathcal{O}(\log n)$ enquanto que $\mathcal{O}(n)$ deve ser o requisito de armazenamento para viabilizar a manipulação de grande coleções de dados (PAN e MANOCHA, 2012, WANG *et al.*, 2015).

Na identificação de plágio, assim como no problema de busca de vizinhos mais próximos, uma família de métodos amplamente utilizados, para lidar com dados de alta dimensionalidade, é o *locality-sensitive hashing (LSH)*. Técnicas baseadas em LSH agrupam documentos similares a partir de famílias de funções de *hash* (BUHLER, 2001, PAN e MANOCHA, 2012) onde, uma função de *hash* é uma função $h : K \mapsto \mathbb{Z}$ que mapeia todos os elementos e_i, de um conjunto K, em valores numéricos $h(e_i)$ de maneira que a probabilidade de dois elementos diferentes serem mapeados para o mesmo valor deve ser baixa (KONHEIM, 2010).

A meta das técnicas LSH (GIONIS *et al.*, 1999, INDYK e MOTWANI, 1998) é aplicar as funções de *hash* em todos os documentos até que documentos similares

estejam mais susceptíveis a estar no mesmo conjunto de valores de *hash* do que os não similares (LESKOVEC *et al.*, 2014). Assim como ocorre no exemplo da Figura 4.1 onde, dois documentos, nas extremidades de um segmento de reta pontilhado, têm pelo menos um valor de *hash* em comum.

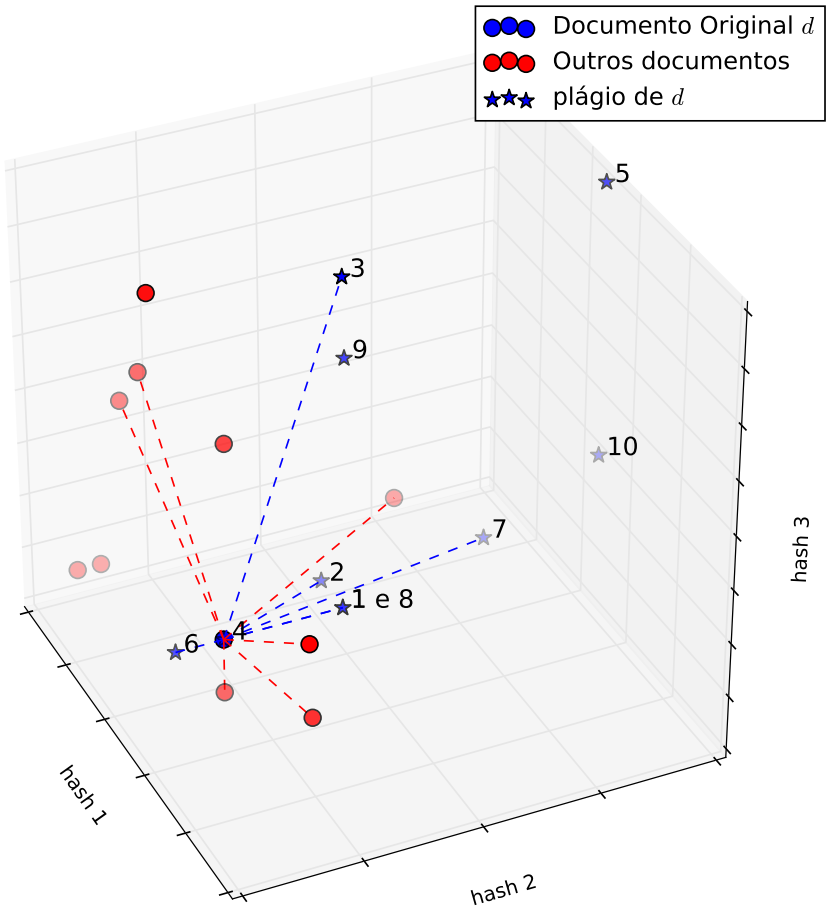


Figura 4.1: Método lsh gerando três valores de *hash* para casos de plágio (estrelas), de um documento (círculo azul), e outros documentos (círculos) selecionados da coleção *Plagiarised Short Answers* (CLOUGH e STEVENSON, 2011)

Em técnicas LSH, o tempo computacional para a busca de vizinhos mais próximos é drasticamente reduzido em troca de uma pequena probabilidade de falhar ao encontrar todos os vizinhos mais próximos (SLANEY e CASEY, 2008). Por exemplo, na Figura 4.1 todos os documentos foram representados a partir de três valores de *hash* gerados por um método LSH. O círculo azul representa um documento que foi copiado por dez documentos que foram representados como estrelas azuis. Além disso, as linhas tracejadas apresentam em suas extremidades pares de documentos com pelo menos um valor de *hash* em comum. Logo, é possível observar que apenas três casos de plágio (5, 9 e 10) não apresentaram *hash* em comum com d. Ademais, o método LSH garante que a probabilidade de colisões é muito maior entre documentos próximos do que para os que estão distantes (GIONIS *et al.*, 1999). Por

conseguinte, para identificar os vizinhos mais próximos de um documento d_j uma abordagem LSH gera valores de hash para d_j que serão comparados com os valores de hash de todos os itens de uma coleção de documentos (GIONIS *et al.*, 1999).

De forma geral, os métodos LSH dividem o texto de cada documento em palavras ou n-gramas e, a partir do conjunto gerado por essa divisão, funções de hash geram valores para representar cada documento. Na Figura 4.1 21 documentos foram divididos em palavras o que gerou um vocabulário (BAEZA-YATES e RIBEIRO-NETO, 1999) de 248 palavras. Logo, é evidente que procurar documentos com pelo menos um valor em comum, entre três, é mais rápido que a mesma busca em 248 valores possíveis. Contudo, conforme o número de documentos na coleção aumenta o vocabulário também aumenta e, portanto, apenas 3 valores de *hash* não serão suficiente para representar, sem perda de informação, a coleção inteira. De fato, o exemplo da figura 4.2 ilustra o impacto do número de funções de *hash* para representar um documento d em uma coleção de 1000 documentos. No exemplo em questão, mediu-se o número de colisões de valores de *hash*, entre d e 16 documentos contendo texto que plagiaram d, na medida em que se aumentava o número de funções de *hash* para representar a coleção. Vale ressaltar que com apenas 60 valores de *hash* foi possível identificar todos os casos de plágio e, portanto, reduzir de 111382 palavras para 60 *hashes*, por documento, o espaço de representação da coleção. Com isso, podemos observar que as funções de *hash*, dos métodos LSH, selecionam e representam características, de forma a reduzir o espaço de representação e o tempo de busca, para encontrar elementos similares de uma coleção.

Na etapa de Recuperação Heurística os métodos LSH devem considerar as características textuais que quantificam e caracterizam os documentos de forma a favorecer a identificação de plágio. As características em questão devem abordar aspectos léxicos, sintáticos, semânticos ou estruturais (ALZAHRANI *et al.*, 2012). Por exemplo, do ponto de vista léxico os documentos podem ser divididos em n-gramas de caracteres, do ponto de vista sintático a ordem da palavra ou a estrutura da frase podem ser utilizadas enquanto que sinônimos e classes gramaticais podem ser utilizados para representar características semânticas. Desse modo, o desafio é prover formas de mapear a combinação dos quatro conjuntos de aspectos em valores de *hash* que aumentem a probabilidade de colisões entre um documento que plagiou e o documento que foi alvo do plágio. Além disso, outro desafio de similar importância é o de conseguir reduzir o tempo de representar os documentos da coleção e o tempo de busca ao se reduzir o tempo de geração de *hashes*.

O presente capítulo é dividido em 6 seções onde a seção 4.1 apresenta o embasamento teórico para os métodos LSH, a seção 4.2 apresenta os tipos de métodos LSH existentes, na seção 4.3 os *pipelines* de execução LSH são discutidos e um pipeline para aplicação de métodos LSH na Recuperação Heurística é apresentado; Os

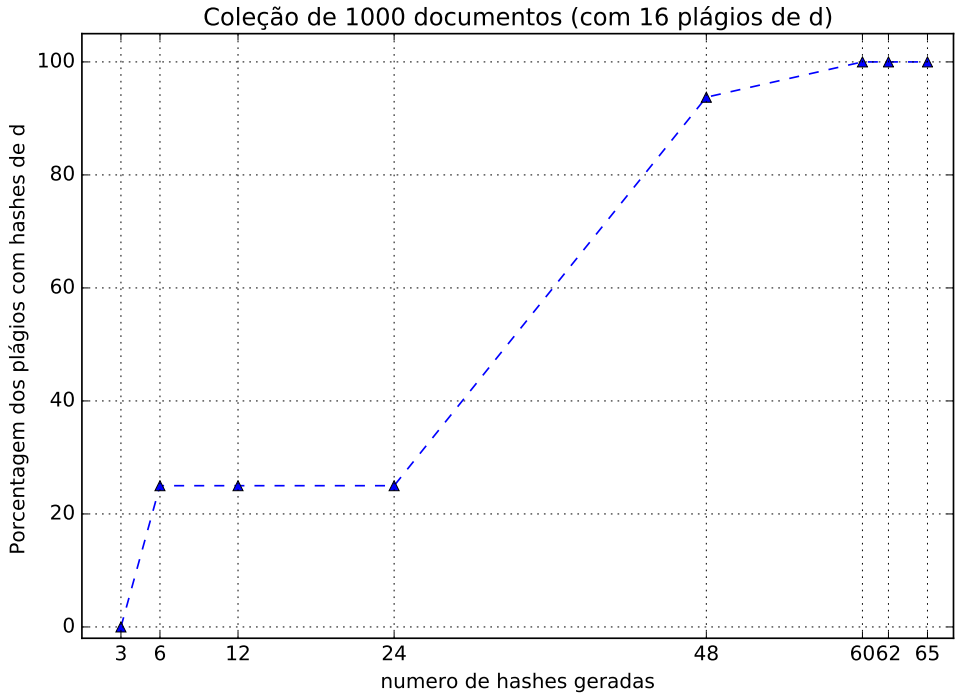


Figura 4.2: Variando o número de *hashes* para representar 1000 documentos da coleção PAN_{10} (POTTHAST *et al.*, 2010b). Os documentos foram particionados em palavras gerando um vocabulário de 111382 palavras distintas.

métodos LSH baseados em permutações conhecidos como *Minwise e Minmaxwise hashing* são apresentados na seção 4.4 enquanto que os algoritmos para indexação e busca usando assinaturas LSH são discutidos na seção 4.5. Por fim, a seção 4.6 apresenta os métodos LSH aplicados na Recuperação Heurística e outros trabalhos relacionados.

4.1 *Semilattices* e LSH

A ideia central por trás dos métodos LSH está em calcular a similaridade entre dois objetos, por exemplo dois textos, duas imagens, dois vídeos e etc., representando-os como conjuntos de valores de *hash*. Logo, é preciso compreender como os métodos LSH preservam a similaridade entre dois objetos ao representá-los como conjuntos de *hash*, o que é explicado ao se compreender o que é um *semilattice* e os conjuntos de ordem parcial. Isto é, os métodos LSH utilizam operações n-árias que definem um conjunto de ordem parcial, para cada objeto, que a partir de propriedades, definidas pelas ordens parciais, é possível avaliar a probabilidade das operações gerarem valores de *hash* iguais para dois elementos similares. Portanto, esta seção irá definir os conceitos de *semilattice*, conjunto de ordem parcial e seus elementos de mínimo e máximo para, em seguida, apresentar como os métodos LSH utilizam dos mesmos para garantir a similaridade entre dois objetos.

A definição 2 formaliza que operações podem ser utilizadas pelos métodos LSH para manipular conjuntos. Isto é, as operações apresentam aridade n e devem gerar valores que estão presentes no conjunto original S. Por exemplo, a função $F(x) = |x|$ é uma operação, de aridade 1, válida nos conjuntos $S = \{-33, 0, 1, 2, 33, 75.9\}$, $S = \mathbb{Z}$ e $S = \mathbb{R}$. Contudo, não é uma operação válida nos conjuntos $S = \mathbb{Z}^-$ ou $S = \mathbb{R}^-$. Outro exemplo de operação, de aridade 4, válida em $S = \mathbb{Z}$ e $S = \mathbb{R}$ é a função $F(x, y, z, w) = x - y - z - w$.

Definição 2. *Em um conjunto S uma operação n-ária é definida como* $F : S^n \mapsto S$[1].

Operações binárias, i.e. operações de aridade 2, são operações muito comuns que, para efeito de simplicidade, podem ser representadas de 3 formas: $F : S^2 \mapsto S$, $F(x, y)$ ou xFy (SIMOVICI e DJERABA, 2014, p.35). Logo, de acordo com SIMOVICI e DJERABA (2014, p.35, def. 1.132), as operações binárias podem ser classificadas como associativas, comutativas ou idempotentes. Isto é, em um conjunto S uma operação F é dita:

1. associativa se $(xFy)Fz = xF(yFz)\ \forall x, y, z \in S$
2. comutativa se $xFy = yFx\ \forall x, y \in S$
3. idempotente se $xFx = x\ \forall x \in S$

Exemplos de operações associativas, comutativas e idempotentes são as operações de interseção e união entre subconjuntos de S enquanto que, as operações de soma e multiplicação, no conjunto de números reais, são apenas associativas e comutativas (SIMOVICI e DJERABA, 2014, p.35).

Um *semilattice* é um semigrupo $\delta = (S, \{F\})$ tal que F é associativa, comutativa e idempotente (SIMOVICI e DJERABA, 2014, p.539, def. 11.1). Exemplos de *semilattices* são a operação de máximo, i.e. $F_1(x, y) = max(x, y)$, e mínimo, i.e. $F_2(x, y) = min(x, y)$, entre dois números reais $(x, y \in \mathbb{R})$. Vale ressaltar que as operações de máximo e mínimos definem *semilattices* $\delta_1 = (S, \{F_1\})$ e $\delta_2 = (S, \{F_2\})$ em qualquer conjunto numérico S tal que $S \subset \mathbb{R}$.

A partir dos *semilattices* conjuntos de ordens parciais podem ser definidos. Isto é, o *semillatice* $\delta = (S, \{F\})$ define um conjunto de ordem parcial $(S, \leq)$ tal que a relação $x \leq y$ é definida por $x = xFy\ \forall x, y \in S$ (SIMOVICI e DJERABA, 2014, p.539, def. 11.3). Ademais, se x é dito um elemento mínimo de $(S, \leq)$ então não existe outro elemento, em S, que seja menor que x. Por outro lado x é dito um elemento máximo de $(S, \leq)$ se não existe elemento de S maior que x (SIMOVICI e

[1](SIMOVICI e DJERABA, 2014, p.35, def. 1.131)

DJERABA, 2014, p.73, teo. 2.23). Contudo, nem todo conjunto de ordem parcial apresenta elementos mínimos ou máximos. Por exemplo, para o conjunto de ordem parcial definido pelo *semilattice* $\delta = (\mathbb{R}, \{F_2\})$, tal que $F_2(x, y) = min(x, y)$, não existem elementos mínimos ou máximos visto que $\mathbb{R}$ contém infinitos valores reais negativos e positivos e, portanto, não é possível encontrar o último menor valor (elemento mínimo) ou o maior valor (elemento máximo) (SIMOVICI e DJERABA, 2014, p.73, teo. 2.23).

O teorema 1 apresenta a relação entre um método LSH e os *semilattices*. Nele é possível estimar a probabilidade da operação F representar dois conjuntos similares S_1 e S_2, com o mesmo valor, a partir dos seus *semilattices* $\delta_1 = (S_1, \{F\})$ e $\delta_2 = (S_2, \{F\})$. Logo, a função de hash de um método LSH será a operação F que define todos os *semilattices* da coleção desde que cada *semilattice* $\delta_i = (S_i, \{F\})$ possa definir um conjunto de ordem parcial $(S_i, \leq)$.

Teorema 1. *Sejam S_1 e S_2 dois conjuntos finitos e não vazios tais que $\delta_1 = (S_1, \{F\})$ e $\delta_2 = (S_2, \{F\})$ são conjuntos de ordens parciais em F. A probabilidade de um elemento "a" que pertence a δ_1, i.e. $a = xFy$ tal que $x, y \in S_1$, ser igual a um elemento "b" que pertence a δ_2, i.e. $b = zFw$ tal que $z, w \in S_2$, é $Pr[a = b] \geq \frac{|S_1 \cap S_2|}{|S_1 \cup S_2|}$*

Demonstração. Seja $a = xFy$ tal que $x, y \in S_1$, $b = zFw$ tal que $z, w \in S_2$, $min_{S_1} \leq x$ tal que $\forall x \in S_1$ e $min_{S_2} \leq z$ tal que $\forall z \in S_2$ tais que S_1 e S_2 são dois conjuntos finitos e não vazios onde $\delta_1 = (S_1, \{F\})$ e $\delta_2 = (S_2, \{F\})$ são conjuntos de ordens parciais em F.

$$Pr[a = b] = \sum_{a \neq min_{S_1}} \Big(Pr[a = b] \Big) + Pr[min_{S_1} = b]$$

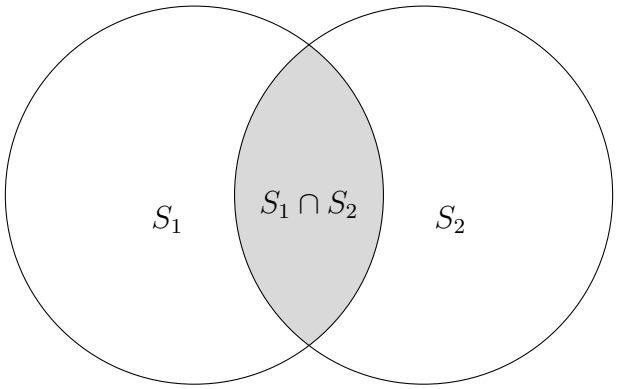
$$Pr[a = b] = \sum_{a \neq min_{S_1}} \Big(Pr[a = b] \Big) + \sum_{b \neq min_{S_2}} \Big(Pr[min_{S_1} = b] \Big) + Pr[min_{S_1} = min_{S_2}]$$

Definindo $\Delta = \sum_{a \neq min_{S_1}} \Big(Pr[a = b] \Big) + \sum_{b \neq min_{S_2}} \Big(Pr[min_{S_1} = b] \Big)$ temos que:

$$Pr[a = b] = \Delta + Pr[min_{S_1} = min_{S_2}] \text{ tal que } 1 > \Delta \geq 0$$
$$\textit{então } Pr[a = b] \geq \Pr[min_{S_1} = min_{S_2}]$$

Ademais, δ_1 e δ_2 serem conjuntos de ordens parciais garante que min_{S_1} e min_{S_2} existem. Logo, basta demonstrar que $\Pr[min_{S_1} = min_{S_2}] = \frac{|S_1 \cap S_2|}{|S_1 \cup S_2|}$ o que é demonstrado em (BRODER, 1997) e reproduzido a seguir.

$min_{S_1} = min_{S_2}$ ocorre quando S_1 e S_2 tem um elemento em comum e, portanto, este elemento pertence a $S_1 \cap S_2$ conforme a figura abaixo ilustra.



Logo, existem $|S_1 \cap S_2|$ elementos que podem ser representados pelo número a tal que $a = min_{S_1}$ e $a = min_{S_2}$. Portanto a $\Pr[min_{S_1} = min_{S_2}]$ é a probabilidade de escolher um elemento na interseção de S_1 e S_2 entre todos os elementos de S_1 e S_2, isto é:

$$\Pr[min_{S_1} = min_{S_2}] = \frac{\text{número de elementos da interseção de } S_1 \text{ e } S_2}{\text{número de elementos de } S_1 \text{ e } S_2} = \frac{|S_1 \cap S_2|}{|S_1 \cup S_2|}$$

□

4.2 Tipos LSH

Os métodos LSH, também conhecidos como métodos de hashing aleatórios, apresentam funções binárias $h(\cdot)$, que mapeiam os pontos de h para um espaço de *semilattice*, tais que a probabilidade de dois pontos terem a mesma assinatura é proporcional a sua similaridade (PAN e MANOCHA, 2012, WANG *et al.*, 2015). Além disso, existe uma diversidade de métodos LSH que, de acordo com WANG *et al.* (2015), podem ser divididos em métodos baseados em projeções aleatórias e métodos baseados em permutações aleatórias.

Os métodos Baseados em Projeções Aleatórias (B_{pro}) apresentam uma alta probabilidade de mapear pontos próximos, no espaço original, para os mesmos valores de hash. Nos métodos B_{pro} a probabilidade de colisão para duas amostras x_i, x_j apresentarem o mesmo valor de hash é determinada pelo ângulo θ_{ij} entre elas, no espaço original, assim como apresentado na equação 4.4. Contudo, as funções de hash dos métodos B_{pro} são independentes dos dados, i.e. ignoram as propriedades específicas de um conjunto de dados, o que pode gerar métodos menos efetivos em comparação aos métodos que aprendem a partir da coleção (WANG *et al.*, 2015).

$$Pr[h_k(x_i) = h_k(x_j)] = 1 - \frac{\theta_{ij}}{\pi} = 1 - \frac{1}{\pi} cos^{-1} \frac{x_i^T x_j}{|x_i||x_j|} \tag{4.4}$$

Os métodos Baseados em Permutações Aleatórias (B_{per}) usam funções de hash $h_i(\cdot)$ para transformar um espaço $\mathbb{R}^D$ em um espaço $\mathbb{Z}^M$ que distribui cada item de dado v em valores $H(v) = (h_1(v), h_2(v), h_3(v), \ldots, h_M(v))$ onde o espaço $\mathbb{Z}^M$ geralmente é implementado a partir de um índice invertido (WANG *et al.*, 2015). Cada função de hash $h_i(\cdot)$ permuta aleatoriamente os valores de v e, em seguida, a operação F que define o *semilattice* $\delta_i = (v, \{F\})$, de um conjunto parcialmente ordenado $(v, \leq)$, seleciona o valor que representará v.

Os métodos LSH B_{per} mais representativos são os baseados na seleção de mínimos a partir de permutações independentes. Entre esses métodos, o *Minwise Hashing* vem sendo amplamente utilizados para medir a similaridade de Jaccard, entre conjuntos ou vetores, de forma aproximada (WANG *et al.*, 2015) e é discutido na seção 4.4.

4.3 Pipeline para Recuperação Heurística com LSH

As abordagens LSH lidam com a tarefa de identificar elementos similares comparando os seus conteúdos em coleções extensas. Para tanto, WANG *et al.* (2015) afirmam que existem três passos para se realizar uma busca aproximada: Modelagem da função de hash, indexação utilizando tabelas de hash e a busca utilizando hashes. Contudo, os três passos anteriores não descrevem, de forma objetiva, o processo para a busca em coleções de texto. A figura 4.3 estende o trabalho de VIEIRA (2016) e descreve as abordagens LSH, aplicadas na Recuperação Heurística, como uma sequência de cinco passos: (*i*) Tokenização, (*ii*) geração de *fingerprint*, (*iii*) permutação de característica, (*iv*) execução da função de seleção e (*v*) avaliação da similaridade. Cada um dos cinco passos possibilitam a comparação de abordagens diferentes e, portanto, esta seção apresenta os passos e conceitos inerentes aos métodos LSH. Um operador é definido na subseção 4.3.1, enquanto os passos (i), (ii), (iii), (iv), and (v) são descritos nas subseções 4.3.2, 4.3.3, 4.3.4, 4.3.5, e 4.3.6, respectivamente.

4.3.1 Operadores

Os operadores são conjuntos de características diferentes que podem ser utilizadas para sumarizar a representação de documentos. A equação 4.5a define um operador o_p como um critério para selecionar elementos de um conjunto $S_i \in S$.

