USANDO PERMUTATION BASED INDEXING NA DETECÇÃO DE PLÁGIO

Hugo Diniz Rebelo

Dissertação de Mestrado 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 Mestre em Engenharia de Sistemas e Computação.

Orientador: Geraldo Bonorino Xexéo

Rio de Janeiro
Setembro de 2019

USANDO PERMUTATION BASED INDEXING NA DETECÇÃO DE PLÁGIO

Hugo Diniz Rebelo

DISSERTAÇÃO 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 MESTRE EM CIÊNCIAS EM ENGENHARIA DE SISTEMAS E COMPUTAÇÃO.

Examinada por:

Prof. Geraldo Bonorino Xexéo, D.Sc.

Prof. Geraldo Zimbrão da Silva, D.Sc.

Prof. Leandro Guimarães Marques Alvim, D.Sc.

RIO DE JANEIRO, RJ – BRASIL
SETEMBRO DE 2019

Rebelo, Hugo Diniz

Usando Permutation Based Indexing na Detecção de Plágio/Hugo Diniz Rebelo. – Rio de Janeiro: UFRJ/COPPE, 2019.

XII, 78 p.: il.; 29, 7cm.

Orientador: Geraldo Bonorino Xexéo

Dissertação (mestrado) – UFRJ/COPPE/Programa de Engenharia de Sistemas e Computação, 2019.

Referências Bibliográficas: p. 63 – 71.

1. Plágio. 2. Permutation Based Indexing. 3. Recuperação de informação. I. Xexéo, Geraldo Bonorino. II. Universidade Federal do Rio de Janeiro, COPPE, Programa de Engenharia de Sistemas e Computação. III. Título.

Em memória de José Francisco Diniz.

Agradecimentos

Primeiramente gostaria de agradecer todo o corpo docente do Programa de Engenharia de Sistemas e Computação(PESC), em especial ao prof. Geraldo Xexéo que sempre teve a total atenção, compreensão e paciência em todo na orientação deste trabalho, Gostaria de agradecer ao Prof. Fellipe Duarte, que muito me ajudou e me orientou para a conclusão desse trabalho, de tal forma que o considero meu co-orientador. Também não poderia deixar de mencionar todo corpo docente do curso de Ciência da Computação da UFRRJ que fez parte da minha formação e que me possibilitou a chegar ao final desta etapa.

Gostaria de dedicar essa vitoria aos meus pais João Francisco e Maria das Graças que devo absolutamente tudo que tenho na minha vida, são as minhas referências, foram graças ao apoio deles que conseguir passar por todas as etapas que me levaram a concluir este trabalho. Ao meu avô José Francisco que não pode vivenciar mas esse fim de ciclo, meus avós Aldevina, Rosa, avós que são como meus pais mais velhos, aos meus tios José, Elísia, Helena, Regina, Paulo, Sílvia e Sílvio que também são como meus pais, meus primos que considero todos meus irmãos, os que cresceram comigo como Lucas e Gabriel, os que hoje são meus vizinhos como a Márcia e Pedro, esse por sinal é meu primo, vizinho e ainda é colega de profissão. Minha prima Juliana, uma pessoa que cresceu junto comigo, passamos pelas mesmas fases da vida juntos, ela é minha irmã gêmea que nasceu alguns dias antes. Minhas primas Mariana e Marina que de distante só as nossas casas.

Não poderia deixar de agradecer aos meus amigos, que sempre estiveram junto comigo, aos meus amigos do Ferreira Viana, Luana, Daniel , Michael, Jessica, Kevyn, Jonathan, amigos tive o prazer conviver durante aqueles belos anos de ensino médio. Ao Kleyton, Raul, Michel, Egberto, Ygor e Julio, amigos que fiz no curso de Ciência da computação, e que permanecemos juntos até a pós graduação PESC. A Luiza, que apesar de não de ser do mesmo curso, foi extremamente importante nesta minha etapa. A Danielle, Victor e o Pedro Henrique, amigos que muito me fizeram crescer como pessoa e como profissional, nos anos que estivemos juntos na COPPETEC.

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

USANDO PERMUTATION BASED INDEXING NA DETECÇÃO DE PLÁGIO

Hugo Diniz Rebelo

Setembro/2019

Orientador: Geraldo Bonorino Xexéo

Programa: Engenharia de Sistemas e Computação

A identificação de plágio Extrínseco é um processo de avaliação de um documento, no qual analisamos o seu conteúdo em busca de um possível plágio comparando de forma direta com possíveis documentos fontes. A identificação de plágio Extrínseco pode ser dividido em três etapas, A Busca Heurística, Analise Detalhada e o Pós-processamento.

Neste trabalho iremos focar na etapa da Busca Heurística, e para isso utilizaremos a abordagem *Permutation Based Indexing* (PBI), que foi proposta como uma nova abordagem para o cálculo de similaridade entre objetos, tendo como diferencial a redução da quantidade de comparações no *dataset*, comparando a consulta somente com os objetos *pivots*, que são objetos do próprio *dataset* escolhidos na etapa de seleção de *pivot*, com a ideia de escolher os objetos que melhor representam o *dataset* como todo. Além da utilização da técnica do PBI, o trabalho terá como uma agregação de valor a criação de variações das técnicas já existentes de *pruning*, baseada numa "poda"dos *pivots*, que retira *pivots* que não tenham muita influência em uma determinada consulta.

Abstract of Dissertation presented to COPPE/UFRJ as a partial fulfillment of the requirements for the degree of Master of Science (M.Sc.)

USING PERMUTATION BASED INDEXING IN PLAGIARISM DETECTION

Hugo Diniz Rebelo

September/2019

Advisor: Geraldo Bonorino Xexéo

Department: Systems Engineering and Computer Science

Extrinsic text plagiarism detection is a document evaluation process, in which we analyze its content for possible plagiarism by comparing directly with potential source documents. The identification of extrinsic plagiarism can be divided into three stages, Heuristic Retrieval, Detailed Analysis and Postprocessing.

This work will focus on the Heuristic Retrieval stage, and for that we will use the Permutation Based Indexing(PBI) approach, which was proposed as a new approach to the calculation of similarity between objects, having as a differential the reduction of the number of comparisons in the dataset, comparing the query only with the pivots objects, which are objects of the dataset itself chosen in the pivot selection step, considering to choose the objects that best represent the dataset as a whole. In addition to using the PBI technique, to add value to this work, this work will create variations on existing pruning techniques, based on a "pruning" of pivots, which removes pivots that do not have much influence on a given query.

Sumário

Lista de Figuras

Lista de Tabelas

Capítulo 1

Introdução

1.1 Problema e Motivação

O plágio pode ter diversas formas distintas de ser ocorrer, variando desde copiar textos sem nenhuma alteração, até adotando as ideias de um determinado trecho do texto sem dar o devido crédito ao autor do documento original (ALZAHRANI *et al.*, 2012). É com a necessidade da detecção dessas fraudes, que podem ser cometidas de forma voluntaria ou não, nasce a ideia da identificação de plágio, que pode ser caracterizado como um processo de avaliar um documento qualquer, com o objetivo de localizar possíveis trechos plagiados de outros documentos (MARTIN, 1994).

Quando o plágio ocorre em forma de texto, existem diversas abordagens na área de Recuperação da informação para esse fim, cada uma delas com o objetivo de resolver um problema da detecção de plágio como todo (BARRÓN-CEDEÑO, 2010; MARTIN, 1994). Uma dessas formas de detecção de plágio, é o processo de detecção de plágio extrínseco, que pode ser caracterizado como a detecção de plágio quando temos acesso aos documentos fontes do possível plágio, e é nessa forma detecção de plágio que esse trabalho se focará, mas exatamente na etapa de Busca heurística na detecção de plágio extrínseco.

A etapa de busca heurística pode ser considerada um filtro de documentos para o restante do processo de detecção de plágio extrínseco, afim que seja possível que um sistema de detecção de plágio responda de forma rápida uma consulta onde a copia se encontra entre as milhões do *dataset* (ZHANG & CHOW, 2011). Para isso se faz um rankeamento afim de filtrar a maior quantidade possível de documentos, e uma das principais formas de se fazer isso é através da medição de similaridade entre documentos (BARRÓN-CEDEÑO, 2010).

A abordagem para a resolução da busca Heurística na qual esse trabalho se focará é o *Permutation Based Indexing*(PBI), abordagem na qual tem chamado a atenção na área de busca por similaridade (AMATO *et al.*, 2015), tendo a ideia básica a

representação do *Dataset* num conjunto de *pivots*, e partindo desse conjunto de *pivots* faz-se a consulta por similaridade entre dois documentos distintos. Como o PBI não utiliza todo o *dataset* para a comparação e sim somente o conjunto de *pivots*, a tendencia é que ele atue de maneira mais eficiente para o tempo de consulta quando comparado a outras técnicas que utilizam o *dataset* como todo (AMATO *et al.*, 2015).

O PBI pode ser separado em 2 partes distintas, a indexação e a consulta. Na parte da indexação podemos sub-dividir em outras duas partes, na transformação do espaço métrico, na qual tem por objetivo transformar cada documento do *dataset* em um espaço aonde seja possível a comparação de distâncias. A seleção de *pivots*, na qual se selecionará n *pivots* mais representativos do *dataset*. A parte da consulta podemos sub-dividir em 3 etapas, para cada documento ordenaremos de forma crescente todos os *pivots*, após isso faremos o *pruning* na qual se selecionará para cada busca de um documento k *pivots* mais similares, afim de retirar os *pivots* que não sejam interessantes na consulta e por fim faremos comparação dois a dois a listas de ordenadas de *pivots* que receberam o *pruning*, no qual chamaremos de calculo de distância entre objetos.

1.2 Objetivo

Este trabalho aplicará a abordagem de busca por similaridade *Permutation Based Indexing*, na etapa de busca heurística para a detecção de plágio extrínseco e apresentará três novas abordagens de quantificação na etapa de comparação dois a dois da abordagem *Permutation Based Indexing*. A função de quantificação é uma importante função no calculo de distância de objetos pois com a permutação perdemos a garantia que nas listas ordenadas de *pivots* teremos os mesmos *pivots*, perdendo a qualidade no calculo.

As 3 novas abordagens, o *Document Quantized Ranking*, *Query Quantized Ranking* e *Fixed Quantized Ranking* são abordagens inspiradas na abordagem *Quantized Ranking* desenvolvida pelo MOHAMED & MARCHAND-MAILLET (2015), na qual abordagem *Quantized Ranking* quantificará os *pivots* no qual em alguma das duas listas. A partir da *Quantized Ranking*, o trabalho propõem o *Document Quantized Ranking*, no qual quantifica somente para a lista ordenada do documento, o *Query Quantized Ranking*, no qual quantifica somente para a lista ordenada da consulta e o *Fixed Ranking*, no qual quantifica a diferença das posições na lista das comparações aonde não exista um dos *pivots*.

Ao longo do trabalho iremos desenvolver os possíveis cenários que ocorrem ao se utilizar o *pruning* no *Permutation Based Indexing*. Após o desenvolvimento do cenários, iremos avaliar de forma empírica o *Permutation Based Index* avaliando

as técnicas de quantificação assim como o *Permutation Based Index* como todo em comparação com duas técnicas já utilizadas na detecção de plágio, o *BM-25* e o *Locality Sensitive Hashing* (DUARTE, 2017; ROBERTSON *et al.*, 1993).

1.3 Estrutura do Trabalho

Este trabalho está organizado em 6 capítulos, o capítulo 1, descreve a motivação, o problema e o objetivo do trabalho. O capítulo 2 tratará da Revisão da literatura acerca do plágio, fazendo sua definição formal, suas formas de aparição, formas de detecção focando na detecção de plágio extrínseco e finalizando com os trabalhos relacionados. O capítulo 3 aborda o *Permutation Based Indexing* apresentando abordagem e as técnicas relacionados a ela. O capítulo 4 é referente a proposta aonde definimos as abordagens de quantificação *Document Quantized Ranking*, *Query Quantized Ranking* e *Fixed Quantized Ranking*, comparando elas com a abordagem de quantificação *Quantized Ranking*. capítulo 5 apresentamos os experimentos e concluímos o trabalho no capítulo 6. No Apêndice A apresentaremos de forma integral todos os resultados dos experimentos em uma tabela.

Capítulo 2

Revisão da Literatura

2.1 Plágio

O plágio na definição de "expropriação da propriedade intelectual", é conhecido há um longo tempo (MAURER *et al.*, 2006), sendo ele um assunto que é visto e estudado em diversas áreas de pesquisa (DUARTE, 2017), tendo que o ato de plagiar pode ser praticado em diversas formas distintas e que cada uma dessas formas podemos compreende-la utilizando a suas características. De uma forma geral, podemos descrever o plágio segundo a *Oxford Dictonary* como o uso de informações, ideias ou expressão de uma outra pessoa para adquirir algum tipo de vantagem, como por exemplo melhores notas, ainda segundo *Oxford Dictonary* a palavra Plágio tem origem do latim *plagiaruis* que pode ser traduzido como "sequestrador, sedutor, saqueador, aquele que sequestra a criança ou escravo de outro"e este termo foi inicialmente utilizado pelo poeta *Martial* no seculo XVII com sentido de "roubo literário"(STEVENSON, 2010). Podemos apontar uma lista de sinônimos da palavra plágio apresentadas na *Oxford Dictonary* : cópia, violação de direitos autorais, pirataria, roubo, furto e apropriação (STEVENSON, 2010).

Segundo o MAURER *et al.* (2006), a fronteira entre o que é plagio ou não é muito nebulosa, um exemplo disso é o que pode ser considerado pesquisa e o que é simplesmente plágio. Em áreas como literárias, judiciarias, um artigo acadêmico está presente centenas de citações de outras fontes para criticar uma tese. Já em áreas como as de engenharia, é preciso citar somente a literatura da área para introduzir os leitores e a após isso demonstrar seu trabalho (MAURER *et al.*, 2006).

Uma importante característica do plágio que também torna a identificação do mesmo algo não trivial, é que a definição de autoria em uma sociedade em um determinado tempo de sua historia são diretamente influenciados por características sociais, econômicas e politicas COMAS & SUREDA (2008). Um exemplo abordado pelo HAYES & INTRONA (2005), é que estudantes Chineses se concentram no

conteúdo de apenas um livro didático e como PENNYCOOK (1996) demonstrou em seu trabalho, para estudantes chineses, utilizar as palavras que o autor utilizou é uma forma de respeito. Isso Consequentemente gera uma dificuldade de demonstrarem sua opinião critica sem que eles cometam plagio pelas definições já aqui citadas.

Além disso a intencionalidade do plágio nem sempre pode existir, o desconhecimento do que realmente é considerado plágio pode levar uma pessoa a comete-lo por exemplo, segundo PERRY (2001) é muito comum a confusão entre o plágio e a paráfrase, e em sua pesquisa ele cita que alguns alunos acreditam que, ao copiar de fontes diversas e depois mescla-lá, isto não se caracteriza plágio e sim pesquisa. Outro exemplo que vale apresentar é a da *cryptomnesia* definida por TAYLOR (1965) como: "A existência de memorias escondidas de sua consciência", essas memorias não são reconhecidas por memorias e sim por algo criado recentemente por si. A intencionalidade do plágio pode ser caracterizada segundo MAURER *et al.* (2006) como:

1. **Intencional**:
    - Um ato doloso de copiar um material de forma completa ou partes do mesmo sem dar o devido crédito ao autor da obra.
2. **Não-Intencional**:
    - Graças a imensa quantidade de informação disponível, ideias igual podem surgir por meio de expressões faladas ou escritas como se fossem originais sem que o autor do plágio saiba da já existência da ideia.
3. **Acidental**:
    - Plágio cometido devido à uma falta de conhecimento sobre plágio e/ou compreensão dos padrões de citação praticados em um determinado instituto.

A característica da intencionalidade do plágio influência diretamente como o ato do plagiar ocorre, e a partir destes atos podemos identificar as características em comum, sendo que podemos listar essas as características mais comuns segundo foi DUARTE (2017); MAURER *et al.* (2006); POTTHAST *et al.* (2011a), como:

1. **Copiar literal**:
    - É a prática de plágio simples e comum de ser feita MAURER *et al.* (2006); POTTHAST *et al.* (2011a). Definida por MARTIN (1994) copia de palavra por palavra sem nenhum tipo de reconhecimento de fonte.
2. **Plágio de fontes secundárias**:

- Foi definido pelo MARTIN (1994) como o autor cita a fonte original do trabalho mas não cita a fonte secundaria, local aonde ele realmente retirou aquele trabalho.

3. **Plágio de código fonte**:

   - O plágio de código foi definido por PARKER & HAMBLEN (1989) como um *software* que foi produzido a partir de outro, com pequenas alterações em seu código fonte, normalmente substituições de texto e outros detalhes que não requerem uma compreensão do código fonte em si.

4. **Plágio de ideia**:

   - Definido pelo MARTIN (1994) e posteriormente por MAURER *et al.* (2006) como um conceito ou ideia original de outro autor é utilizado, mas sem qualquer tipo de reconhecimento. O conceito de plágio de ideias é um conceito mais geral que pode abranger outros atos de plagio.

5. **Plágio Artístico**:

   - É definido por MAURER *et al.* (2006) como "apresentação de um trabalho de outra pessoa usando mídias diferentes, como texto, imagens, voz ou vídeo."

6. **Plágio de tradução**:

   - Um plágio de tradução são casos de tradução de um texto de uma língua para outra, sendo depois integrado como uma escrita do autor do plágio POTTHAST *et al.* (2011a).

7. **Paráfrase**:

   - Paráfrase é uma reformulação do significado de um texto utilizando outras palavras MERRIAM-WEBSTER ONLINE (2009). O plágio por paráfrase ocorre quando não ocorre a citação do autor original.

     Exemplo de Paráfrase STUDY.COM (2017):

     - *"Her life spanned years of incredible change for women."*
     - *"Mary lived through an era of liberating reform for women."*

8. **Coleções misturadas**:

   - Técnica que copia-se parágrafos de diversas fontes distintas, randomicamente os ordena gerando um plagiado novo. WEBER-WULFF (2010)

9. **Auto Plágio**:

- O auto plágio ocorre quando o autor utiliza seus trabalhos produzidos e publicados anteriormente em seu atual trabalho sem as devidas citações.(MAURER *et al.*, 2006)

10. **GhostWriting**:

    - O *ghostwriting* diferente do plágio normal, o autor,conhecido nessa pratica como "fantasma"está voluntariamente escrevendo para uma outra pessoa receber o crédito. O "fantasma"consente com o uso de seu trabalho por outro e permite que o outro represente o trabalho como seu (MAURER *et al.*, 2006).

2.2 Plágio em texto

Com advento da internet e consequentemente a facilitação da criação e disponibilidade de grandes quantidades de texto, o plágio tornou-se muito simples LOSE (2011). Uma pesquisa apresentada por MCCABE *et al.* (2001) com 18.000 alunos demonstraram que cerca de 50% dos alunos admitiram já ter plagiado documentos estranhos. Segundo MAURER *et al.* (2006), ocorreu um aumento 10% para 40% dos estudantes que já admitiram ter cometido plágio nas pesquisas conduzidas pelo MCCABE (2005) em 1999 e 2005. Uma outra pesquisa, agora feita pelo CALDWELL (2010) ele aponta para 98% dos alunos já admitiram ter cometido plágio demonstrando a magnitude do problema nas escolas e academias.

Segundo MAURER *et al.* (2006), o plágio é umas das mais serias má condutas em escolas e academias, aonde muitos professores estão atuando ativamente, oferecendo guias, tutorias para demonstrar e explicar o plágio e como evita-lo. Apesar de ser umas das mais serias má conduta praticadas ativamente pelos alunos, existem fatores do sistema educacional que influenciam o aluno a cometer o plágio (COMAS & SUREDA, 2008), fatores esses que que também devem ser combatidos pelas escolas e academias. Esses fatores foram agrupados em 3 áreas pelo BARRÓN-CEDEÑO (2010).

1. Problemas existentes nas estrategias de ensino e na forma aonde os professores atribuem tarefas aos aluno;

2. Problemas existentes na relação do aluno, instituição de ensino e a visão do aluno ao processo de aprendizagem e

3. Problemas na falta de uma politica clara e concisa por parte da instituição educacional em relação ao plágio.

Algumas instituições educacionais responderam a esses fatores com campanhas e formas politicas de combate ao plágio, como por exemplo a criação de um códigos de honra formal implementada por algumas universidades segundo MAURER *et al.* (2006); PARK (2003) com a intenção de enfatizar valores sociais PARK (2003) e possíveis punições ao autor do plágio. Outras universidades como *Stanford* PARK (2003) e *Massey University* GODDARD & RUDZKI (2005) começaram a utilizar e *softwares* para a detecção de plágio.

Softwares para detecção de plágio se tornam cada vez mais importante e mais efetivos na sua tarefa, já que o plágio em versão eletrônica(também definido na literatura por plágio digital)(DUARTE, 2017; GODDARD & RUDZKI, 2005; PARK, 2003) é problemático enquanto facilita a a copia de diversos materiais, ele facilita a na comparação via softwares e possibilitam que esses softwares tenham uma gigantesca base de dados comparativas PARK (2003).

2.3 Identificação do plágio em texto

A identificação do plágio é um processo de avaliar o documento, analisando o seu conteúdo, revelando partes de texto que podem ter sido plagiadas trazendo os possíveis documentos plagiados, caso eles estejam disponíveis ao usuário (ALZAHRANI *et al.*, 2012). E existem diversas formas de realizar esta avaliação a partir do que diversos autores chamam de taxonomia do plágio (ALZAHRANI *et al.*, 2012; CESKA *et al.*, 2008; DUARTE, 2017; MAURER *et al.*, 2006; PARK, 2003).

Como exemplificado na figura 2.1, a taxonomia do plágio em texto pode ser dividida em 2 grupos, o plágio Inteligente e o Literal. Segundo ALZAHRANI *et al.* (2012), o plágio literal é uma pratica comum no qual, quem cometeu o autor do plágio não gasta muito tempo e recursos escondendo o delito. Normalmente o plágio literal ocorrem copias literais do texto plagiados de forma inteira ou parte dele, no máximo pequenas alterações como junções ou divisões de sentenças,reordenamento de frase (MAURER *et al.*, 2006).

Já o plágio inteligente pode ser considerado um delito mais grave, pois houve um gasto de recursos e tempo afim de enganar o leitor. Essa pratica de tentar esconder o plágio de maneira inteligentes como manipulação de texto(Paráfrase, restruturação do texto, sumarização), tradução do texto, podendo ser feita por um tradutor automatizado, com pequenas correções (EHSAN & SHAKERY, 2016) e até mesmo na reescrita de um texto ou parte dele tomando somente a sua ideia e/ou contribuição.

Para plágio literal que consiste basicamente de copias do texto completo, ou de grande parte deles, consegue se aplicar modelos de comparação de documentos clássicos como por exemplo modelo vetorial (ALZAHRANI *et al.*, 2012; BAEZA-

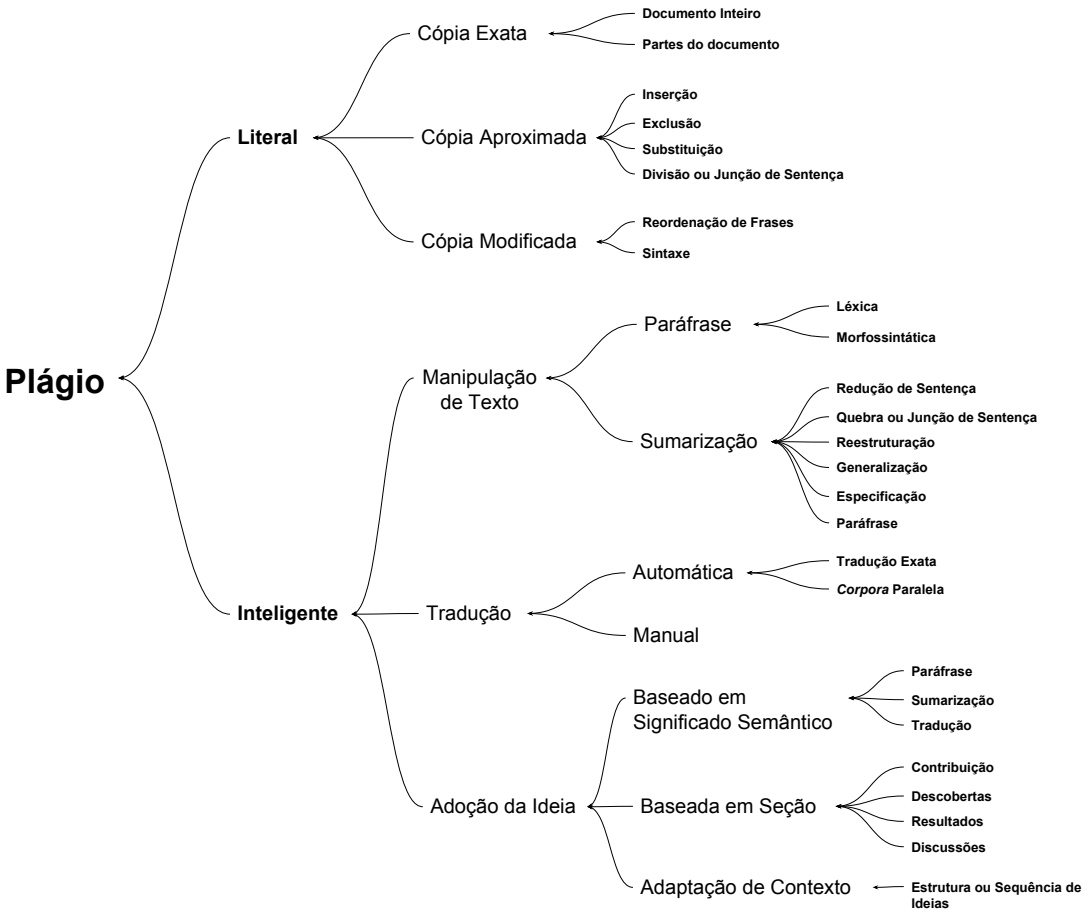

Figura 2.1 – Taxonomia do plágio, adaptada do ALZAHRANI *et al.* (2012).

YATES *et al.*, 1999; DUARTE, 2017), quando se envolve plágio inteligente no qual se traduz um documento ou se utiliza uma pequena parte do documento e alguns artifícios de transformação do texto, pode se utilizar modelos que assim como modelo vetorial, utilizam corpus de referencia. Todas essas soluções utilizam corpus de referência afim de extrair atributos destes corpus para a comparação destes documentos de maneira a auxiliar a detecção do plágio. Esses atributos são conhecidos como atributos externos ou extrínsecos e foram definidos por BARRÓN-CEDEÑO (2010) como atributos utilizados para comparar um texto com um documento suspeito com intuito de encontrar conteúdos "emprestados". Geralmente estes atributos são utilizados para localizar similaridades entres fragmentos de texto, estrutura ou até mesmo erros.

Quando na analise não é possível a utilização de um corpus de referência, se utiliza a estrategia de uma analise de estilo, que se refere a busca de atributos intrínsecos no texto. Atributos intrínsecos pode ser definido como atributos relacionados com a evolução do texto de um documento suspeito. Variações dessa evolução do texto podem ser causadas por adição de um texto de um fonte externa BARRÓN-CEDEÑO (2010).

2.3.1 Sistemas para a detecção de plágio

Mesmo que um humano consiga fazer a analise de um documento suspeito com um outro documento ou mesmo fazer uma analise da escrita daquele documento, isto requer um esforço muito grande pra fazer analise de diversos documentos e está ciente de uma quantidade enorme de fontes, que torna necessário sistemas automatizados de busca e detecção de plagio para uma detecção de plágio satisfatória dada a grande quantidade de fontes suspeitas e originais ALZAHRANI *et al.* (2012).

A detecção de plágio pode ser dividas em duas tarefas. A extrínseca e a intrínseca POTTHAST *et al.* (2009). O primeiro analisa um documento suspeito com um ou mais documentos de um corpus de referência. Já o segundo analisa somente o documento suspeito procurando variações na escrita. Essas duas tarefas podem trabalhar de forma mono-lingual, ou seja com a mesma língua de escrita ou multi-lingual, que analisa documentos escrita em diversas línguas.

O processo utilizado na detecção de plágio extrínseco multi-lingual apresentado pelo BARRÓN-CEDEÑO (2010), é muito similar ao modelo de plágio extrínseco(detalhado na subseção 2.3.1.2), tendo como adição ao modelo a forma de sistema de tradução e extração de palavras chaves pra aí então termos a busca heurística e a analise detalhada ALZAHRANI *et al.* (2012); BARRÓN-CEDEÑO (2010). Para a detecção de plágio intrínseco multi-lingual podemos também utilizar um modelo muito similar ao modelo de plagio intrínseco (definido na subseção 2.3.1.1) que foi representado na figura 2.2 com a diferença que na detecção de plágio intrínseco multi-lingual, estamos interessados em identificar uma possível tradução (BARRÓN-CEDEÑO, 2010).

2.3.1.1 Plagio intrínseco

Podemos caracterizar o plágio intrínseco como a forma de detectar o plágio procurando uma variação de sua escrita, sem a utilização de corpus de referencia, buscando inconsistência na escrita (STAMATATOS, 2009). Essas inconsistências em um documento podem ser uma mudança do vocabulário utilizado, complexidade, fluxo do texto entre outras (POTTHAST *et al.*, 2009). Para a tarefa de identificação de plágio intrínseco, o POTTHAST *et al.* (2011a) definiu um conjunto de passos representado na figura 2.2.

O primeiro passo é chamado de Segmentação, ou estratégia de segmentação ALZAHRANI *et al.* (2012); DUARTE (2017), esse passo é destinado para a divisão de um texto, em segmentos. Normalmente é utilizada uma técnica chamada n-gramas, no qual se divide o texto em tamanho iguais, e com sobreposição de palavras.

A partir dos segmentos de texto já separados passamos a etapa do Modelo de recuperação de escrita, que é a etapa aonde mapeamos os segmentos e quantificamos

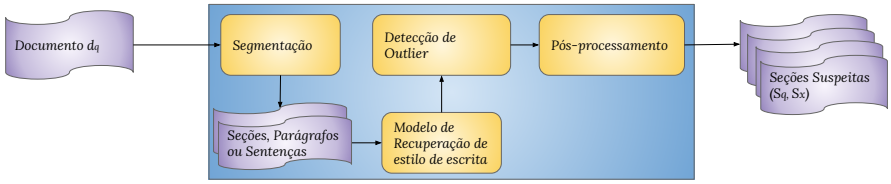

Figura 2.2 – Representação da detecção de plágio intrínseco, adaptado do ALZAHRANI *et al.* (2012).

o estilo de escrita daquele texto, podemos nomear essa quantificação como medidas estilométricas BARRÓN-CEDEÑO (2010). A maioria das medidas estilométricas podem ser classificadas em 5 categorias segundo ZU EISSEN & STEIN (2006).

1. Estatísticas de texto, extraídas a nível de palavras;
2. Características sintáticas, que medem o estilo de escrita no nível da sentença;
3. parte recursos de fala para quantificar o uso de classes de palavras;
4. conjuntos de palavras de classe fechada para contar palavras especiais e
5. características estruturais, características da organização do texto.

2.3.1.2 Plagio Extrínseco

Segundo o autor VANI & GUPTA (2014), o processo de identificação de plágio extrínseco tem como principio, que os documentos fontes estão todos disponíveis e acessíveis, isto significa que se um documento d_q contém algum plágio, a fonte original do trecho plagiado está contido em um ou mais documentos de um conjunto D que se tem acesso. A partir disto podemos separar em 3 formas na qual um método de detecção de plágio extrínseco pode atuar. Na análise de estilo, no qual faz-se a análise de escrita de cada pessoa, busca do documento suspeito por toda sua coleção de documentos em buscas e a terceira na forma de qual se busca por fragmentos de texto ao invés de todo seu documento. A partir disto o autor ALZAHRANI *et al.* (2012); DUARTE (2017) separaram a tarefa de identificação de plágio em 3 etapas, A busca heurística, analise detalhada e pós-processamento, como exemplificado na figura 2.3.

A ultima etapa e mais simples de ser descrita, a etapa de pós processamento tempo por objetivo tornar os resultados obtidos na etapa de analise detalhada legível para o ser humano. Ou seja ela pode ser entendida como a etapa aonde o sistema demonstra de forma visual, em que trecho do texto houve o plágio e qual foi o trecho do documento foi plagiado (ALZAHRANI *et al.*, 2012).

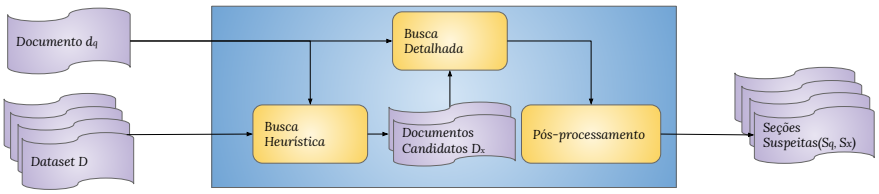

Figura 2.3 – Representação da detecção de plágio extrínseco, adaptado do ALZAHRANI *et al.* (2012).

A analise Detalhada, pode ser descrita como uma importante etapa da detecção de plágio extrínseco, no qual tem por objetivo comparar as passagens d_q de um documento suspeito com um *dataset* de documentos possivelmente plagiados, utilizando abordagens de detecção de plágio inteligente (DUARTE, 2017), no qual são consideradas abordagens computacionalmente custosas, demandando um tempo alto de execução para a comparação.

É a partir da dificuldade de se identificar plágio de forma eficiente, respondendo de rápida uma consulta onde a copia se encontra entre as milhões do *dataset*, sendo que além da quantidade massiva de documentos que um *dataset* pode ter, cada documento tem em média centenas de palavras (ZHANG & CHOW, 2011), que utilizamos a etapa de Busca Heurística, também nomeada de recuperação heurística (DUARTE, 2017) e representada na figura 2.4. Podemos definir a etapa de busca heurística na detecção de plágio extrínseco como uma etapa na qual tem o objetivo de reduzir o espaço de comparação de documentos na etapa de Analise Detalhada, retirando os falsos positivos, assim reduzindo o *dataset* a ser comparado Analise detalhada.

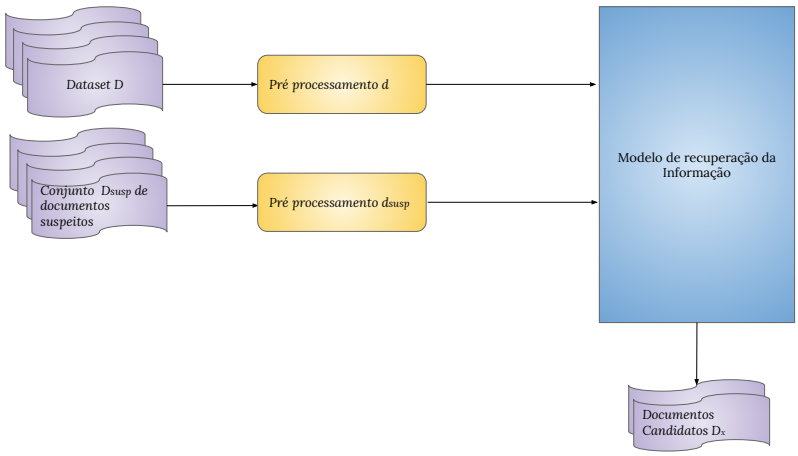

Figura 2.4 – Representação da busca Heurística, adaptada de POTTHAST *et al.* (2013)

.

Formalmente representamos a etapa de busca heurística como uma tupla $[D, D_{susp}, F_r, R(d_q, d_j)]$, no qual D é o *dataset* dos documentos, D_{susp} é o conjunto de consultas(documentos suspeitos), F_r é a forma de representação dos documentos e consultas e $R(d_q, d_j)$ é a função de rankeamento que avalia qual a probabilidade de $d \in D$ ser um documento fonte para $d_q \in D_{susp}$ (BAEZA-YATES *et al.*, 1999; DUARTE, 2017).

Para a função de rankeamento, podemos utilizar abordagens fazem a busca por similaridade (DUARTE, 2017), na qual apresentaremos seus conceitos nas próximas subseções, juntamente com a seção 2.5 trabalhos relacionados.

2.4 Busca por similaridade

A busca por similaridade foi estabelecida como um paradigma para diversas aplicações modernas CHÁVEZ & NAVARRO (2008), como a detecção de reuso de texto, detecção de plágio, reconhecimentos de padrões, reconhecimento de biometria entre outras FISCHETTI1Ú *et al.* (2014). Podemos definir segundo o autor PATELLA & CIACCIA (2009) sendo objetivo de encontrar, em um *dataset*, aqueles objetos que são mais similares a um determinado objeto. Sendo que esta similaridade pode ser avaliada por uma função de distância $F_d(x, y)$, no qual valores baixos de distância correspondem a altos graus de similaridade.

Para seguirmos no contexto da busca por similaridade é importante primeiro definirmos o conceito de espaço métrico pois as técnicas que falaremos a seguir se baseiam no conceito de um espaço métrico. Então podemos definir um espaço métrico como um par (X, d), onde X é o conjuntos de objetos $d : XxX \rightarrow \mathbb{R}^+$, no qual d é uma função de distancia que satisfaz as 4 propriedades. A primeira é a simetria no qual $d(x, y) = d(y, x)$, a reflexividade no qual $d(x, x) = 0$, a positividade $d(x, y) > 0 \Leftrightarrow x \neq y$, e por fim a desigualdade triangular, $d(x, y) \leqslant d(x, z) + d(z, y), \forall x, y, z \in X$ (CHÁVEZ *et al.*, 2005).

$$\begin{aligned} d(x, y) &= d(y, x) \\ d(x, x) &= 0 \\ d(x, y) > 0 &\Leftrightarrow x \neq y \\ d(x, y) \leqslant d(x, z) &+ d(z, y), \forall x, y, z \in X \end{aligned} \tag{2.1}$$

Na busca por similaridade, podemos considerar segundo o autor PATELLA & CIACCIA (2009), dois tipos muito comuns de consulta. A consulta por intervalo, no qual a partir de um limite de distância a qualquer, retorne qualquer objeto pertencente ao *dataset* X no qual $F_d(x, y) \leqslant a$ sendo que $x, y \in X$. A outra consulta

por similaridade muito comum é o *K-Nearest Neighbors*(*K-NN*), onde são retornados os k objetos mais similares pertencente ao conjunto X, de um objeto qualquer. Este trabalho se focará no estudo do *K-NN* e suas variações, principalmente porque segundo o autor PATELLA & CIACCIA (2009), o usuário é capaz de controlar a seletividade da consulta, ou seja, a cardinalidade do conjunto de resultados.

2.4.1 K-Nearest Neighbors

O clássico algorítimo *K-Nearest Neighbors* (*K-NN*), amplamente utilizado em diversas áreas como Inteligencia artificial, sistemas de recomendação, clusterização, reconhecimentos de padrões e entre outras (COVER, 1968; FISCHETTI1Ú *et al.*, 2014), é um algorítimo de busca por proximidade chamado pelo autor CHÁVEZ *et al.* (2005) como um *Kernel-Methods* podendo ser bastante atraente para diversos pesquisadores, sendo utilizados por eles para também a classificação e regressão.