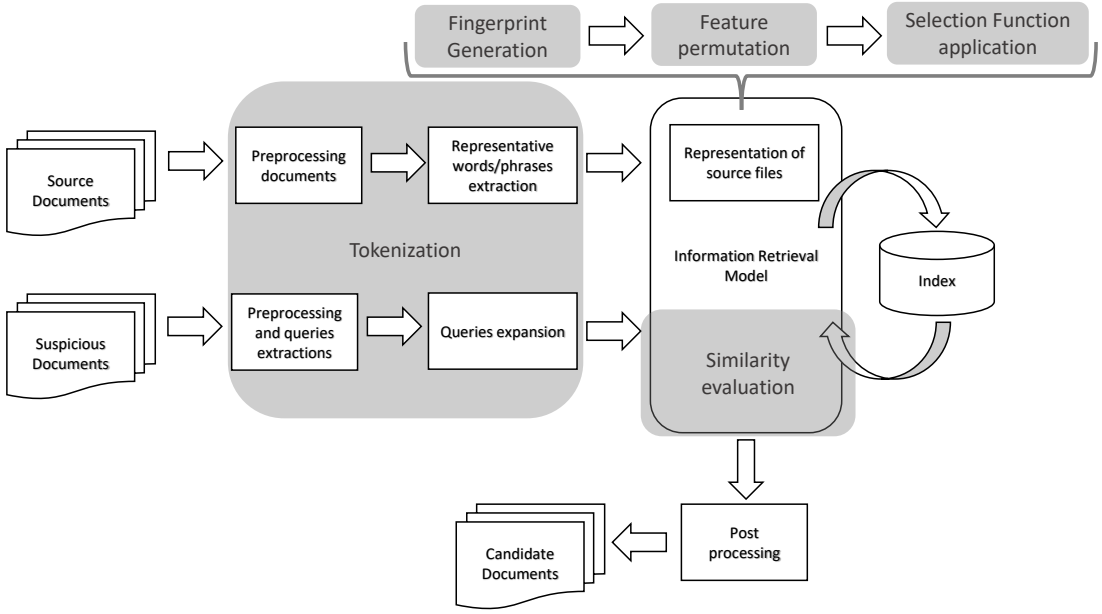


Figura 4.3: Sequência de passos para métodos baseados em operadores

Um exemplo é o método *Minwise hashing* (BRODER, 1997), que usa o operador mínimo, como definido na equação 4.5b. Embora o método *Minwise hashing* explore apenas o operador mínimo, existe uma extensa variedade de operadores que podem ser explorados; por exemplo, o máximo, o i-ézimo mínimo e o j-ézimo máximo. Logo, assumindo que O é o universo de operadores, a Tabela 4.1 apresenta alguns exemplos de subconjuntos de operadores $O_k \subset O$ que podem se utilizados para compactar a representação de documentos.

$$o_p(S_i) = \begin{cases} \textit{conjunto de } p \in S_i \textit{ que atendem aos } \text{critérios de } o_p, \textit{ se } |S_i| \geq p \\ S_i, \textit{ outros casos} \end{cases} \tag{4.5a}$$

$$min_p(S_i) = \begin{cases} \textit{o conjunto dos } p \textit{ menores elementos em } S_i, \textit{ se } |S_i| \geq p \\ S_i, \textit{ outros casos} \end{cases} \tag{4.5b}$$

4.3.2 (i) Tokenização

Este passo extrai uma matriz termo-documento do conteúdo de todos os documentos da coleção D, de tamanho $|D|$. Para tanto, uma matriz binária $M \in$

Tabela 4.1: Examples of $O_k \subset O$.

k	O_k	Critério
1	menor valor	$\{min_1\}$
2	maior valor	$\{max_1\}$
3	menor e maior valores	$\{min_1, max_1\}$
4	primeiro e segundo menores valores	$\{min_1, min_2\}$
5	primeiro e segundo maiores valores	$\{max_1, max_2\}$
6	Média de todos os valores	$\{mean\}$
7	segundo menor valor e maior valor	$\{min_2, max_1\}$
8	p-ézimo menor valor	$\{min_P\}$

$\{0,1\}^{J\times I}$, conhecida como matriz de características M, é construída após cada documento de D ser mapeado em um conjunto de termos. Cada coluna i de M, gerada a partir de um documento genérico d_i, corresponde a um conjunto S_i; enquanto cada linha de M corresponde a um termo t_j do vocabulário T da coleção de documentos. Assim sendo, para cada $d_i \in D$ com $i \in [1, |D|]$ existirá $S_i \subset S$, tal que S é o universo definido pelo grupo que inclui todos os conjuntos S_i.

Como ilustração, suponha uma coleção de documentos $D = \{d_1, d_2\} =$ { *"too good thing"*, *"much of a thing"*} e que, para fins de simplicidade, cada termo extraído será uma palavra presente em D. Logo, teremos um vocabulário $T =$ { *"too"*, *"much"*, *"of"*, *"a"*, *"good"*, *"thing"*}, visto que $d_1 =$ { *"too"*, *"good"*, *"thing"*} e $d_2 =$ { *"much"*, *"of"*, *"a"*, *"thing"*} e, portanto, D apresentará uma matriz M:

$$\mathrm{M} = \begin{array}{c} \\ too \\ much \\ of \\ a \\ good \\ thing \end{array} \begin{array}{c} \begin{array}{cc} d_1 & d_2 \end{array} \\ \begin{pmatrix} 1 & 0 \\ 0 & 1 \\ 0 & 1 \\ 0 & 1 \\ 1 & 0 \\ 1 & 1 \end{pmatrix} \end{array}$$

4.3.3 (*ii*) Geração de *fingerprint*

O segundo passo gera uma *fingerprint* númerica g para cada termo $t_j \in T$. Logo, a *fingerprint* g_j pode ser gerada através de uma transformação $\psi(t_j)$, conforme definido na equação 4.6.

$$L = [\psi(t_j) \text{ para todo } t_j \in T], \text{onde } \psi(t_j) : T \to \mathbb{N} \tag{4.6}$$

O propósito deste passo é mapear todos os termos em inteiros positivos e não nulos que serão utilizados como entrada para o passo de permutação de característica

(4.3.4). Logo, a equação 4.6 gera L como uma representação discreta, contínua e única para o vocabulário da coleção T.

Aplicando, no exemplo anterior, uma transformação $\psi(t_j)$ que representa t_j em T como o seu índice j gerará $L = [1, 2, 3, 4, 5, 6]$. Logo, usando L é possível representar cada documento de D em M gerando a matriz M':

$$M' = \begin{array}{c} \\ \psi(too) \\ \psi(much) \\ \psi(of) \\ \psi(a) \\ \psi(good) \\ \psi(thing) \end{array} \begin{array}{c} \begin{array}{cc} d_1 & d_2 \end{array} \\ \begin{pmatrix} 1 & 0 \\ 0 & 2 \\ 0 & 3 \\ 0 & 4 \\ 5 & 0 \\ 6 & 6 \end{pmatrix} \end{array}$$

4.3.4 (*iii*) Permutação de Características

Considerando L como uma sequência de *fingerprints* geradas por $\psi(t_j)$, no passo anterior, a ordem de L corresponderá a ordem presente nas linhas de M. Em seguida, a equação 4.7 define L_n como a sequência obtida da permutação π_n aplicada em L.

$$L_n = \pi_n(L) \tag{4.7}$$

A permutação genérica π_n é executada atráves de funções de hash que reordenam L de forma aleatória. Logo, este passo produz uma nova matriz M_n, em que as linhas são ordenadas de acordo com a ordem da sequência L_n. A equação 4.8 define, no universo finito L^P de todas as permutações possíveis de L, π_n^{max} como a posição de ordem máxima para um elemento de uma sequência $L_n \subset L^P$ ordenada por π_n, i.e. π_n^{max} corresponderá ao maior valor que é possível representar um termo de L. Portanto, para um vocabulário de termos T, de uma coleção de documentos D, qualquer função de permutação π_n tem $\pi_n^{max} = |T|$.

$$\pi_n^{max} = \max_{L_n \in L^P}(\pi_n(L)) \tag{4.8}$$

Logo, para o exemplo anterior, é possível escolher duas permutações π_1 e π_2 como $L_1 = [1, 2, 3, 4, 5, 6]$ e $L_2 = [6, 5, 4, 3, 2, 1]$ para permutar os conjuntos de assinaturas, de $d_i \in D$, representados como as colunas de M'. Gerando assim as matrizes M'', para π_1, e M''', para π_2.

$$M'' = \begin{array}{c} \\ \psi(too) \\ \psi(much) \\ \psi(of) \\ \psi(a) \\ \psi(good) \\ \psi(thing) \end{array} \begin{array}{c} \begin{array}{cc} \pi_1(d_1) & \pi_1(d_2) \end{array} \\ \begin{pmatrix} \mathbf{1} & 0 \\ 0 & \mathbf{2} \\ 0 & 3 \\ 0 & 4 \\ 5 & 0 \\ 6 & 6 \end{pmatrix} \end{array} \times M''' = \begin{array}{c} \\ \psi(too) \\ \psi(much) \\ \psi(of) \\ \psi(a) \\ \psi(good) \\ \psi(thing) \end{array} \begin{array}{c} \begin{array}{cc} \pi_2(d_1) & \pi_2(d_2) \end{array} \\ \begin{pmatrix} 6 & 0 \\ 0 & 5 \\ 0 & 4 \\ 0 & 3 \\ 2 & 0 \\ \mathbf{1} & \mathbf{1} \end{pmatrix} \end{array}$$

4.3.5 (iv) Execução de Função de Seleção

As funções de seleção μ são responsáveis por sumarizar a representação de um conjunto de documentos após serem permutados. Logo, este passo tem como objetivo prover suporte para o cálculo de similaridade entre os conjuntos de características de dois documentos. Isto é possível pois a seleção dos subconjuntos de elementos mais representativos desses documentos, leva à seleção de elementos na interseção entre eles. Portanto, a similaridade entre eles aumenta (BRODER, 1997) se eles compartilham muitos elementos em comum.

A sumarização das permutações dos conjuntos de características é realizada ao se escolher uma função de seleção μ (Equação 4.9) baseada no conjunto de operadores O_k que será explorado; por exemplo, o mínimo, o máximo ou qualquer outro operador escolhido de O. As permutações levam a ordens diferentes em L e, portanto, é possível gerar valores diferentes para cada operador escolhido de acordo com cada ordem. Ademais, um conjunto de todos os elementos representativos, montado por μ, é conhecido como conjunto de assinaturas Sig. As equações (4.9) e (4.10) formalizam a assinatura $s_{i,n,k} \subset Sig$ como os elementos representativos, gerados a partir de permutações π_n e selecionados por μ de acordo com um conjunto de operadores $O_k \subset O$.

$$\mu(d_i, \pi_n, O_k) : T \mapsto Sig \tag{4.9}$$

$$s_{i,n,k} = \mu(d_i, \pi_n, O_k) \tag{4.10}$$

No exemplo anterior, após a geração de *fingerprint* e a permutação de características, foi aplicada a função de seleção μ_1. μ_1 apresenta um conjunto de operadores O_k composto do operador mínimo $o_{p=1} \in O_k$ (Tabela 4.1) e, portanto, a função de seleção $\mu_1(d_1, \pi_1, o_{p=1})$ retornará o termo "*too*" como elemento representativo de d_1. Isto ocorre por que os termos de d_1, mapeados de acordo com π_1, tem como seu menor valor em L_1 a *fingerprint* 1. Em contrapartida, o elemento representativo de d_1, para a permutação π_2 é "*thing*". De forma similar, μ_1 afirma que o conjunto d_2 tem *"much"* e *"thing"* como seus elementos representativos, de acordo com o

operador mínimo executado a partir das permutações π_1 e π_2, respectivamente.

Um mecanismo de índice invertido orientado a assinaturas indexa D para acelerar a tarefa de busca. A aceleração é realizada pois o índice invertido escolhe, para comparar par-a-par, apenas documentos $d_i \in D$ com pelo menos uma assinatura igual as da consulta d_q. A tabela 4.2 apresenta o índice invertido da coleção D, do exemplo anterior, e para uma consulta $d_3 = \{$ *"too", "good"* $\}$ com $s_{3,1,1} = 1$, $s_{3,2,1} = 5$ apenas d_1 será comparado com d_3.

Tabela 4.2: Ocorrências de d_1 e d_2 no índice invertido baseado em assinaturas

Elementos representativos	$s_{i,n,k}$	Ocorrências no índice invertido
too	$s_{i,1,1} = 1$	$[d_1]$
much	$s_{i,1,1} = 2$	$[d_2]$
thing	$s_{i,2,1} = 1$	$[d_1, d_2]$

4.3.6 (v) Avaliação de similaridade

A similaridade entre dois documentos d_i e d_j é calculada a partir das representações de d_i e d_j na matriz de assinaturas Sig. Logo, a equação 4.11 calcula a similaridade $sim_m(d_i, d_j)$, entre d_i e d_j, de acordo com a métrica m.

$$sim_m(d_i, d_j) : Sig \times Sig \mapsto \mathbb{R} \tag{4.11}$$

4.4 Métodos *Minwise*, *Maxwise*, and *Min-maxwise hashing*

O coeficiente de similaridade de Jaccard é amplamente utilizado para medir similaridade entre dois conjuntos (JI *et al.*, 2013). Também conhecido como "semelhança" (LI e KÖNIG, 2011), o coeficiente de Jaccard $J(A, B)$ entre dois conjuntos A e B é definido de acordo com a equação 4.12.

$$J(A, B) = \frac{|A \cap B|}{|A \cup B|}, \tag{4.12}$$

Os métodos baseados em hashing aleatório (e.g. LSH) usam a similaridade de Jaccard como fundamentação teórica para esboçar os dados, enquanto preserva a similaridade par-a-par (DONG *et al.*, 2008, JI *et al.*, 2013). Além disso, para dados esparsos representados como conjunto finitos (e.g. documentos como conjuntos de termos), manter um esboço compacto dos dados é uma forma efetiva de lidar com a maldição da alta dimensionalidade (INDYK e MOTWANI, 1998, JI *et al.*, 2013).

Dado que dois conjuntos parcialmente ordenados A e B são as representações de dois documentos d_a e d_b, *Minwise hashing* (BRODER, 1997) é definido como um método de hashing baseado na propriedade (4.13), entre A e B, depois de se executar a permutação aleatória π_n. Ademais, em uma coleção de documentos D, tal que $d_i \in D$, com um vocabulário T de tamanho $|T|$, existem $|T|!$ permutações distintas (π_n) para a representação S_i de um documento, isto é $1 \leq n \leq |T|!$.

$$Pr(min(\pi_n(A)) = min(\pi_n(B))) = \frac{|A \cap B|}{|A \cup B|} = J(A,B), \text{ where}$$
$$\pi_n(S_i) \text{ é a ordem dos elementos de } d_i \text{ de acordo com } L_n \tag{4.13}$$

Formalmente, na equação 4.14, cada permutação aleatória e independente π_n apresenta uma variável aleatória X_{π_n}. Consequentemente, $X_\pi = \frac{1}{N}\sum_{n=1}^{N} X_{\pi_n}$ é um estimador sem viés de $J(A,B)$ com variância $Var[X_\pi] = \frac{1}{N}\, J(A,B)(1 - J(A,B))$, visto que $\mathbb{E}[X_{\pi_n}] = \frac{1}{N}\sum_{n=1}^{N} Pr(X_{\pi_n} = 1) = J(A,B)$ após N permutações aleatórias e independentes (JI *et al.*, 2013). Além disso, o *Minwise hashing* indexa cada conjunto S_i com N valores inteiros $min(\pi_n(S_i)) \mid n \in \{1, 2, \ldots, N\}$.

$$X_{\pi_n} = \begin{cases} 1, & \text{se } min(\pi_n(A)) = min(\pi_n(B)) \\ 0, & \text{caso contrário} \end{cases} \tag{4.14}$$

Contudo, não existe razão para esperar que o mínimo é a única função com esta propriedade. De fato, a demonstração disponível em (BRODER, 1997) foi repetida para a função de máximo, que apresenta o mesmo comportamento do mínimo (JI *et al.*, 2013) e, portanto, a equação 4.15 formaliza uma variável aleatória Z_{π_n} para a função de máximo (JI *et al.*, 2013).

$$Z_{\pi_n} = \begin{cases} 1, & \text{se } max(\pi_n(A)) = max(\pi_n(B)) \\ 0, & \text{caso contrário} \end{cases} \tag{4.15}$$

Vale ressaltar que, a subseção 4.1 definiu e discutiu que algumas operações definidas a partir de *semilattices*, assim como o mínimo e o máximo, podem ser utilizadas como funções de hash para métodos LSH. Portanto, a equação 4.16 generaliza, a partir do teorema 1, a forma de calcular a probabilidade de colisão utilizando *semilattices* com funções de seleção.

$$Pr(o_i(\pi_n(S_1)) = o_i(\pi_n(S_2))) \geq \frac{|S_1 \cap S_2|}{|S_1 \cup S_2|} \tag{4.16}$$

4.4.1 Melhorias ao se combinar os operadores de máximo e mínimo

A função de seleção utilizada pelo método *Minwise hashing*, $\mu(d_i, \pi_n, O_1)$[2], é apresentada no algoritmo 1 e pode ser facilmente adaptada para o método *Maxwise hashing*. Contudo, ao invés de manter apenas o menor ou o maior valor de cada permutação π_n, o método *Minmaxwise hashing* (O_3) (JI *et al.*, 2013) combina as vantagens de selecionar o mínimo e o máximo assim como apresentado no algoritmo 2. JI *et al.* (2013) prova que O_3 provê um estimador sem viés para a similaridade de Jaccard, com resultados ligeiramente melhores que os de O_1. Além disso, O_3 reduz o tempo de O_1 pela metade pois, para selecionar n valores para d_i, $\mu(d_i, \pi_n, O_3)$ precisa de apenas $N/2$ permutações, enquanto que $\mu(d_i, \pi_n, O_1)$ necessita de N permutações (JI *et al.*, 2013).

Algorithm 1: *Minwise hash* (O_1) PARA π_n EM UM CONJUNTO S_i

Input: $\pi_n(S_i) = \{s_1, s_2, s_3, \ldots, s_t\}$

```
1 Initialize: minId = s_1
2 begin
3     for i ← 1 to t do
4         if minId > s_i then
5             minId ← s_i
6         end
7     end
8 end
```

Output: $minId$

A função de seleção do método *Minmaxwise hashing* é baseada em operações, a partir de um *lattice* $\delta = (S, \{min, max\})$, que definem um conjunto parcialmente ordenado. Onde, assim como um *semilattice* $(S, \{F\})$ garante que o conjunto parcialmente ordenado $(S, \leq)$ apresenta um valor ínfimo ou um valor supremo, um *lattice* garante que $(S, \leq)$ apresenta ambos os valores ínfimo e supremo (SIMOVICI e DJERABA, 2014).

[2]Tabela 4.1.

Algorithm 2: *Minmaxwise hash* (O_1) PARA π_n EM UM CONJUNTO S_i

Input: $\pi_n(S_i) = \{s_1, s_2, s_3, \ldots, s_t\}$

Initialize: $minId = s_1, maxId = s_1$

begin

  for $i \leftarrow 1$ ***to*** t **do**

    if $minId > s_i$ **then**

      $minId \leftarrow s_i$

    end

    if $maxId < s_i$ **then**

      $maxId \leftarrow s_i$

    end

  end

end

Output: $minId, maxId$

O intervalo $[min(\pi_n(S_i)), max(\pi_n(S_i))]$ de um *lattice* $\delta = (\pi_n(S_i), \{min, max\})$ é o intervalo de valores com fronteiras $a, b \in \pi_n(S_i)$ tais que $t \in \pi_n(S_i) \mid a \leq t \leq b$. Logo, apesar do método *Minmaxwise hashing* medir a similaridade comparando os valores de fronteira do intervalo de $(\pi_n(S_i), min, max)$, ele não lida com outras propriedades do intervalo de $\pi_n(S_i)$ como o tamanho do intervalo. Contudo, analisar apenas o tamanho do intervalo, de cada permutação aleatória π_n, não preserva a similaridade par-a-par pois, por exemplo, dois conjuntos diferentes, A (equação 4.17a) e E (equação 4.17e), podem apresentar intervalos diferentes com o mesmo tamanho de intervalo na permutação π_3[3]. Portanto, se faz necessário explorar abordagens que possam sumarizar o intervalo dos *lattices*, para representar conjuntos (como A (4.17a) e B (4.17b)), enquanto a similaridade entre os conjuntos é mantida.

$$\pi_3\{A\} = \{5, 3, 1, 2, 4\} \implies \text{Intervalo de valores: } [1, 5], \text{ de tamanho} = 5. \quad (4.17a)$$

$$\pi_3\{B\} = \{5, 1, 2, 4\} \implies \text{Intervalo de valores: } [1, 5], \text{ de tamanho} = 5. \quad (4.17b)$$

$$\pi_3\{C\} = \{5, 3, 1, 2, 4, 9\} \implies \text{Intervalo de valores: } [1, 9], \text{ de tamanho} = 9. \quad (4.17c)$$

$$\pi_3\{D\} = \{5, 3, 2, 4\} \implies \text{Intervalo de valores: } [2, 5], \text{ de tamanho} = 4. \quad (4.17d)$$

$$\pi_3\{E\} = \{45, 42, 41\} \implies \text{Intervalo de valores: } [41, 45], \text{ de tamanho} = 5. \quad (4.17e)$$

[3]suponha que π_3 é uma permutação gerada aleatoriamente de um intervalo de valores de 1 a 45

4.5 Algoritmos para Indexação e Busca

A subseção 4.3.5 discute como a função de seleção $\mu(d_i, \pi_n, O_k)$ é utilizada para: a geração do índice invertido e a busca dos documentos suspeitos. Logo, o algoritmo de indexação de documentos fonte dos métodos LSH é formalizado no algoritmo 3 enquanto que o algoritmo 4 formaliza a busca de documentos candidatos a fonte de documentos suspeitos de plágio. Ademais, o algoritmo 3 cria um Índice Invertido Completo λ, visto que é possível recompor o documento original, a partir de λ, assim como discutido na subseção 3.2.

O algoritmo 4 busca em λ o conjunto de documentos candidatos a fonte D^*_{susp} de um documento suspeito de plágio. Como entrada ele recebe um conjunto de consultas d_{susp}, geradas pelos passos (i) e (ii) e discutidos nas subseções 4.3.2 e 4.3.3, e, em seguida, gera, permuta e seleciona as *fingerprints* que serão utilizadas como elementos de busca em λ.

Algorithm 3: Criação o índice invertido dos documentos fonte $|D|$ usando LSH. Passos (iii) e (iv)

Input: $D = \{d_1, d_2, d_3, \ldots, d_{|D|}\}$, P = número de permutações

1 **Initialize:** $\lambda =$ índice invertido
2 **begin**
3 **for** $i \leftarrow 1$ ***to*** $|D|$ **do**
4 **for** $n \leftarrow 1$ ***to*** P **do**
5 $S_{i,n,k} \leftarrow \mu(d_i, \pi_n, O_k)$
6 **foreach** $sig \in S_{i,n,k}$ **do**
7 $\lambda.sig \leftarrow \lambda.sig \cup \{d_i\}$
8 **end**
9 **end**
10 **end**
11 **end**

Output: λ

Algorithm 4: Busca um um conjunto de consultas d_{susp}, geradas a partir de um documentos suspeito, no índice invertido dos documentos fonte D usando LSH.

Input: $d_{susp} = \{q_1, q_2, q_3, \ldots, q_{|d_{susp}|}\}$, λ = índice invertido,
P = número de permutações

```
1  Initialize: D*_susp = {}
2  begin
3      for i ← 1 to |d_susp| do
4          for n ← 1 to P do
5              S_{i,n,k} ← μ(q_i, π_n, O_k)
6              foreach sig ∈ S_{i,n,k} do
7                  foreach d_src ∈ λ.sig do
8                      D*_susp ← D*_susp ∪ {d_src}
9                  end
10             end
11         end
12     end
13 end
```

Output: D^*_{susp}

Vale ressaltar que, do ponto de vista de execução paralela, ambos os algoritmos podem ser implementados usando paralelismo na tarefa ou paralelismo nos dados. No algoritmo 3 o paralelismo nos dados pode ser efetuado a partir dos documentos de $d_i \in D$ ou de cada permutação a ser aplicada por $\mu(d_i, \pi_n, O_k)$ em d_i enquanto que, no algoritmo 4, o mesmo paralelismo é aplicável em cada consulta $q_i \in d_{susp}$ ou em cada permutação a ser aplicada por $\mu(q_i, \pi_n, O_k)$ em q_i. Exemplos de métodos LSH implementados com execução paralela são (SZMIT, 2013, ZHANG *et al.*, 2016).

4.6 Trabalhos relacionados na Recuperação Heurística

Boa parte das abordagens aplicadas na Recuperação Heurística tem como contribuição central alterações nas etapas periféricas ao modelo de recuperação de informação como o Pré-processamento de documentos suspeitos, a extração de consultas e a expansão de consultas. Isto é, assumem como premissa que a tarefa de Recuperação Heurística é definida como:

Resgate todos os documentos fontes, reduzindo o custo de Busca e Recu-

peração de cada documento, usando um Motor de Busca que armazena todos os documentos que podem ter sido plagiados (POTTHAST *et al.*, 2011).

De fato, a competição anual de identificação de plágio conhecida como PAN[4] enumera e classifica estes tipos de contribuições em *chunking, Keyphrase extraction, Query Formulation, Search Control* e *Download filtering* (POTTHAST *et al.*, 2011). Logo, um motor de busca baseado em LSH também poderá se beneficiar dessas contribuições o que merece um estudo dedicado e, portanto, está fora do escopo do presente trabalho.

Os modelos de recuperação de informação relacionados com os LSH são divididos em *Fingerprinting* e baseados em *Hash* (ALZAHRANI *et al.*, 2012). Os modelos de *Fingerprinting* extraem trechos do texto de um documento para identificá-lo unicamente, no índice invertido, enquanto que os modelos baseados em *Hash* aplicam funções de *Hash* nos trechos de texto extraídos do documento, que geram dimensões de um vetor que são a base para a criação do índice invertido (ALZAHRANI *et al.*, 2012). A busca é realizada a partir das *Fingerprint*s de uma consulta, ou dos seus valores de *hash*, que são submetidas ao índice invertido para selecionar o conjunto de documentos que também as tem, gerando assim um conjunto de documentos que apresentam pelo menos uma *Fingerprint* em comum com a consulta.