Podemos definir *K-NN* como uma função $F_{knn}(o, X)$ no qual o é o objeto no qual queremos localizar o k objetos mais similares, X é o *Dataset*, e o retorno dessa função é os k objetos que pertençam a X que minimizam a função de distância $F_d(o, y)$. Olhando a definição do *K-NN* é possível ver um problema do *K-NN*, que é o custo computacional para sua execução, já que para a sua execução é preciso percorrer todo o *dataset* executando a função de distancia qualquer, que segundo o autor PATELLA & CIACCIA (2009) se torna um problema quando trabalhamos uma grande massa de dados. Outro problema que ocorre com certa frequência no *K-NN* segundo o autor CHÁVEZ *et al.* (2005) é a maldição da dimensionalidade.

A maldição da dimensionalidade é um problema bastante conhecido da literatura que afeta de forma quase geral as técnicas de busca, classificação e regressão quando tem trabalhar com dados em altas dimensões. O termo foi cunhado por BELLMAN (1961), ao descrever a dificuldade gerada com aumento de dimensões no espaço euclidiano.

Apesar de ser muito utilizado em problemas que envolvem o espaço vetorial, é interessante ressaltar que o conceito de "dimensionalidade" também pode ser utilizado em espaços métricos. Segundo o autor CHÁVEZ & NAVARRO (2001), a característica comum dos espaços vetoriais de alta dimensão é que a distribuição de probabilidade das distâncias entre os elementos desse espaço, tem um histograma muito concentrado. Em outras palavras quando as dimensões de um espaço vetorial aumenta, o histograma das probabilidades das distancias entre os documentos concentra. Os autores CHÁVEZ & NAVARRO (2000); CHÁVEZ *et al.* (2001), utilizam ma definição de dimensionalidade intrínseca para espaços métricos gerais, que pode ser dita como "A dimensão intrínseca de um conjunto de dados em um espaço

métrico é $\rho = \frac{\mu^2}{2\sigma^2}$, onde μ e σ são a média e a variância de seu histograma de distância."

É sobre a definição de dimensionalidade intrínseca, que o autor CHÁVEZ & NAVARRO (2001) um *dataset* de vetores aleatórios com um dimensão k e com coordenadas uniformemente distribuídas possui uma dimensão intrínseca $\rho = k$, sendo que a partir dessa definição a maldição da dimensionalidade pode ser estendida naturalmente para os espaços métricos. Além de uma explicação teórica sobre o fenômeno os autores CHÁVEZ & NAVARRO (2000); CHÁVEZ *et al.* (2001) a partir de experimentos, fazem uma análise empírica no qual demonstram que os algorítimos degradam a medida que a dimensão intrinsica ρ do espaço aumenta.

Portanto, em problemas de grande base de dados e com alta dimensionalidade, se torna problemática a utilização do *K-NN* , pois como o autor MARIMONT & SHAPIRO (1979) exemplifica, numa consulta aonde temos um grande *dataset* e com alta dimensionalidade como de doenças com atributos de sintomas, tipo de paciente entre outros possíveis atributos, uma busca utilizando o *K-NN*, que é calcular sua a distancia do sintoma do paciente para os sintomas ocasionados por todas as doenças no *dataset*, pode ser um calculo bem demorado, e se essa consulta precisar de uma resposta em tempo real, o uso do *K-NN* se torna inviável. Para acelerar a busca por similaridade, diversas formas de acessos otimizados aos dados nos mais diversos campos de aplicação, como métodos de acesso multidimensionais (espaciais) e métricos (HINNEBURG *et al.*, 2000; WEBER *et al.*, 1998). Só que segundo o autor PATELLA & CIACCIA (2009) o uso de tais estruturas de acesso às vezes não é muito eficiente, pois ainda existe uma gigante quantidade de comparações feitas para grandes *datasets*. Pra esse casos, pode ser interessante, uma busca aproximada, no qual se utiliza alguma heurística para tornar acelerar a busca, com o efeito de uma degradação na qualidade do resultado, ou seja, um erro em relação ao caso exato.

2.4.2 Busca Aproximada

A busca aproximada, como o próprio no nome sugere, é uma forma de fazer uma busca aonde não se busca o resultado exato, e sim um resultado que seja satisfatório. Como já dito a busca aproximada é muito utilizada quando a busca exata se torna inviável, normalmente ocorridos pela maldição da dimensionalidade e/ou uma massa de dados muito grande que torna inviável a execução em um tempo aceitável. O funcionamento da busca aproximada tem como base, o relaxando das restrição em relação a uma busca exata, ou seja, os k objetos no resultado aproximado podem não ser os k mais próximos. E segundo o PATELLA & CIACCIA (2009), os principais

argumentos que normalmente são utilizados para defender a busca aproximada em relação a esse relaxamento são:

1. **A não existência de um resultado exato**:

    - Segundo SANTINI & JAIN (1998), existe uma lacuna entre a similaridade percebida pelo usuário e a implementada através da função distância. O resultado "exato" de uma consulta, em muitos casos, pode na verdade ser considerado incorreto pelo usuário, esse caso comumente ocorre em buscas de similaridade em dados de multimídia, em sistemas de recomendação (DO CARMO, 2018), entre outros.

2. **O processo de busca pode ser iterativo**:

    - Pela mesma razão, o processo de busca de similaridade é tipicamente iterativo, porque o usuário pode estar pesquisando, usando um gerando um *feedback* positivo ou negativo, para o objeto da busca.

3. **O tempo de busca**:

    - Mesmo quando a função de distância e o objeto de consulta são adequados, o usuário ainda pode preferir obter rapidamente um resultado aproximado, em vez de esperar mais tempo pela resposta exata. Por muitas vezes o tempo de resposta do sistema é essencial para a solução, por exemplo a alteração de uma rota do GPS (PATELLA & CIACCIA, 2009).

O sucesso de uma busca aproximada vai depender diretamente da relação de qualidade do resultado com tempo de execução dessa busca, sendo que essa relação entre qualidade do resultado com tempo de execução da busca é distinto para cada área, é a partir dessas diferentes relações começa ter diferentes formas de construção de uma busca aproximada. Foi analisando as diferentes formas de construção que o PATELLA & CIACCIA (2009) desenvolveu uma forma de classificar as mais diversas formas de busca aproximada, sendo que classificação visa somente avaliar a aplicabilidade das técnicas e não sua qualidade ou fraqueza. Podemos definir separar essa classificação em 4 partes, em que tipo de espaço a técnica se aplica, como a aproximação é obtida, quais são as garantias de resultado e qual é o grau de interação da técnica com o usuário.

1. **Tipo de espaço em que técnica se aplica**:

    - Espaço Métrico.
    - Espaço Vetorial com distancia L_p.

- Espaço Vetorial.

A primeira parte da classificação, se propõem a classificar as técnicas com base no tipo de dados em que elas se aplicam. O primeiro e mais genérico, são os métodos que se aplicam a espaços métricos. Um pouco mais especifico temos as abordagens que se aplicam a espaços vetoriais quaisquer, sendo assim qualquer função de distancia entre vetoriais poderia ser utilizada por exemplo. E por fim temos as abordagens por espaços vetoriais com a distancia L_p, no qual são abordagens que são consideradas objetos em um espaço vetorial D-dimensional aonde não existe nenhuma correlação entre coordenadas, aonde são utilizadas uma métrica L_p qualquer para comparação.

Uma abordagem que podemos exemplificar como espaço métrico é *Permutation Based Index* no qual detalharemos melhor seu funcionamento no capítulo 3. Podemos exemplificar o *VA-Low* como uma abordagem que dados em um espaço vetorial, que em podemos resumir a técnica como uma transformação do espaço afim de reduzir de compactar as informações (WEBER *et al.*, 1998). E por fim podemos classificar o *Locality Sensitive Hashing* como uma abordagem que utiliza Espaço vetorial com distancia L_p, sendo que essa abordagem será explicada na seção 2.5.1

2. **Tipo de aproximação**:

   - Transformação do espaço.
   - Redução de comparação.
     - *Pruning Agressivo*
     - Parada antecipada.

A segunda parte da classificação é de que forma ocorrem a aproximação, ou seja é aonde ocorre a flexibilização da busca. Nessa parte temos duas formas de flexibilizar a busca, reduzindo o numero de comparações e transformando do espaço. Podemos definir as abordagens que são classificadas como transformação de espaço, abordagens nas quais alteram de alguma forma o espaço métrico, seja alterando a distância usada para comparar objetos ou modificando o espaço do objeto. Podemos citar como exemplo abordagens que fazem uma redução de dimensionalidade, antes da aplicação de uma busca exata.

As abordagens que classificadas como as que reduzem o numero de comparações, elas utilizam o espaço sem nenhuma transformação e aceleram a consulta sem fazer todas as comparações. Sendo que essa classificação pode ser subdivididas em outras duas, a parada antecipada e a remoção de forma agressiva. Na remoção de forma agressiva, a abordagem decide de uma forma qualquer a remoção de parte do conjunto de dados que seriam comparados. Um exemplo disso são abordagens que podam parte do espaço usando alguns limite probabilísticos. Já as abordagens

de parada antecipada, elas decidem não visitar certas regiões do espaço somente no tempo de execução, normalmente parando a execução quando um custo máximo foi atingido e/ou valor aceitável de distancia dos objetos até então selecionados foi atingido.

3. **Garantias do Resultado**:

    - Sem garantias.
    - Garantias determinísticas.
    - Garantias probabilísticas.

A terceira parte da classificação de PATELLA & CIACCIA (2009), é a garantia que a abordagem consegue entregar um resultado com uma determinada qualidade. Três classificações são possíveis neste caso, a que não tem garantia nenhuma, ou seja as abordagens trabalham somente condições heurísticas e não tem formalmente nenhuma forma de demonstrar o erro introduzido utilizando a abordagem. Podemos classificar também pelas as garantias determinísticas, ou seja a abordagem é capaz de vincular deterministicamente o erro introduzido pela abordagem. Uma abordagem que exemplifica a as garantias determinísticas é a abordagem para pesquisas em *BBD-tree* proposta em ARYA *et al.* (1998), no qual a abordagem trata os dados como uma arvores binarias na qual representa uma decomposição recursiva do espaço, tendo que seu erro não ultrapassa a medida definida no trabalho ZEZULA *et al.* (1998).

A ultima classificação das garantias de erro, as abordagens que podemos classificar com garantias probabilísticas, são a abordagens que garantem que a uma faixa das consultas realizadas terão a qualidade atendidas, normalmente as abordagens com garantias probabilísticas definem essas garantias nas suas distribuições de dados, sendo que podemos subdividir ela em abordagens que garantem a qualidade assumindo a distribuição de dados por meios paramétricos, sendo que esses dados parametrizados são recuperados normalmente pegando uma amostra dos dados, e as abordagens que fazem garantias sem a utilizar distribuição dos dados, ou poucas suposições sobre essa distribuição, sem utilizar nenhum dado parametrizável.

4. **Interação com o usuário**:

    - Abordagem estática.
    - Abordagem interativa.

A quarta parta da classificação das buscas por proximidade apresentada pelo PATELLA & CIACCIA (2009), envolve a interação com o usuário em tempo de consulta. Ou seja ela é relativa se o usuário é capaz de alterar parâmetros com a

abordagem em execução para uma determinada consulta. Por exemplo a quantidade mínima da distancia para uma parada antecipada do busca. Nesta classificação nós teremos duas formas de classificar, as abordagens que são interativas, ou seja permitem de alguma forma alterar parâmetros em tempo de execução ou as estáticas, que não permitem a alteração de algum parâmetro em tempo de execução.

2.5 Trabalhos Relacionados

As abordagens de busca aproximada que normalmente são utilizadas na etapa da busca heurística, podem ser classificadas como abordagens que buscam em um espaço métrico de forma genérica devido a forma de transformação dos dados amplamente adotados (APOSTOLICO *et al.*, 2006; BAEZA-YATES *et al.*, 1999; DUARTE, 2017; VIEIRA, 2016), tendo como o tipo de aproximação a transformação do espaço das diversas garantias de resultados.

2.5.1 Locality Sensitive Hashing

Locality Sensitive Hashing(*LSH*) é um conjunto de abordagens desenvolvida para a busca aproximada para problemas de alta dimensionalidade computacional. O autor INDYK & MOTWANI (1998) motiva a utilização desta técnica pois com a utilização do *LSH* o tempo computacional é reduzida drasticamente com uma probabilidade bem baixa de não se encontrar os vizinhos mais próximos desejados. A ideia básica do *LSH* utilizado para busca por similaridade em texto é separar em objetos e a partir disso gerar *hashes* para cada objeto, afim de que objetos iguais tenham a mesma *hash*, sendo que se um texto é próximo do outro eles compartilharam de diversas *hashes*.

O algorítimo pelo conjunto de abordagens *LSH* pode ser dividido em duas partes, o pré-processamento e a busca para encontrar os k vizinhos mais próximos. O pré-processamento tem como a entrada um conjunto D e a variável N que podemos representar como o numero de funções *hash* a utilizadas no *LSH*. São as funções de *hash*, as responsáveis por agrupar objetos similares , ou seja que tenham valor de *hash*. E que na fase da consulta são utilizados esses agrupamentos criados na fase de pré-processamento para localização dos k vizinhos mais próximos.

É a partir do *LSH*, que surgem algumas abordagens que utilizam a funções de *hash*. Uma das primeiras, o *Minwise Hashing*(*min-hash*), que surge motivação segundo o BRODER (1997) pelo o aumento da quantidade de dados *online* disponíveis, sendo que as técnicas que fazem busca por similaridade utilizando as distancias já conhecidas como *Hamming*, *Levenshteing*, não eram fazer a análise desses conjuntos de dados por demandarem alto tempo computacional.

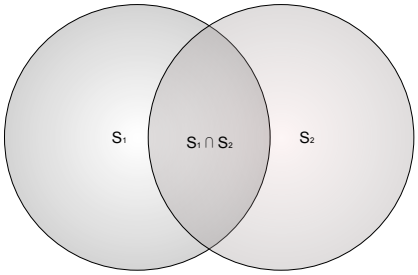

Figura 2.5 – Interseção de dois objetos.

A base da abordagem é tentar descobrir a similaridade entre os documentos por meio de um problema de interseção de conjuntos, que pode ser resolvido por uma processo de amostragem aleatória de forma independente para cada conjunto (BRODER, 1997). Ou seja a ideia básica da abordagem se da quanto mais similares são os documentos maior é a interseção entre eles, sendo assim se buscarmos amostras aleatórias destes documentos, podemos fazer essa comparação utilizando a ideia da interseção de conjuntos.

Antes de definirmos formalmente o *Min-Hash*, é importante definirmos a técnica chamada *Shingling* desenvolvida pelo BRODER (1997), na qual ela ao associa um conjunto de subsequencias de *tokens* ao um documento, A partir disso ela reduz a listas de *tokens* a uma lista de *hashes*, que essas podem ser comparadas diretamente utilizando diferença, união e interseção de conjuntos para determinar a dissimilaridade ou distancia, que pode ser considerada como o inverso da similaridade. (BUTTLER, 2004; VIEIRA, 2016). Normalmente a métrica utilizada para essa comparação é o coeficiente de *Jaccard* (BUTTLER, 2004; DUARTE, 2017; JI *et al.*, 2012; VIEIRA, 2016), sendo criado pelo JACCARD (1908), no qual pode ser definido como visto na equação 2.2, a interseção de dois conjuntos sobre a união dos mesmos.

$$Jaccard\,(C_i, C_j) = \frac{|C_i \cap C_j|}{|C_i \cup C_j|} \tag{2.2}$$

Formalmente podemos definir abordagem segundo JACCARD (1908) o *min-hash* como uma abordagem derivada do *LSH*, aonde a partir de um universo C, se empregará uma permutação que podemos definir como π, dos grupos de objetos pertencentes a $C_i \in C$ pela uma função *hash*. A partir de cada permutação, o menor valor de assinatura de cada conjunto C_i será igual ao primeiro menor elemento pertencente a C_i. Os trabalhos CHUM *et al.* (2008); VIEIRA (2016) simplificam a explicação da abordagem *min-hash* para a seguinte definição: Seja t_1 e t_2 dois termos distintos do conjunto C_i, as funções hash devem seguir duas seguintes condições,

$h(t_1) \neq h(t_2)$ e $P(h(t_1) > h(t_2)) = 0,5$, sendo que podemos expressar a o *min-hash* matematicamente através da equação 2.3.

$$min(C_1, h) = argmin_{t \in C_i}\ h(t) \tag{2.3}$$

Quando o objeto $o_k = min(C_i, C_j, h)$, como a função h é uma função de *hash* aleatória, cada elemento de $C_i \cup C_j$ tem a mesma probabilidade de ser o menor. Quando o o_k pertencer a C_i e C_j, teremos que $min(C_i, h) = min(C_j, h)$, caso não ocorra podemos garantir que $min(C_i, h) \neq min(C_j, h)$.

Assim sendo, podemos escolher N permutações aleatórias induzidas por N funções *hashes*, podemos assumir aproximar a similaridade entre C_i e C_j utilizando os conjuntos $\bar{C_i}$ e $\bar{C_j}$ apresentados na equação 2.4.

$$\begin{aligned} \bar{C_i} &= (min(C_i, h_1), min(C_i, h_2), ..., min(C_i, h_N)) \\ \bar{C_j} &= (min(C_j, h_1), min(C_j, h_2), ..., min(C_j, h_N)) \end{aligned} \tag{2.4}$$

A partir do *min-hash*, variantes dessa abordagem surgiram com o intuito de aumentar como ao invés de recuperar a menor permutação, se recuperar a maior permutação (JI *et al.*, 2013b; VIEIRA, 2016), e recuperar a menor e maior permutação, essa ultima conhecida como *Min-Max Hashing*, tendo como uma de suas vantagens explicadas no trabalho VIEIRA (2016) de poder aumentar a quantidade de informações obtidas do documento para o mesmo custo computacional, já que ao invés de recuperar somente a permutação com menor valor, se recupera também a permutação de maior valor. A partir da equação 2.4, podemos aproximar a similaridade entre conjuntos C_i e C_j utilizando os conjuntos $\bar{C_i}$ e $\bar{C_j}$ apresentados na equação 2.5.

$$\begin{aligned} \bar{C_i} &= (min(C_i, h_1), max(C_i, h_1), ..., min(C_i, h_N), max(C_i, h_N)) \\ \bar{C_j} &= (min(C_j, h_1), max(C_j, h_1), ..., min(C_j, h_N), max(C_j, h_N)) \end{aligned} \tag{2.5}$$

2.5.2 Okapi BM-25

Outra abordagem muito utilizada na comparação entre documentos, o *Okapi BM-25* é uma função de busca por similaridade (WAN *et al.*, 2008), na qual classifica o documento a partir da relevância dos termos com base nas ocorrências dos termos no *dataset* como um todo (ROBERTSON *et al.*, 2009). Criado pelo ROBERTSON *et al.* (1995), o *Okapi BM-25*(*BM-25*) foi desenvolvido para gerar um *ranking* que estima os documentos mais relevantes para uma determinada busca.

O *BM-25* é uma adaptação da abordagem probabilística *Relevance Weighting Scheme* desenvolvida pelo ROBERTSON & JONES (1976), , motivado pelo desenvolvimento de uma abordagem que lhe permitisse a busca por termos. É a partir do trabalho SPARCK JONES (1972), que se apresenta a ideia da *inverse frequency term*, também conhecida como frequência inversa do coumento(IDF), apresenta a ideia que um bom termo que representa a consulta, ou seja deve receber um peso maior na consulta é aquele que aparece em poucos documentos. Podemos definir formalmente o IDF na equação 2.6, aonde q_i é um termo qualquer, a função n retorna a quantidade de documentos aonde q_i é encontrado, N é a quantidade de documentos.

$$IDF\left(q_i\right) = log\left(\frac{N - n\left(q_i\right) + 0.5}{n\left(q_i\right) + 0.5}\right) \tag{2.6}$$

Baseado no IDF, agora podemos definir formalmente o *Relevance Weighting Scheme* segundo ROBERTSON *et al.* (1993) na equação 2.7 no qual q_i é um termo qualquer, a função n retorna a quantidade de documentos aonde q_i é encontrado, N é a quantidade de documentos, R é a quantidade Total de documentos relevantes e a função r é a quantidade de documentos relevantes para o termo q_i.

$$W\left(q_i\right) = log\left(\frac{\frac{(0.5+r)}{R-r+0.5}}{\frac{n(q_i)-r+0.5}{N-n(q_i)-R+r+0.5}}\right) \tag{2.7}$$

Caso não não se utilize o os documentos relevantes, podemos refazer a equação 2.7 utilizando $R = r = 0$ de forma a gerar a equação 2.8 (LV & ZHAI, 2011; ROBERTSON *et al.*, 1993).

$$W^2\left(q_i\right) = log\left(\frac{N + 1}{n\left(q_i\right) + 0.5}\right) \tag{2.8}$$

É a partir do sistema *Okapi*, desenvolvido pela *City University* em Londres (ROBERTSON, 1997) que nasce o *BM-25*, que segundo ROBERTSON *et al.* (1993) começa utilizando *Relevance Weighting Scheme* e a junção de outras duas abordagens também derivadas do *Relevance Weighting Scheme* o *BM-11* e o *BM-15*, sendo que podemos classificar todos essas abordagens como abordagens da classe *Best-Matching*.

$$BM25\left(D,Q\right)=\sum_{i=1}^{n}W^{2}\left(q_{i}\right)\frac{f\left(q_{i},D\right)\left(k_{1}+1\right)}{f\left(q_{i},D\right)+k_{1}\left(1-b+b\frac{|D|}{avgdl}\right)} \tag{2.9}$$

Apesar de ter sido criado posteriormente, por questões didáticas primeiramente definiremos o *BM-25* e a partir deles iremos correlacioná-lo com os outros métodos. Definimos o *BM-25* na equação 2.9 na qual D e Q são os documentos no qual se quer comparar, q_i é termo i no qual $q \in Q$, $f\left(q_i, D\right)$ é a função de frequência do termo de q_i no documento D, $|D|$ é a quantidade de termos no documento D, k_1 é uma constante na qual normalmente é utilizado um valor entre 1.2 e 2.0 (SANDERSON, 2010), b é uma constante no qual o valor normalmente utilizado é 0.75 (ROBERTSON *et al.*, 1993; SANDERSON, 2010) e $avgdl$ é a quantidade média de termos dentro de um documento.

$$BM11\left(D,Q\right)=\sum_{i=1}^{n}W^{2}\left(q_{i}\right)\frac{f\left(q_{i},D\right)\left(k_{1}+1\right)}{f\left(q_{i},D\right)+k_{1}\left(\frac{|D|}{avgdl}\right)} \tag{2.10}$$

Podemos Definir o *BM-11* a partir do *BM-25* como um caso especifico aonde o $b = 1$. A partir disso podemos montar a mesma equação do *BM-25* para o *BM-11* visto na equação 2.10 (ROBERTSON *et al.*, 1993).

$$BM15\left(D,Q\right)=\sum_{i=1}^{n}W^{2}\left(q_{i}\right)\frac{f\left(q_{i},D\right)\left(k_{1}+1\right)}{f\left(q_{i},D\right)+k_{1}} \tag{2.11}$$

Da mesma forma que o *BM-11*, podemos definir o *BM-15* como um caso especifico do *BM-25* no qual o $b = 0$, A partir disso vemos o *BM-15* na equação 2.11. E por ultimo podemos definir o *BM-25F* na equação 2.12 que é uma variação do *BM-25* no qual soma-se um valor δ qualquer (CRASWELL *et al.*, 2005).

$$BM25F\left(D,Q\right)=\sum_{i=1}^{n}W^{2}\left(q_{i}\right)\frac{f\left(q_{i},D\right)\left(k_{1}+1\right)}{f\left(q_{i},D\right)+k_{1}\left(1-b+b\frac{|D|}{avgdl}\right)}+\delta \tag{2.12}$$

Capítulo 3

Permutation Based Indexing

Embora existam soluções para o problemas de busca em um espaço métrico sem a necessidade de busca aproximada como o *k-nearest neighbors*, com o aumento da dimensionalidade e da grande massa de dados, essas buscas se tornam cada vez mais custosa, tornando a utilização dela inviável, por essa motivação levou a primeira aparição do *Permutation Based Indexing* em 2008, como uma modificação do algoritmo *LAESA* baseada em pivô apresentado por CHAVEZ GONZALEZ *et al.* (2008) para resolver problemas de indexação e busca de similaridade entre dois ou mais itens, fazendo a busca em um espaço vetorial na área de reconhecimento de faces e na indexação de documentos. LUISA MICÓ & ONCINA (1994).

Utilizando a forma de classificação criada pelo PATELLA & CIACCIA (2009) para a classificação de busca aproximada, podemos classificar este método como uma técnica que utiliza o espaço métrico na sua base de dados, seu tipo de aproximação é a redução de comparação com um *Pruning* agressivo, sem garantia formal de resultados e é uma abordagem iterativa.

Com o avanço na pesquisa na área de multimédia (Imagem e Video), A pesquisa em torno do *Permutation Based Indexing* aumentou, sendo ele muito utilizado na busca e indexação de conteúdos multimédias (FIGUEROA *et al.*, 2015; KRULIŠ *et al.*, 2015a). O uso do *Permutation Based Indexing* em problemas com conteúdos de multimédia ocorrem, pois eles devem ser classificados, previstos, filtrados ou organizados, e que para isso, a utilização do *K-Nearest Neighbors* se torna inviável, pelo tamanho do *dataset* nessa área que faz a sua execução que varre todo *dataset* ser muito custoso (CHÁVEZ & NAVARRO, 2005), algo que o *Permutation Based Indexing* se torna extremamente atrativo pois ele tem a capacidade de reduzir o espaço de busca. A utilização do *Permutation Based Indexing* não se restringe a utilização na área de multimédia, sendo possível ser utilizado em outras áreas, como texto (FIGUEROA & FREDIKSSON, 2009).

Podemos separar o *Permutation Based Indexing* em duas partes, a indexação e busca (CHÁVEZ *et al.*, 2005), sendo a busca, o ato de comparar um objeto a todo

o conjunto de dados em busca de objetos mais similares. A indexação é a parte que definimos um tratamento no nosso conjunto de dados afim que otimize essa busca e a seleção de *pivots* que melhor representem o *dataset*.

Um dos pontos interessante em relação do *Permutation Based Indexing* é a sua implementação de forma paralela em CPU ou mesmo em GPU, como o trabalho do autor MOHAMED *et al.* (2014b) tem como motivação a otimização do tempo de indexação, e com isso trabalha no desenvolvimento de uma variação do processo de indexação utilizando paralelismo em CPU e GPU e de forma similar os trabalhos dos autores FIGUEROA *et al.* (2015); KRULIŠ *et al.* (2015b) também propõem uma alternativa para indexação em GPU.

3.1 Indexação

A parte da indexação pode ser explicada como qualquer execução que se demande anteriormente a utilização de qualquer documento suspeito, que na prática é a parte que se faz toda a transformação do *dataset*, cuja a única restrição é que essa transformação se respeite o espaço métrico, para que se possa comparar distâncias entre dois objetos distintos. A indexação segundo MOHAMED *et al.* (2014b) é custosa, principalmente na etapa de seleção de *pivot*, no qual existem abordagens que contornam a dificuldade utilizando paralelismo em *CPU* e em *GPU* (KRULIŠ *et al.*, 2015b) mas que não serão vistos neste trabalho por não ser o foco.

Na parte da indexação nós representamos um conjunto D de objetos em um Espaço métrico no qual d_i, $D = d_1, d_2, d_3...d_n$ no qual identificamos cada ponto documento d_i como pontos x_i em um espaço multidimensional R^m. A Partir desse conjunto de Objetos representados no espaço vetorial nós selecionamos k objetos para tornar-se Objetos de Referencia desse conjunto D de documentos, esses objetos de referencias são conhecido como *pivots*. Esse conjunto *pivots* tem como função representar o Conjunto de Objetos D, e é partir da distancia deles em relação a dois ou mais objetos que se calcula a similaridade entre eles.

Algorithm 1: Indexação com Permutation Based Index

Data: Conjunto de Objetos D

Result: Conjunto P' de conjuntos Pivots ordenados

1 D = Representar o Conjunto O de Objetos em um Espaço Vetorial;
2 P = Selecionar k objetos do conjunto D para o conjunto de Pivots ;
3 **for** $n = 1, ..., N$ **do**
4 $\quad dist_{d_n}$ = Calcular a distância do objeto d_n a cada $p_k \in P$;
5 $\quad P_{d_n}$ = Ordenar o conjunto de *pivots* utilizando a $dist_{d_n}$

3.1.1 Transformação para um Espaço Métrico

A etapa de transformação para um espaço métrico é um importante passo na transformação de um objeto de forma que se torne viável a comparação da distância entre eles. Neste trabalho, iremos abordar formas de transformação para documentos em forma de texto. Faremos a transformada de todo documento em um conjunto de valores numéricos, sendo que anteriormente a este passo, pode-se fazer Pré-processamentos no texto para melhor desempenho dos algorítimos de busca heurística, como a conversão de letras maiúsculas para minusculas, pontuações em geral e palavras raras e muito utilizadas. Pode-se também utilizar a identificação de classes gramaticais, identificação de radicais das palavras e remoção de afixos (EHSAN & SHAKERY (2016)) e a retirada de *stop-words*, que o autor WILBUR & SIROTKIN (1992) define como palavras que não tenham relevância para a busca, podendo ela ter o mesmo tipo de ocorrência nos documentos que não são relevantes para uma busca, como naqueles documentos relevantes para a busca.

Uma técnica muito utilizada para a comparação de texto e que se utiliza para transformação do texto em um conjunto de valores numéricos é a ideia de dividir o texto em um conjunto de elementos únicos do texto nomeado de *fingerprint* BARRÓN-CEDEÑO (2010), e essa técnica foi desenvolvida para encontrar similaridades em documentos a partir de trechos do documento, sendo essa técnica utilizada pelos autores DUARTE (2017); VIEIRA (2016) para o pré-processamento em problemas de busca heurística no contexto de plágio e reuso de texto.

A ideia é que uma *fingerprint* seja uma função matemática qualquer,no qual essa função retornar a mesma saída para a mesma entrada, e para entrada distintas, a saída deve ser também distinta. Podemos definir formalmente uma função que gera uma *fingerprint* $f_{fing}(d_i) = fingerprint$, sendo d_i um subconjunto qualquer e $f_{fing}(d_i) \neq f_{fing}(d_j)$, normalmente utiliza-se uma função de *hash* como função que gera uma *fingerprint* (VIEIRA, 2016). Neste trabalho focaremos na geração de *fingerprints* para um subconjunto de texto, e esse subconjunto de texto pode ser composto de caracteres, palavras ou até mesmo sentenças.

Com a ideia da *fingerprint* podemos representar o nosso conjunto de documentos de forma matricial no qual as colunas representam o conjunto de *fingerprints* existentes no conjunto de documentos e as linhas representam os documentos desse conjunto. Cada elemento $e_{l,c}$ pertencente a matriz de L linhas e C colunas podem ser definidas de algumas formas, a *booleana*, *não-booleana* e a *term frequency–inverse document frequency*(*TF-IDF*).

A *boolena* é a ideia de uma representação da existência ou não da daquela *fingerprints* no documento, podemos definir formalmente $e_{l,c} = 1$ se a *fingerprints* c estiver contida no documento l, caso contrario $e_{l,c} = 0$. Exemplificando: Seja os documen-

	a	b	c	d	e	f
D_1	1	1	1	1	0	0
D_2	0	1	1	0	0	1
D_3	1	1	0	0	1	0

Tabela 3.1 – Exemplo de Matriz Booleana

tos D_1, D_2 e D_3 dados pelo conjunto de *fingerprints* $D_1 = a, b, c, d, b$, $D_2 = b, c, f$ e $D_3 = a, b, e, a, b$, tendo o conjunto total de *fingerprints* nos documentos igual a a, b, c, d, e, f. A representação matricial é dada pela 3.3

A forma *não-boolena* é a ideia de uma representação da quantidade de vezes que aquela *fingerprint* apareceu no documento. Definimos como $e_{l,c} = q$, sendo q a quantidade de vezes que a $e_{l,c}$ apareceu no documento l. Utilizando o mesmo exemplo da forma *boolena*, a representação matricial é a dada pela tabela 3.3.

	a	b	c	d	e	f
D_1	1	2	1	1	0	0
D_2	0	1	1	0	0	1
D_3	2	2	0	0	1	0

Tabela 3.2 – Exemplo de Matriz não booleana

A forma *TF-IDF* é uma variação representação da quantidade de vezes que aquela *fingerprint* apareceu no documento, só que agora levando em consideração a indicar a importância dessa *fingerprint* para a coleção. Podemos Separar o *TF-IDF* no *TF*(*Term-Frequency*) que podemos definir como a quantidade de vezes que a *fingerprint* aparece no texto, o *idf* aparece pela primeira vez no trabalho do autor LUHN (1957). O *idf*(*inverse document frequency*) que pode ser definido como $idf = \log \frac{N}{fp_D+1}$ sendo N a quantidade de documentos da coleção, $fp_{f,D}$ a quantidade de documentos da coleção que a *fingerprint* f aparece. Utilizando o mesmo exemplo da forma *boolena*, a representação matricial é a dada pela tabela 3.2.

	a	b	c	d	e	f
D_1	0.403	0.626	0.4035	0.530	0	0
D_2	0	0.425	0.54783	0	0	0.720
D_3	0.700	0.544	0	0	0.460	0

Tabela 3.3 – Exemplo de matriz com o *TF-IDF*

3.1.2 Seleção de Pivots

Os métodos que utilizam *pivots*, classificados como *pivot-based*, tem sido um tópico bastante recorrente na área de busca de similaridade CHEN *et al.* (2017), sendo que

boa parte desses métodos utilizam a seleção de *pivots* para a redução na quantidade de comparações feitas na consulta por similaridade , que é o caso do *Permutation Based Indexing* e que segundo o autor AMATO *et al.* (2015), uma boa seleção de *pivots* para o *Permutation Based Indexing* é um dos fatores que diretamente responsável pela a qualidade dos resultados da técnica.

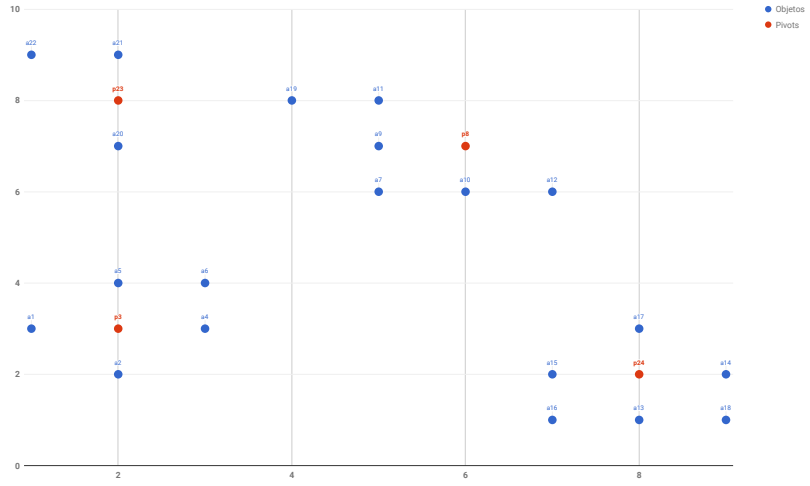

Figura 3.1 – Seleção do Conjunto de *pivots*.

Existem diversas formas de escolha de *pivots* AMATO *et al.* (2015), os principais trabalhos para a seleção de *pivots* visam a escolha de um *pivot* de forma randômica, pela sua localização no *dataset* considerando a distancia que dele em relação aos outros *pivots* de forma a tentar espalhar os *pivots* pelo *dataset* como todo, utilizando a ideia que o *dataset* é um conjunto de objetos heterogênicos e que podemos separa-los em sub-conjuntos aonde os objetos são homogêneos.

Podemos definir a seleção de *pivots* como uma função $F_{sp}(D, k)$ na qual recebe o *dataset* D e k sendo um numero natural na qual representa a quantidade de *pivots* e retorno dessa função será um conjunto P de *pivots* de tamanho k.

3.1.2.1 Randômico

Segundo BUSTOS *et al.* (2001), apesar de sabido que uma boa escolha de *pivots* tende a tornar a abordagem mais eficiente, algorítimos de busca de proximidade em baseadas em *pivots*, costumam-se utilizar o método aleatório de seleção,por isso o algorítimo que podemos considerar um *baseline* para a seleção de *pivots*, é o algorítimo de seleção randômica, tem por ideia inicial, na qual não se precisa escolher o objeto no qual seja mais representativo do *dataset*, e sim só precisa-se escolher uma quantidade boa de objetos para uma boa representação do *dataset* como todo.

3.1.2.2 Pivoted Space Incremental Selection

Afim de melhorar a seleção de *pivots* que na maioria das vezes são randômicas, o autor BUSTOS *et al.* (2001), apresentou o *Pivoted space incremental selection* (PSIS) com a ideia de se selecionar os *pivots* com tendo uma distancia miníma (também nomeado de *threeshold*) para os outros *pivots* da lista.

A motivação de utilizar o *Pivoted space incremental selection*(*PSIS*), segundo BUSTOS *et al.* (2001), é que bons pivôs estão distantes uns dos outros, ou seja, a distancia entre os *pivots* são maiores do que a distância média entre objetos aleatórios do mesmo espaço e bons *pivots* estão longe do resto dos elementos do espaço. Os objetos que satisfazem as propriedades de bons *pivots* segundo o BUSTOS *et al.* (2001) são chamados objetos chamados *outliers*, pois a definição de *outliers* segundo os autores BRIN (1995); FARAGÓ *et al.* (1993); YIANILOS (1993) são bem similares que a definição de *outiliers*.

Podemos definir o *PSIS* como uma técnica que é baseada distribuição de distância do espaço métrico, a partir do conjunto D de objetos, selecionamos um *pivot* p_1 qualquer, após isso selecionamos um *pivot* p_2 que satisfaça a distancia miníma para o p_1, depois selecionamos um *pivot* p_3 que satisfaça a distancia miníma para o conjunto p_1, p_2 e assim repetidamente até selecionar os k *pivots*.

Algorithm 2: Seleção de Pivot PSIS

Result: Array de Pivots P

Data: Conjunto de Objetos D, quantidade de pivtos k, distancia minima θ

```
1 p_1 = selecionar um pivot qualquer do D;
2 P = Array de pivots;
3 for n = 1, ..., k do
4     p_n = selecionar um pivot qualquer satisfaça a distancia máxime a
        distancia mínima entre todos os pivots P;
5     adicionar p_n ao P;
```

3.1.2.3 Farthest-first traversal

A técnica *Farthest-first traversal* é uma variação da técnica de seleção de objetos em um espaço métrico que é utilizada em diversos problemas chamada de, sendo utilizado a principio pelo ROSENKRANTZ *et al.* (1977) *Farthest Insert* para a resolução do problema do caixeiro viajante, e que esse problema é definido como um problema que tenta determinar a menor rota para percorrer quantidade qualquer de cidades, somente a visitando cada cidade uma única vez, e retornando à cidade de origem LAWLER *et al.* (1985). O problema do caixeiro viajante é um problema de otimização NP-difícil, pode ser considerado uma heurística que tenta resolver esse problema em tempo linear.