Os modelos baseados em *Hash* podem combinar métricas de similaridade entre vetores com o processo de busca para melhorar o resultado. A tabela 4.3 apresenta um conjunto de métricas de similaridades entre vetores levantada por ALZAHRANI *et al.* (2012). Ademais, vale ressaltar que, além da similaridade usando as assinaturas, conforme discutido na subseção 4.3.6, também é possível combinar qualquer métrica de similaridade da tabela 4.3 com os modelos LSH desde que o índice invertido a ser utilizado seja o completo.

[4] pan.webis.de

Tabela 4.3: Lista de métricas de similaridade entre vetores adaptada de ALZAHRANI *et al.* (2012)

Métricas de similaridade	descrição	Equação
Coeficiente *Matching*	similar a distância de Hamming mas entre vetores de mesmo tamanho	$M(x,y) = \|x\| - \|x \cap y\|$
Coeficiente Jaccard	mede o número de termos compartilhados	$J(x,y) = \frac{\|x \cap y\|}{\|x \cup y\|}$
Coeficiente de Dice	similar ao Jaccard mas reduz o efeito dos termos compartilhados	$D(x,y) = \frac{2\|x \cap y\|}{\|x \cup y\|}$
Coeficiente de Sobreposição	se x é um subconjunto de y ou vice-versa o valor é máximo	$O(x,y) = \frac{\|x \cap y\|}{min(\|x\|,\|y\|)}$
Similaridade do cosseno	mede o cosseno do angulo entre x e y	$cos(x,y) = \frac{\sum_i x_i \times y_i}{\sqrt{\sum_i (x_i)^2} \times \sqrt{\sum_i (y_i)^2}}$
Distância euclidiana	mede a distância geométrica entre dois vetores	$Ec(x,y) = \sqrt{\sum_i \|x_i - y_i\|^2}$
Distância euclidiana quadrática	pesos maiores para vetores mais distantes	$SEc(x,y) = \sum_i (x_i - y_i)^2$
Distância de Manhattan	mede a diferença média entre dimensões	$Manh(x,y) = \sum_i \|x_i - y_i\|^2$

Capítulo 5

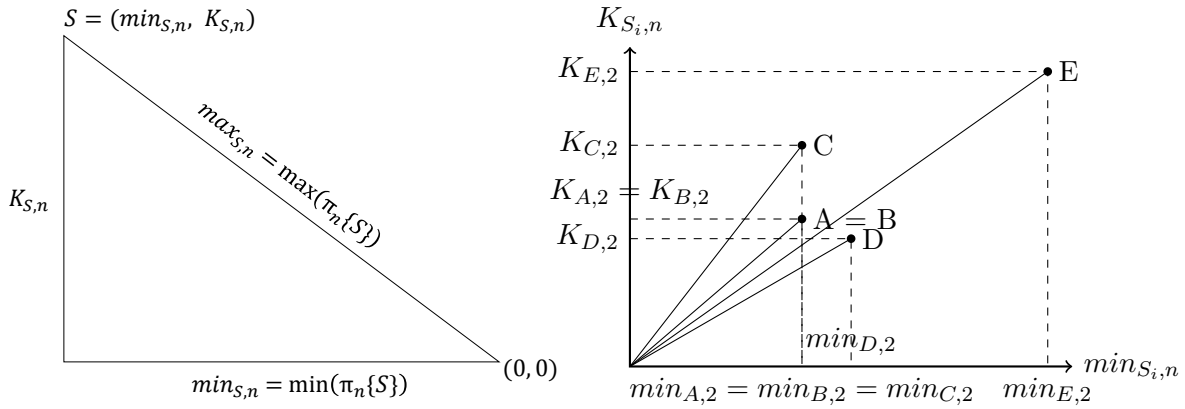
Locality-Sensitive Hashing Usando Arcos de Setores Circulares

At the point of view of individual scholars,
credit for ideas is vital in career terms and,
typically, even more so in terms of self-image.
— Brian Martin, (MARTIN, 1994)

No exemplo (4.17) foram apresentados cinco conjuntos permutados assim como a dificuldade de como representá-los em poucos valores numéricos tentando preservar a similaridade entre eles. Portanto, é desejável que os métodos LSH selecionem valores de assinaturas que garantam uma probabilidade colisão maior entre conjuntos que apresentem interseção e que essa probabilidade aumente proporcionalmente ao tamanho da interseção dos conjuntos. Para tanto, o presente trabalho interpreta cada conjunto de valores (S) como um triângulo retângulo, conforme apresentado na figura 5.1a, e em seguida utiliza do valor cateto $K_{S,n}$ como assinaturas para representar o intervalo de valores de S. Logo, os conjuntos do exemplo (4.17) são representados conforme ilustrado na figura 5.1b e os valores $K_{A,2}$, $K_{B,2}$, $K_{C,2}$, $K_{D,2}$ e $K_{E,2}$ serão as assinaturas selecionadas para os conjuntos A,B,C,D e E na permutação 2. Contudo, esse método, que alcunhamos de propriedade triangular, consegue contemplar apenas os casos em que os valores de mínimo e máximo de dois conjuntos são iguais. Por exemplo, A e B tem a mesma assinatura enquanto os conjuntos C e D são representados com assinaturas diferentes da de A reduzindo assim a chance de se encontrar C e D ao se buscar A.

Uma forma de se aumentar o alcance da propriedade triangular seria selecionar valores de assinaturas que representassem conjuntos que pertencem a um mesmo círculo como A, B e D que pertencem a um circulo de centro (0,0) e raio $max_{A,2} =$

$max_{B,2} = max_{D,2} = 5$. Além disso, outra característica desejável é que a assinatura consiga contemplar conjuntos como o mesmo mínimo como o caso de A, B e C. Logo, os Arcos de Setores Circulares (ASC) foram propostos para aumentar o alcance a propriedade triangular. Os ASC aumentam a chance de representar conjuntos similares a A com a mesma assinatura de A desde que pertençam ao mesmo setor circular de A ou a setores circulares que apresentam o mesmo mínimo de A na região cinza da figura 5.2.



(a) Propriedade triangular de π_n em um conjunto S_i

(b) Propriedade triangular no exemplo (4.17)

Figura 5.1: Interpretação geométrica da Propriedade triangular

A ideia por trás dos métodos propostos neste capítulo é representar um conjunto S_i, sumarizando o intervalo de valores de $\pi_n(S_i)$ em um número que é geometricamente interpretado a partir de propriedades inerentes a Arcos de Setores Circulares (ASC). De modo a esclarecer esta solução, a seção 5.1 apresenta a propriedade triangular, para codificar o intervalo do *lattice* em um único valor numérico, enquanto que a seção 5.2 apresenta como os Arcos de Setores Circulares melhoram a eficácia da propriedade triangular; a seção 5.3 formaliza os estimadores propostos e discute a similaridade de Jaccard de cada estimador; e, por fim, a seção 5.4 apresenta os algoritmos propostos, para cada estimador de 5.3, e avalia as melhorias apresentadas por cada algoritmo em comparação com os dos métodos *Minwise Hash* e *Minmaxwise Hash*.

5.1 Propriedade triangular: Codificando o intervalo de um *lattice*

A propriedade triangular(P_{tri}) representa o intervalo $[min(\pi_n(S_i)), max(\pi_n(S_i))]$ como o valor do cateto oposto do triângulo retângulo, ilustrado na figura 5.1a e

formalizado na equação (5.1). Em outras palavras, para cada permutação π_n, a P_{tri} codifica o intervalo do *lattice* $(\pi_n(S_i), min, max)$ em um valor numérico $K_{S_i,n}$. A figura 5.1b ilustra como os conjuntos do exemplo (4.17) seriam representados pela equação (5.1), onde todos os conjuntos que apresentam o intervalo $[1, 5]$ estarão no mesmo triângulo retângulo de A e, portanto, apresentarão o mesmo valor de $K_{S_i,3}$ assim como ocorreu com B. Além disso, vale ressaltar que conjuntos com intervalos diferentes, mas com comprimentos iguais ao de A, não apresentarão o mesmo valor para $K_{S_i,3}$ como ocorre com E.

$$K_{S_i,n} = \sqrt{max_{S_i,n}^2 - min_{S_i,n}^2} \tag{5.1}$$

Para um conjunto S_i, que pertence a universo de conjuntos finitos $S \subset \mathbb{Z}^+$, a P_{tri} permite definir axiomas e demonstrar os lemas a seguir:

Axioma 1. $1 \leq min_{S_i,n} \leq max_{S_i,n}$

Axioma 2. $\exists \pi_n^{max} \in \mathbb{Z}^+ | max_{S_i,n} \leq \pi_n^{max}$ *(Equação* (4.8)*)*

Lema 1. $0 \leq K_{S_i,n} < max_{S_i,n}$

Demonstração. Substituindo o axioma 1 na equação (5.1) teremos que o menor valor possível de $K_{S_i,n}$ é $\sqrt{max_{S_i,n}^2 - max_{S_i,n}^2}$ enquanto o maior valor é $\sqrt{max_{S_i,n}^2 - 1}$. Logo:

$$\begin{aligned}
&\sqrt{max_{S_i,n}^2 - max_{S_i,n}^2} \leq K_{S_i,n} \leq \sqrt{max_{S_i,n}^2 - 1} \\
&\iff 0 \leq K_{S_i,n} \leq \sqrt{max_{S_i,n}^2 - 1} \\
&\iff 0 \leq K_{S_i,n} \leq \sqrt{max_{S_i,n}^2 - 1} < \sqrt{max_{S_i,n}^2} \\
&\overset{max_{S_i,n} \geq 1}{\iff} 0 \leq K_{S_i,n} < max_{S_i,n}
\end{aligned}$$

Vale destacar que $max_{S_i,n} \geq 1$ garante que $\sqrt{max_{S_i,n}^2 - 1} \geq 0$ é possível e, portanto, $0 \leq K_{S_i,n} < max_{S_i,n} \implies 0 \leq K_{S_i,n} \leq \sqrt{max_{S_i,n}^2 - 1} < \sqrt{max_{S_i,n}^2}$. □

Lema 2. $1 \leq min_{S_i,n} \leq max_{S_i,n} \leq \pi_n^{max}$

Demonstração. Basta substituir o axioma 2 no axioma 1. □

Lema 3. $Pr[max_{S_i,n} = x] = \frac{1}{\pi_n^{max}} \mid 1 \leq x \leq \pi_n^{max}$

Demonstração.

$$Pr[max_{S_i,n} = x] = \frac{\text{número de vezes que } max_{S_i,n} = x}{\text{quantos valores } max_{S_i,n} \text{ consegue assumir}}$$

Suponha, sem perda de generalidade, que x tem um valor fixo ε. Logo:

$$Pr[max_{S_i,n} = x] = Pr[max_{S_i,n} = \varepsilon] = \frac{\text{número de vezes que } max_{S_i,n} = \varepsilon}{\text{quantos valores } max_{S_i,n} \text{ consegue assumir}}$$
$$Pr[max_{S_i,n} = x] = Pr[max_{S_i,n} = \varepsilon] = \frac{1}{\text{quantos valores } max_{S_i,n} \text{ consegue assumir}}$$

Pelo lema 2 temos que $1 \leq max_{S_i,n} \leq \pi_n^{max}$ e, portanto, $max_{S_i,n}$ consegue assumir $\pi_n^{max} - 1 + 1 = \pi_n^{max}$ valores. Logo:

$$Pr[max_{S_i,n} = x] = \frac{1}{\text{quantos valores } max_{S_i,n} \text{ consegue assumir}}$$
$$Pr[max_{S_i,n} = x] = \frac{1}{\pi_n^{max}}$$

□

Lema 4. *Para todo valor $K_{S_i,n}$ existe um círculo de raio $max_{S_i,n}$ e centro na origem a que S_i pertence.*

Demonstração. Da equação (5.1):

$$K_{S_i,n} = \sqrt{max_{S_i,n}^2 - min_{S_i,n}^2} \overset{K_{S_i,n} \geq 0}{\iff} (K_{S_i,n})^2 = (\sqrt{max_{S_i,n}^2 - min_{S_i,n}^2})^2$$
$$\iff (K_{S_i,n})^2 = max_{S_i,n}^2 - min_{S_i,n}^2 \iff (K_{S_i,n})^2 + (min_{S_i,n})^2 = (max_{S_i,n})^2$$
$$\iff (K_{S_i,n} - 0)^2 + (min_{S_i,n} - 0)^2 = (max_{S_i,n})^2$$
$$\iff \text{existe um círculo de raio } max_{S_i,n} \text{ que passa por } S_i \text{ em um sistema cartesiano de eixos } K_{S_i,n} \times min_{S_i,n}$$

□

5.2 Aumentando o alcance através de Arcos de Setores Circulares

A propriedade triangular codifica a permutação de dois conjuntos (S_1 e S_2), que têm os mesmos valores de extremidade, em um único valor $K_{S_i,n}$; isto é, se $min_{S_1,n} = min_{S_2,n}$ e $max_{S_1,n} = max_{S_2,n}$ então $K_{S_1,n} = K_{S_2,n}$. Contudo, P_{tri} tem uma desvantagem: conjuntos similares que apresentam apenas um valor igual de extremidade serão codificados em valores diferentes; por exemplo, os conjuntos A e C ou os conjuntos A e D da figura 5.1b. Para minimizar essa desvantagem o presente trabalho propõe dois métodos, baseados em Arcos de Setores Circulares (ASC), que codificam um intervalo de valores de $K_{S_i,n}$ em um número inteiro, conforme definido nas equações (5.7a) e (5.7b). Esta seção apresenta as definições e as demonstrações

apenas para a equação (5.7a) visto que o processo inteiro é análogo para a equação (5.7b).

$$K_{S_i,n}^{inf} = \lfloor K_{S_i,n} \rfloor \tag{5.7a}$$
$$K_{S_i,n}^{sup} = \lceil K_{S_i,n} \rceil \tag{5.7b}$$

Suponha que existe ε_i tal que $0 \leq \varepsilon_i < 1$ logo, podemos reescrever a equação (5.7a) como a equação (5.8a) tal que $K_{S_i,n}^{inf}$ representa um intervalo de valores enquanto que $K_{S_i,n}$ representa apenas um valor. Por conseguinte, um conjunto S_1 apresentará um mesmo valor de $K_{S_1,n}^{inf}$ com conjuntos S_i que respeitam a relação da equação (5.9a).

$$\begin{aligned} &K_{S_i,n}^{inf} = \lfloor K_{S_i,n} \rfloor \overset{0 \leq \varepsilon_i < 1}{\iff} K_{S_i,n} = K_{S_i,n}^{inf} + \varepsilon_i \\ &\iff K_{S_i,n}^{inf} = K_{S_i,n} - \varepsilon_i \mid 0 \leq \varepsilon_i < 1 \end{aligned} \tag{5.8a}$$

$$\begin{aligned} &K_{S_1,n}^{inf} = K_{S_1,n} - \varepsilon_1 = K_{S_i,n}^{inf} = K_{S_i,n} - \varepsilon_i \mid 0 \leq \varepsilon_1, \varepsilon_i < 1 \\ &\iff K_{S_i,n} = K_{S_1,n} - \varepsilon_1 + \varepsilon_i \mid 0 \leq \varepsilon_1, \varepsilon_i < 1 \end{aligned} \tag{5.9a}$$

Teorema 2. *Para todo conjunto S_1 com lattice $(\pi_n(S_1), min, max)$ existe pelo menos um setor circular de raio $max_{S_i,n}$ tal que $K_{S_i,n}^{inf} = K_{S_1,n}^{inf}$.*

Demonstração. Pelo lema 4, cada $K_{S_i,n}$ pertence a um circulo de raio $max_{S_i,n}$ logo:

$$\exists\theta | \text{seno}(\theta) = \frac{K_{S_i,n}}{max_{S_i,n}} \overset{eq.(5.9a)}{=} \frac{K_{S_1,n} - \varepsilon_1 + \varepsilon_i}{max_{S_i,n}}$$

Da equação (5.9a) temos que $K_{S_i,n}$ varia de acordo com $0 \leq \varepsilon_i < 1$ e, portanto:

$$\frac{K_{S_1,n} - \varepsilon_1}{max_{S_i,n}} \leq \text{seno}(\theta) < \frac{K_{S_1,n} - \varepsilon_1 + 1}{max_{S_i,n}} | 1 \leq max_{S_i,n} \leq \pi_n^{max}$$

Logo, para cada valor de $max_{S_i,n}$ tal que $K_{S_i,n}^{inf} = K_{S_1,n}^{inf}$, teremos um setor circular de angulo no intervalo:

$$\text{arcsen}\left(\frac{K_{S_1,n} - \varepsilon_1}{max_{S_i,n}}\right) \leq \theta < \text{arcsen}\left(\frac{K_{S_1,n} - \varepsilon_1 + 1}{max_{S_i,n}}\right)$$

Contudo, é preciso garantir que existe pelo menos um valor de S_i que apresente

sempre um setor circular, no intervalo proposto, em relação a S_1. O ponto em questão é o próprio S_1. □

A Figura 5.2 apresenta a interpretação geométrica das equações (5.7a), (5.8a) e (5.9a), para o conjunto A do exemplo 4.17. O raio de cada círculo representa os valores de $max_{S_i,n}$ que um conjunto S_i pode ter, i.e. todos os conjuntos com o mesmo valor de $max_{S_i,n}$ pertencem ao mesmo círculo. No exemplo, apesar de D estar presente no mesmo círculo de A, a P_{tri} não consegue representar D com o mesmo valor de A. A região em cinza ilustra o resultado de a equação (5.9a) usando A como ponto de referência ($S_1 = A$) onde se encontram todos os conjuntos tais que $K^{inf}_{S_i,3} = K^{inf}_{A,3}$ de tal forma que $K^{inf}_{D,3} = K^{inf}_{A,3}$.

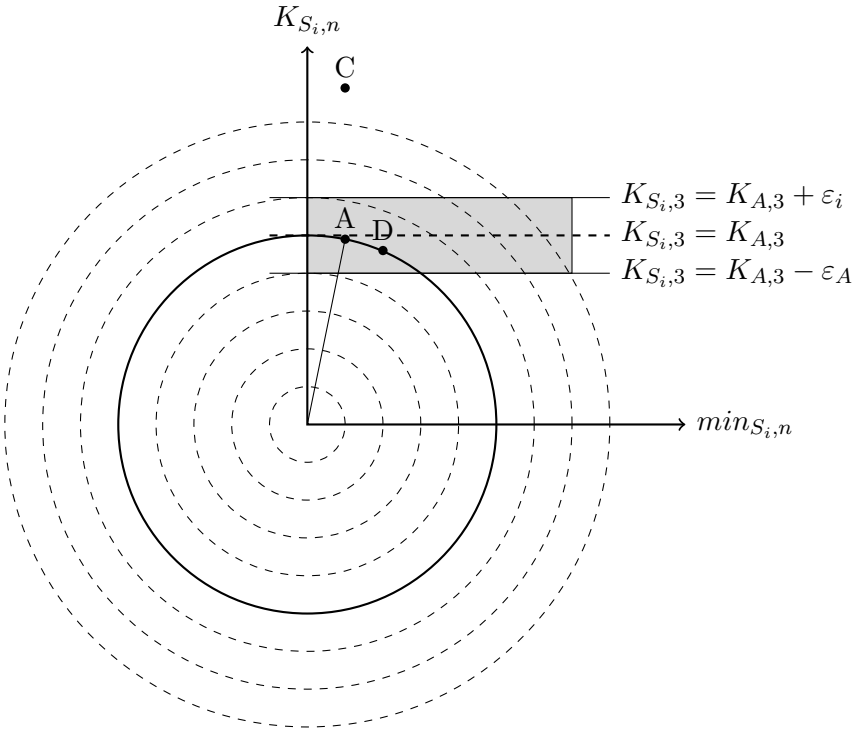


Figura 5.2: Equação (5.7a): Ilustrando os Arcos de Setores Circulares para $\lfloor K_{S_i,3} \rfloor$ do exemplo (4.17)

5.3 Estimando a similaridade de Jaccard

O teorema 1 discutiu como a probabilidade de se gerar as mesmas *fingerprints* para dois conjuntos está diretamente relacionada com a similaridade de Jaccard entre eles. Logo, esta seção define e discute como os métodos propostos estimam Jaccard. Isto é, tendo em vista que o propósito é reduzir o número de permutações, selecionando mais valores por permutação, esta seção avalia os estimadores W^{inf}, que seleciona $\min(\pi_n(S_i))$, $\max(\pi_n(S_i))$ e $K^{inf}_{S_i,n}$, e W que seleciona o $\min(\pi_n(S_i))$, $\max(\pi_n(S_i))$, $K^{inf}_{S_i,n}$ e $K^{sup}_{S_i,n}$.

As equações (5.14) e (5.15) definem duas variáveis aleatórias $W^{inf}_{\pi_n}$ e $W^{sup}_{\pi_n}$ que estimam o comportamento dos métodos propostos em cada permutação π_n. Isto é, indicam quando o método gera o mesmo valor para dois conjuntos A e B em π_n. Enquanto que as equações (5.16) e (5.17) estimam, respectivamente, o comportamento ao se combinar $\min(\pi_n(S_i))$, $\max(\pi_n(S_i))$ e $K^{inf}_{S_i,n}$ para $n = \{1, 2, \ldots, \frac{K}{3}\}$ permutações e o comportamento de combinar $\min(\pi_n(S_i))$, $\max(\pi_n(S_i))$, $K^{inf}_{S_i,n}$ e $K^{sup}_{S_i,n}$ para $n = \{1, 2, \ldots, \frac{K}{4}\}$ permutações. Os lemas 11, 12, 13 e 14 servem como base para realizarmos a análise desta seção. Isto é, para realizar a análise é necessário relacionar W^{inf} e W com a similaridade de Jaccard entre dois conjuntos A e B. Para tanto, foi necessário provar os lemas 5 a 10 que utilizam as equações (5.18) e (5.19) definidas e demonstradas em (JI *et al.*, 2013).

$$K_{\pi_n} = \begin{cases} 1, & \text{if } K_{A,n} = K_{B,n} \\ 0, & \text{caso contrário} \end{cases} \tag{5.13}$$

$$W^{inf}_{\pi_n} = \begin{cases} 1, & \text{if } K^{inf}_{A,n} = K^{inf}_{B,n} \\ 0, & \text{caso contrário} \end{cases} \tag{5.14}$$

$$W^{sup}_{\pi_n} = \begin{cases} 1, & \text{if } K^{sup}_{A,n} = K^{sup}_{B,n} \\ 0, & \text{caso contrário} \end{cases} \tag{5.15}$$

$$W^{inf} = \frac{1}{K} \sum_{i=1}^{K/3} (X_{\pi_i} + Z_{\pi_i} + W^{inf}_{\pi_i}) \tag{5.16}$$

$$W = \frac{1}{K} \sum_{i=1}^{K/4} (X_{\pi_i} + Z_{\pi_i} + W^{inf}_{\pi_i} + W^{sup}_{\pi_i}) \tag{5.17}$$

$$\mathbb{E}[X_{\pi_n}] = \mathbb{E}[Z_{\pi_n}] = Pr[X_{\pi_n} = 1] = Pr[Z_{\pi_n} = 1] = J(A, B) \tag{5.18}$$

$$Pr[Z_{\pi_n} = 1 | X_{\pi_n} = 1] = Pr[X_{\pi_n} = 1 | Z_{\pi_n} = 1] = \frac{|A \cap B| - 1}{|A \cup B| - 1} \tag{5.19}$$

Lema 5. $Pr[W^{inf}_{\pi_n} = 1 | X_{\pi_n} = 1, Z_{\pi_n} = 1] = Pr[W^{sup}_{\pi_n} = 1 | Z_{\pi_n} = 1, X_{\pi_n} = 1] = 1$

Demonstração. esta demonstração equivale a estimar o número de conjuntos que se encontram na região definida pela equação (5.9a) com mesmo valor de mínimo e máximo. Por exemplo, a região em cinza da figura 5.2 em que $min_{B,n} = min_{A,n}$ e $max_{B,n} = max_{A,n}$. Portanto, $Pr[W^{inf}_{\pi_n} = 1 | X_{\pi_n} = 1, Z_{\pi_n} = 1]$, para dois conjuntos

A e B, é calculada como:

$$\frac{\Delta_{k_{B,n}^{inf}}}{\Delta_{all}} = \frac{\text{número de vezes em que } k_{B,n}^{inf} = k_{A,n}^{inf} \text{ se } min_{B,n} = min_{A,n} \text{ e } max_{B,n} = max_{A,n}}{\text{número de vezes em que } min_{B,n} = min_{A,n} \text{ e } max_{B,n} = max_{A,n}}$$
$$\frac{\Delta_{k_{B,n}^{inf}}}{\Delta_{all}} = \frac{\text{número vezes em que } k_{B,n}^{inf} = k_{A,n}^{inf} \text{ se } X_{\pi_n} = 1 \text{ e } Z_{\pi_n} = 1}{\text{número de vezes em que } X_{\pi_n} = 1 \text{ e } Z_{\pi_n} = 1}$$