O *Farthest Insert* foi definido pelo ROSENKRANTZ *et al.* (1977) como uma técnica a partir de um conjunto de T_i vértices em um grafo, escolhe-se um vértice de T_i que maximiza a distancia para o vértice a_i. O MENDEL & NAOR (2006) definiu uma variação do *Farthest Insert*, chamada de *Farthest-first traversal*, na qual não se escolhe mais o vértice de T_i mais distante para um vértice a_1 qualquer, e começa a escolher um vertice de T_i que tenha no mínimo uma distancia θ qualquer, essa distancia miníma foi chamada de *Threshold*.

$$FarthestInsert(T_i, a_i) \;=\; max\left(d(a_i, T)\right) \tag{3.1}$$

O *Farthest-first traversal*(FFT) é utilizado inicialmente pelo AMATO *et al.* (2015) para a seleção de *pivots*, motivado por causa que *Farthest-first traversal*, tenta maximar a distancia miníma entre os *objetos*, uma característica muito interessante para uma técnica de seleção de *pivots*. De fato isso é uma característica interessante pois isso pode ajudar obter como já falado aqui neste trabalho, *pivots* que melhor representem o *dataset*, tendem a ser distantes um dos outros. Podemos definir a técnica do *Farthest-first traversal* utilizada para a seleção de *pivot* no algorítimo 4.

Algorithm 3: Seleção de *Pivot* Farthest-first traversal

Result: Lista ordenada de *pivots* P

Data: Conjunto de Objetos D, quantidade de *pivots* k, distancia miníma θ

1 p_1 = selecionar um *pivot* qualquer do D;
2 P = Lista ordenada de *pivots*;
3 **for** *n = 1, ..., k* **do**
4 $\quad p_n$ = selecionar um pivot qualquer satisfaça a distancia miníma θ de todos os pivots P;
5 $\quad$ adicionar p_n ao P;

3.1.2.4 K-Medoids

O *K-Medoids* é uma técnica de agrupamento de objetos similares, no qual o autor definiu como um conjunto de técnicas que tem por objetivo encontrar um conjunto de objetos homogêneos dentro de um conjunto aonde se tem objetos heterogêneos. Como conhecida na literatura como técnicas de*clusterização*, ela na pratica pode ser explicada como um ato de subdividir um conjunto em outros sub-conjuntos com uma hipótese especifica na qual visa-se explicar ou sugerir esses agrupamentos .

O *K-medoids* também conhecido como *Partitioning Around Medoids (PAM)*, foi proposto inicialmente pelo ROUSSEEUW & KAUFMAN (1990), podendo ser descrito como um agrupamento de k *cluster* aonde o centro de cada *cluster* é um objeto chamado de *medoid*, no qual ele pertencente a este mesmo *cluster* ROUSSEEUW &

KAUFMAN (1990). O *K-Medoids* tem uma estreita relação com outro algorítimo de clusterização, o chamado *K-Means*, no qual ele agrupa em k *cluster*, objetos similares em torno de um centroide de um espaço vetorial, com a diferença que este sento é criado na mesma etapa aonde se define os *cluster*, esta etapa é comumente chamada de treinamento.

A partir da implementação feita pelo ROUSSEEUW & KAUFMAN (1990), diversos outros autores trabalhos desenvolveram otimizações para o *K-Medoids* de forma o tornar-lo viável em relação ao tempo computacional, para a utilização em grandes *datasets*, preocupação que aparece com o autor . Como definido pelo autor que propõem uma solução otimizada para o *k-medoids* utilizada neste trabalho, podemos definir o algorítimo em duas etapas, a seleção inicial do *k-medoids*, as atualizações dos *medoides*.

Algorithm 4: K-Medoids

Result: Lista ordenada de *pivots* P

Data: Conjunto de Objetos D, quantidade de *pivots* k, distancia miníma θ

for *j = 1, ..., n* **do**
  for *i = 1, ..., n* **do**
    $d_{ij} = F_d(o_i, o_j)$
for *j = 1, ..., n* **do**
  $v_j = \sum_{i=1}^{n} \frac{d_{ij}}{\sum_{l=1}^{n} d_{il}}$
medoids[] = A partir dos fatores v_j, escolher os k objetos com menores somas da distancia.
Atribua cada objeto ao *medoid* mais próximos formando os novos *clusters*
Obtenha o resultado inicial do *cluster* atribuindo cada objeto ao *medoid* mais próximo.
S_m =Soma das distâncias de cada *medoid* com todos os objetos do *cluster*.
while $S_m != S_{\bar{m}}$ **do**
  Encontre o novo *medoid* para cada *cluster*
  Atribua cada objeto ao *medoid* mais próximos formando os novos *clusters*
  $S_{\bar{m}}$ =Soma das distâncias de cada *medoid* com todos os objetos do *cluster*.

A primeira etapa irá calcular a distância par a par de todos os objetos, com base na medida uma função de dissimilaridade $F_d(o_i, o_j)$ qualquer. Após irá se calcular o fator v_j para cada objeto o_j, no qual o fator v_j é o soma das distâncias do objeto o_j para o objeto o_i divido com a soma das distancias de todos os objetos com o objeto o_i. Ao final do calculo dos fatores v_j, irá escolher os k objetos o_j com os menores v_j como *medoids* e criar os clusters atribuindo um cluster para cada *medoid* e atribuir o objeto o_j para o cluster que contenha o *medoid* mais próximo, por final calcule a soma das distâncias do *medoid* para cada objeto do cluster.

A segunda etapa começa encontrando um novo *medoid* que minimize a soma das distancias do *medoid* para cada objeto do *cluster*. após isso fará novamente a atribuição de cada objeto para o *cluster* no qual tem o *medoid* mais próximo e recalculará a soma das distancias do *medoid* para cada objeto do *cluster*. Essa etapa se repetirá até a soma das distancias do *medoid* para cada objeto do *cluster* ser igual da etapa anterior.

A seleção de *pivots* utilizará os *medoids* de cada *cluster* como *pivot*, pois como definido no trabalho , o papel dos *pivots* selecionados pelo *K-Medoids* é o de representar a área com maior densidade de uma parte do espaço aonde ele está localizado, sendo a grande diferença entre o *Pivoted Space Incremental Selection* e *Farthest-first traversal* é que o *k-medoids* tenta minimizar a distância média dos objetos de seu pivô mais próximo.

3.2 Busca

O conceito básico da busca como definida pelo autor CHÁVEZ *et al.* (2005), tem a ideia de prever a proximidade comparando com os objetos *pivots* e utilizando a ordem de suas distancias e não a proximidade real dos objetos a serem comparados. Essa etapa pode ser caracterizada como o passo de a partir do conjunto de *pivots* já selecionados pela função $F_{sp}(D)$,uma consulta q qualquer e o *dataset* D qualquer, será representada pela função $B(q, d, P)$ na qual definimos que irá a partir do conjunto de *pivots* previamente selecionados, irá retornar os documentos mais similares de q de forma ordenada FIGUEROA *et al.* (2015). Podemos separar a busca em duas etapas, a ordenação da listas de *pivots* e o calculo da distancia entre os objetos.

Algorithm 5: Seleção com Permutation Based Indexing

Data: Conjunto de Objetos O

Result: Conjunto P' de conjuntos Pivots ordenados

1 q = Representar a Consulta q em um Espaço Métrico

2 d_n = Calcular a distância do objeto q_n a cada $p_k \in P$

3 P_q = Ordenar o conjunto de *pivots* utilizando a distância d_n **for** $n = 1, ..., N$ **do**

4 $\quad q_{d_n}$ = comparar a distancia de P_q com o conjunto P_{d_n}

3.2.1 Ordenação das listas de pivots

Podemos definir a primeira etapa da busca, a ordenação da listas de *pivots*, como o ato gerar duas listas ordenadas dos conjuntos de *pivots*, no qual a primeira será do documento d, no qual essa lista será ordenada de forma crescente por distancia utilizando uma função de ordenação $L(d, P)$ qualquer e a segunda lista ordenada

será gerada de forma similar a primeira utilizando a mesma função de ordenação $L(q, P)$ só que utilizando a consulta q.

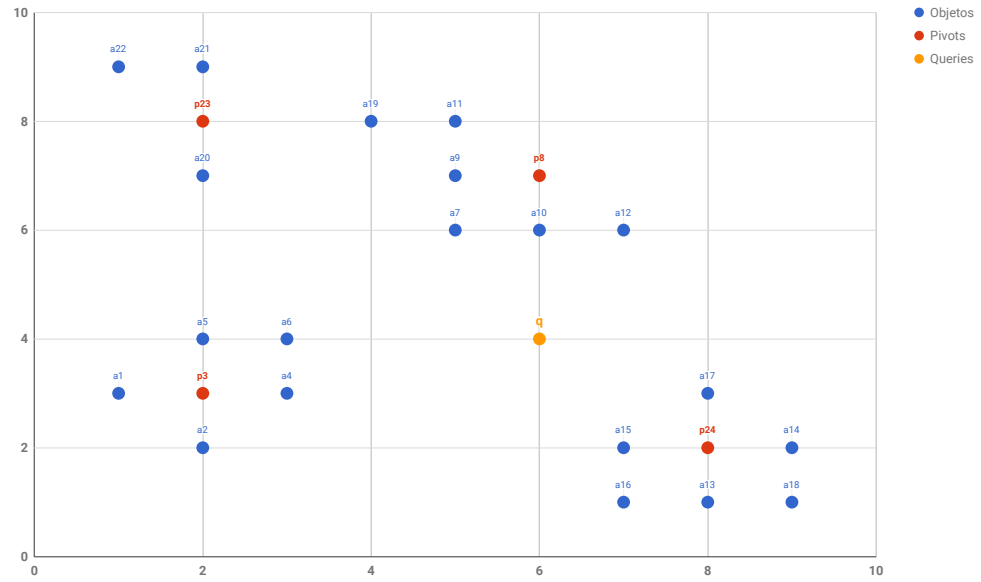

Figura 3.2 – Conjunto de *Pivots* e a Consulta q.

As funções de ordenamento de listas, utilizam funções de comparação de distâncias, no qual podem ser qualquer função que comparem a distancia entre dois objetos, e que respeitem os requisitos de uma função do espaço métrico (CHÁVEZ & NAVARRO, 2005) explicada na seção 2.4. Uma das funções funções de distância utilizadas no *Permutation Based Indexing* é a distância euclidiana, que pode ser definida como $\sqrt{\sum_{i=1}^{n}(a_i - b_i)^2}$ no qual a_i e b_i são pontos no espaço n dimensional dos objetos a e b (CHÁVEZ & NAVARRO, 2005). A partir de um exemplo criado na figura 3.2, podemos visualizar a ordenação de *pivots* de uma lista para o ponto q na equação 3.2, na qual F_D é a função de distância euclidiana

$$\begin{aligned} F_D(p_{23}, q) &= \sqrt{(4-8)^2 + (6-2)^2} = \sqrt{32} \\ F_D(p_{24}, q) &= \sqrt{(4-2)^2 + (6-8)^2} = \sqrt{8} \\ F_D(p_8, q) &= \sqrt{(4-7)^2 + (6-6)^2} = 3 \\ F_D(p_3, q) &= \sqrt{(4-3)^2 + (6-2)^2} = \sqrt{17} \\ L(q, P) &= \{p_{24}, p_8, p_3, p_{23}\} \end{aligned} \tag{3.2}$$

3.2.2 Cálculo de distância entre objetos

Após o processo de seleção e ordenação do conjunto P_q *pivots* para os objetos a serem comparados, deve se medir as diferenças entre as listas ordenadas, pois o *Per-*

mutation Based Indexing considera que lista similares, demonstram uma tendência que o documento d e a consulta q estão próximas, ou pelo menos na mesma região do espaço. Para esta comparação de listas ordenadas, utilizaremos métricas *Kendall Tau*, *Spearman's Footrule* e *Spearman's Rho* para a comparação de conjuntos ordenados que foram estudados no livro KENDALL (1955), no DIACONIS (1988) e o FAGIN *et al.* (2003) com um estudo mais condensado olhando para os problemas dos motores de busca.

Começando pela a métrica que aparece primeiro na literatura pelo autor KENDALL (1938) e motivado pela resolução do problema de correlação de *rank*, a Distância de *Kendall Tau* é uma métrica de comparação de listas ordenadas, onde se soma a distância usando a diferença entre as posições dos *pivots* nos Conjuntos. Quanto maior a discordância entre as posições desses conjunto maior será a distancia.

Algorithm 6: Seleção com Permutation Based Index

Data: Os Conjuntos P' P'' de *pivots* ordenados

```
1 score = 0
2 for x = 1, ..., k do
3     if P'[x] != P''[x] then
4         score = score + 1
5 ,
```

Result: Calculo de distancia entre 2 objetos

Segundo o FAGIN *et al.* (2003) podemos também definir a Distancia de *Kendall Tau* como o numero de inversões feita com a técnica de ordenação *Bubble Sort* para colocar as permutações de um conjunto na mesma posição de outro.

Assim como *Kendall Tau*, *Spearman's Footrule* também é uma técnica de comparação de de listas ordenadas, mas que diferentemente dela, *Spearman's Footrule* leva em consideração a diferença entre as posições dos pivots entre as listas ordenadas.

$$Spearman'sFootrule(l_q, l_d) = \sum_{p \epsilon P} |pos(p, l_q) - pos(p, l_d)| \tag{3.3}$$

Dado l_d e l_q como listas ordenada de *pivots* do documento d e consulta q que foram ordenadas pelas funções $L(q, P)$ e $L(q, P)$, no qual $d, q \in P$, sendo P o conjunto de *pivots*, podemos definir *Spearman's Footrule* como a soma das diferenças entre as posições dos *pivots* das listas ordenadas l_d e l_q.

Utilizada em diversos trabalhos da literatura junto com o *Permutation Based Indexing*, o *Sperman Rho Distance* é baseado no coeficiente de correlação de postos de *Spearman* ou ou rô de *Spearman*, no qual é pode ser definido como a correlação de que avalia a relação linear entre duas variáveis contínuas (FAGIN *et al.*, 2003).

$$SpearmanRho(l_q, l_d) = \sum_{p \epsilon P} \sqrt{|pos(p, l_d)^2 - pos(p, l_q)^2|} \tag{3.4}$$

Podemos definir o cálculo do *SpearmanRho* para a comparação das listas ordenadas de *pivots*, de forma para cada *pivot* $p_i \in P$, no qual P é o conjunto de *pivots* utilizado no *SpearmanRho* no qual todos os *pivots* que pertençam ao conjunto de P deveram pertencer ao as listas ordenadas l_d e l_q, no qual foram ordenadas pelas funções $L(q, P)$ e $L(q, P)$. O calculo será feito pelo somatório das diferenças de posições do *pivot* p_i elevada ao quadrado nas listas ordenadas l_d e l_q, fazendo a raiz quadrática da diferença (FAGIN *et al.*, 2003).

Podemos ver uma exemplificação de como é execução da comparação das listas ordenadas com *SpearmamRho*, *Spearman's Footrule* e *Kendall Tau* na equação 3.5, que além de demonstrar o funcionamento de cada uma, demonstra a diferença entre elas.

$$\begin{aligned}
L(d, P) &= [p_0, p_1, p_2, p_3] \\
L(q, P) &= [p_2, p_1, p_0, p_4] \\
SpearmanRho(q, d) &= \sqrt{(0^2+(-2)^2)+(1^2+(-1)^2)+(2^2+(-0)^2)+(3^2+(-4)^2)} \\
&= \sqrt{15} \\
Spearman'sFootrule(d, q) &= |0-2| + |1-1| + |2-0| + |3-4| \\
&= 5 \\
KendallTau(d, q) &= 1 + 0 + 1 + 1 \\
&= 3
\end{aligned} \tag{3.5}$$

3.2.3 Pruning

Uma forma de reduzir o tempo de execução da consulta busca segundo MOHAMED & MARCHAND-MAILLET (2015) é a etapa conhecida como *pruning*. Esta etapa consiste que partir de um conjunto P de *pivots* já selecionados, selecionamos os ϕ *pivots* mais próximos dos objetos a serem buscados.

A partir de um documento d e q, utiliza-se uma função qualquer $L(d, P)$ e $L(q, P)$ aonde ordena-se de forma decrescente os *pivots* mais similares, após isso, faz-se a seleção dos ϕ *pivots* mais próximos das listas ordenadas $L(d, P)$ e $L(q, P)$, gerando as listas ordenadas $L_\phi(d, P)$ e $L_\phi(q, P)$. A Equação 3.6, demonstra um exemplo do funcionamento do *Permutation Based Indexing* com *pruning* e o calculo de distância *Sperman Rho*, baseada na figura 3.2.

$$
\begin{aligned}
\phi &= 3 \\
L\left(a_{17}, P\right) &= \{p_{24}, p_3, p_8, \cancel{p_{23}}\} \\
L\left(q, P\right) &= \{p_8, p_{24}, p_3, \cancel{p_{23}}\} \\
L_\phi\left(a_{17}, P\right) &= \{p_{24}, p_3, p_8\} \\
L_\phi\left(q, P\right) &= \{p_8, p_{24}, p_3\}
\end{aligned}
\tag{3.6}
$$

$$
L_\phi\left(a_{17}, P\right) - L_\phi\left(q, P\right) = |0-1| + |1-2| + |2-0| + \cancel{|3-3|} = 4
$$

No exemplo 3.6 teremos uma quantidade de *pivots* igual a ϕ, já que o p_{23} é retirado da busca, e a operação de comparação de *pivots* é feita normalmente, mas no caso que os *pivots* podados sejam distintos, como no exemplo 3.7, teremos *pivots* que não poderiam ser utilizados, que é o caso do p_{23} e p_3, já que $p_{23} \notin L\left(q, P\right)$ & $p_{23} \in L\left(a_{17}, P\right)$ enquanto $p_3 \notin L\left(a_{17}, P\right)$ & $p_3 \in L\left(q, P\right)$.

$$
\begin{aligned}
\phi &= 3 \\
L\left(a_{19}, P\right) &= \{\cancel{p_{23}}, p_{24}, p_8, \cancel{p_3}\} \\
L\left(q, P\right) &= \{p_8, p_{24}, \cancel{p_3}, \cancel{p_{23}}\} \\
L_\phi\left(a_{19}, P\right) &= \{\cancel{p_{23}}, p_{24}, p_8\} \\
L_\phi\left(q, P\right) &= \{p_8, p_{24}, \cancel{p_3}\}
\end{aligned}
\tag{3.7}
$$

$$
L_\phi\left(a_{17}, P\right) - L_\phi\left(q, P\right) = \cancel{|0-3|} + |0-0| + |2-0| + \cancel{|3-2|} = 2
$$

Para a utilização desses *pivots*, os autores MOHAMED & MARCHAND-MAILLET (2015) propõem a utilização da abordagem *Quantized Ranking*, que é uma forma de quantificar a posição dos *pivots* no qual não estejam dentro de uma lista ordenada, aumentando a quantidade de *pivots* na busca. Antes definirmos o *Quantized Ranking* na seção é importante definimos o ato de *quantificar*, como o ato mensurar um valor desconhecido.