A partir da equação (5.1) temos que :

$$k_{B,n} = \sqrt{max_{B,n}^2 - min_{B,n}^2} \overset{X_{\pi_n}=1}{\Longrightarrow} k_{B,n} = \sqrt{max_{B,n}^2 - min_{A,n}^2}$$
$$\overset{Z_{\pi_n}=1}{\Longrightarrow} k_{B,n} = \sqrt{max_{A,n}^2 - min_{A,n}^2} \Longrightarrow k_{B,n} = k_{A,n}$$

Logo :

$$k_{B,n} = k_{A,n} \Longrightarrow \begin{cases} \Delta_{all} = 1 \\ \lfloor k_{B,n} \rfloor = \lfloor k_{A,n} \rfloor \overset{eq.(5.7a)}{\Longrightarrow} k_{B,n}^{inf} = k_{A,n}^{inf} \Longrightarrow \Delta_{k_{B,n}^{inf}} = 1 \\ \lceil k_{B,n} \rceil = \lceil k_{A,n} \rceil \overset{eq.(5.7b)}{\Longrightarrow} k_{B,n}^{sup} = k_{A,n}^{sup} \Longrightarrow \Delta_{k_{B,n}^{sup}} = 1 \end{cases}$$

e, portanto:

$$\frac{\Delta_{k_{B,n}^{inf}}}{\Delta_{all}} = \frac{\Delta_{k_{B,n}^{sup}}}{\Delta_{all}} = 1 \Longrightarrow \begin{cases} Pr[W_{\pi_n}^{inf} = 1 | X_{\pi_n} = 1, Z_{\pi_n} = 1] = 1 \\ Pr[W_{\pi_n}^{sup} = 1 | X_{\pi_n} = 1, Z_{\pi_n} = 1] = 1 \end{cases}$$

□

Lema 6. $Pr[W_{\pi_n}^{inf} = 1 | K_{\pi_n} = 1] = Pr[W_{\pi_n}^{sup} = 1 | K_{\pi_n} = 1] = 1$

Demonstração. A partir da equação (5.13) temos que :

$$K_{\pi_n} = 1 \Longrightarrow k_{B,n} = k_{A,n} \Longrightarrow \begin{cases} \lfloor k_{B,n} \rfloor = \lfloor k_{A,n} \rfloor \overset{eq.(5.7a)}{\Longrightarrow} k_{B,n}^{inf} = k_{A,n}^{inf} \Longrightarrow W_{k_{B,n}^{inf}} = 1 \\ \lceil k_{B,n} \rceil = \lceil k_{A,n} \rceil \overset{eq.(5.7b)}{\Longrightarrow} k_{B,n}^{sup} = k_{A,n}^{sup} \Longrightarrow W_{k_{B,n}^{sup}} = 1 \end{cases}$$

□

Lema 7. $Pr[K_{\pi_n} = 1 | Z_{\pi_n} = 0, x_{\pi_n} = 1] = Pr[K_{\pi_n} = 1 | Z_{\pi_n} = 1, x_{\pi_n} = 0] = 0$

Demonstração.

$$X_{\pi_n} = 1 \overset{eq.(5.1)}{\Longrightarrow} \begin{cases} k_{B,n} = \sqrt{max_{B,n}^2 - min_{B,n}^2} \Longrightarrow k_{B,n}^2 = max_{B,n}^2 - min_{B,n}^2 \\ k_{A,n} = \sqrt{max_{A,n}^2 - min_{B,n}^2} \Longrightarrow k_{A,n}^2 = max_{A,n}^2 - min_{B,n}^2 \end{cases}$$

$$\implies k_{B,n}^2 - k_{A,n}^2 = max_{B,n}^2 - max_{A,n}^2 \overset{Z_{\pi_n}=0}{\implies} k_{B,n}^2 - k_{A,n}^2 \neq 0 \implies k_{B,n}^2 \neq k_{A,n}^2$$
$$\overset{k_{B,n},k_{A,n}\geq 0}{\implies} k_{B,n} \neq k_{A,n} \implies K_{\pi_n} = 0$$

A demonstração para $Pr[K_{\pi_n} = 1|Z_{\pi_n} = 1, x_{\pi_n} = 0] = 0$ é equivalente a de $Pr[K_{\pi_n} = 1|Z_{\pi_n} = 0, x_{\pi_n} = 1] = 0$ □

Lema 8. $Pr[K_{\pi_n} = 1] \geq J(A, B) \times \left(\frac{|A\cap B|-1}{|A\cup B|-1}\right)$

Demonstração. a partir do Teorema da Probabilidade Total (TPT) temos que:

$$Pr[K_{\pi_n} = 1] \overset{TPT}{=} Pr[K_{\pi_n} = 1|X_{\pi_n} = 1] \times Pr[X_{\pi_n} = 1] + Pr[K_{\pi_n} = 1|X_{\pi_n} = 0] \times Pr[X_{\pi_n} = 0]$$
$$Pr[K_{\pi_n} = 1] \overset{TPT}{=} Pr[X_{\pi_n} = 1] \times \Big(Pr[K_{\pi_n} = 1|Z_{\pi_n} = 1, X_{\pi_n} = 1] \times Pr[Z_{\pi_n} = 1|X_{\pi_n} = 1]$$
$$+ Pr[K_{\pi_n} = 1|Z_{\pi_n} = 0, X_{\pi_n} = 1] \times Pr[Z_{\pi_n} = 0|X_{\pi_n} = 1] \Big)$$
$$+ Pr[X_{\pi_n} = 0] \times \Big(Pr[K_{\pi_n} = 1|Z_{\pi_n} = 1, X_{\pi_n} = 0] \times Pr[Z_{\pi_n} = 1|X_{\pi_n} = 0]$$
$$+ Pr[K_{\pi_n} = 1|Z_{\pi_n} = 0, X_{\pi_n} = 0] \times Pr[Z_{\pi_n} = 0|X_{\pi_n} = 0] \Big)$$

Substituindo os trechos em azul pelo lema 7:

$$Pr[K_{\pi_n} = 1] = Pr[X_{\pi_n} = 1] \times \Big(Pr[K_{\pi_n} = 1|Z_{\pi_n} = 1, X_{\pi_n} = 1] \times Pr[Z_{\pi_n} = 1|X_{\pi_n} = 1] \Big)$$
$$+ Pr[X_{\pi_n} = 0] \times \Big(Pr[K_{\pi_n} = 1|Z_{\pi_n} = 0, X_{\pi_n} = 0] \times Pr[Z_{\pi_n} = 0|X_{\pi_n} = 0] \Big)$$
$$Pr[K_{\pi_n} = 1] \overset{lema\ 5}{=} Pr[X_{\pi_n} = 1] \times \Big(1 \times Pr[Z_{\pi_n} = 1|X_{\pi_n} = 1] \Big)$$
$$+ Pr[X_{\pi_n} = 0] \times \Big(Pr[K_{\pi_n} = 1|Z_{\pi_n} = 0, X_{\pi_n} = 0] \times Pr[Z_{\pi_n} = 0|X_{\pi_n} = 0] \Big)$$
$$\implies Pr[K_{\pi_n} = 1] \geq Pr[X_{\pi_n} = 1] \times \Big(Pr[Z_{\pi_n} = 1|X_{\pi_n} = 1] \Big)$$
$$\overset{eq.(5.18)}{\implies} Pr[K_{\pi_n} = 1] \geq J(A, B) \times \Big(Pr[Z_{\pi_n} = 1|X_{\pi_n} = 1] \Big)$$
$$\overset{eq.(5.19)}{\implies} Pr[K_{\pi_n} = 1] \geq J(A, B) \times \left(\frac{|A \cap B| - 1}{|A \cup B| - 1} \right)$$

□

Lema 9. $Pr[W_{\pi_n}^{inf} = 1] \geq Pr[K_{\pi_n} = 1] \geq J(A, B) \times \left(\frac{|A\cap B|-1}{|A\cup B|-1}\right)$

Demonstração. A partir do Teorema da Probabilidade Total (TPT) temos que:

$$\begin{aligned} Pr[W_{\pi_n}^{inf}=1] &\overset{TPT}{=} Pr[W_{\pi_n}^{inf}=1|K_{\pi_n}=1]\times Pr[K_{\pi_n}=1] \\ &\quad + Pr[W_{\pi_n}^{inf}=1|K_{\pi_n}=0]\times Pr[K_{\pi_n}=0] \\ Pr[W_{\pi_n}^{inf}=1] &\overset{lema\ 6}{=} 1\times Pr[K_{\pi_n}=1] \\ &\quad + Pr[W_{\pi_n}^{inf}=1|K_{\pi_n}=0]\times Pr[K_{\pi_n}=0] \\ \implies Pr[W_{\pi_n}^{inf}=1] &\geq 1\times Pr[K_{\pi_n}=1] \\ \overset{lema\ 8}{\implies} Pr[W_{\pi_n}^{inf}=1] &\geq J(A,B)\times\left(\frac{|A\cap B|-1}{|A\cup B|-1}\right) \end{aligned}$$

□

Lema 10. $Pr[W_{\pi_n}^{sup}=1] \geq Pr[K_{\pi_n}=1] \geq J(A,B)\times\left(\frac{|A\cap B|-1}{|A\cup B|-1}\right)$

Demonstração. A demonstração do lema 10 é equivalente a do lema 9. □

Lema 11. $\mathbb{E}[W_{\pi_n}^{inf}] = Pr[W_{\pi_n}^{inf}=1]$

Demonstração.

$$\begin{aligned} \mathbb{E}[W_{\pi_n}^{inf}] &= \frac{1}{N}\sum_{n=1}^{N} W_{\pi_n}^{inf} = \frac{1}{N}\sum_{n=1}^{N} 1\times Pr[W_{\pi_n}^{inf}=1] \\ \overset{\pi_n \text{é iid}}{\implies} \mathbb{E}[W_{\pi_n}^{inf}] &= \frac{1}{N}(N\times Pr[W_{\pi_n}^{inf}=1]) = Pr[W_{\pi_n}^{inf}=1] \end{aligned}$$

□

Lema 12. $\mathbb{E}[W_{\pi_n}^{sup}] = Pr[W_{\pi_n}^{sup}=1]$

Demonstração. A demonstração do lema 12 é equivalente a do lema 11. □

Lema 13. $\mathbb{E}[W^{inf}] \geq \frac{J(A,B)}{3}\times\left(2+\frac{|A\cap B|-1}{|A\cup B|-1}\right) > \frac{2\times J(A,B)}{3}$

Demonstração. A partir da equação (5.16) temos que:

$$\mathbb{E}[W^{inf}] = \frac{1}{K}\sum_{i=1}^{K/3}(\mathbb{E}[X_{\pi_i}] + \mathbb{E}[Z_{\pi_i}] + \mathbb{E}[W^{inf}_{\pi_i}]) \overset{eq.(5.18)}{=} \frac{1}{K}\sum_{i=1}^{K/3}(J(A,B) + J(A,B) + \mathbb{E}[W^{inf}_{\pi_i}])$$

$$\mathbb{E}[W^{inf}] \overset{lema\ 11}{=} \frac{1}{K}\sum_{i=1}^{K/3}(J(A,B) + J(A,B) + Pr[W^{inf}_{\pi_n} = 1])$$

$$\overset{lema\ 9}{\Longrightarrow} \mathbb{E}[W^{inf}] \geq \frac{1}{K}\sum_{i=1}^{K/3}\left(J(A,B) + J(A,B) + J(A,B) \times \left(\frac{|A \cap B| - 1}{|A \cup B| - 1}\right)\right)$$

$$\mathbb{E}[W^{inf}] \geq J(A,B) \times \left(2 + \left(\frac{|A \cap B| - 1}{|A \cup B| - 1}\right)\right) \times \frac{1}{K}\sum_{i=1}^{K/3}(1)$$

$$\mathbb{E}[W^{inf}] \geq J(A,B) \times \left(2 + \left(\frac{|A \cap B| - 1}{|A \cup B| - 1}\right)\right) \times \frac{1}{3}$$

$$\mathbb{E}[W^{inf}] \geq \frac{J(A,B)}{3} \times \left(2 + \left(\frac{|A \cap B| - 1}{|A \cup B| - 1}\right)\right) > \frac{2 \times J(A,B)}{3}$$

□

Lema 14. $\mathbb{E}[W] \geq \frac{J(A,B)}{2} \times \left(1 + \frac{|A \cap B| - 1}{|A \cup B| - 1}\right) > \frac{J(A,B)}{2}$

Demonstração. A partir da equação (5.17) temos que:

$$\mathbb{E}[W] = \frac{1}{K}\sum_{i=1}^{K/4}(\mathbb{E}[X_{\pi_i}] + \mathbb{E}[Z_{\pi_i}] + \mathbb{E}[W^{inf}_{\pi_i}] + \mathbb{E}[W^{sup}_{\pi_i}])$$

$$\mathbb{E}[W] \overset{eq.(5.18)}{=} \frac{1}{K}\sum_{i=1}^{K/4}(J(A,B) + J(A,B) + \mathbb{E}[W^{inf}_{\pi_i}] + \mathbb{E}[W^{sup}_{\pi_i}])$$

$$\mathbb{E}[W] \overset{lema\ 11}{=} \frac{1}{K}\sum_{i=1}^{K/4}(J(A,B) + J(A,B) + Pr[W^{inf}_{\pi_n} = 1] + \mathbb{E}[W^{sup}_{\pi_i}])$$

$$\overset{lema\ 9}{\Longrightarrow} \mathbb{E}[W^{inf}] \geq \frac{1}{K}\sum_{i=1}^{K/4}\left(J(A,B) + J(A,B) + J(A,B) \times \left(\frac{|A \cap B| - 1}{|A \cup B| - 1}\right) + \mathbb{E}[W^{sup}_{\pi_i}]\right)$$

$$\overset{lema\ 10}{\Longrightarrow} \mathbb{E}[W^{inf}] \geq \frac{1}{K}\sum_{i=1}^{K/4}\left(J(A,B) + J(A,B) + J(A,B) \times \left(\frac{|A \cap B| - 1}{|A \cup B| - 1}\right) \times 2\right)$$

$$\mathbb{E}[W] \geq J(A,B) \times \left(2 + 2 \times \left(\frac{|A \cap B| - 1}{|A \cup B| - 1}\right)\right) \times \frac{1}{K}\sum_{i=1}^{K/4}(1)$$

$$\mathbb{E}[W] \geq J(A,B) \times \left(1 + \left(\frac{|A \cap B| - 1}{|A \cup B| - 1}\right)\right) \times \frac{2}{4}$$

$$\mathbb{E}[W] \geq \frac{J(A,B)}{2} \times \left(1 + \left(\frac{|A \cap B| - 1}{|A \cup B| - 1}\right)\right) > \frac{J(A,B)}{2}$$

□

Os lemas 9 e 10 estimam a probabilidade de $W_{\pi_n}^{inf}$ e $W_{\pi_n}^{sup}$ afirmar, na permutação π_n, que A e B são iguais. Os dois estimadores apresentam uma probabilidade maior ou igual a de K_{π_n}, i.e., no pior caso, $W_{\pi_n}^{inf}$ e $W_{\pi_n}^{sup}$ se comportam de forma igual a K_{π_n} e, nos outros casos, os dois estimadores podem selecionar mais pares A e B que K_{π_n} não conseguiria. Contudo, como $W_{\pi_n}^{inf}$ e $W_{\pi_n}^{sup}$ se comportam ao se permutar mais de uma vez os conjuntos? além disso, como ele se comportam ao combiná-los com X_{π_n} e Z_{π_n}? Os lemas 11 e 12 respondem a primeira questão enquanto que os lemas 13 e 14 respondem a segunda questão. Além disso, com o intuindo de elucidar as perguntas acima a tabela 5.1 foi criada. Cada estimador é avaliado de acordo com valores de similaridade de Jaccard onde, por exemplo, para $J(A, B) = 0.6$ a probabilidade de afirmar que A e B são similares por W^{inf} é maior que 40 % enquanto que para W é maior que 30 %.

Os métodos propostos no presente trabalho têm como objetivo reduzir o custo computacional selecionando mais valores em cada permutação; Contudo, eles apresentam uma desvantagem: K_{π_n}, W^{inf} e W não são estimadores da similaridade de Jaccard sem viés. Em vez disso, o viés está relacionado com a própria similaridade de Jaccard, como exemplificado na tabela 5.1, pois $\mathbb{E}[W_{\pi_n}^{inf}]$, $\mathbb{E}[W^{inf}]$, e $\mathbb{E}[W]$ são esperanças menos consistentes, com valores menores de $J(A, B)$, e, portanto, as suas variâncias tendem a ser maiores.

Tabela 5.1: Comparando os estimadores: Limiar mínimos das esperanças.

$J(A,B)$	$\mathbb{E}[X]$	$\mathbb{E}[W^{inf}]$	$\mathbb{E}[W]$
0.10	0.10	> 0.07	> 0.05
0.20	0.20	> 0.13	> 0.10
0.30	0.30	> 0.20	> 0.15
0.40	0.40	> 0.27	> 0.20
0.50	0.50	> 0.33	> 0.25
0.60	0.60	> 0.40	> 0.30
0.70	0.70	> 0.47	> 0.35
0.80	0.80	> 0.53	> 0.40
0.85	0.85	> 0.57	> 0.43
0.90	0.90	> 0.60	> 0.45
0.92	0.92	> 0.61	> 0.46
0.94	0.94	> 0.63	> 0.47
0.96	0.96	> 0.64	> 0.48
0.98	0.98	> 0.65	> 0.49
1.00	1.00	> 0.67	> 0.50

5.4 Algoritmos e seu custo computacional

A presente seção analisa e compara o custo computacional dos métodos baseados na propriedade triangular com o método *MinMaxwise hash*. O algoritmo CSA_L, acrônimo de ***C**ircular **S**ector **A**rc **L**ower bounded*, gera o valor $k^{inf}_{S_i,n}$ para cada permutação π_n de um conjunto S_i. Já o algoritmo $MinMaxCSA_L$ (***Min**, **Max** and **C**ircular **S**ector **A**rc **L**ower bounded*) combina os valores de mínimo, de máximo e de $k^{inf}_{S_i,n}$ enquanto que o algoritmo $MinMaxCSA$(***Min**, **Max** and **C**ircular **S**ector **A**rc full bounded*) combina o mínimo, o máximo, $k^{inf}_{S_i,n}$ e $k^{sup}_{S_i,n}$.

O algoritmo 12 apresenta como CSA_L é computado. Em síntese, para uma permutação $\pi_n(S_i)$, de um conjunto $\{s_1, s_2, s_3, \ldots, s_t\}$, o algoritmo seleciona os valores máximo e mínimo, presentes em $\pi_n(S_i)$, e, em seguida, retorna $K^{inf}_{S_i,n}$ (5.7a). A única diferença entre os três algoritmos está na quantidade de valores que serão retornados, i. e. CSA_L retorna apenas $\lfloor triProp \rfloor(K^{inf}_{S_i,n})$ enquanto que $MinMaxCSA_L$ retorna $minId$, $maxId$ e $\lfloor triProp \rfloor(K^{inf}_{S_i,n})$ e $MinMaxCSA$ retorna $minId$, $maxId$, $\lfloor triProp \rfloor(K^{inf}_{S_i,n})$ e $\lceil triProp \rceil(K^{sup}_{S_i,n})$.

Algorithm 5: CSA_L

Input: $\pi_n(S_i) = \{s_1, s_2, s_3, \ldots, s_t\}$

Initialize: $minId = s_1, maxId = s_1$

begin

  for $i \leftarrow 1$ ***to*** t **do**

    if $minId > s_i$ **then**

      $minId \leftarrow s_i$

    end

    if $maxId < s_i$ **then**

      $maxId \leftarrow s_i$

    end

  end

  $triProp = \sqrt{maxId^2 - minId^2}$

end

Output: $\lfloor triProp \rfloor$

CSA_L nos permite avaliar o custo computacional do método baseado apenas em Arcos de Setores Circulares enquanto que os outros métodos permitem avaliar a combinação os métodos propostos com os valores de mínimo e máximo. Conforme JI *et al.* (2013) afirma, aplicar funções de permutação é mais custoso que aplicar operações de comparação e de atribuição e, portanto, quanto menor for o número de permutações mais rápido será o método. A tabela 5.2 apresenta o custo compu-

tacional para se representar um conjunto S_i, com P valores de *fingerprints*, entre os métodos propostos. $MinMaxCSA_L$ e $MinMaxCSA$ selecionam, respectivamente, 2 e 3 vezes mais valores que o *Minwise Hash* e, portanto, reduzem a quantidade de tempo gasta por *Minwise Hash* em, aproximadamente, 66 and 75% enquanto que *MinMaxwise Hash* reduz apenas 50%.

Tabela 5.2: Comparando o custo computacional para produzir valores de P hash.

Método	custo por permutação	$min_{S_i,n}$	$max_{S_i,n}$	$K^{inf}_{S_i,n}$	$K^{sup}_{S_i,n}$
Minwise Hash	$P \times \lvert S_i \rvert$	Sim	-	-	-
Minmaxwise Hash	$({}^{P}/_{2}) \times \lvert S_i \rvert$	Sim	Sim	-	-
CSA_L	$P \times \lvert S_i \rvert$	-	-	Sim	-
$MinMaxCSA_L$	$({}^{P}/_{3}) \times \lvert S_i \rvert$	Sim	Sim	Sim	-
MinMaxCSA	$({}^{P}/_{4}) \times \lvert S_i \rvert$	Sim	Sim	Sim	Sim

Capítulo 6

Avaliação Experimental

A little less conversation, a little more action please

— Elvis Presley interpretando (DAVIS e STRANGE, 2002)

Este capítulo descreve dois grupos de experimentos conduzidos para avaliar a efetividade e a eficiência dos métodos propostos. Os experimentos foram feitos com base na sequência de passos (Figura 6.1), proposta na seção 4.3, com ênfase no passo de "Execução da função de seleção" (iv)(Subseção 4.3.5). Isto é, os experimentos aqui descritos foram propostos para avaliar os métodos propostos como alternativas de estimadores para o passo (iv). Para tanto, a análise dos experimentos comparou os métodos propostos com as abordagens tradicionais, isto é índice invertido usando palavras, *Minwise Hashing* e *MinMaxwise Hashing*.

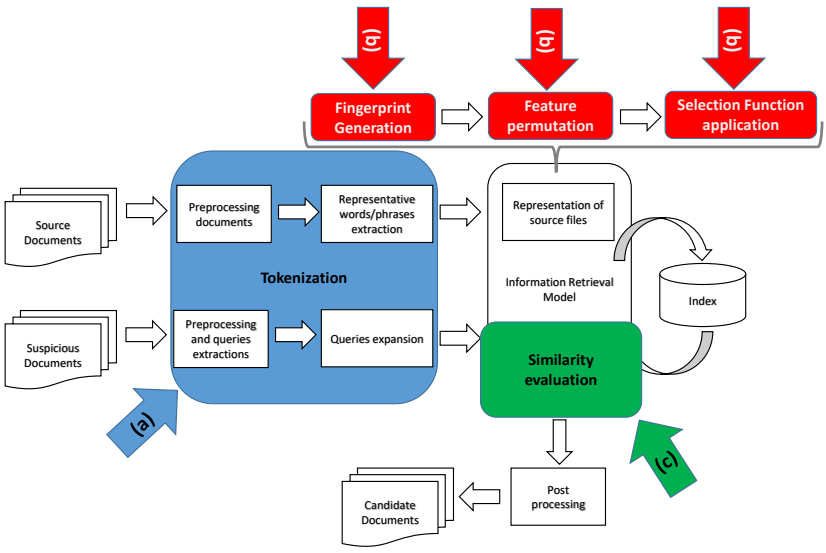


Figura 6.1: Localizando passos do experimento na Figura 4.3

Com o intuito de simplificar a notação a tabela 6.1 apresenta a nomeclatura que será utilizadas nos gráficos e nas tabelas dos resultados dos experimentos.

Tabela 6.1: nomeclatura dos métodos utilizada nos experimentos

Sigla	Nome do método
Min	*Minwise hashing* (i.e. mínimo)
$Minmax$	*Minmaxwise hashing* (i.e. mínimo e máximo)
CSA_L	$MinMaxCSA_L$ (i.e. mínimo, máximo e CSA_L)
CSA	$MinMaxCSA$ (i.e. mínimo, máximo e CSA)

O primeiro grupo de experimentos foi denominado de "Estimando a Similaridade de Jaccard par-a-par ($ESJP$)". $ESJP$ foi proposto de forma similar ao de JI *et al.* (2013) e avalia como os métodos propostos estimam a similaridade de Jaccard. O segundo grupo de experimentos ($RHPE$) está relacionado com o passo da Recuperação Heurística do Plágio Externo que avaliou se os métodos propostos são aplicáveis no problema em questão, isto é no plágio externo. Além disso, todos os experimentos foram conduzidos usando Python, em um computador com 16 GB de RAM e processador Intel i7 (3.4 Ghz).