3.2.4 Quantized Ranking

Para a resolução dos problema de uma perda excessiva de *pivots*, tornando o *pruning* uma ação na qual gera-se uma degradação alta na qualidade da consulta, o autor MOHAMED & MARCHAND-MAILLET (2015) propõem a quantificação, das posições dos posição dos *pivots* no qual não estejam incluídos em uma das listas ordenadas $L_\phi\left(d, P\right)$ e $L_\phi\left(q, P\right)$, na qual são listas ordenadas em ordem decrescente

por sua similaridade em relação ao d e q, na qual após a ordenação sofreram o processo de *pruning*, selecionando os ϕ *pivots* mais próximos.

$$\sum_{\substack{p \in L(q,P) \\ p \in L(p,P)}} |b_{p_a} - b_{p_q}| + \sum_{\substack{p \in L(q,P) \\ p \notin L(a,P)}} |b_{p_a} - \beta| + \sum_{\substack{p \notin L(q,P) \\ p \in L(a,P)}} |b_{p_q} - \beta| \tag{3.8}$$

A equação do *Quantized Ranking* pode ser definida para todo $p \in P$, aonde P é o conjunto de todos os *pivots*, tal que cada $p \in L_\phi(d, P) \;\; \& \;\; p \in L_\phi(q, P)$ se faz o calculo de distância entre dois *pivots*, caso $p \in L_\phi(d, P) \;\; \& \;\; p \notin L_\phi(q, P)$, será utilizado um $\beta > \phi$ para quantificará a posição de p da lista $L_\phi(q, P)$, caso $p \in L_\phi(q, P) \;\; \& \;\; p \notin L_\phi(d, P)$, será utilizado um $\beta > \phi$ para quantificará a posição de p da lista $L_\phi(d, P)$. Essa definição pode ser visualizada na equação 3.8, na qual a e d são os documentos, p é um *pivot* b_{p_a} é a posição do *pivot* p na lista ordenada $L_\phi(a, P)$ e b_{p_q} é a posição do *pivot* p na lista ordenada $L_\phi(q, P)$.

Essa definição do *Quantized Ranking* feita pelo MOHAMED & MARCHAND-MAILLET (2015) é desenvolvida em cima da função *Spearman's Footrule* que é baseada na diferença simples das posições entre os vetores, mas o *Quantized Ranking* pode ser facilmente transportado para outras funções de Cálculo de distância entre objetos. Baseado nos Exemplos 3.6 e 3.7, veremos no exemplo 3.9 a utilização da abordagem *Quantized Ranking* com um $\beta = 4$

$$\begin{aligned} \phi &= 3 \\ \beta &= 4 \\ L_\phi(a_{19}, P) &= \{p_{23}, p_{24}, p_8\} \\ L_\phi(q, P) &= \{p_8, p_{24}, p_3\} \end{aligned} \tag{3.9}$$

$$\begin{aligned} L_\phi(a_{17}, P) - L_\phi(q, P) &= |0 - \beta| + |0 - 0| + |2 - 0| + |\beta - 2| \\ L_\phi(a_{17}, P) - L_\phi(q, P) &= |0 - 4| + |0 - 0| + |2 - 0| + |4 - 2| = 8 \end{aligned}$$

Capítulo 4

Proposta

Como já explicado na seção 3.1.2, os *pivots* são essenciais numa boa representação do seu conjunto de dados, e que a quantidade de *pivots* tem relação direta a qualidade do resultado. E como também já explicado na seção 3.2.4, pra uma redução na quantidade de *pivots* utilizando o *pruning* para recuperando somente os *pivots* mais relevantes, se passou a quantificar qual seria a posição de um determinado *pivot* de forma manter a qualidade da consulta mesmo reduzindo a quantidade de *pivots*, consequentemente tornando a consulta mais rápida e eficaz.

Nessa capítulo, apresentamos três variações da abordagem *Quantized Ranking*, o *Document Quantized Ranking* e o *Query Quantized Ranking* e a *Fixed Ranking*, variações no qual espera-se reduzir ainda mais o tempo da consulta sem uma grande degradação do resultado da busca.

4.1 Query Quantized Ranking

O *Query Quantized Ranking* é uma variação da técnica *Quantized Ranking* que irá quantificar a posição somente dos *pivots* que pertençam a lista de *pivots* da consulta, não quantificando a posição dos *pivots* que somente estejam na lista de *pivots* do Documento comparado.

$$\sum_{\substack{d \in L_\phi(q,P) \\ q \in L_\phi(d,P)}} |b_d - b_q| \quad + \sum_{\substack{d \in L_\phi(q,P) \\ q \notin L_\phi(d,P)}} |b_d - \beta| \tag{4.1}$$

Podemos assim definir o *Query Quantized Ranking* como uma técnica de quantificação que após a execução do *pruning* representado por $L_\phi(q,P)$ & $L_\phi(d,P)$ para uma consulta q qualquer e um documento d qualquer, utilizaremos um $\beta > \phi$ para quantificar a posição na lista $L(d,P)$ do *pivot* $p \in L_\phi(q,P)$ & $p \notin L_\phi(d,P)$, e para todo p no qual $p \in L_\phi(q,P)$ & $p \in L_\phi(d,P)$, se fará a diferença das posições

dos *pivots* em cada uma das listas, sendo essas posições representadas na equação 4.1 como b_d para *pivots* da lista $L_\phi(d, P)$ e b_q para *pivots* da lista $L_\phi(q, P)$.

4.2 Document Quantized Ranking

O inverso da abordagem *Query Quantized Ranking*, temos o *Document Quantized Ranking*, que também sendo uma variação da técnica *Quantized Ranking*, ele irá quantificar a posição somente dos *pivots* que pertençam a lista de *pivots* do documento, não quantificando a posição dos *pivots* que somente estejam somente na lista de *pivots* da Consulta.

$$\sum_{\substack{d \in L_\phi(q,P) \\ q \in L_\phi(d,P)}} |b_d - b_q| \; + \sum_{\substack{d \notin L_\phi(q,P) \\ q \in L_\phi(d,P)}} |b_d - \beta| \tag{4.2}$$

De forma muito similar ao *Query Quantized Ranking*, definimos o *Document Quantized Ranking* como uma técnica de quantificação que após a execução do *pruning* representado por $L_\phi(q, P)$ & $L_\phi(d, P)$ para uma consulta q_i qualquer e um documento d_i qualquer, utilizaremos um $\beta > \phi$ para quantificar a posição na lista $L(q, P)$ do *pivot* $p \in L_\phi(d, P)$ & $p \notin L_\phi(q, P)$, e para todo p no qual $p \in L_\phi(q, P)$ & $p \in L_\phi(d, P)$, se fará a diferença das posições dos *pivots* em cada uma das listas, sendo essas posições representadas na equação 4.2 como b_d para *pivots* da lista $L_\phi(d, P)$ e b_q para *pivots* da lista $L_\phi(q, P)$.

4.3 Fixed Ranking

O *Fixed Ranking* é uma abordagem que também tenta representar todos os subconjuntos de *pivots* não selecionados na ação do *pruning*, mas que diferentemente do *Quantized Ranking*, *Query Quantized Ranking* e o *Document Quantized Ranking*, se propõe uma solução que não tenta quantificar a posição dos *pivots* não selecionados. A ideia do *Fixed Ranking*, se quantifique utilizando um β qualquer para cada comparação dos *pivots* que não se enquadrem na interseção dos elementos selecionados pela a ação do *pruning*, ou seja quantifique a diferença entre as posições dos *pivots*, e não a posição do *pivot*.

$$\sum_{\substack{d \in L_\phi(q,P) \\ q \in L_\phi(d,P)}} |b_d - b_q| \; + \sum_{\substack{d \notin L_\phi(q,P) \\ q \in L_\phi(d,P)}} |\beta| + \sum_{\substack{d \in L_\phi(q,P) \\ q \notin L_\phi(d,P)}} |\beta| \tag{4.3}$$

Podemos definir a equação 4.3, que a partir da ação de *pruning* faz-se o calculo já explicado na seção 3.2.3 normalmente para todo *pivot* p no qual $p \in L_\phi(q,P)$ & $p \in L_\phi(d,P)$. Para todo *pivot* p, no qual $p \notin L_\phi(q,P)$ & $q \in L_\phi(d,P)$ ou $p \in L_\phi(q,P)$ & $q \notin L_\phi(d,P)$, utiliza-se um valor β qualquer.

Como não necessita-se de *pivots* para o calculo caso um *pivot* p qualquer, no qual $p \notin L_\phi(q,P) \vee p \notin L_\phi(d,P)$, Utilizaremos a diferença da quantidade de *pivots* da lista $L_\phi(d,P)$ com a quantidade de elementos da interseção das listas $L_\phi(q,P)$ e $L_\phi(d,P)$, na qual chamaremos de w, simplificando a equação do *Fixed Ranking*, agora representada de forma simplificada na equação 4.4.

$$\sum_{\substack{d \in L_\phi(q,P) \\ q \in L_\phi(d,P)}} |b_d - b_q| + (w - q)\,\beta \tag{4.4}$$

4.4 Diferenças entres as técnicas de Quantização

As técnicas de quantificação *Quantized Ranking*, *Query Quantized Ranking* e o *Document Quantized Ranking*, todos tem em comum a quantificação das diferenças das posições entre elementos da lista de *pivots*, mas essas formas de quantificar são distintas, e essas diferenças impactam de forma direta no tempo e na qualidade da consulta. Pra definirmos formalmente as diferenças entre essas abordagens, iremos avaliar a partir da diferença de cada uma, quais *pivots* serão utilizados nos melhores e piores cenários de cada uma das abordagens, finalizando com uma análise de um cenário médio, para cada abordagem.

Considerando as listas ordenadas de *pivots* $L(d,P)$ e $L(q,P)$ qualquer, podemos considerar que ao passar pelo ato do *pruning*, separamos os em quatro listas ordenadas. as listas $L_\phi(d,P)$ e $L_\phi(q,P)$ que são as listas dos *pivots* selecionados na ação de *pruning* e as listas ordenadas $\bar{L}_\phi(d,P)$ e $\bar{L}_\phi(q,P)$ que não foram selecionados na ação do *pruning*.

$$L(d,P) = \overbrace{p_0, p_1, p_2, p_3, p_4, p_5, p_6}^{L_\phi(d,P)} \overbrace{p_7, p_8, p_9}^{\bar{L}_\phi(d,P)}$$

$$L(q,P) = \underbrace{p_0, p_9, p_7, p_3, p_5, p_4, p_6}_{L_\phi(q,P)} \underbrace{p_2, p_8, p_1}_{\bar{L}_\phi(q,P)}$$

Figura 4.1 – Exemplo de separação em Listas ordenadas.

Para a demonstração dos cenários e suas consequências em cada abordagem distinta, visualizaremos cada listas ordenada como um conjunto. Para isto, alteraremos a nomenclatura de cada lista ordenada enquanto estivermos nos referindo a ela como um conjunto. Sendo as novas nomenclaturas definidas na equação 4.5.

$$\begin{aligned} L\left(d,P\right) &= I \\ L\left(q,P\right) &= J \\ L_{\phi}\left(d,P\right) &= I_{\phi} \\ L_{\phi}\left(q,P\right) &= J_{\phi} \\ \bar{L}_{\phi}\left(d,P\right) &= \bar{I}_{\phi} \\ \bar{L}_{\phi}\left(q,P\right) &= \bar{J}_{\phi} \end{aligned} \tag{4.5}$$

As listas ordenadas $L\left(d,P\right)$ e $L\left(q,P\right)$ como já explicadas anteriormente na seção 3.2, são o mesmo conjunto P de *pivots* que foram ordenados de forma crescente pela distância entre cada *pivot* p_i e o documento a_i, no caso da lista $L\left(d,P\right)$ ou q_i no caso da lista $L\left(q,P\right)$. Outra característica presente nesses conjuntos, é que os conjuntos $\bar{I}_{\phi}$ e $\bar{J}_{\phi}$ são conjuntos complementares de I_{ϕ} e J_{ϕ}, no qual a união entre eles serão iguais aos conjuntos I e J e suas interseções serão um conjunto vazio, já que em nenhum momento haverá um *pivot* qualquer selecionado e não selecionado ao mesmo tempo pela ação de *pruning*

$$\begin{aligned} I_{\phi} \cup \bar{I}_{\phi} &= I \\ J_{\phi} \cup \bar{J}_{\phi} &= J \\ I_{\phi} \cap \bar{I}_{\phi} &= \emptyset \\ J_{\phi} \cap \bar{J}_{\phi} &= \emptyset \\ I &= P \\ J &= P \end{aligned} \tag{4.6}$$

Podemos visualizar as relações entre os conjunto I_{ϕ}, $\bar{I}_{\phi}$, J_{ϕ} e $\bar{J}_{\phi}$ na figura 4.2, na qual representamos as interseções desses conjuntos, $I_{\phi} \cap J_{\phi} = A$, $I_{\phi} \cap \bar{J}_{\phi} = B$, $\bar{I}_{\phi} \cap J_{\phi} = C$ e $\bar{I}_{\phi} \cap \bar{J}_{\phi} D$. Será a partir dessas interseções, que podemos avaliar o comportamento da busca, para cada técnica de representação avaliando a cada cenário.

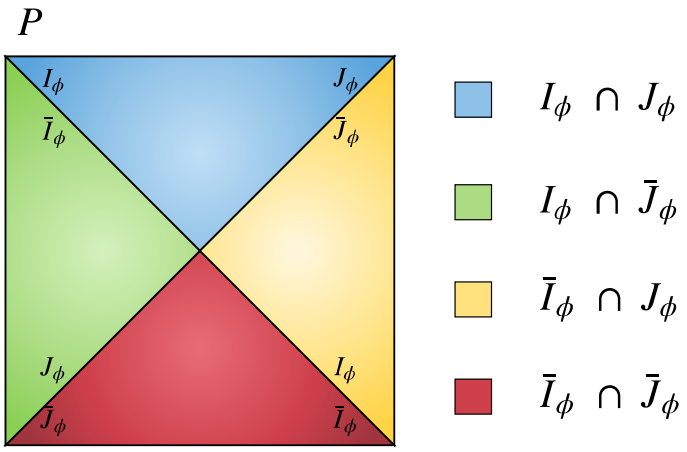

Figura 4.2 – Representação do conjunto P de *pivots*.

$$\textbf{Cenário Hipotético 1:} \quad p \in P \ : \ p \in I_\phi \cap J_\phi \tag{4.7}$$

Olhando para o cenário no qual chamaremos de cenário Hipotético 1, que podemos defini-lo a partir de qualquer *pivot* $p \in P$, onde $p \notin I_\phi$ e $p \notin J_\phi$, assim teremos um cenário hipotético, aonde não haverá busca em nenhuma das variações do *Permutation Based Indexing* com *pruning*, pois por definição $\bar{I}_\phi$ e $\bar{I}_\phi$ são conjuntos complementares I_ϕ, J_ϕ respectivamente, que foram retirados da busca.

$$\textbf{Cenário Hipotético 2:} \quad p \in P \ : \ p \in \bar{I}_\phi \cap \bar{J}_\phi \tag{4.8}$$

Um Cenário oposto ao cenário hipotético 1, no qual podemos definir a partir de qualquer *pivot* $p \in P$, onde $p \in I_\phi$ e $p \in J_\phi$, isso só seria seria possível em um cenário $I_\phi = J_\phi = P$, isso só seria possível caso $\phi = k$, aonde k é o numero de *pivots* do conjunto P, logo concluímos que para esse cenário a execução haverá a execução do *pruning* dentro do *Permutation Based Indexing*, que isso foge do escopo dessa seção que é verificar as diferenças de comportamento da execução do *Permutation Based Indexing* com *pruning* somente, ou com o *pruning* e alguma técnica de quantificação.

$$\textbf{Cenário 1:} \quad p \in P \ : \ p \in I_\phi \cap J_\phi \vee p \in \bar{I}_\phi \cap \bar{J}_\phi \tag{4.9}$$

Considerando o cenário 1 aonde o conjunto I_ϕ seja igual ao J_ϕ, poderemos definir esse cenário para qualquer *pivot* $p \in P$ aonde $p \in \bar{I}_\phi \cap \bar{J}_\phi lorp \in \bar{I}_\phi \cap \bar{J}_\phi$. Para esse cenário todas o *Permutation Based Indexing* com *pruning* utilizando qualquer uma

das técnicas de quantificação apresentadas nesse trabalho ou somente com o *Permutation Based Indexing* com *pruning* mas sem nenhuma técnica de identificação, terão com comportamentos idênticos, já que para esse cenário não teremos por definição nenhum *pivot* $p \notin I_\phi \ \& \in J_\phi$ ou $p \in I_\phi \ \& \notin J_\phi$.

$$\textbf{Cenário 2:} \quad p \in P \ : \ p \in I_\phi \cap J_\phi \ \vee \ p \in I_\phi \cap \bar{J}_\phi \ \vee p \ \in \bar{I}_\phi \cap J_\phi \tag{4.10}$$

Olhando para um cenário aonde exista pelo menos algum *pivot* p nas interseções dos conjuntos $I_\phi \cap J_\phi$, $I_\phi \cap \bar{J}_\phi$, $\bar{I}_\phi \cap J_\phi$ e $\bar{I}_\phi \cap \bar{J}_\phi$, a execução do *Permutation Based Indexing* somente com o *pruning* sem nenhuma técnica de quantificação, haverá busca somente no campo $I_\phi \cap J_\phi$ por causa não há nenhuma forma de quantificação para os *pivots* que não estejam nos conjuntos que sofreram a ação de *pruning*.

Olhando para mesmo cenário agora para a execução do *Permutation Based Indexing* com a técnica de quantificação *Quantized Ranking*, já que pela sua definição na seção 3.2.4, além das busca já possiveis de ser realizadas, caso exista algum *pivot* p que esteja na $p \in I_\phi \ \& \ p \notin J_\phi$ ou $p \notin I_\phi \ \& \ p \in J_\phi$ haverá uma quantificação para a posição do *pivot* que não esteja incluído em um dos conjuntos. logo definiremos que a busca para essa técnica como os conjuntos A, B e C.

O *Fixed Ranking*, apesar de ser uma técnica de quantifica diferença das posições do *pivots*, não a posição do *pivot* que não esteja em um dos conjuntos, analisando os cenários em relação quando há busca na execução do *Permutation Based Indexing*, ele se comportará de forma igual a mesma execução do *Permutation Based Indexing* utilizando o *Quantized Ranking*, pois segundo sua definição já vista na seção 4.3, a técnica irá quantificar a diferença de posições quando um *pivot* $p \in I_\phi \ \& \ p \notin J_\phi$ ou $p \notin I_\phi \ \& \ p \in J_\phi$.

Diferentemente da Execução do *Permutation Based Indexing* com a técnica *Quantized Ranking*, a execução do *Permutation Based Indexing* com a técnica quantização *Document Quantized Ranking*, haverá uma busca mais restrita, já que como já explicitados nas seções 4.2 haverá quantificação para a posição do *pivot* p somente quando, $p \in I_\phi \ \& \ p \notin J_\phi$, e de forma similar, a execução do *Permutation Based Indexing* com a técnica quantização *Query Quantized Ranking* haverá uma redução na busca, havendo somente a quantificação para a posição do *pivot* p somente quando $p \notin I_\phi \ \& \ p \in J_\phi$.

$$\textbf{Cenário 3:} \quad p \in P \ : \ p \in I_\phi \cap \bar{J}_\phi \ \vee p \ \in \bar{I}_\phi \cap J_\phi \tag{4.11}$$

Ao analisar a busca do cenário 3, verificamos que a execução do *Permutation Based Indexing* com o *pruning* sem nenhuma técnica de quantificação, não haverá busca, pois pela definição do *pruning* na seção 3.2.3, é possível utilizar na busca somente os *pivots* $p \in I_\phi \ \& \ p \in J_\phi$. Já a execução do *Permutation Based Indexing* com a técnica de quantificação *Quantized Ranking*, terá um resultado bem distinto, já que haverá quantificação pra qualquer *pivot* que $p \in I_\phi$ ou $p \in J_\phi$ como descrito na seção 3.2.4.

Fazendo a mesma análise da busca no cenário 3 para a execução do *Permutation Based Indexing* com a técnica de quantificação *Document Quantized Ranking*, veremos que pela definição na seção 4.2 a busca ocorrerá quando *pivot* p qualquer, no qual $p \in I_\phi \ \& \ p \notin J_\phi$ fazendo a quantização da posição do *pivot* p na lista ordenada na $L\,(q, P)$. O mesmo ocorrerá para a execução do *Permutation Based Indexing* com a técnica de quantificação *Query Quantized Ranking* com a diferença que a busca ocorrerá para um *pivot* p qualquer no qual $p \notin I_\phi \ \& \ p \in J_\phi$ fazendo a quantização da posição do *pivot* p na lista ordenada na $L\,(d, P)$.

$$\begin{cases} \textbf{Cenário 4:} \quad p \in P \ : \ p \in I_\phi \cap J_\phi \ \vee \ p \in I_\phi \cap \bar{J}_\phi \ \vee p \ \in \bar{I}_\phi \cap J_\phi \ \vee \ p \in \bar{I}_\phi \cap \bar{J}_\phi \\ \textbf{Cenário 5:} \quad p \in P \ : \ p \in I_\phi \cap \bar{J}_\phi \ \vee p \ \in \bar{I}_\phi \cap J_\phi \ \vee \ p \in \bar{I}_\phi \cap \bar{J}_\phi \end{cases} \tag{4.12}$$

O cenário 4 é bem similar ao cenário 2, tendo como diferença que existirá pelo menos um *pivot* $p \in P$, no qual $p \in \bar{I}_\phi \cap \bar{J}_\phi$, essa diferenças entre os cenários não se refletirá algum efeito na busca, já que a diferença entre eles parte da interseção dos conjuntos aonde não ocorrerá a busca em nenhuma das execuções do *Permutation Based Indexing*, como já visto no cenário hipotético 1. Essa situação será similar ao cenário 5 comparado ao cenário 3.

$$\begin{cases} \textbf{Cenário 6:} \quad p \in P \ : \ p \in I_\phi \cap \bar{J}_\phi \\ \textbf{Cenário 7:} \quad p \in P \ : \ p \ \in \bar{I}_\phi \cap J_\phi \\ \textbf{Cenário 8:} \quad p \in P \ : \ p \in I_\phi \cap J_\phi \ \vee p \ \in \bar{I}_\phi \cap J_\phi \\ \textbf{Cenário 9:} \quad p \in P \ : \ p \in I_\phi \cap J_\phi \ \vee \ p \in I_\phi \cap \bar{J}_\phi \end{cases} \tag{4.13}$$

Já o cenário 6, aonde no qual todo *pivot* $p \in P$, no qual $p \in I_\phi \cap \bar{J}_\phi$, se torna impossível, já que por definição vista na seção 3.2.3, o o conjunto I_ϕ tem ϕ elementos, e $\bar{J}_\phi$ tem $k - \phi$ *pivots*, sendo k a quantidade de elementos no conjunto P, pois como já visto, $\bar{J}_\phi$ é um conjunto complementar de J_ϕ. Nesse momento temos um problema

já que para todos *pivots* estarem contidos no $I_\phi \cap \bar{J}_\phi$ os conjuntos $I_\phi = \bar{J}_\phi$, logo $\bar{J}_\phi = \phi$ e sabendo que $\bar{J}_\phi = k$, $k = \phi - \phi$, $P = \emptyset$, logo não existe *pivots* no $I_\phi \cap \bar{J}_\phi$ e não existirá busca neste cenário.

Já o cenário 8 torna-se impossível de ocorrer, já que pela sua própria definição vista na equação 4.13, teremos p pertencendo a $I_\phi \cap J_\phi$ ou $\bar{I}_\phi \cap J_\phi$, que pode ser simplificado para J_ϕ, e que para esse cenário ocorrer $\phi = k$, sendo k o numero de *pivots* do conjunto P, caindo no cenário similar ao do cenário hipotético 2. De forma similar irá ocorrer ao cenário 9, aonde teremos p pertencendo a $I_\phi \cap J_\phi$ ou $I_\phi \cap \bar{J}_\phi$, que pode ser simplificado para I_ϕ, gerando novamente um cenário muito similar ao do cenário hipotético 2.

$$\begin{cases} \textbf{Cenário 10:} & p \in I_\phi \cap \bar{J}_\phi \vee \ p \in \bar{I}_\phi \cap \bar{J}_\phi \\ \textbf{Cenário 11:} & p \ \in \bar{I}_\phi \cap J_\phi \ \vee \ p \in \bar{I}_\phi \cap \bar{J}_\phi \end{cases} \tag{4.14}$$

Ocorrerá de forma próxima ao que ocorreu aos cenários 8 e 9 para o cenário 10, já que pela sua definição vista na equação 4.14, teremos p pertencendo a $\bar{I}_\phi \cap \bar{J}_\phi$ ou $I_\phi \cap \bar{J}_\phi$, que simplificando $\bar{I}_\phi$, isso só será possível como visto de forma similar ao cenário hipotético 1, se $\phi = 0$, e caso isso ocorra o $I_\phi = \emptyset$, entrando novamente em um similar ao cenário hipotético 1, na qual não haveŕá busca. O mesmo ocorrerá no cenário 11 já que teremos teremos p pertencendo a $\bar{I}_\phi \cap \bar{J}_\phi$ ou $\bar{I}_\phi \cap J_\phi$, , que simplificando $\bar{I}_\phi$, isso só será possível já explicado não haverá busca.

A partir das definições de todos os cenários e de cada comportamento em relação a busca de cada execução do *Permutation Based Indexing* com somente o *Pruning* ou com as 3 técnicas de quantificação, podemos avaliar a quantidade de *pivots* de cada busca, utilizando os conjuntos utilizados para a mesma. Para esse comparativo representaremos na tabela 4.1 a interseção dos conjuntos $I_\phi \cap J_\phi = A$, $I_\phi \cap \bar{J}_\phi = B$, $\bar{I}_\phi \cap J_\phi = C$.

	Pruning	**QR**	**DQR**	**QQR**	**FR**
Cenário 1	$\|A\|$	$\|A\|$	$\|A\|$	$\|A\|$	$\|A\|$
Cenário 2	$\|A\|$	$\|A\| + \|B\| + \|C\|$	$\|A\| + \|B\|$	$\|A\| + \|C\|$	$\|A\| + \|B\| + \|C\|$
Cenário 3	$\emptyset$	$\|B\| + \|C\|$	$\|B\|$	$\|C\|$	$\|B\| + \|C\|$
Cenário 4	$\|A\|$	$\|A\| + \|B\| + \|C\|$	$\|A\| + \|B\|$	$\|A\| + \|C\|$	$\|A\| + \|B\| + \|C\|$
Cenário 5	$\emptyset$	$\|B\| + \|C\|$	$\|B\|$	$\|C\|$	$\|B\| + \|C\|$

Tabela 4.1 – *Pivots* utilizados na consulta.

O comparativo visto na tabela 4.1, vemos que a execução do *Permutation Based Indexing* com somente o *pruning*, representado na tabela somente com *pruning*, demonstrando que o *pruning* dependendo do cenário aonde as poderá, até mesmo não ocorrer buscas, como no cenário 3 e 5, sendo que no cenário 2 e 4 ele faŕa uma busca com uma quantidade inferior de *pivots* em relação as execuções com as

técnicas de quantificação, e o cenário 1, ele terá o mesmo comportamento em relação as outras técnicas.

Já execução do *Permutation Based Indexing* com a técnica de quantificação *Quantized Ranking*, representado na tabela 4.1 como $Q.R$, terá um comportamento bem distinto em relação a execução do *Permutation Based Indexing* somente com *Pruning*, uma maior quantidade de *pivots* nos cenários 2,3,4 e 5 comparados a execução do *Permutation Based Indexing* com somente o *pruning* ou com também as técnicas de *Document Quantized Ranking* e *Query Quantized Ranking*. no cenário 1 como já dito ele se comportará de forma igual as outras.

De forma idêntica ao do *Quantized Ranking*, o a execução do *Permutation Based Indexing* com o *Fixed Ranking*, funcionará em relação a quantidade de *pivots* na busca em qualquer cenário, sendo sua diferença já dita na seção 4.3, que ele não quantificará a posição de um *pivot* e sim a diferença entre a posições dos pivots gerando um comportamento na consulta não em relação a quantidade *pivots* utilizados e sim na forma do calculo da distancia.

Capítulo 5

Experimentos

Nesta Seção iremos avaliar de forma empírica o *Recall*, tempo de consulta e vazão para o *Permutation Based Indexing* e suas variações de seleção de *pivots* afim de uma escolha de parâmetros, a partir disso avaliaremos o *pruning* e as diferentes técnicas de quantificação apresentadas nesses trabalho visando uma avaliação empírica das abordagens e posteriormente uma comparação com os cenários obtidos na seção 4.4 e por fim compararemos o *Permutation Based Indexing* com o *BM-25* e abordagem *Min-Max*, que pertence a família de abordagens *Locality Sensitive Hashing*.

Para a primeira comparação avaliaremos a variação da quantidade de k *pivtos* e diferença gerada em relação ao tempo de consulta, tempo de indexação e o *Recall*. Após esta comparação avaliaremos a variação do pruning com a técnica *Quantized Ranking* e as propostas dessa dissertação, o *Fixed Ranking*, *Document Quantized Ranking* e *Query Quantized Ranking*. Essa avaliação tem por objetivo medir a degradação enquanto se reduz a quantidade de *pivots* da técnica utilizando o *pruning*, e o tempo de consulta de cada variação do *Permutation Based Indexing* gerado pela mudança de da utilização da técnica de representação. E a ultima avaliação se dará pela comparação do *Permutation Based Indexing* com as técnicas mais utilizadas na literatura para busca por similaridade no contexto de plágio *Locality Sensitive Hashing* e *BM-25*. Essa avaliação terá como métricas o *Recall*, tempo de consulta e a vazão.

5.1 Recuperação Heurística do Plágio Extrínseco

Para a comparação entre as diferentes variações do *Permutation Based Indexing*, *BM25* e Técnicas da Família do *Locality Sensitive Hashing*, utilizaremos um experimento já utilizado anteriormente na comparação da técnica de técnicas da família *Locality Sensitive Hasing* e na própria etapa de busca heurística nos trabalhos DUARTE (2017); JI *et al.* (2013a). Este experimento tem como avaliar a busca heu-

rística como um todo, que tem como objetivo de reduzir o número de documentos para comparação de pares de documentos na próxima etapa a análise detalhada.

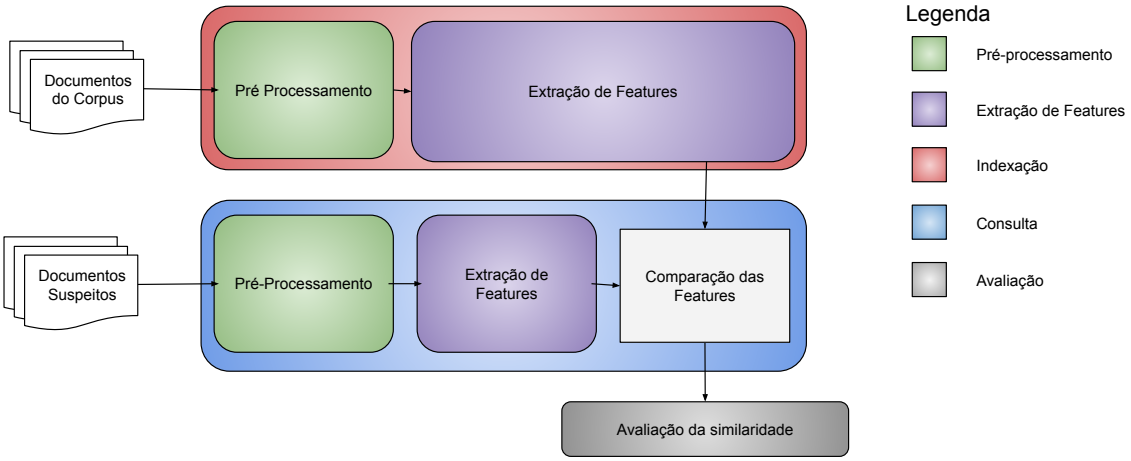

Figura 5.1 – Visualização da Recuperação Heurística do Plágio Extrínseco (DUARTE, 2017).

O experimento de Recuperação Heurística do Plágio Extrínseco, pode ser separado como visto na figura 5.1, em quatro partes. O pré processamento, extração de *features*, comparação entre as *Features* e avaliação da similaridade. O pré-processamento envolverá qualquer processamento do texto anterior a qualquer tipo de transformação por parte de qualquer técnica avaliada, esta etapa ela deve ser similar a qualquer técnica, afim de manter a equidade para a comparação entre as técnicas.

A etapa de extração de *feature* e comparação entre as *features* pode ser considerada etapas que se diferenciam a cada técnica, tendo como padrão a saída e entrada de cada etapa. É a partir disso podemos visualizar na figura 5.2 como e aonde ocorrerá cada etapa do *Permutation Based Indexing* no experimento.

As etapas do experimento para o *Permutation Based Indexing*, podem ser separadas na parte de extração de *features* como seleção de *pivots* e geração das listas de *pivots* para o documento e o *pruning*. A seleção de *pivots* e a própria etapa de seleção de *pivots* do *Permutation Based Indexing*, já a geração deas listas de *pivots* para o documento e o *pruning*, se refere após termos os *pivots* selecionados se executará para cada documento d no qual $d \in D$ de *dataset*, a função de ordenação e *pruning* $L_\phi\,(d, P)$.

Será dividida a análise dos experimentos em três etapas, a primeira a extração de *Feature* com as etapas de de seleção de *pivot* e a geração de listas de *pivots* para o documento e o *pruning*, no qual teremos de fazer a melhor escolha de parâmetros para execução das próximas etapas. Após isso avaliaremos o tempo de consulta com o *recall* para as diferentes técnicas de quantificação, afim de avaliarmos s o de forma

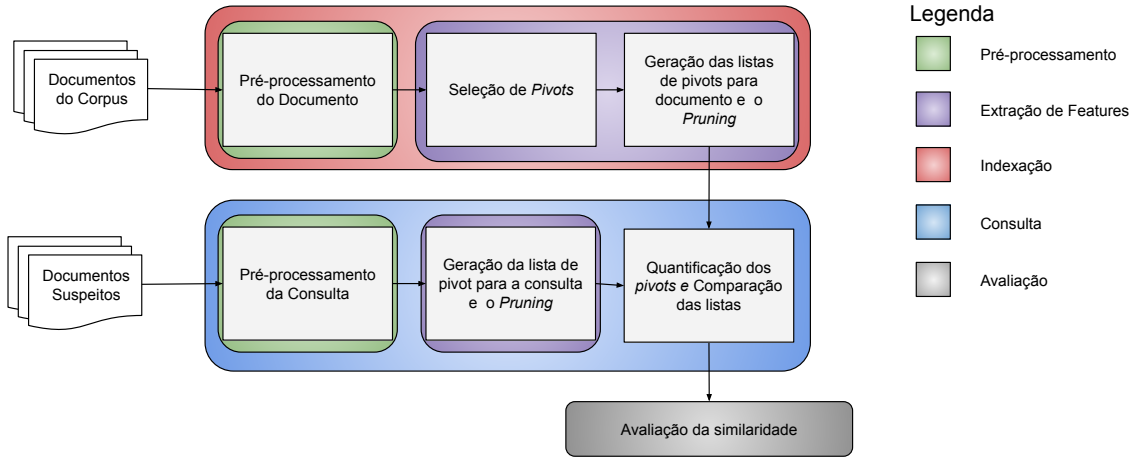

Figura 5.2 – Visualização da Recuperação Heurística do Plágio Extrínseco para especificamente para o *Permutation Based Indexing* (DUARTE, 2017).

empírica as execuções de cada técnica, podendo analisar com os cenários visto na seção 4.4. Por fim avaliaremos a execução do *Permutation Based Indexing* como um todo em relação ao *BM25* e as técnicas da família *Locality Sensitive Hashing*.

5.1.1 Dataset

Para o experimento, foi utilizado o *dataset* PAN-11, que foi criado para a competição *Uncovering Plagiarism, Authorship, and Social Software Misus*(*PAN*), no qual existe as tarefas de detecção de plágio Extrínseco e intrínseco (POTTHAST *et al.*, 2011b). O *PAN-11* contém 26.939 documentos, com 61.064 trechos de texto demarcados como plágio, um vocabulário extraído de 686.668.842 ocorrências de palavras com 1.207.741 palavras distintas. Além disso, os documentos no PAN11 são classificados como 50% documentos fontes (documentos originais), dos quais o texto é plagiado, 25% como falsos positivos (documentos suspeitos sem nenhum caso de plágio) e 25% verdadeiros positivos, no qual são documentos plagiados que devem ser encontrados (POTTHAST *et al.*, 2011b).

5.1.2 Métricas de Avaliação

Como métricas de avaliação dos 3 experimentos, será utilizado para avaliar a qualidade das técnicas será utilizado a métrica *Recall* que avalia a quantidade de documentos relevantes, podendo ser definido documentos relevantes retornados P sobre todos os documentos relevantes, sendo que documentos relevantes pode ser definido como a união dos documentos relevantes retornados P e documentos relevantes não retornados F_P.

$$Recall \ = \ \frac{P}{P \cup F_P} \tag{5.1}$$

Outra métrica utilizada será a vazão, que podemos defini-la como a quantidade de documentos que a etapa de indexação, que é formada pelas etapas de pré-processamentos do *dataset*, a etapa de seleção de *pivots* e a geração das listas de *pivots* para documento e o *Pruning* no caso de uma técnica do *Permutation Based Indexing*. A finalidade dessa métrica é avaliar todo o tempo de processamento até começar a realmente até a avaliação da primeira consulta. E a ultima métrica é o tempo de cada consulta em segundos.

5.1.3 Implementação

Para o experimento foi utilizado uma maquina com dedicação exclusiva ao experimento no tem por características, um processador Intel *i5-5575R*, com 16Gb *DDR-3* com Frequência de 1333 *Mhz*, um Disco Rígido de 500 GB com velocidade do eixo de 7200 *RPM*, Sistema Operacional Ubuntu 16.04, executando o gerenciador de contêiner *Docker* na versão *17.06.2-ee-15*. Para a implementação e execução dos experimentos foi utilizado o Python versão 3.5.2, instalado pelo gerenciador de pacotes Anaconda na versão 4.1.1 e pip 8.1.2 com as bibliotecas *Numpy* 1.15.1, *scikit-learn* 0.17.1 e *scipy* 0.17.1.

5.2 Experimento 1

O experimento 1, tem por objetivo, escolher os parâmetros aceitáveis para a execução do *Permutation Based Indexing* no dataset *PAN-11*, avaliaremos como parâmetro para qual função de seleção de *pivot*, quantidade *pivots* p, e variação do *Threshold* θ para as técnicas *Pivoted space incremental selection* e o *Farthest-first traversal*.

Assim como desenvolvidos nos trabalhos de DUARTE (2017); JI *et al.* (2013a); VIEIRA (2016), iremos fixar os parâmetros de Pré-processamento, as formas de comparação par-a-par e comparação de distancia entre as listas de *pivots*, pois o trabalho foca somente demonstração da viabilidade da técnica *Permutation Based Indexing* e as contribuições na parte de quantificação e não na otimização dos parâmetros para o *dataset*.

Neste modo utilizaremos *Jaccard* para a comparação par-a-par em todas as técnicas de seleção de *pivot*, na comparação da distancia do documento d e consulta q para cada *pivot* p. Essa escolha se da pois é encontrado na literatura outros trabalhos com detecção de plágio e reuso de texto utilizando a mesma métrica DU-

ARTE (2017); VIEIRA (2016). Utilizaremos *SpearmanRho* como comparação de distancia entre listas, pois temos já é bem consolidado a grande semelhança das técnicas *Spearmam Footrule* com *SpearmamRho*, e da vantagem delas em relação a *Kendall Tau* quando essas técnicas são utilizadas com o o *Permutation Based Indexing* (CHÁVEZ & NAVARRO, 2008; FIGUEROA *et al.*, 2015; MOHAMED & MARCHAND-MAILLET, 2015; VON LÜCKEN *et al.*, 2015).