O resto deste capítulo se organiza em 4 seções. A seção 6.1 apresenta a descrição das coleções de dados utilizadas nos experimentos, como a tabela 6.2 que lista o número de documentos, a contagem de incidências de palavras e o número de palavras distintas, conhecido como vocabulário, que cada coleção apresenta. Na seção 6.2 as métricas utilizadas nos experimentos são apresentadas e a seção 6.3, descreve os experimentos $ESJP$ onde os estimadores baseados em arcos de setores circulares foram comparados em termos de tempo e do erro médio quadrático, do Jaccard, entre pares de documentos. Já a seção 6.4, apresenta os experimentos $RHPE$ onde os métodos baseados em arcos de setores circulares também são avaliados em termos de tamanho do índice de documentos (i.e. o número de assinaturas necessárias para representar a coleção de documentos D) e do tempo de recuperação do documento (i.e. a quantidade total de tempo gasta para representar cada documento suspeito e resgatar suas fontes de plágio).

6.1 Coleções de dados

Nos experimentos foram utilizadas as coleções de dados *PAN plagiarism corpus 2011 (PAN-PC-11)* e *Plagiarised Short Answers (PSA)*, criadas especificamente para a tarefa de identificação de plágio, além da coleção METER que foi criada para o estudo de reúso de texto. A coleção PAN-PC-11 (POTTHAST *et al.*, 2010b) foi criada para avaliar algoritmos automáticos de identificação de plágio. Ela é composta de 26.939 documentos contendo 61.064 casos de plágio. Além disso, 50% desses

documentos são documentos fonte enquanto a outra metade é composta de casos suspeitos de plágio tais que 25% dos documentos, da coleção, apresentam plágio enquanto os outros 25% não apresentam. Os conjunto de documentos com plágio incluem casos de ofuscação (82%) que envolvem paráfrase ou plágio de tradução realizados tanto de forma manual quanto de forma automática.

A coleção PSA (CLOUGH e STEVENSON, 2011) é uma coleção composta de respostas de questões curtas, na área de ciência da computação, indagadas as 19 participantes. Esta coleção simula plágio a partir de respostas baseadas em revisões ou cópias aproximadas de documentos fonte, extraídos da Wikipedia[1], assim como respostas feitas sem aplicação de técnicas de plágio. A coleção é composta de 5 artigos da wikipedia, como documentos fonte, e 95 respostas escritas pelos participantes.

A coleção METER (CLOUGH *et al.*, 2002) divide notícias, escritas em artigos, em dois grupos de documentos. O primeiro grupo é composto de 773 artigos de noticias originais escritas pela *Press Association (PA)*. O segundo grupo de documentos é composto de 944 artigos onde 300 foram classificados como totalmente derivados, 438 como parcialmente derivados e 206 como não derivados dos 773 artigos originais.

Tabela 6.2: Estatísticas das coleções de dados utilizadas nos experimentos

Coleção de dados	Número de documentos	Tamanho do vocabulário	Total de palavras
Meter	1.717	17.431	507.894
Plagiarised Short Answers	100	2.254	20.679
PAN-PC-11	26.939	1.207.741	686.668.842

6.2 Métricas de avaliação

A presente seção apresenta as métricas que foram utilizadas nos experimentos, elas são: *Root Mean Squared Error (RMSE)*, *Mean Absolute Error (MAE)*, *Mean Error (ME)*, *Recall* e CRT. *Root Mean Squared Error (RMSE)* e *Mean Absolute Error (MAE)* foram utilizados nos experimentos do primeiro grupo isto é nos experimentos $ESJP$ enquanto que *Mean Error (ME)*, *Recall* e CRT foram utilizados nos experimentos $RHPE$.

Suponha que, y_j é o valor da similaridade de Jaccard entre dois documentos d_a e d_b e que $\hat{y}_j$ é o valor da similaridade de Jaccard calculado para o conjunto de assinaturas gerados para d_a e d_b. Logo, a equação 6.1 apresenta o *Mean Absolute Error (MAE)* que é a média da diferença absoluta entre o Jaccard observado e o Jaccard gerado pelos conjuntos de assinaturas onde todas as diferenças de valores

[1] wikipedia.org

individuais (y_j e $\hat{y}_j$) têm o mesmo peso na média final. Já a equação 6.2 apresenta *Root Mean Squared Error (RMSE)* que é a raiz quadrada da média do quadrado da diferença entre entre o Jaccard observado e o Jaccard gerado. A regra de pontuação quadrática do $RMSE$ dá pesos maiores a erros maiores e, portanto, o $RMSE$ é mais influenciado pela variância de valores de erros que o MAE. Logo, o presente trabalho utilizou do $RMSE$ para avaliar o comportamento geral dos métodos, incluindo também os resultados fora da média, enquanto que o comportamento médio foi avaliado a partir do MAE.

$$MAE = \frac{1}{n}\sum_{j=1}^{n}|y_j - \hat{y}_j| \tag{6.1}$$

$$RMSE = \sqrt{\frac{1}{n}\sum_{j=1}^{n}(y_j - \hat{y}_j)^2} \tag{6.2}$$

Suponha, para os experimentos $RHPE$, que $d_q \in D_{susp}$ é um documento suspeito e que a meta é encontra um conjunto de documentos F_q que foram copiados por d_q e estão disponíveis em D (i.e. $F_q \subset D$). Para tanto, um motor de busca retornará um conjunto de documentos $R \subset D$ e, portanto, o *recall* (revocação) medirá a quantidade de documentos relevantes (de F_q) que foram retornados em R para d_q assim como apresentado na equação 6.3.

$$Recall = \frac{1}{|D_{susp}|}\sum_{d_q \in D_{susp}}\frac{|F_q \cap R|}{|F_q|} \tag{6.3}$$

Os experimentos $RHPE$ comparam os métodos propostos com o *Minwise Hashing* onde o *Mean Error (ME)* apresenta a diferença do *recall* do método *Minwise Hashing* com os outros métodos. A equação 6.4 formaliza o cálculo do *Mean Error (ME)* onde $Rec_{min}(d_q)$ é o recall gerado pelo método *Minwise Hashing* para o documento d_q enquanto que $Rec_{metodo_i}(d_q)$ é o recall gerado por um método($metodo_i$). Contudo, o *Mean Error (ME)* não consegue avaliar a proporção que em o recall de uma abordagem aumentou ou diminuiu em relação ao recall do método *Minwise Hashing*. Para tanto, Diferença Percentual Relativa (DRP)(equação 6.7) foi utilizada[2] onde valores positivos de DRP siginificam que o método avaliado apresentou um recall menor que o do método *Minwise Hashing* enquanto que valores negativos significam que o método apresentou valores de recall maiores. Ademais, com o intuito de avaliar o impacto no tempo para algumas das tarefas, por exemplo extração de consulta, indexação ou busca, a Comparação Relativa de Tempo (CRT) em relação ao método *Minwise Hashing* também foi medida na equação 6.5. Por

[2]visto que o recall do método *Minwise Hashing* pode ser zero e, portanto, a taxa percentual de erro não pode ser avaliada

fim, com o intuito de avaliar a capacidade de indexar um grande volume de documentos a vazão, em documentos por hora, de cada método é medida assim como formalizado na equação 6.6.

$$ME = \frac{1}{n}\sum_{j=1}^{n}(Rec_{min}(d_q) - Rec_{metodo_i}(d_q)) \tag{6.4}$$

$$CRT(metodo_i) = \frac{\Delta_t \text{ médio para } \textit{Minwise Hashing} \text{ executar a tarefa}}{\Delta_t \text{ médio para o } metodo_i \text{ executar a tarefa}} \tag{6.5}$$

$$Vazão(metodo_i) = \frac{|D|}{\sum\limits_{d_i \in D} \Delta_t \text{ para o } metodo_i \text{ indexar } d_i} \tag{6.6}$$

$$DPR(metodo_i) = \frac{Rec_{min}(d_q) - Rec_{metodo_i}(d_q)}{Rec_{min}(d_q)} \tag{6.7}$$

6.3 Estimando a Similaridade de Jaccard Par-a-par ($ESJP$)

O objetivo do experimento $ESJP$ é avaliar como os métodos propostos estimam a similaridade de Jaccard. Para tanto, o erro do Jaccard, entre dois documentos, sem a utilização dos métodos e após a utilização dos métodos é avaliado. A avaliação é feita de duas formas: Medir erro médio (MAE) que não sofre influência dos resultados fora da média, i.e. erros muito grandes que não aparecem tanto, e medir o erro médio ($RMSE$) que é influenciado pela variância dos erros e, portanto, erros maiores tem maior peso no $RMSE$.

Os experimentos $ESJP$ foram conduzidos de forma similar aos de JI *et al.* (2013). Logo, a configuração dos experimentos envolveu três passos: (a) computar a similaridade de Jaccard para cada par de documentos de um conjunto D; (b) estimar a similaridade para cada estimador de método (e.g., Minwise hash or Minmaxwise hash, com k valores de hash); e (c) avaliar usando as métricas $RMSE$ e MAE, entre as similaridades de Jaccard geradas por (a) e (b), assim como o tempo (em segundos) gasto para gerar as assinaturas de cada amostra, i.e. documento, em D.

Os experimentos são mapeados no conjunto de passos LSH, proposto na seção 4.3, como: A tokenização (4.3.2), a geração de *fingerprint* (4.3.3), a permutação de características (4.3.4) e a execução da função de seleção (4.3.5) são executadas no passo (a) enquanto que a avaliação da similaridade (4.3.6) ocorre no passo (b). Durante a tokenização o vocabulário (T) da coleção é extraído a partir do modelo *bag-of-words*, removendo *stopwords*, para, em seguida, a matriz termo-documento

ser gerada. A geração de *fingerprint* é realizada ao se mapear cada termo da matriz no seu índice em T. A permutação de características efetua permutações aleatórias de T e atualiza os índices atribuídos aos termos na matriz. Por fim, a similaridade de Jaccard é utilizada para efetuar a similaridade entre pares de documentos representados como conjuntos de assinaturas gerados pelos passos anteriores.

No passo (b) foi utilizado o mesmo número de assinaturas para estimar a similaridade da cada método analisado. As assinaturas são selecionadas a partir de funções de seleção aplicadas a permutações aleatórias das representações numéricas das amostras da coleção. Para tanto, cada estimador recebeu um número diferente de permutações (N) de acordo com o número de assinaturas que é selecionado por ele. Por exemplo, o método *Minwise hashing* seleciona apenas um valor de assinatura (o mínimo) para cada permutação realizada e, portanto, é preciso $N = k$ permutações aleatórias para obter k valores de assinatura. Já o método *Minmaxwise hashing* seleciona os valores de mínimo e máximo, para cada permutação, fazendo com que ele necessite de metade das permutações ($N = k/2$), do método *Minwise hashing*, para gerar k assinaturas. De maneira similar, os métodos CSA_L e CSA usam, respectivamente, $N = k/3$ e $N = k/4$ permutações. A tabela 6.3 apresenta um exemplo de como o passo (b) seleciona 300 assinaturas para cada método.

Tabela 6.3: Exemplo de número de permutações necessário para selecionar 300 assinaturas. A coluna k apresenta um número de assinaturas selecionados e a coluna N apresenta o número de permutações necessárias para cada método selecionar k assinaturas.

Método	k	N
Minwise Hash	300	$k = 300$
Minmaxwise Hash	300	$^k/_2 = 150$
CSA_L	300	$^k/_3 = 100$
CSA	300	$^k/_4 = 75$

Os passos $(a), (b)$ e (c) foram realizados para todas as coleções de dados da tabela 6.2. Para a coleção PAN-PC-11 apenas uma amostra de 2.000 documentos foi utilizada, assim como feito em (JI *et al.*, 2013) para comparar o método $Minwise$ com o $Minmaxwise$. Já nas coleções METER e PSA todos os documentos foram utilizados. Os experimentos foram repetidos 100 vezes, para cada k, e, em seguida, foram medidos o RMSE, o MAE e o tempo de geração de assinaturas médio para cada estimador. As tabelas 6.4, 6.5 e 6.6 listam os resultados para as coleções PSA, METER, e PAN-PC-11, respectivamente.

Da tabela 6.4 é possível observar que o método *Minmaxwise hash* apresentou o melhor resultado de RMSE para todos os valores de K ($k \in \{100, 200, 400, 800\}$) avaliados no experimento. Além disso, o método CSA é o método mais rápido, em

todos os valores de k, enquanto que o *Minmaxwise hashing* foi o mais lento. De fato, o tempo de execução do método *Minmaxwise hashing* é, aproximadamente, 33% mais lento que o CSA enquanto que o método CSA_L é de 40% a 50% mais rápido que o *Minmaxwise hashing*.

Tabela 6.4: RMSE da similaridade de Jaccard, assim como o tempo de seleção de assinaturas (em segundos), para a coleção PSA.

	Minmax		CSA_L		CSA	
k	RMSE	Tempo	RMSE	Tempo	RMSE	Tempo
100	**0.0327**	131	0.0409	96	0.0499	**77**
200	**0.0235**	257	0.0298	176	0.0356	**138**
400	**0.0164**	548	0.0210	362	0.0258	**269**
800	**0.0114**	1174	0.0153	783	0.0188	**579**

Na avaliação da coleção METER (tabela 6.5), o RMSE do método *Minmaxwise* alcançou os melhores resultados entre todos os estimadores analisados, seguido pelo RMSE do método CSA_L que apresentou a mesma ordem de magnitude dos valores do *Minmaxwise hashing*. Como esperado, os métodos CSA_L e CSA são, respectivamente, 33% e 50% mais rápidos que o método *Minmaxwise hashing* em todos os valores de k. Além disso, na coleção PAN-PC-11 (tabela 6.6), o método CSA, apesar de ser menos preciso que o *Minmaxwise hashing* (RMSE aproximadamente 42% maior), apresenta a mesma ordem de magnitude do RMSE apresentado pelo *Minmaxwise hashing*. Portanto, novamente, para todos os valores de k os métodos CSA_L e CSA são aproximadamente 33 e 50% mais rápidos, que o *Minmaxwise hash*.

Tabela 6.5: RMSE da similaridade de Jaccard, assim como o tempo de seleção de assinaturas (em segundos), para a coleção METER.

	Minmax		CSA_L		CSA	
k	RMSE	Tempo	RMSE	Tempo	RMSE	Tempo
100	**0.0286**	3575	0.0348	2436	0.0427	**1881**
200	**0.0205**	7517	0.0270	4982	0.0335	**3755**
400	**0.0142**	15410	0.0205	10312	0.0271	**7666**
800	**0.0100**	32502	0.0169	21697	0.0232	**16129**

Tabela 6.6: RMSE da similaridade de Jaccard, assim como o tempo de seleção de assinaturas (em segundos), para a coleção *PAN plagiarism corpus 2011*.

	Minmax		CSA_L		CSA	
k	RMSE	Tempo	RMSE	Tempo	RMSE	Tempo
100	**0.0338**	109278	0.0429	71250	0.0518	**53343**
200	**0.0232**	221512	0.0301	145299	0.0369	**110106**
400	**0.0163**	444097	0.0214	293985	0.0264	**220230**
800	**0.0117**	892653	0.0163	594500	0.0205	**446404**

As figuras 6.2, 6.3 e 6.4 apresentam os resultados do MAE para cada método. Os retângulos vermelhos são os valores da média isto é o MAE para cada abordagem enquanto as barras horizontais são o maior e o menor valor de erro observado em cada método. Vale ressaltar que a precisão, de todos os métodos, aumenta conforme k aumenta visto que os intervalos de valores de MAE, representados pelos quartis, reduzem com o aumento de k. Além disso, todas as abordagens apresentam valores de médias próximos mas, conforme esperado, os intervalos de valores para os métodos CSA_L e CSA são maiores que os do método *Minmaxwise hashing*. Também vale ressaltar que, o intervalo de valores dos métodos *Minmaxwise hashing*, CSA_L e CSA convergem para o mesmo intervalo de valores, em todos as coleções de dados, ao se aumentar k. Por exemplo, a figura 6.4 mostra que: para $k = 100$, o maior valor de MAE para o método CSA ($\simeq 0,12$) é 50% maior que o do método *Minmaxwise hashing* ($\simeq 0,08$) enquanto que para $k = 800$ a diferença reduz para $\simeq 34\%$.

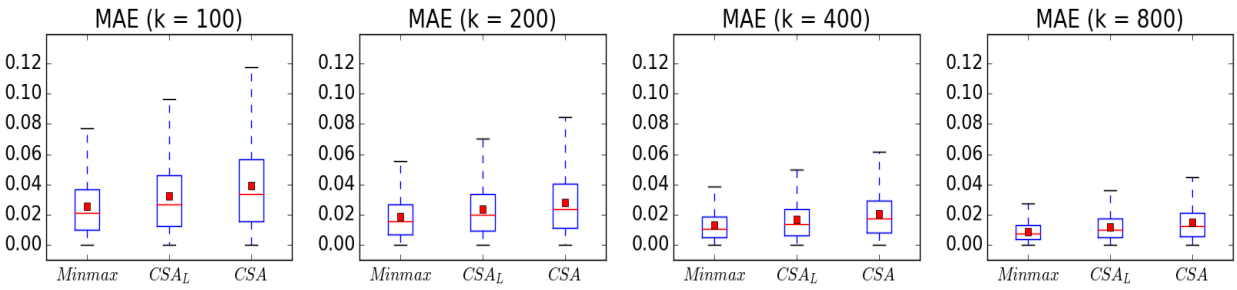


Figura 6.2: Pairwise Jaccard similarity MAE results for PSA corpus

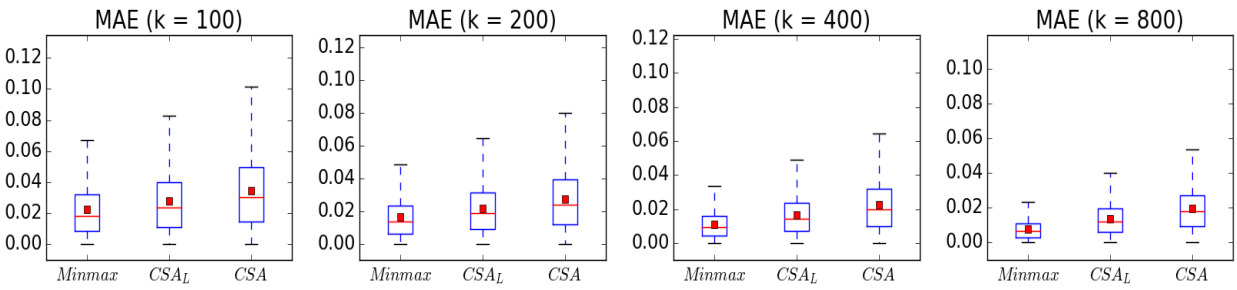


Figura 6.3: Pairwise Jaccard similarity MAE results for METER corpus

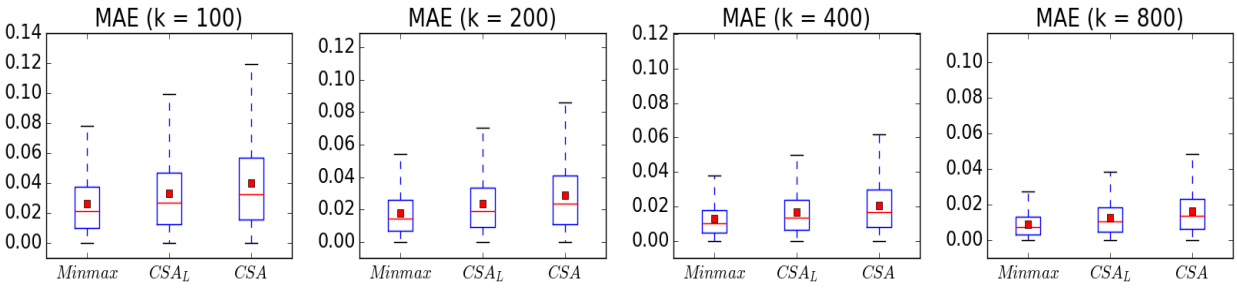


Figura 6.4: Pairwise Jaccard similarity MAE results for PAN plagiarism corpus

6.4 Recuperação Heurística do Plágio Externo ($RHPE$)

Os experimentos para avaliar os métodos propostos no passo da Recuperação Heurística da tarefa de Plágio Externo são descritos nesta seção. Este estágio tem como objetivo reduzir o número de documentos para comparação par-a-par do próximo passo e, portanto, deve retornar o menor conjunto possível de documentos candidatos a fonte de um documento suspeito de plágio.

Os experimentos $RHPE$ foram conduzidos em 20.958 documentos, escritos em inglês, da coleção PAN-PC11. O conjunto de documentos foi dividido em dois subconjuntos D e D_{susp}. O primeiro subconjunto é composto de documentos fonte, de tamanho $|D| = 15.966$, enquanto que D_{susp} ($|D_{susp}| = 4.992$) representa o subconjunto de consultas, i.e. documentos suspeitos de plágio, com pelo menos um documento fonte em D. As configurações dos experimentos envolvem: (a) pré-processamento do texto, (b) indexação das assinaturas de D e (c) busca e recuperação dos documentos, que apresentam pelo menos uma assinatura em comum com o documento suspeito $d_q \in D_{susp}$.

A tarefa $RHPE$ é mapeada nos passos da figura 6.1 da seguinte forma: A tokenização (4.3.2) é feita no passo (a)(azul); A geração de *fingerprint* (4.3.3), a permutação de características (4.3.4) e a execução da função de seleção (4.3.5) ocorrem em (b)(vermelho). Já a avaliação de similaridade (4.3.6) corresponde ao passo (c)(verde).

Os documentos foram tokenizados a partir do modelo *bag-of-words* onde foram removidas palavras que aparecem em apenas um documento, visto que elas nunca ocorrerão entre dois documentos e, portanto, são inúteis na busca a partir do índice invertido. A remoção resultou em um vocabulário (T) com 457.092 palavras. Em seguida, *fingerprints* são geradas extraindo o índice de cada termo em T e a permutação de características é gerada aleatoriamente. No passo de seleção foram empregados os métodos *Minwise*, *MinMaxwise*, CSA_L e CSA que computaram matrizes de assinaturas $\times$ documentos para cada estimador. Por fim, os documentos

$d_{src} \in D_{src}^{d_q}$ são documentos de D que apresentam pelo menos uma assinatura em comum com o documento suspeito $d_q \in D_{susp}$. Vale ressaltar que, para efeito de simplicidade, o cálculo da similaridade adotado foi o mesmo de JI *et al.* (2013) onde a similaridade é o tamanho do conjunto de assinaturas em comum entre d_{src} e q_q, i.e. $|d_{src} \cap d_q|$, e, portanto, só serão selecionados documentos tais que $J(d_{src}, d_q) > 0$.

Os experimentos foram executados para 48, 96, 192, 384 e 768 assinaturas (k) onde cada método foi avaliado ao se retornar no máximo $10\% \times |D| = 1597$, $25\% \times |D| = 3991$, $50\% \times |D| = 7983$ e $75\% \times |D| = 11.975$ dos documentos da coleção. Além disso, os resultados do método $Minwise$, em todos os gráficos, foram associados a losangos enquanto que os resultados dos métodos $Minmaxwise$, CSA_L e CSA foram associados, respectivamente, a estrelas, círculos e triângulos.

Na subseção 6.4.2 a busca é avaliada do ponto de vista de eficiência, i.e. são comparados o tempo médio para indexar um documento ou para extrair consultas, de um documento suspeito, e busca-las em todas as abordagens. A subseção 6.4.3 avalia o quão eficaz cada método é para identificar plágio enquanto que a subseção 6.4.1 avalia cada método em relação ao tempo, de criação do índice gerado, assim como também avalia a eficiência ao se extrair as consultas (i.e. o tempo médio para se extrair uma consulta). Já a última subseção apresenta as considerações geradas a partir da analises dos resultados anteriores.

6.4.1 Criação do índice e extração consultas

A missão da Recuperação Heurística é reduzir o espaço de comparação do próximo passo da identificação de Plágio Externo no menor tempo possível. Para tanto, o tempo de resposta a uma consulta é de fundamental importância visto que, um documento suspeito pode gerar várias consultas. Portanto, a figura 6.5 apresenta a comparação do tempo de criar uma consulta em relação ao método $Minwise$. O método $Minmaxwise$ foi o método com menor CRT, em relação ao método $Minwise$, enquanto que o método CSA apresentou a melhor CRT. Contudo, isto não significa que o método $Mimaxwise$ não apresentou bons resultados. De fato, a CRT do $Mimaxwise$ ($\simeq 2.0$) significa que, em média, o método leva a metade do tempo para extrair uma consulta. Além disso, os métodos CSA_L e CSA levaram aproximadamente um terço e um quarto do tempo do método $Minwise$ para gerar uma consulta.

Figura 6.5: CRT entre o $Minmax$, o $MinmaxCSA$ e o $MinmaxCSA_L$ e o $Minwise$ para extrair uma consulta e o tempo médio, em segundos, para o $Minwise$ extrair uma consulta

k	Tempo de extração de consultas médio (Minwise)	CRT		
		$Minmax$	CSA_L	CSA
48	13.13	1.97	2.96	3.87
96	26.43	1.99	2.99	3.9
192	52.47	1.98	2.97	3.9
384	104.77	1.96	2.98	3.91
768	208.17	1.97	2.95	3.9

Um motor de busca de plágio também deve ser capaz de efetuar buscas em grandes coleções de dados e, portanto, um outro fator relevante é a capacidade de um método de indexar um grande número de documentos. Em vista disso, a tabela 6.7 avalia a capacidade de indexação de documentos por hora, i.e. apresenta a vazão de cada método para indexar documentos a cada hora. Já a figura 6.6 ilustra os valores de vazão da tabela 6.7 permitindo uma melhor compreensão da tendência que os resultados de cada método apresenta. É possível observar que a vazão de todos os métodos é inversamente proporcional ao número de assinaturas que cada documento deve ser representado onde, por exemplo, o método $Minwise$ consegue indexar 99,6 documentos por hora com 48 assinaturas enquanto que para indexar documentos com o dobro de assinaturas o mesmo método apresenta a metade da vazão, i.e. ($\simeq$) 49 documentos por hora. O método $Minwise$ apresenta os menores valores de vazão enquanto que os maiores valores são apresentados pelo método CSA seguido pelo método CSA_L. De forma geral, a vazão do método $Minmaxwise$ é, aproximadamente, 20% maior que a do método $Minwise$ enquanto que os métodos CSA_L e CSA apresentam vazões 31% e 36% maiores que a do $Minwise$.

Tabela 6.7: Vazão (em documentos por hora) para indexar os documentos fontes da coleção PAN-PC-11

k	Vazão Min $(doc./hora)$	Vazão $Minmax$ $(doc./hora)$	Vazão CSA_L $(doc./hora)$	Vazão CSA $(doc./hora)$
48	99.6	119.3	130.13	135.6
96	49.99	60.19	65.24	67.75
192	25	30.06	32.76	33.91
384	12.47	14.83	16.33	16.95
768	6.18	7.36	8.06	8.3

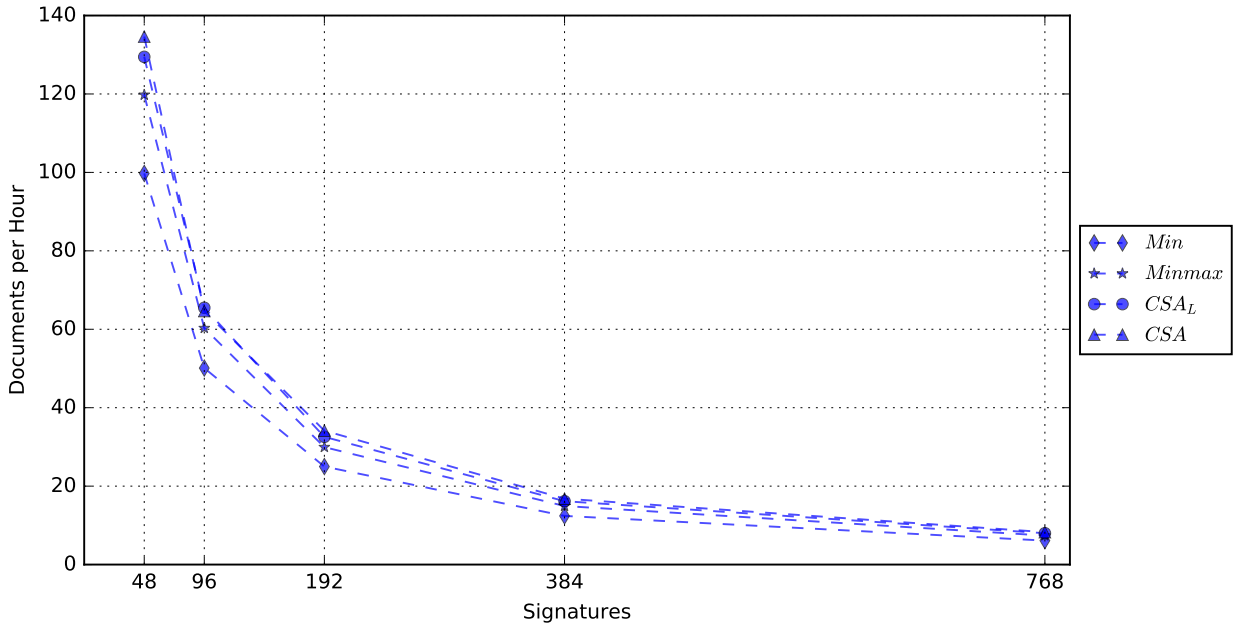


Figura 6.6: Vazão (em documentos por hora) para indexar os documentos fontes da coleção PAN-PC-11

6.4.2 Eficiência da busca

As figuras 6.7 e 6.8 apresentam a avaliação dos métodos em relação ao tempo de indexação de documentos e ao tempo de extração e busca de consultas. Para tanto, a figura 6.7 apresenta, em azul, o tempo médio que cada método leva para indexar um documento ($\Delta_{t,index}$) enquanto que o tempo médio para extrair e buscar ($\Delta_{t,ret}$) uma consulta, a partir de um documento suspeito, é apresentado em vermelho. Já a figura 6.8 apresenta tabelas feitas com o intuito de avaliar as melhorias no tempo médio $\Delta_{t,ret}$ dos métodos em relação ao $Minwise$.

Na figura 6.7 é possível observar que o tempo $\Delta_{t,ret}$ é muito mais influenciável pelo número de assinaturas selecionados que o tempo $\Delta_{t,index}$. Isto é, quanto maior for o número de assinaturas utilizado maior será o tempo médio de extração e busca de uma consulta. Contudo, apesar de $\Delta_{t,index}$ apresentar um aumento, o tempo de indexação médio $\Delta_{t,index}$ sofre uma influência bem menor que o de $\Delta_{t,ret}$. Também é possível observar que, em todos os casos, o método $Minwise$ apresenta os maiores tempos $\Delta_{t,index}$ e $\Delta_{t,ret}$ enquanto que os menores tempos são apresentados pelo método CSA. Ademais, os métodos CSA e CSA_L apresentam tempos menores que o $Minmaxwise$ em todos os casos apresentados.

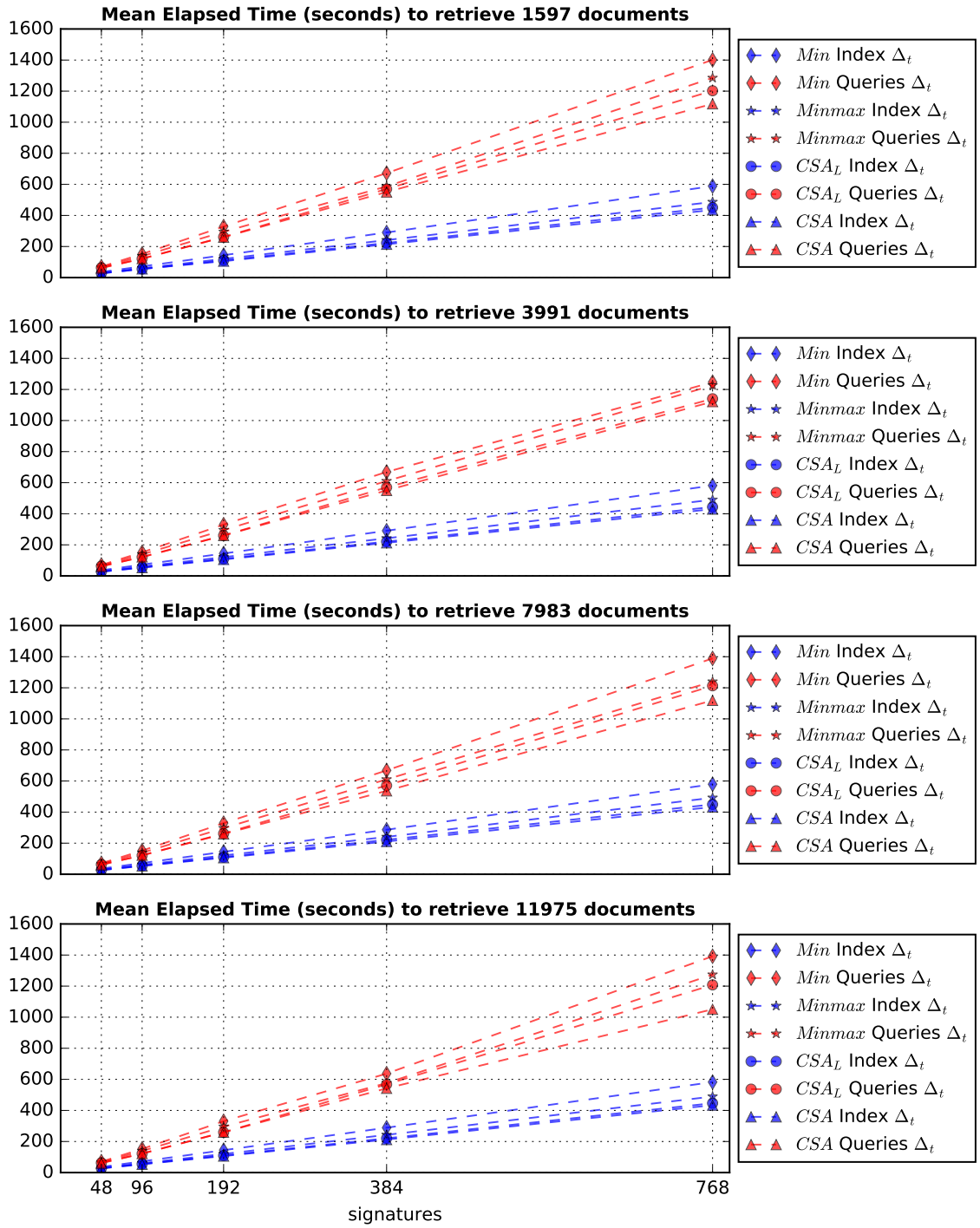


Figura 6.7: Tempo médio, na coleção PAN plagiarism corpus, de indexação (da coleção) e de extração e busca de consultas.

As tabelas da figura 6.8 apresentam a CRT de cada método em relação ao método *Minwise*. Assim como evidenciado na figura 6.7 todos os métodos apresentam um tempo de extração e busca de consultas menor que o do método *Minwise* e, portanto, apresentam CRT $> 1,0$. De fato, o pior CRT dos métodos *Minmaxwise*, CSA_L e CSA são, respectivamente, $1,02$, $1,09$ e $1,11$. Em contra partida, os melhores valores de CRT são, respectivamente, $1,15$, $1,27$ e $1,27$. Vale ressaltar

que, na maioria dos casos, a CRT do método $Minmaxwise$ é menor que o do CSA_L enquanto que a CRT do CSA_L é menor ou igual ao do método CSA.

Figura 6.8: Calculando a CRT do entre o $Minmax$, o $MinmaxCSA$ e o $MinmaxCSA_L$ e o $Minwise$ para retornar: 10% (recall@3991), 25% (recall@1597), 50% (recall@7983) e 75% (recall@11975) dos documentos do índice.

Extração e Busca(@1597)				
k	Tempo médio do Min (segundos)	CRT		
		$Minmax$	CSA_L	CSA
48	70.15	1.08	1.11	1.15
96	151.96	1.1	1.25	1.24
192	328.52	1.11	1.26	1.26
384	672.34	1.15	1.18	1.22
768	1,401.26	1.09	1.17	1.25

Tabela 6.8: CRT para 1597

Extração e Busca(@3991)				
k	Tempo médio do Min (segundos)	CRT		
		$Minmax$	CSA_L	CSA
48	70.26	1.08	1.11	1.16
96	151.95	1.1	1.26	1.24
192	330.18	1.12	1.27	1.27
384	669.15	1.1	1.17	1.22
768	1,246.69	1.02	1.09	1.11

Tabela 6.9: CRT para 3991

Extração e Busca(@7983)				
k	Tempo médio do Min (segundos)	CRT		
		$Minmax$	CSA_L	CSA
48	70.53	1.08	1.12	1.17
96	152.09	1.1	1.26	1.24
192	329.01	1.11	1.26	1.27
384	668.05	1.09	1.17	1.24
768	1,389.31	1.12	1.15	1.24

Tabela 6.10: CRT para 7983

Extração e Busca(@11975)				
k	Tempo médio do Min (segundos)	CRT		
		$Minmax$	CSA_L	CSA
48	70.35	1.09	1.12	1.17
96	152.47	1.1	1.25	1.25
192	330.46	1.12	1.28	1.27
384	636.83	1.11	1.12	1.17
768	1,392.25	1.09	1.15	1.32

Tabela 6.11: CRT para 11975

6.4.3 Eficácia da busca

A presente seção tem como objetivo avaliar a eficácia de um método ao se busca, em uma coleção de documentos $|D|$, os documentos que foram plagiados por um documentos suspeito. Para tanto, o resultados da figura 6.9 foi avaliado, para cada documento suspeito $d_q \in D_{susp}$, em termos de "recalls at pos" (recall@k) para $pos \in [1597, 3991, 7983, 11975]$. Em seguida os resultados de recall@pos dos métodos $Minmaxwise$, CSA e CSA_L foram comparados com os resultados do método $Minwise$ onde, as tabelas 6.12, 6.13, 6.14 e 6.15 apresentam o erro médio e o desvio padrão de cada método em relação ao $Minwise$. A porcentagem de valores de erro negativos também estão apresentados nas tabelas 6.13, 6.14 e 6.15 onde o objetivo é medir a porcentagem de casos em que determinada abordagem apresentou um recall@pos maior que o $Minwise$. Já as figuras 6.10, 6.11, 6.12 e 6.13

representam a Diferença Percentual Relativa (DPR) do recall@pos de cada método em relação *Minwise* onde cada círculo apresenta um valor de RPD e o seu raio apresenta a quantidade de vezes que o métodos apresentou um valor de DRP.

Na figura 6.9, cada valor de *pos* apresenta uma cor associada e, portanto, todos os resultados em azul claro são resultados do *recall*@11975, para todos os métodos avaliados, enquanto que os resultados vermelhos, azuis e verdes estão associados ao *recall*@7983, *recall*@1597 e *recall*@3991, respectivamente. Além disso, é possível observar que os valores de recall melhoram na medida em que o número de assinaturas geradas aumenta. Também é possível observar que, ao se aumentar o número de assinaturas, os resultados de recall dos métodos *Minmaxwise*, *CSA* e CSA_L convergem para valores próximos aos apresentados pelo *Minwise*. Afirmação que é corroborada pela diminuição dos valores de erro médio (Média) e do seu Desvio padrão (D.P.) nas tabelas 6.12, 6.13, 6.14 e 6.15.

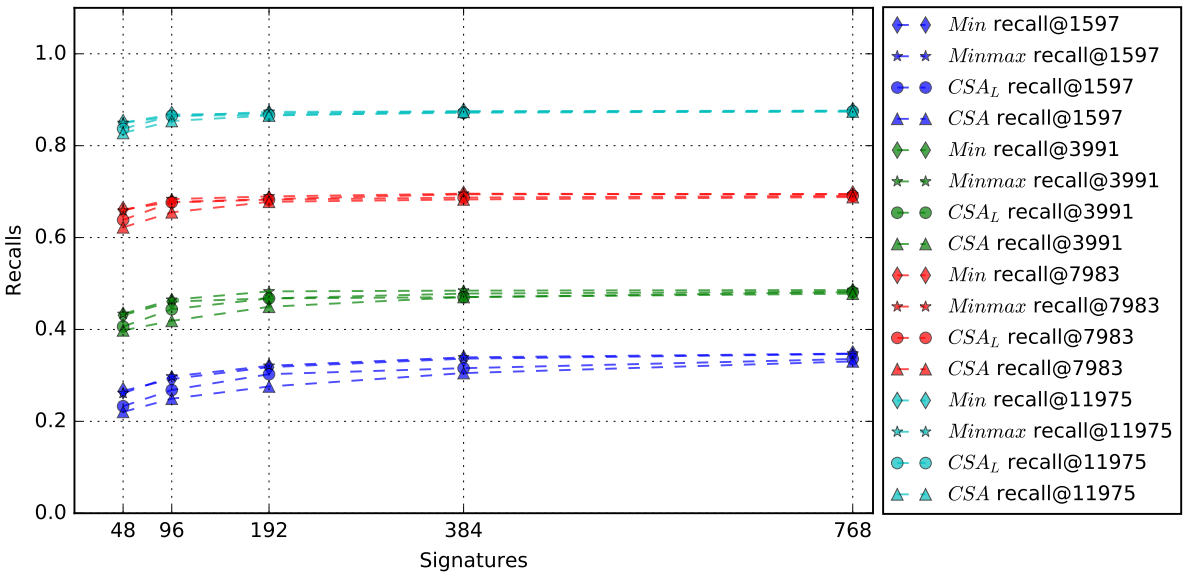


Figura 6.9: recall@pos do resultado da Recuperação Heurística na coleção *PAN plagiarism corpus*

De forma geral, os melhores resultados de erro médio foram apresentados pelo método *Minmaxwise* enquanto que os piores foram apresentados pelo método *CSA*. De fato, na maioria dos casos, o método *Minmaxwise* apresentou erro médio negativo o que significa que ele apresentou resultados maiores, e portanto melhores, que os do método *Minwise*. Ademais, a porcentagem de erro médio negativo do *Minmaxwise* variou de 83,67 % a 96,3 % enquanto que o método *CSA* variou de 76,96 % até 93,63 % e em nenhum caso o método *Minmaxwise* apresentou % menor que as dos outros dois métodos.

Tabela 6.12: Erro médio para recall@1597, seu desvio padrão (D.P.) e a porcentagem de valores de erro negativos (% neg.)

k	$Minmax$			CSA_L			CSA		
	Média	D.P.	% neg.	Média	D. P.	% neg.	Média	D. P.	% neg.
48	$5.06 \cdot 10^{-3}$	0.47	84.46	$3.33 \cdot 10^{-2}$	0.52	79.39	$4.58 \cdot 10^{-2}$	0.55	76.96
96	$-5.54 \cdot 10^{-3}$	0.43	86.94	$2.4 \cdot 10^{-2}$	0.49	81.61	$4.29 \cdot 10^{-2}$	0.53	77.9
192	$-2.99 \cdot 10^{-3}$	0.39	88.86	$1.53 \cdot 10^{-2}$	0.45	84.23	$4.2 \cdot 10^{-2}$	0.48	80.21
384	$-2.86 \cdot 10^{-3}$	0.34	91.27	$2.07 \cdot 10^{-2}$	0.4	86.16	$3.14 \cdot 10^{-2}$	0.43	83.51
768	$-8.95 \cdot 10^{-4}$	0.3	92.63	$1.09 \cdot 10^{-2}$	0.36	88.78	$1.62 \cdot 10^{-2}$	0.39	87.38

Tabela 6.13: Erro médio para recall@3991, seu desvio padrão (D.P.) e a porcentagem de valores de erro negativos (% neg.)

k	$Minmax$			CSA_L			CSA		
	Média	D.P.	% neg.	Média	D. P.	% neg.	Média	D. P.	% neg.
48	$-2.9 \cdot 10^{-3}$	0.44	83.67	$2.46 \cdot 10^{-2}$	0.5	77.94	$3.3 \cdot 10^{-2}$	0.52	75.64
96	$-3.91 \cdot 10^{-3}$	0.4	86.32	$1.67 \cdot 10^{-2}$	0.45	81.39	$4.2 \cdot 10^{-2}$	0.49	77.5
192	$-1.52 \cdot 10^{-2}$	0.36	89.44	$2.09 \cdot 10^{-5}$	0.41	85.16	$1.83 \cdot 10^{-2}$	0.44	81.13
384	$-6.49 \cdot 10^{-3}$	0.3	92.09	$7.76 \cdot 10^{-3}$	0.36	87.74	$7.49 \cdot 10^{-3}$	0.4	84.78
768	$-3.5 \cdot 10^{-3}$	0.26	93.35	$1.21 \cdot 10^{-3}$	0.31	90.46	$4.54 \cdot 10^{-3}$	0.35	88.38

Tabela 6.14: Erro médio para recall@7983, seu desvio padrão (D.P.) e a porcentagem de valores de erro negativos (% neg.)

k	$Minmax$			CSA_L			CSA		
	Média	D.P.	% neg.	Média	D. P.	% neg.	Média	D. P.	% neg.
48	$2.36 \cdot 10^{-3}$	0.34	84.46	$2.25 \cdot 10^{-2}$	0.39	79.89	$3.88 \cdot 10^{-2}$	0.41	77.32
96	$-6.52 \cdot 10^{-3}$	0.31	88.4	$5.29 \cdot 10^{-4}$	0.34	84.94	$2.23 \cdot 10^{-2}$	0.38	80.79
192	$-5.74 \cdot 10^{-3}$	0.28	90.56	$1.28 \cdot 10^{-3}$	0.31	87.26	$5.97 \cdot 10^{-3}$	0.33	85.18
384	$-1.28 \cdot 10^{-3}$	0.23	92.87	$7.2 \cdot 10^{-3}$	0.28	90.02	$1.13 \cdot 10^{-2}$	0.3	87.42
768	$9.48 \cdot 10^{-4}$	0.2	93.85	$4.18 \cdot 10^{-3}$	0.24	91.29	$6.92 \cdot 10^{-3}$	0.26	89.9

Tabela 6.15: Erro médio para recall@11975, seu desvio padrão (D.P.) e a porcentagem de valores de erro negativos (% neg.)

k	$Minmax$			CSA_L			CSA		
	Média	D.P.	% neg.	Média	D. P.	% neg.	Média	D. P.	% neg.
48	$8.78 \cdot 10^{-4}$	0.25	87.9	$1.34 \cdot 10^{-2}$	0.27	84.82	$2.26 \cdot 10^{-2}$	0.29	82.57
96	$3.75 \cdot 10^{-3}$	0.22	90.48	$2.74 \cdot 10^{-3}$	0.23	88.84	$1.41 \cdot 10^{-2}$	0.25	86.14
192	$-3.35 \cdot 10^{-3}$	0.19	93.53	$3.47 \cdot 10^{-3}$	0.21	90.6	$4.76 \cdot 10^{-3}$	0.23	89.5
384	$2.48 \cdot 10^{-4}$	0.16	94.91	$3.23 \cdot 10^{-3}$	0.18	93.13	$1.32 \cdot 10^{-3}$	0.2	91.99
768	$1.31 \cdot 10^{-3}$	0.13	96.03	$1.63 \cdot 10^{-3}$	0.16	94.73	$2.07 \cdot 10^{-3}$	0.17	93.63

A partir dos valores de DPR apresentados nas figuras 6.10, 6.11, 6.12 e 6.13 observa-se o aumento percentual dos valores de recall@pos dos métodos $Minmaxwise$, CSA_L e CSA em relação aos obtidos pelo método $Minwise$. Os valores de DPR iguais a 2 significam que o método $Minwise$ apresentou um valor de $recall@pos > 0$ enquanto que o método comparado apresentou $recall@pos = 0$. Por exemplo, na figura 6.10 para 48 permutações o método CSA_L apresentou um círculo de centro no valor de $DPR = 2$ e raio maior que o método $Minmaxwise$. Portanto, CSA_L apresentou mais valores de recall@1597 nulos que o $Minmaxwise$. De forma similar, os valores de DPR iguais a -2 significam que o método $Minwise$ apresentou $recall@pos = 0$ enquanto que o método avaliado não. Logo, os círculos de centro em valores negativos apresentam que determinado método apresentou um $recall@pos$ melhor que o $Minwise$, os de centro em valores positivos apresentam recalls piores que o $Minwise$ e os de centro no valor zero representam que determinado método apresentou o mesmo valor de recall que o $Minwise$. Os resultados de DPR apresentados indicam que todos os métodos apresentaram muitos valores de recall@pos próximos aos do $Minwise$ e, portanto, os resultados do $Minmaxwise$ são maiores porém próximos aos apresentados pelo método $Minwise$.

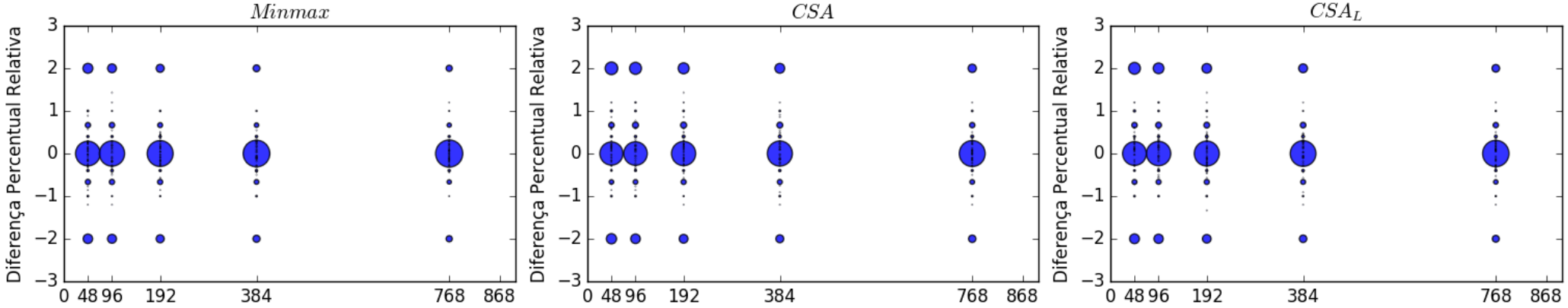


Figura 6.10: DRP para os métodos $Minmax$, CSA e CS_L calculadas a partir dos valores de recall@1597

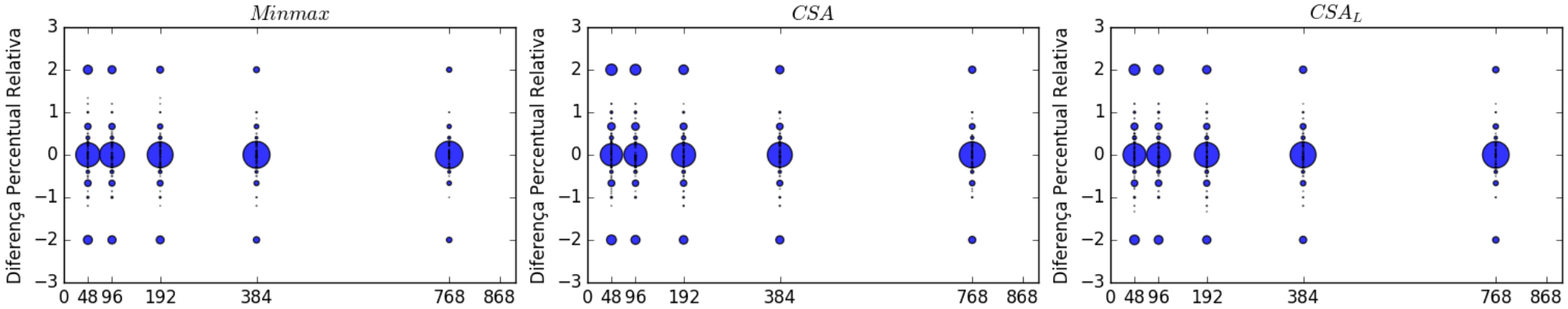


Figura 6.11: DRP para os métodos $Minmax$, CSA e CS_L calculadas a partir dos valores de recall@3991

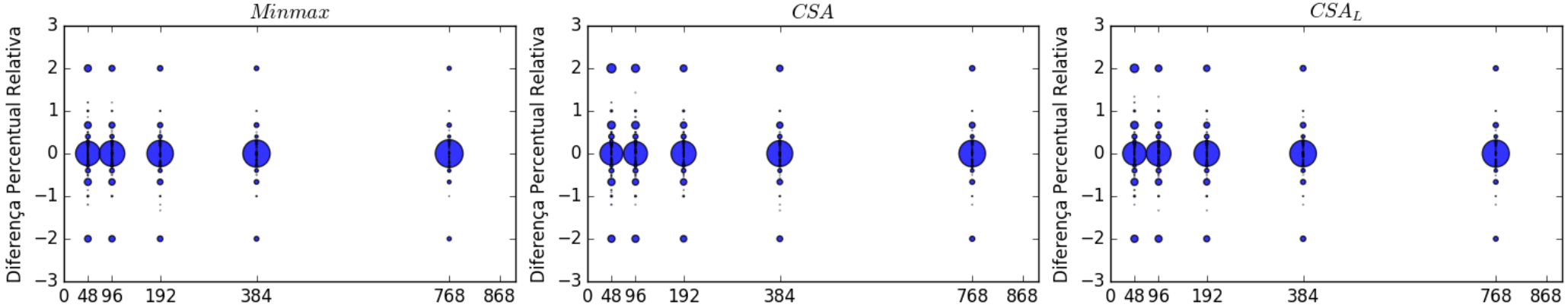


Figura 6.12: DRP para os métodos $Minmax$, CSA e CS_L calculadas a partir dos valores de recall@7983

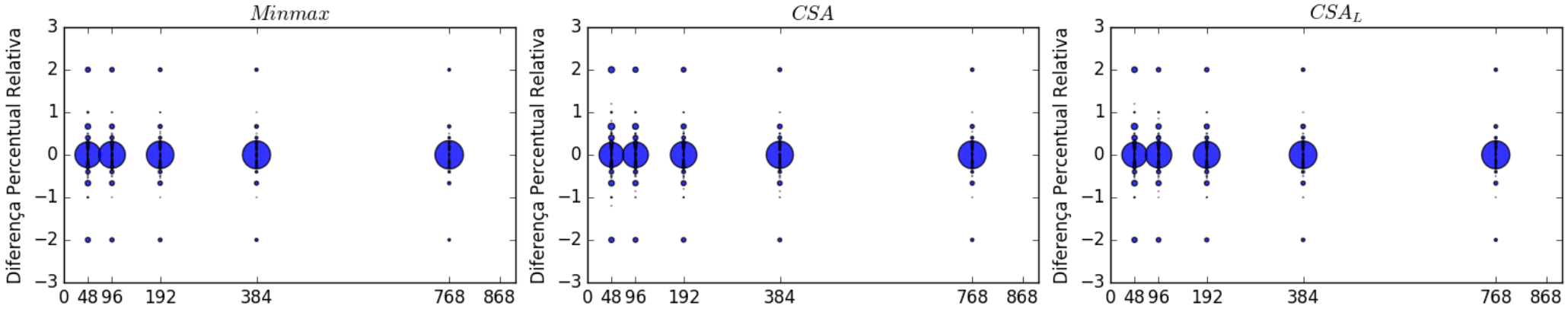


Figura 6.13: DRP para os métodos $Minmax$, CSA e CS_L calculadas a partir dos valores de recall@11975

6.4.4 Considerações sobre o experimento

Os métodos foram analisados de acordo com quatro aspectos relevantes para motores de busca: (*i*) A capacidade de indexação de documentos (medido pela vazão), (*ii*) o tempo empregado para extrair consultas (de um documento suspeito), (*iii*) o tempo para se realizar uma busca (extraindo a consulta e realizando a busca) e (*iv*) quão eficaz é a busca para retornar documentos relevantes (i.e. os documentos fonte do plágio).

Na avaliação (*i*) o método $Minmaxwise$ apresentou uma vazão 20% superior em relação a do método $Minwise$, assim como os métodos CSA_L e CSA que apresentaram, respectivamente, 31% e 36% de melhoria em relação ao $Minwise$. Portanto, os métodos CSA e CSA_L conseguem usar um tempo de CPU expressivamente menor que os métodos $Minwise$ e $Minmaxwise$ para indexar uma coleção com muitos documentos. Por exemplo, suponha que o método $Minwise$ apresente uma vazão de 100 documentos/hora e, portanto, a vazão dos métodos $Minmaxwise$, CSA_L e CSA seria, respectivamente, 120 documentos/hora, 131 documentos/hora e 136 documentos/hora. Logo, para indexar 550 mil documentos, i.e. indexar todos os novos artigos criados em um mês[3] na wikipedia (WIKIMEDIA, 2016), o método $Minwise$ precisaria de 5500 horas de CPU enquanto que os métodos $Minmaxwise$, CSA_L e CSA economizariam aproximadamente 916(16%), 1301(23%) e 1455(26%) horas de CPU.

Do ponto de vista da extração de consultas (*ii*) os métodos apresentaram tempo de extração com valores diretamente proporcionais ao número de assinaturas a ser gerado. Isto é, o tempo de extração de uma consulta dobra ou quadruplica ao se dobra ou quadruplicar o número de assinaturas, por documento, conforme apresentado na tabela 6.5. Além disso, o número de assinaturas influencia de forma mais acentuada o tempo de extração e busca (*iii*), como evidencia a figura 6.7. Por exemplo, a tabela 6.9 apresenta o tempo de extração e busca médio para o método $Minwise$ retornar, no máximo, 3991 documentos. Dobrando o número de permutações o método $Minwise$ apresentou um tempo $2,16$ vezes maior enquanto que, ao se quadruplicar, o tempo aumenta por um fator de $4,7$. Finalmente no que diz respeito ao fator de aumento os resultados evidenciam que para se gerar e buscar consultas com $768(48 \times 16)$ assinaturas o método $Minwise$ apresenta um aumento de $17,74$ no seu tempo.

Ao se comparar os outros métodos com o $Minwise$ é possível observar que, conforme o discutido na seção 5.4, o método $Minmaxwise$ leva metade do tempo (CRT $\simeq 2$), enquanto que os métodos CSA_L e CSA levam um terço (CRT $\simeq 3$) e um quarto (CRT $\simeq 4$) do tempo, para extrair consultas com o mesmo número de

[3]neste caso, setembro de 2016

assinaturas. Contudo, a CRT de todos os métodos é mais modesta ao se extrair e buscar as consultas. Isto é, o $Minmaxwise$ apresenta um CRT entre $1,08$ e $1,15$ enquanto que os métodos CSA_L e CSA apresentam valores entre $1,09$ e $1,28$ e $1,15$ e $1,32$, respectivamente. Isto se deve ao fato de que boa parte do tempo de busca é gasto unificando a lista de todos os documentos que apresentam as mesmas assinaturas da consulta. Ação que aumenta o tempo total de consulta e reduz a contribuição de CRT feita durante a extração da consulta.

Os resultados de (iv) demonstraram que os métodos CSA e CSA_L apresentaram valores de recall menores que os dos métodos $Minwise$ e $Minmaxwise$. Contudo, ao se aumentar o número de assinaturas a diferença entre eles e o $Minwise$ diminuiu. De fato, ao se observar o gráfico 6.9 é possível notar que os valores de recall dos métodos tendem a ficar mais próximos ao se aumentar o número de assinaturas e o número máximo de documentos retornados. Ademais, além de corroborar com as afirmações anteriores, as tabelas 6.12, 6.13, 6.14 e 6.15 também informam que o erro médio do recall entre os métodos $Minmaxwise$, CSA e CSA_L e o método $Minwise$ é pequeno visto que, o maior erro médio do método $Minmaxwise$ é $0,00506$ enquanto que o maior erro médio dos métodos CSA_L e CSA são $0,0333$ e $0,0458$ respectivamente. Isto é, se o método $Minwise$ apresentar, por exemplo, um recall de $0,6$ os métodos $Minmaxwise$, CSA e CSA_L apresentariam, no pior caso (48 assinaturas retornando 1597 documentos), um recall de $0,594$, $0,5677$ e $0,554$.

Capítulo 7

Conclusões

> *A ship is safe in harbor, but that's not what ships are for.*
>
> — William Shedd

Nesta tese foi estudado o problema da Recuperação Heurística no Plágio Externo. O estágio de Recuperação Heurística é uma tarefa de recuperação de informação que tem como objetivo reduzir o espaço de comparação do estágio posterior, que é demorado e computacionalmente custoso. Foram encontrados diversos trabalhos que lidam com o problema de Recuperação Heurística reduzindo o tempo de Busca e Recuperação de cada documento a partir de um Motor de Busca baseado no modelo probabilístico BM25 (POTTHAST *et al.*, 2011). Entretanto, a literatura também apresenta Motores de Busca baseados na família de modelos LSH que associam assinaturas aos documentos com o objetivo de reduzir o espaço de representação. Além disso, os modelos LSH auxiliam na criação de índices invertidos aumentando a capacidade de indexar grandes coleções de documentos e evitando a maldição da alta dimensionalidade.

A indexação e busca usando LSH foi apresentada no capítulo 4 onde, também foi apresentada uma adaptação do *pipeline* de execução de métodos LSH na Recuperação Heurística. Em seguida, algoritmos para indexação e busca, aplicáveis na Recuperação Heurística, foram apresentados. Entretanto, além dos métodos *Minwise* e *Minmaxwise Hashing* não foram encontrados, na literatura, outros métodos LSH para gerar mais assinaturas por permutação. Nesse cenário, esta tese apresenta métodos LSH baseados na propriedade geométrica alcunhada de Arcos de Setores Circulares e este capítulo sumariza as contribuições enumerando as direções dos trabalhos futuros desta tese.

O capítulo 5 apresentou a propriedade triangular que representa o intervalo de valores de um lattice em um valor numérico. Para tanto, algumas propriedades derivadas de axiomas foram propostas e demonstradas. Em seguida, com o intuito de melhorar a capacidade de representação da propriedade triangular, dois métodos

baseados em Arcos de Setores Circulares foram propostos, seus algoritmos apresentados e avaliados da perspectiva da similaridade de Jaccard e do tempo de geração de assinaturas.

No capítulo 6 foram realizadas duas avaliações experimentais nos métodos $MinmaxwiseCSA_L$, $MinmaxCSA$ e *Minmaxwise Hashing*. O primeiro grupo de experimentos tinha como objetivo avaliar como a similaridade de Jaccard se comporta ao se utilizar os métodos propostos e foi realizada nas coleções de dados Meter, Plagiarised Short Answers e PAN-PC-11. O segundo grupo de experimentos foi realizado na coleção PAN-PC-11 e avaliou se os métodos propostos são aplicáveis no estágio de Busca Heurística do Plágio Externo. Para tanto, em cada método foi avaliado o tempo de criação do índice de documentos, o tempo de extração das consultas, o tempo de busca de uma consulta e, por fim, quão eficaz cada método foi para recuperar os documentos fontes do plágio.

No primeiro conjunto de experimentos ($ESJP$) os métodos $MinmaxwiseCSA_L$ e $MinmaxCSA$ apresentaram, como esperado, uma redução do tempo de 33% e 50% em relação ao *Minmaxwise Hashing* em troca de uma pequena imprecisão ao se medir Jaccard. Contudo, foi observado que ao se aumentar o número de permutações essa imprecisão reduziu.

O segundo grupo de experimentos também apresenta uma pequena imprecisão nos valores de recall assim como o comportamento da imprecisão se reduz ao se aumentar o número de permutações. Todavia, apesar da melhoria ser modesta ao se comparar com o método *Minmaxwise hashing*, os métodos $MinmaxwiseCSA_L$ e $MinmaxCSA$ apresentaram um tempo médio de extração e busca de consultas menor que o do método *Minmaxwise hashing*. Além disso, os métodos $MinmaxwiseCSA_L$ e $MinmaxCSA$ apresentaram melhorias consideráveis no tempo médio de indexação de documentos assim como no tempo de extração de consultas o que é de fundamental importância para um motor de busca que deve indexar grandes coleções de dados.

Os métodos $MinmaxwiseCSA_L$ e $MinmaxCSA$ mostraram ser aplicáveis no problema da Recuperação Heurística do Plágio Externo. Os tempo de indexação e busca são as contribuições centrais dos métodos propostos em troca de uma pequena imprecisão. A partir dos experimentos realizados, foram obtidos resultados que forneceram elementos que corroboram com o objetivo central da tese que é: reduzir custo computacional para indexar e buscar no espaço de busca da Recuperação Heurística representando a similaridade entre dois documentos como um problema de intersecção de conjuntos.

7.1 Sumário das contribuições

As principais contribuições desta tese podem ser sumarizadas da seguinte forma:

1. **Avaliação dos métodos *Minwise e Minmaxwise hashing* na Recuperação Heurística do Plágio Externo:** Até o momento os dois métodos foram avaliados e comparados em problemas de busca relacionados com reúso de texto e busca de imagens e não no problema de plágio externo. Os experimentos permitiram avaliar a eficácia e eficiência dos dois métodos na Recuperação Heurística onde vários aspectos diferentes como o tempo de indexação, o tempo de geração de consultas e de busca foram analisados detalhadamente, o que não foi observado na literatura nos experimentos do reúso de texto.

2. **Propriedade triangular e suas expansões baseadas em Arcos de Setores Circulares:** Foi proposto um arcabouço teórico que codifica o intervalo de valores um lattice em assinaturas que servem para identificar o objeto que gerou o lattice. Além disso, todas as propriedades necessárias foram formalizadas e devidamente demonstradas.

3. **Métodos LSH $MinMaxCSA_L$ e $MinMaxCSA$ e seus algoritmos de indexação e busca:** Foram propostos dois métodos, baseados no arcabouço teórico da propriedade triangular e dos Arcos de Setores Circulares para a Busca Heurística do plágio externo. Os algoritmos de indexação e busca são determinísticos, baseados em índices invertidos e apresentaram tempos menores que os dos métodos *Minwise e Minmaxwise hashing*.

7.2 Trabalhos futuros

Durante o desenvolvimento da presente tese alguns estudos foram relacionados para um futuro estudo. Eles estão divididos em 3 grupos de trabalhos futuros: (a) exploração de abordagens periféricas ao motor de busca, (b) paralelismo e (c) aplicabilidade em outros problemas.

7.2.1 (a) exploração de abordagens periféricas ao motor de busca

Assim como discutido no capítulo 4 existem várias contribuições que são aplicadas em etapas periféricas ao modelo de busca e recuperação como a tokenização, a extração de consultas (i.e. gerar várias consultas a partir de um documento suspeito) e o controle da busca (i.e. decisão de quais consultas valem a pena serem executadas). Logo os seguintes trabalhos foram identificados:

(a.1) estudar possíveis estratégias de tokenização que favoreçam os métodos LSH a identificar plágio explorando os aspectos léxicos, sintáticos e semânticos do texto. Por exemplo: extrair n-gramas, sentenças, classes gramaticais, sinônimos ou hiperônimos.

(a.2) estudar a combinação outras métricas de similaridade, além da similaridade de Jaccard, com as assinaturas geradas que favoreçam a identificação de plágio pelos métodos LSH.

7.2.2 (b) paralelismo

Os método LSH baseados em permutações podem ser paralelizados de diversas maneiras e, portanto, foram enumerados alguns trabalhos futuros da presente tese na perspectiva do paralelismo:

(b.1) Avaliar o impacto da técnica de *banding* nos resultados dos métodos durante a Busca Heurística.

(b.2) Avaliar a possibilidade de paralelizar ou distribuir a técnica de *banding* e o seu impacto nos resultados do métodos.

(b.3) Avaliar e propor abordagens de paralelismo a ser aplicado nas permutações

(b.4) Avaliar e propor abordagens de paralelismo a ser aplicado nas permutações e na indexação ao mesmo tempo.

(b.5) Avaliar e propor abordagens de paralelismo a ser aplicado na geração de *fingerprint*, nas permutações e na indexação de forma simultânea. Isto é, como paralelizar para cada documento ou consulta a geração de *fingerprint* e a indexação de suas permutações.

(b.6) Avaliar a otimização das operações que geram assinaturas utilizando GPU.

(b.7) Avaliar a otimização das operações ao se incorporar em cada thread da GPU as permutações e a seleção de assinaturas.

(b.8) Avaliar e propor métodos para particionar o vocabulário gerado aumentando assim o número de assinaturas gerado por permutação.

7.2.3 (c) aplicabilidade em outros problemas

O método *Minwise hashing* é muito explorado em problemas de como busca e classificação de imagens, vídeos e streams e, portanto, possíveis trabalhos relacionados com as propostas da tese são:

(c.1) Avaliar a aplicação dos métodos $Minmax$, $MinmaxCSA$ e $MinmaxCSA_L$ em problemas de busca e classificação de imagens.

(c.2) Avaliar a aplicação dos métodos $Minmax$, $MinmaxCSA$ e $MinmaxCSA_L$ em problemas de busca e classificação de música e áudio.

(c.3) Avaliar a aplicação dos métodos $Minmax$, $MinmaxCSA$ e $MinmaxCSA_L$ em problemas de busca e classificação de vídeos.

(c.4) Avaliar a aplicação dos métodos $Minmax$, $MinmaxCSA$ e $MinmaxCSA_L$ em problemas de busca e classificação de *streamming* de dados.

(c.5) Avaliar a aplicação dos métodos $Minmax$, $MinmaxCSA$ e $MinmaxCSA_L$ como métodos de solução de *Big data*.

Apêndice A

Exemplo de plágio

In object-oriented programming, inheritance is a way to form new classes (instances of which are called objects) using classes that have already been defined. The inheritance concept was invented in 1967 for Simula.

The new classes, known as derived classes, take over (or inherit) attributes and behavior of the pre-existing classes, which are referred to as base classes (or ancestor classes). It is intended to help reuse existing code with little or no modification.

Inheritance provides the support for representation by categorization in computer languages. Categorization is a powerful mechanism number of information processing, crucial to human learning by means of generalization (what is known about specific entities is applied to a wider group given a belongs relation can be established) and cognitive economy (less information needs to be stored about each specific entity, only its particularities).

Inheritance is also sometimes called generalization, because the is-a relationships represent a hierarchy between classes of objects. For instance, a "fruit"is a generalization of "apple", "orange", "mango"and many others. One can consider fruit to be an abstraction of apple, orange, etc. Conversely, since apples are fruit (i.e., an apple is-a fruit), apples may naturally inherit all the properties common to all fruit, such as being a fleshy container for the seed of a plant.

An advantage of inheritance is that modules with sufficiently similar interfaces can share a lot of code, reducing the complexity of the program. Inheritance therefore has another view, a dual, called polymorphism, which describes many pieces of code being controlled by shared control code. Inheritance is typically accomplished either by overriding (replacing) one or more methods exposed by ancestor, or by adding new methods to those exposed by an ancestor.

Complex inheritance, or inheritance used within a design that is not sufficiently mature, may lead to the Yo-yo problem.

Figura A.1: Documento fonte extraído da coleção *Plagiarised Short Answers (PSA)* (CLOUGH e STEVENSON, 2011)

In object-oriented programming, inheritance is a way to form new classes (instances of which are called objects) using classes that have already been defined. The inheritance concept was invented in 1967 for Simula. The new classes, known as derived classes, take over (or inherit) attribute and behaviour of the pre-existing classes, which are referred to as base classes (or ancestor classes). It is intended to help reuse existing code with little or no modification. Inheritance provides the support for representation by categorization in computer languages. Categorization is a powerful mechanism number of information processing, crucial to human learning by means of generalization (what is known about specific entities is applied to a wider group given a belongs relation can be established) and cognitive economy (less information needs to be stored about each specific entity, only its particularities). Inheritance is also sometimes called generalization, because the is-a relationships represent a hierarchy between classes of objects. For instance, a "fruit"is a generalization of "apple", "orange", "mango"and many others. One can consider fruit to be an abstraction of apple, orange, etc. Conversely, since apples are fruit (i.e., an apple is-a fruit), apples may naturally inherit all the properties common to all fruit, such as being a fleshy container for the seed of a plant. An advantage of inheritance is that modules with sufficiently similar interfaces can share a lot of code, reducing the complexity of the program. Inheritance therefore has another view, a dual, called polymorphism, which describes many pieces of code being controlled by shared control code. Inheritance is typically accomplished either by overriding (replacing) one or more methods exposed by ancestor, or by adding new methods to those exposed by an ancestor.

Figura A.2: Plágio de A.1, com poucas alterações, extraído da coleção *Plagiarised Short Answers (PSA)* (CLOUGH e STEVENSON, 2011)

inheritance in object oriented programming is where a new class is formed using classes which have allready been defined. These classes have have some of the behavior and attributes which where existent in the classes that it inherited from. The peropos of inheritance in object oriented programming is to minimize the reuse of existing code without modification.

Inheritance allowes classes to be categorized, similer to the way humans catagorize. It also provides a way to generalize du to the "is a"relationship between classes. For example a "cow"is a generalization of änimalsïmilarly so are "pigs"& cheaters". Defeining classes in this way, allows us to define attributes and behaviours which are commen to all animals in one class, so cheaters would natuarly inheart properities commen to all animals.

The advantage of inheritance is that classes which would otherwise have alot of similar code , can instead shair the same code, thus reducing the complexity of the program. Inheritance, therefore, can also be refered to as polymorphism which is where many pieces of code are controled by shared control code.

Inheritance can be accomplished by overriding methods in its ancestor, or by adding new methods.

Figura A.3: Plágio de A.1, com muitas alterações, extraído da coleção *Plagiarised Short Answers (PSA)* (CLOUGH e STEVENSON, 2011)

Referências Bibliográficas

ABDI, A., IDRIS, N., ALGULIYEV, R. M., et al., 2015a, "PDLK: Plagiarism detection using linguistic knowledge", *Expert Systems with Applications*, v. 42, n. 22, pp. 8936 – 8946. ISSN: 0957-4174. doi: http://dx.doi.org/10.1016/j.eswa.2015.07.048. Disponível em: <http://www.sciencedirect.com/science/article/pii/S0957417415005084>.

ABDI, A., IDRIS, N., ALGULIYEV, R. M., et al., 2015b, "PDLK: Plagiarism detection using linguistic knowledge", *Expert Systems with Applications*, v. 42, n. 22, pp. 8936 – 8946. ISSN: 0957-4174. doi: http://dx.doi.org/10.1016/j.eswa.2015.07.048. Disponível em: <http://www.sciencedirect.com/science/article/pii/S0957417415005084>.

ADEEL NAWAB, R. M., STEVENSON, M., CLOUGH, P., 2012, "Detecting Text Reuse with Modified and Weighted N-grams". SemEval '12, pp. 54–58, Stroudsburg, PA, USA. Association for Computational Linguistics. Disponível em: <http://dl.acm.org/citation.cfm?id=2387636.2387646>.

AFP, T., 2013, "Taiwan defence minister quits after plagiarism allegation", *AFP*, (ago.). Disponível em: <http://www.foxnews.com/world/2013/08/06/taiwan-defence-minister-quits-after-plagiarism-allegation/>.

ALZAHRANI, S. M., SALIM, N., ABRAHAM, A., 2012, "Understanding plagiarism linguistic patterns, textual features, and detection methods", *Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on*, v. 42, n. 2, pp. 133–149.

ALZAHRANI, S. M., SALIM, N., PALADE, V., 2015, "Uncovering highly obfuscated plagiarism cases using fuzzy semantic-based similarity model", *Journal of King Saud University - Computer and Information Sciences*, v. 27, n. 3, pp. 248 – 268. ISSN: 1319-1578. doi: http://dx.doi.org/10.1016/j.jksuci.2014.12.001. Disponível em: <http://www.sciencedirect.com/science/article/pii/S1319157815000361>.

AURÉLIO, O., 2001, *plagiarism definition in dicionario do aurelio*. Aurélio, online. Disponível em: <https://dicionariodoaurelio.com/plagios>.

BAEZA-YATES, R. A., RIBEIRO-NETO, B., 1999, *Modern Information Retrieval*. Boston, MA, USA, Addison-Wesley Longman Publishing Co., Inc. ISBN: 020139829X.

BAKER, A., GHOSH, S., KUMAR, A., et al., 2008, "Apology [Plagiarism]", *Circuits and Systems Magazine, IEEE*, v. 8, n. 3 (Third), pp. 95–95. ISSN: 1531-636X. doi: 10.1109/MCAS.2008.923978.

BARRÓN-CEDEÑO, A., 2012. "On the Mono- and Cross-Language Detection of Text Re-Use and Plagiarism". .

BARRÓN-CEDEÑO, A., GUPTA, P., ROSSO, P., 2013, "Methods for cross-language plagiarism detection", *Knowledge-Based Systems*, v. 50, pp. 211 – 217. ISSN: 0950-7051. doi: http://dx.doi.org/10.1016/j.knosys.2013.06.018. Disponível em: <http://www.sciencedirect.com/science/article/pii/S0950705113002001>.

BENGIO, Y., DUCHARME, R., VINCENT, P., et al., 2003, "A Neural Probabilistic Language Model", *J. Mach. Learn. Res.*, v. 3 (mar.), pp. 1137–1155. ISSN: 1532-4435. Disponível em: <http://dl.acm.org/citation.cfm?id=944919.944966>.

BENTLEY, J. L., 1975, "Multidimensional Binary Search Trees Used for Associative Searching", *Commun. ACM*, v. 18, n. 9 (set.), pp. 509–517. ISSN: 0001-0782. doi: 10.1145/361002.361007. Disponível em: <http://doi.acm.org/10.1145/361002.361007>.

BERNSTEIN, Y., ZOBEL, J., 2004, "A Scalable System for Identifying Co-derivative Documents". In: Apostolico, A., Melucci, M. (Eds.), *String Processing and Information Retrieval*, v. 3246, *Lecture Notes in Computer Science*, Springer Berlin Heidelberg, pp. 55–67. ISBN: 978-3-540-23210-0. doi: 10.1007/978-3-540-30213-1_6. Disponível em: <http://dx.doi.org/10.1007/978-3-540-30213-1_6>.

BERRY, C., DE LA FUENTE, J., MULLIN, M., et al., 2007, "Notice of Violation of IEEE Publication Principles Nuclear Localization of HIV-1 Tat Functionalized Gold Nanoparticles", *NanoBioscience, IEEE Transactions on*, v. 6, n. 4 (Dec), pp. 262–269. ISSN: 1536-1241. doi: 10.1109/TNB.2007.908973.

BRODER, A. Z., 1997, "On the resemblance and containment of documents". In: *Compression and Complexity of Sequences 1997. Proceedings*, pp. 21–29. IEEE.

BUHLER, J., 2001, "Efficient large-scale sequence comparison by locality-sensitive hashing", *Bioinformatics*, v. 17, n. 5, pp. 419–428.

CAVANILLAS, S., 2008, "Cyberplagiarism in University Regulations", *The e-journal produced by the UOC's Languages and Cultures, and Humanities Departments*. ISSN: 1575-2275.

CENTER FOR YOUTH ETHICS, C., 2012, *2012 Report Card on the Ethics of American Youth*. , Josephson Institute of Ethics. Disponível em: <https://charactercounts.org/wp-content/uploads/2014/02/ReportCard-2012-DataTables.pdf>.

CESKA, Z., TOMAN, M., JEZEK, K., 2008, "Multilingual Plagiarism Detection". In: *Proceedings of the 13th International Conference on Artificial Intelligence: Methodology, Systems, and Applications*, AIMSA '08, pp. 83–92, Berlin, Heidelberg. Springer-Verlag. ISBN: 978-3-540-85775-4. doi: 10.1007/978-3-540-85776-1_8. Disponível em: <http://dx.doi.org/10.1007/978-3-540-85776-1_8>.

CHOWDHURY, A., FRIEDER, O., GROSSMAN, D., et al., 2002, "Collection Statistics for Fast Duplicate Document Detection", *ACM Trans. Inf. Syst.*, v. 20, n. 2 (abr.), pp. 171–191. ISSN: 1046-8188. doi: 10.1145/506309.506311. Disponível em: <http://doi.acm.org/10.1145/506309.506311>.

CLOUGH, P., 2000, *Plagiarism in natural and programming languages: an overview of current tools and technologies*. Relatório técnico, University of Sheffield.

CLOUGH, P., STEVENSON, M., 2011, "Developing a Corpus of Plagiarised Short Answers", *Lang. Resour. Eval.*, v. 45, n. 1 (mar.), pp. 5–24. ISSN: 1574-020X. doi: 10.1007/s10579-009-9112-1. Disponível em: <http://dx.doi.org/10.1007/s10579-009-9112-1>.

CLOUGH, P., STUDIES, D. O. I., 2003. "Old and new challenges in automatic plagiarism detection". .

CLOUGH, P., GAIZAUSKAS, R. J., PIAO, S. S., 2002, "Building and annotating a corpus for the study of journalistic text reuse". In: *LREC*.

DAVIS, M., STRANGE, B., 2002. "Elvis vs. JXL: a little less conversation". June.

DICTIONARY.COM, 2017. "plagiarism definition in dictionary.reference.com". Disponível em: <http://dictionary.reference.com/browse/plagiarism?s=t>.

DONG, W., WANG, Z., JOSEPHSON, W., et al., 2008, "Modeling LSH for Performance Tuning". In: *Proceedings of the 17th ACM Conference on Information and Knowledge Management*, CIKM '08, pp. 669–678, New York, NY, USA. ACM. ISBN: 978-1-59593-991-3. doi: 10.1145/1458082.1458172. Disponível em: <http://doi.acm.org/10.1145/1458082.1458172>.

EHSAN, N., SHAKERY, A., 2016, "Candidate document retrieval for cross-lingual plagiarism detection using two-level proximity information", *Information Processing & Management*, v. 52, n. 6, pp. 1004 – 1017. ISSN: 0306-4573. doi: http://dx.doi.org/10.1016/j.ipm.2016.04.006. Disponível em: <http://www.sciencedirect.com/science/article/pii/S0306457316300784>.

EISSEN, S., STEIN, B., 2006, "Intrinsic Plagiarism Detection". In: Lalmas, M., MacFarlane, A., Rüger, S., et al. (Eds.), *Advances in Information Retrieval*, v. 3936, *Lecture Notes in Computer Science*, pp. 565–569. Springer Berlin Heidelberg. ISBN: 978-3-540-33347-0. doi: 10.1007/11735106_66. Disponível em: <http://dx.doi.org/10.1007/11735106_66>.

EKBAL, A., SAHA, S., CHOUDHARY, G., 2012, "Plagiarism detection in text using Vector Space Model". In: *2012 12th International Conference on Hybrid Intelligent Systems (HIS)*, pp. 366–371, Dec. doi: 10.1109/HIS.2012.6421362.

FACSAR, F., 2012, "Hungary's president quits over alleged plagiarism", *Cable News Network (CNN)*, (abr.). Disponível em: <http://www.cnn.com/2012/04/02/world/europe/hungary-president-resigns/index.html>.

FRANCO-SALVADOR, M., ROSSO, P., Y GÓMEZ, M. M., 2016, "A systematic study of knowledge graph analysis for cross-language plagiarism detection", *Information Processing & Management*, v. 52, n. 4, pp. 550 – 570. ISSN: 0306-4573. doi: http://dx.doi.org/10.1016/j.ipm.2015.12.004. Disponível em: <http://www.sciencedirect.com/science/article/pii/S0306457315001417>.

GIBALDI, J., 1999, *MLA Handbook for Writers of Research Papers*. MLA Handbook for Writers of Research Papers. Modern Language Association of America. ISBN: 9780873529754. Disponível em: <http://books.google.com.br/books?id=JtVdt30MF68C>.

GIONIS, A., INDYK, P., MOTWANI, R., et al., 1999, "Similarity search in high dimensions via hashing". In: *VLDB*, v. 99, pp. 518–529.

GOMAA, W. H., FAHMY, A. A., 2013, "Article: A Survey of Text Similarity Approaches", *International Journal of Computer Applications*, v. 68, n. 13 (April), pp. 13–18. Full text available.

HARPER, D., 2001, *plagiarism definition in Online etymology dictionary*. Douglas Harper. Disponível em: <http://www.etymonline.com/index.php?term=plagiarism>.

HOAD, T. C., ZOBEL, J., 2003, "Methods for identifying versioned and plagiarized documents", *Journal of the American Society for Information Science and Technology*, v. 54, n. 3, pp. 203–215. ISSN: 1532-2890. doi: 10.1002/asi.10170. Disponível em: <http://dx.doi.org/10.1002/asi.10170>.

IFLA, 1998, *Functional Requirements for Bibliographic Records: Final Report*. K. G. Saur. Disponível em: <http://www.ifla.org/files/assets/cataloguing/frbr/frbr.pdf>.

INDYK, P., MOTWANI, R., 1998, "Approximate Nearest Neighbors: Towards Removing the Curse of Dimensionality". In: *Proceedings of the Thirtieth Annual ACM Symposium on Theory of Computing*, STOC '98, pp. 604–613, New York, NY, USA. ACM. ISBN: 0-89791-962-9. doi: 10.1145/276698.276876. Disponível em: <http://doi.acm.org/10.1145/276698.276876>.

JI, J., LI, J., YAN, S., et al., 2013, "Min-Max Hash for Jaccard Similarity". In: *2013 IEEE 13th International Conference on Data Mining*, pp. 301–309, Dec. doi: 10.1109/ICDM.2013.119.

JUOLA, P., 2006, "Authorship Attribution", *Found. Trends Inf. Retr.*, v. 1, n. 3 (dez.), pp. 233–334. ISSN: 1554-0669. doi: 10.1561/1500000005. Disponível em: <http://dx.doi.org/10.1561/1500000005>.

JUOLA, P., 2007, "Future Trends in Authorship Attribution". In: Craiger, P., Shenoi, S. (Eds.), *Advances in Digital Forensics III*, v. 242, *IFIP — The International Federation for Information Processing*, Springer New York,

pp. 119–132. ISBN: 978-0-387-73741-6. doi: 10.1007/978-0-387-73742-3_8. Disponível em: <http://dx.doi.org/10.1007/978-0-387-73742-3_8>.

K, V., GUPTA, D., 2017a, "Detection of idea plagiarism using syntax–Semantic concept extractions with genetic algorithm", *Expert Systems with Applications*, v. 73, pp. 11 – 26. ISSN: 0957-4174. doi: http://dx.doi.org/10.1016/j.eswa.2016.12.022. Disponível em: <http://www.sciencedirect.com/science/article/pii/S0957417416306960>.

K, V., GUPTA, D., 2017b, "Detection of idea plagiarism using syntaxSemantic concept extractions with genetic algorithm", *Expert Systems with Applications*, v. 73, pp. 11 – 26. ISSN: 0957-4174. doi: http://dx.doi.org/10.1016/j.eswa.2016.12.022. Disponível em: <http://www.sciencedirect.com/science/article/pii/S0957417416306960>.

KONHEIM, A. G., 2010, *Hashing in Computer Science: Fifty Years of Slicing and Dicing*. John Wiley & Sons.

KOPPEL, M., SCHLER, J., ARGAMON, S., 2009, "Computational Methods in Authorship Attribution", *J. Am. Soc. Inf. Sci. Technol.*, v. 60, n. 1 (jan.), pp. 9–26. ISSN: 1532-2882. doi: 10.1002/asi.v60:1. Disponível em: <http://dx.doi.org/10.1002/asi.v60:1>.

KUMAR, J. P., GOVINDARAJULU, P., 2009, "Duplicate and Near Duplicate Documents Detection: A Review", *European Journal of Scientific Research*, v. 32, pp. 514–527.

LESKOVEC, J., RAJARAMAN, A., ULLMAN, J. D., 2014, *Mining of massive datasets*. Cambridge University Press.

LEUNG, C.-H., CHAN, Y.-Y., 2007, "A Natural Language Processing Approach to Automatic Plagiarism Detection". In: *Proceedings of the 8th ACM SIGITE Conference on Information Technology Education*, SIGITE '07, pp. 213–218, New York, NY, USA. ACM. ISBN: 978-1-59593-920-3. doi: 10.1145/1324302.1324348. Disponível em: <http://doi.acm.org/10.1145/1324302.1324348>.

LI, P., KÖNIG, A. C., 2011, "Theory and applications of b-bit minwise hashing", *Communications of the ACM*, v. 54, n. 8, pp. 101–109.

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

MARTIN, B., 1994. "Plagiarism: a misplaced emphasis". Disponível em: <https://www.uow.edu.au/~bmartin/pubs/94jie.html>.

MAURER, H., KAPPE, F., ZAKA, B., 2006, "Plagiarism - A Survey", *Journal of Universal Computer Science*, v. 12, n. 8 (aug), pp. 1050–1084. Disponível em: <http://www.jucs.org/jucs_12_8/plagiarism_a_survey>.

MERRIAM-WEBSTER, 2017, *signal definition in merriam-webster dictionary*. Merriam-Webster. Disponível em: <http://www.merriam-webster.com/dictionary/plagiarism>.

MEUSCHKE, N., GIPP, B., 2014, "Reducing computational effort for plagiarism detection by using citation characteristics to limit retrieval space". In: *IEEE/ACM Joint Conference on Digital Libraries*, pp. 197–200, Sept. doi: 10.1109/JCDL.2014.6970168.

MIHALCEA, R., CORLEY, C., STRAPPARAVA, C., 2006, "Corpus-based and Knowledge-based Measures of Text Semantic Similarity". In: *Proceedings of the 21st National Conference on Artificial Intelligence - Volume 1*, AAAI'06, pp. 775–780. AAAI Press. ISBN: 978-1-57735-281-5. Disponível em: <http://dl.acm.org/citation.cfm?id=1597538.1597662>.

MOHAMED, H., MARCHAND-MAILLET, S., 2015, "Quantized ranking for permutation-based indexing", *Information Systems*, v. 52, pp. 163 – 175. ISSN: 0306-4379. doi: http://dx.doi.org/10.1016/j.is.2015.01.009. Disponível em: <http://www.sciencedirect.com/science/article/pii/S0306437915000198>. Special Issue on Selected Papers from {SISAP} 2013.

MUHA, D., 2003. "New Study Confirms Internet Plagiarism Is Prevalent". ago. Disponível em: <http://urwebsrv.rutgers.edu/medrel/viewArticle.html?ArticleID=3408>.

NAWAB, R. M. A., STEVENSON, M., CLOUGH, P., 2016, "An IR-based Approach Utilising Query Expansion for Plagiarism Detection in MEDLINE", *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, v. PP, n. 99, pp. 1–1. ISSN: 1545-5963. doi: 10.1109/TCBB.2016.2542803.

OSMAN, A. H., SALIM, N., BINWAHLAN, M. S., et al., 2012, "An improved plagiarism detection scheme based on semantic role labeling", *Applied Soft Computing*, v. 12, n. 5, pp. 1493 – 1502. ISSN: 1568-4946. doi: http://dx.doi.org/10.1016/j.asoc.2011.12.021. Disponível em: <http://www.sciencedirect.com/science/article/pii/S1568494612000087>.

OXFORD, 1999, *The Cambridge Dictionary of Philosophy*. Oxford University Press. Disponível em: <http://www.oxforddictionaries.com/us/definition/american_english/plagiarism>.

PAN, J., MANOCHA, D., 2012, "Bi-level Locality Sensitive Hashing for k-Nearest Neighbor Computation". In: *2012 IEEE 28th International Conference on Data Engineering*, pp. 378–389, April. doi: 10.1109/ICDE.2012.40.

PARKER, A., HAMBLEN, J., 1989, "Computer Algorithms for Plagiarism Detection". In: *Proceedings of the 1989 IEEE Transactions on education*, IEEE Transactions on education. IEEE.

PAUL, M., JAMAL, S., 2015, "An Improved {SRL} Based Plagiarism Detection Technique Using Sentence Ranking", *Procedia Computer Science*, v. 46, pp. 223 – 230. ISSN: 1877-0509. doi: http://dx.doi.org/10.1016/j.procs.2015.02.015. Disponível em: <http://www.sciencedirect.com/science/article/pii/S1877050915000794>. Proceedings of the International Conference on Information and Communication Technologies, {ICICT} 2014, 3-5 December 2014 at Bolgatty Palace & Island Resort, Kochi, India.

PEREIRA, A. R., ZIVIANI, N., 2003, "Syntactic similarity of Web documents". In: *Proceedings of the IEEE/LEOS 3rd International Conference on Numerical Simulation of Semiconductor Optoelectronic Devices (IEEE Cat. No.03EX726)*, pp. 194–200, Nov. doi: 10.1109/LAWEB.2003.1250297.

PIDD, H., 2011, "German defence minister resigns in PhD plagiarism row", *The Guardian*, (mar.). ISSN: 0261-3077. Disponível em: <http://www.theguardian.com/world/2011/mar/01/german-defence-minister-resigns-plagiarism>.

POSNER, R. A., 2007, *The little book of plagiarism*. Pantheon.

POTTHAST, M., BARRÓN-CEDEÑO, A., EISELT, A., et al., 2010a, "Overview of the 2nd international competition on plagiarism detection". In: *In Proceedings of the SEPLN'10 Workshop on Uncovering Plagiarism, Authorship and Social Software Misuse*, a.

POTTHAST, M., STEIN, B., BARRÓN-CEDEÑO, A., et al., 2010b, "An Evaluation Framework for Plagiarism Detection". In: *Proceedings of the 23rd International Conference on Computational Linguistics (COLING 2010)*, Beijing, China, ago.b. Association for Computational Linguistics.

POTTHAST, M., EISELT, A., BARRÓN-CEDEÑO, A., et al., 2011, "Overview of the 3th International Competition on Plagiarism Detection". In: *Notebook Papers of CLEF 2011 LABs and Workshops (CLEF-2011)*.

POTTHAST, M., HAGEN, M., GOLLUB, T., et al., 2013, "Overview of the 5th international competition on plagiarism detection". In: *CLEF Conference on Multilingual and Multimodal Information Access Evaluation*, pp. 301–331. CELCT.

SALTON, G., WONG, A., YANG, C. S., 1975, "A Vector Space Model for Automatic Indexing", *Commun. ACM*, v. 18, n. 11 (nov.), pp. 613–620. ISSN: 0001-0782. doi: 10.1145/361219.361220. Disponível em: <http://doi.acm.org/10.1145/361219.361220>.

SAMUELSON, P., 1994, "Self-plagiarism or Fair Use", *Commun. ACM*, v. 37, n. 8 (ago.), pp. 21–25. ISSN: 0001-0782. doi: 10.1145/179606.179731. Disponível em: <http://doi.acm.org/10.1145/179606.179731>.

SEO, J., CROFT, W. B., 2008, "Local Text Reuse Detection". In: *Proceedings of the 31st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '08, pp. 571–578, New York, NY, USA. ACM. ISBN: 978-1-60558-164-4. doi: 10.1145/1390334.1390432. Disponível em: <http://doi.acm.org/10.1145/1390334.1390432>.

SIMOVICI, D. A., DJERABA, C., 2014, *Mathematical Tools for Data Mining: Set Theory, Partial Orders, Combinatorics*. Advanced information and knowledge processing. Springer-Verlag London. ISBN: 978-1-4471-6406-7,978-1-4471-6407-4,1447164075,1447164067.

SLANEY, M., CASEY, M., 2008, "Locality-sensitive hashing for finding nearest neighbors [lecture notes]", *Signal Processing Magazine, IEEE*, v. 25, n. 2, pp. 128–131.

SOLEMAN, S., PURWARIANTI, A., 2014, "Experiments on the Indonesian plagiarism detection using latent semantic analysis". In: *2014 2nd International Conference on Information and Communication Technology (ICoICT)*, pp. 413–418, May. doi: 10.1109/ICoICT.2014.6914098.

STAFF, AGENCIES, 2013, "German education minister quits over PhD plagiarism", *The Guardian*, (fev.). ISSN: 0261-3077. Disponível em: <http://www.theguardian.com/world/2013/feb/09/german-education-minister-quits-phd-plagiarism>.

STAMATATOS, E., 2009a, "Intrinsic Plagiarism Detection Using Character n-gram Profiles". In: *In: 3rd PAN Workshop. Uncovering Plagiarism, Authorship and Social Software Misuse*, pp. 38–46, a.

STAMATATOS, E., 2009b, "A Survey of Modern Authorship Attribution Methods", *J. Am. Soc. Inf. Sci. Technol.*, v. 60, n. 3 (mar.), pp. 538–556. ISSN: 1532-2882. doi: 10.1002/asi.v60:3. Disponível em: <http://dx.doi.org/10.1002/asi.v60:3>.

STEIN, B., EISSEN, S., 2007, "Intrinsic Plagiarism Analysis with Meta Learning". In: *Proceedings of Workshop on plagiarism analysis, authorship identification and near-duplicated detection*, SIGIR '07.

STEIN, B., ZU EISSEN, S. M., POTTHAST, M., 2007, "Strategies for Retrieving Plagiarized Documents". In: *Proceedings of the 30th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '07, pp. 825–826, New York, NY, USA. ACM. ISBN: 978-1-59593-597-7. doi: 10.1145/1277741.1277928. Disponível em: <http://doi.acm.org/10.1145/1277741.1277928>.

SUREDA, J., COMAS, R., 2008, "Academic Cyberplagiarism: Tracing the causes to reach solutions", *The e-journal produced by the UOC's Languages and Cultures, and Humanities Departments*. ISSN: 1575-2275.

SZMIT, R., 2013, "Locality Sensitive Hashing for Similarity Search Using MapReduce on Large Scale Data". In: Kłopotek, M. A., Koronacki, J., Marciniak, M., et al. (Eds.), *Language Processing and Intelligent Information Systems: 20th International Conference, IIS 2013, Warsaw, Poland, June 17-18, 2013. Proceedings*, pp. 171–178, Berlin, Heidelberg, Springer Berlin Heidelberg. ISBN: 978-3-642-38634-3. doi: 10.1007/978-3-642-38634-3_19. Disponível em: <http://dx.doi.org/10.1007/978-3-642-38634-3_19>.

TAYLOR, F. K., 1965, "Cryptomnesia and Plagiarism", *The British Journal of Psychiatry*, v. 111, n. 480, pp. 1111–1118. doi: 10.1192/bjp.111.480.1111. Disponível em: <http://bjp.rcpsych.org/content/111/480/1111.abstract>.

THOMPSON, V. U., PANCHEV, C., OAKES, M., 2015, "Performance evaluation of similarity measures on similar and dissimilar text retrieval". In: *2015 7th International Joint Conference on Knowledge Discovery, Knowledge Engineering and Knowledge Management (IC3K)*, v. 01, pp. 577–584, Nov.

VANI, K., GUPTA, D., 2014, "Using K-means cluster based techniques in external plagiarism detection". In: *2014 International Conference on Contemporary Computing and Informatics (IC3I)*, pp. 1268–1273, Nov. doi: 10.1109/IC3I.2014.7019659.

VIEIRA, D. C., 2016, *Abordagens de Técnicas de LSH Aplicadas ao Problema de Similaridade de Documentos*. Tese de Mestrado, Programa de Engenharia de Sistemas, COPPE, UFRJ, http://www.cos.ufrj.br/index.php/pt-BR/publicacoes-pesquisa/details/15/2600, 2.

WANG, J., LIU, W., KUMAR, S., et al., 2015, "Learning to Hash for Indexing Big Data - A Survey", *CoRR*, v. abs/1509.05472. Disponível em: <`http://arxiv.org/abs/1509.05472`>.

WEBER-WULFF, D., 2010, "Test Cases for Plagiarism Detection Software". In: *In Proceedings of the 4th International Plagiarism Conference, Newcastle upon Tyne, UK, 2010*.

WIKIMEDIA, 2016, *Wikipedia statistics All languages*. Relatório técnico. Disponível em: <`https://stats.wikimedia.org/EN/TablesWikipediaZZ.htm`>.

ZHANG, H., CHOW, T. W., 2011, "A coarse-to-fine framework to efficiently thwart plagiarism", *Pattern Recognition*, v. 44, n. 2, pp. 471 – 487. ISSN: 0031-3203. doi: http://dx.doi.org/10.1016/j.patcog.2010.08.023. Disponível em: <`http://www.sciencedirect.com/science/article/pii/S0031320310004097`>.

ZHANG, W., LI, D., XU, Y., et al., 2016, "Shuffle-efficient distributed Locality Sensitive Hashing on spark". In: *2016 IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*, pp. 766–767, April. doi: 10.1109/INFCOMW.2016.7562179.