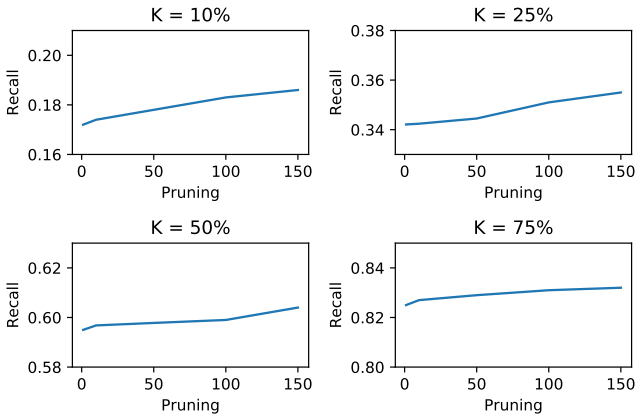

Figura 5.3 – Avaliação do θ para o *Farthest-first traversal* em relação ao *Recall*.

Avaliando o *treshold* θ em relação ao tempo de consulta e o *Recall*, fixa-se o numero de *pivots* igual a 100 sem utilizar o *pruning*. Podemos ver um pequeno aumento constante do *recall* a partir do θ igual a 50 até o $\theta = 50$, sendo que esse aumento é próximo de 0.1 comparando o menor valor de $\theta = 50$ e , $\theta = 150$. Para os valores entre $\theta = 1$ e $\theta = 50$ existe uma pequena variação de comportamento quando o numero de documentos retornados representa 10% e 25% do *dataset*, temos uma aumento quase imperceptível na figura 5.3, mas quando vemos para 50% e 75% do *dataset* retornado, vemos uma aumento constante bem parecido com o resto da curva.

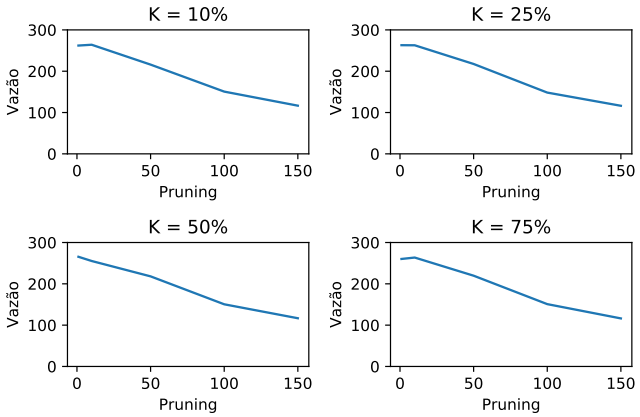

Figura 5.4 – Avaliação do θ para o *Farthest-first traversal* em relação a Quantidades de documentos indexados por segundo.

Agora avaliando as mesmas variações mas olhando para a vazão, é perceptível uma queda na vazão, quando o $\theta = 1$ o valor é próximo de 260 documentos, caindo a a valores próximos de 115 documentos indexados por segundo independente da quantidade de documentos retornados do *dataset*.

Avaliando o *treshold* θ em relação ao tempo de consulta e o *Recall* agora para a técnica de quantização *Pivoted space incremental selection*, de forma igual a execução para o *Farthest-first traversal*, fixa-se o numero de *pivots* igual a 100 sem a utilização do *pruning*. Podemos ver na figura 5.6 um pequenas variações do *recall* de forma inconstante para o partir do θ até 50 e um aumento é próximo de 0.1 comparando o menor valor de θ $= 50$ e , θ $= 150$, quando o numero de documentos retornados representa 25%, 50% e 75% do *dataset*, e no caso de 10% dos documentos retornados vemos uma leve aumento no aumento do θ de 1 a 150.

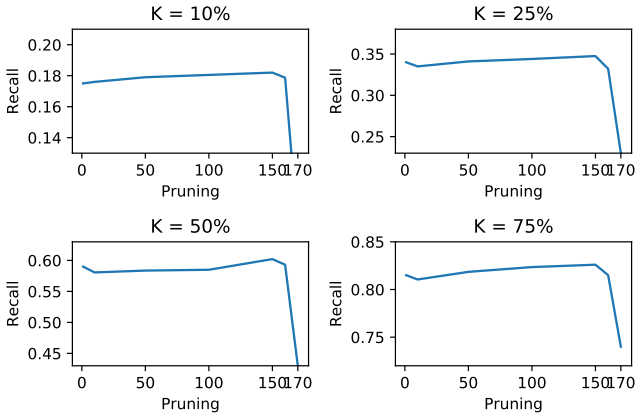

Figura 5.5 – Avaliação do θ para o *Pivoted space incremental selection* em relação ao *Recall*

Para o $\theta > 150$ começamos a ver uma perda significativa no *Recall*, que podem ser explicadas por causa que não foi possível selecionar os 100 *pivots* para a consulta, por causa do *Threshold* muito alto, e isso é possível de ser avaliado já que podemos ver na tabela 5.1 a quantidade de *pivots* recuperados quando o $\theta = 150$, $\theta = 160$ e $\theta = 170$.

Threshold θ	K
150	100
160	65
170	6

Tabela 5.1 – Quantidade K de *pivots* selecionados com *Threshold theta na execução do Pivoted space incremental selection*

Analisando as mesmas variações mas olhando para a vazão, é perceptível na figura 5.6 uma aumento na vazão, diferentemente do que ocorre no mesmo experimento para o *Farthest-first traversal*, quando o $\theta = 1$ o valor é próximo de 1.5

documentos, aumentando até a a valores próximos de 15 documentos indexados por segundo quando utilizamos o $\theta = 150$ independente da quantidade de documentos retornados do *dataset*. Uma das explicações que podem ser dadas para esse aumento, é que como explicado na seção 3.1.2.3, ele tem sempre quando altera um *pivot*, ele tem que adicionar adicionar o novo *pivot* na posição correta da lista, na qual mantenha a lista de *pivots* ordenados.

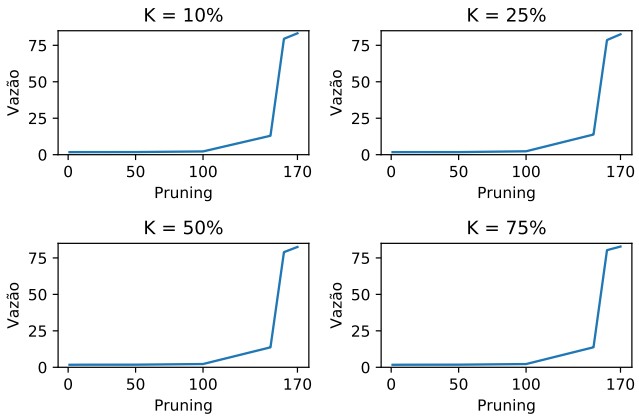

Figura 5.6 – Avaliação do θ para o *Pivoted space incremental selection* em relação a Quantidades de documentos indexados por segundo.

A partir que já foi analisado as variações do *threshold* para as técnicas de seleção de *Farthest-first traversal* e *Pivoted space incremental selection*, analisaremos a quantidade de *Pivots* para cada técnica de seleção de *pivot* distanta, comparando o *recall* e a vazão entre essas técnicas. Para essa execução será utilizado o *threshold* $\theta = 1$ para a técnica *Farthest-first traversal*, já que não houve uma perda significativa no *recall* e um ganho na vazão, já para a técnica *Pivoted space incremental selection*, foi utilizado o *threshold* $\theta = 150$ pois houve uma pequeno ganho no *recall* e um significativo aumento no numero de documentos indexados por segundo e para esse experimento não será utilizado o *pruning*.

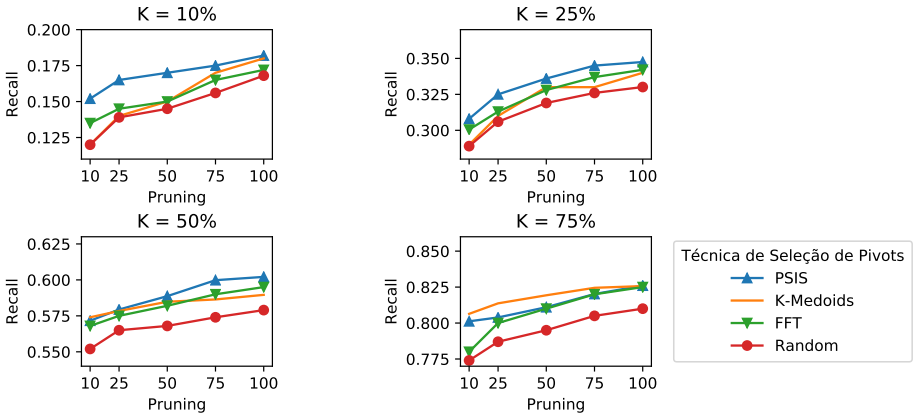

Figura 5.7 – Avaliação das técnicas seleção de *pivot* em relação ao *recall*

Analisando as técnicas de seleção de *pivot* em relação ao *Recall* podemos perceber uma curva crescente para todas as técnicas de seleção de *pivots*, quando se aumenta a quantidade de *pivots*, sendo que esse aumento vai se reduzindo quanto maior é quantidade *pivots*. Outra coisa que degrada o aumento do *recall* para as técnicas de seleção de *pivot* é o tamanho da quantidade de documentos retornados na busca, quanto maior a quantidade de documentos retornados na busca, menor é o aumento do *recall* quando aumenta de *pivots*. Comparando as técnicas em relação ao *Recall*, é visto uma sobreposição de todas as técnicas, com uma vantagem quase insignificante ao *Pivoted space incremental selection*, que está um pouco melhor quando é retornado 25% ,50% e 75% dos documentos na busca e a técnica seleção de *Randomized* tendo um recall um pouco abaixo em relação as outras técnica, mas também quase imperceptível.

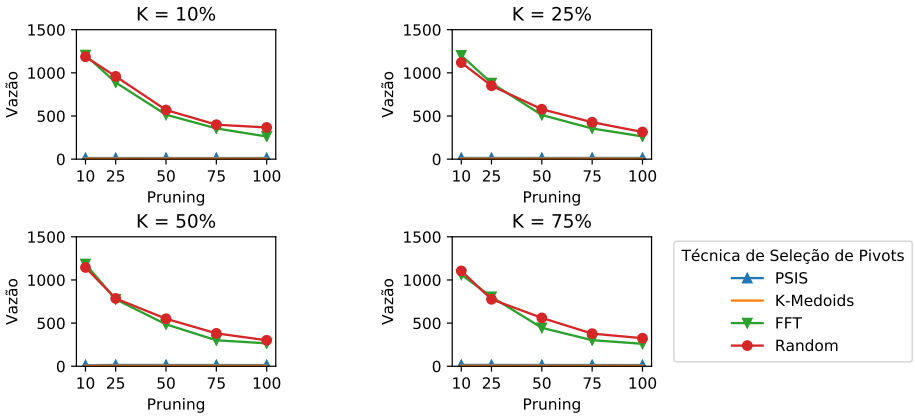

Figura 5.8 – Numero de documentos indexados por segundo para cada técnica de seleção de *pivot*.

Avaliando em relação ao tempo de indexação, vemos uma discrepância bastante elevada, tendo as técnicas*Farthest-first traversal* e *Randomized* independente da quantidade de *pivots*, é sempre superior ao numero de documentos indexados por segundo, sendo a diferença variando na quantidade de *pivots*, pois*Farthest-first traversal* e *Randomized* sofrem uma perda significativa da quantidade de documentos retornados por segundo, enquanto as técnicas *K-Medoids* e não sofrem isso *Pivoted space incremental selection*, isso pode ser explicado pois essas duas técnicas tem uma maior grau de sofisticação na escolha do seu *pivots* como apresentado na 3.1.2.

Analisando o tempo de consulta visto na figura 5.9, verificamos que ele aumenta quando aumenta a quantidade de *pivots* e de forma igual as técnicas de seleção de *pivot*, pois como visto na seção 5.2, terá relação somente as técnicas de quantificação, *pruning*, pré-processamento das consultas e quantidade de *pivots*, sendo ele o única variável do experimento 1, as variações de *pruning* e das técnicas de quantificação serão vistas no experimento 2..

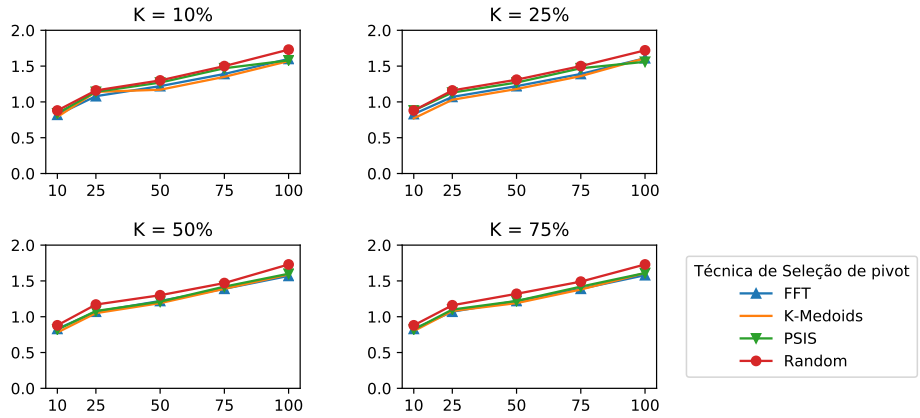

Figura 5.9 – Avaliação das técnicas seleção de *pivot* em relação ao tempo de consulta

5.3 Experimento 2

No experimento 2 se analisará, as variações de *pruning* e as técnicas de quantificação apresentadas na seção 3.2.4 e no capítulo 4, em relação ao *Recall* e a possível degradação ao reduzir a quantidade de *pivots* usando o *pruning* e o tempo de consulta de cada variação da técnica do *Permutation Based Indexing*, tendo uma análise empírica dos efeitos vistos na seção 4.4 .

Para esse experimento fixa-se o pré-processamento com a representação não booleana, sem utilização n-grama, utilizando o *TF-IDF*, com a quantidade miníma 2 documentos de forma idêntica ao do experimento 1. Será utilizado *Jaccard* para a comparação par-a-par na técnicas de seleção de *pivotFarthest-first traversal* com *threshold* $\theta = 1$ e com 75 *pivots*, e também será utilizado o *Jaccard* na comparação da distancia do documento d e consulta q para cada *pivot* p, pelos mesmos motivos explicados já explicados na seção 5.2.

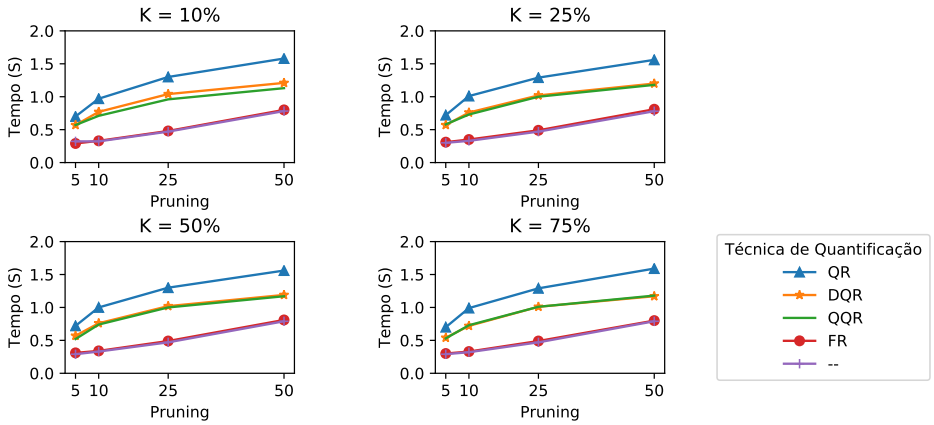

Figura 5.10 – Avaliação das técnicas de quantificação com a variação do *pruning* em relação ao tempo de consulta.

Analisando os resultados do experimento do tempo de consulta apresentados na figura 5.10, no qual *QR* é a representação de *Quantized Ranking*, *LQR* é igual *Query Quantized Ranking*, *RQR* é *Document Quantized Ranking*, *FR* é Fixed Ranking e – é quando não foi utilizada nenhuma técnica de quantificação. É visível que o tempo de consulta do *Quantized Ranking* é superior ao das outras técnicas de quantificação sendo que essa distancia se aproxima quando se reduz de 10 *pivots* para 5 *pivots* o *pruning*, sendo que este comportamento indefere na quantidade de documentos retornados pela consulta.

A técnica *Query Quantized Ranking* e *Document Quantized Ranking* estão com o tempo de consulta sobrepostos na figura *2* indiferentemente da quantidade de documentos retornados pela consulta. O tempo de consulta do *Query Quantized Ranking* e *Document Quantized Ranking* são inferiores ao do *Quantized Ranking* mas são superiores ao do *Fixed Ranking* e de quando não se utiliza nenhuma técnica de quantificação. Já a do *Fixed Ranking* se mostrou mais rápido que as técnicas de quantificação e com tempo de consulta sobreposto quando não se utiliza nenhuma técnica de quantificação.

A diferença entre os tempos de consulta entre as diferentes técnicas de quantificação era esperada, pois como se viu na seção 4.4, as diferentes técnicas de quantificação tem seus diversos cenários de busca, tendo o *Quantized Ranking*, junto ao *Fixed Ranking* os maiores cenários de buscas. Isso pode ser um dos fatores que fazem o *Quantized Ranking* ser mais comparativamente mais lento as demais técnicas, já em relação *Fixed Ranking* sendo o mais rápido entre as técnicas, pode ser explicado no fato de que ele quantifica como explicado na seção 4.3, a diferença entre os dois elementos da lista ordenada de *pivots*, enquanto o *Quantized Ranking*, *Query Quantized Ranking* e *Document Quantized Ranking* quantifica a posição de um *pivot* na lista ordenada.

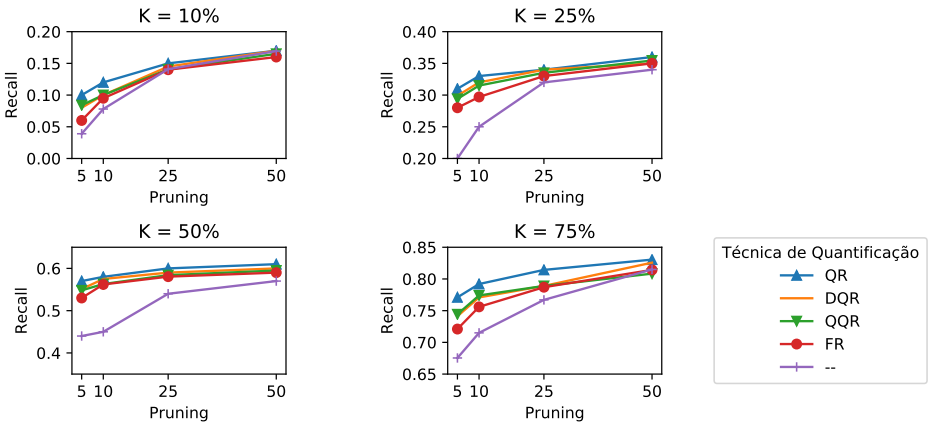

Figura 5.11 – Avaliação das técnicas de quantificação com a variação do *pruning* em relação ao *Recall*

Diferentemente da analise pelo tempo de consulta, quando analisamos pelo *recall*, vemos que a técnica de *Quantized Ranking*, representada pela sigla *QR* na figura 5.11, com uma degradação um pouco menor em comparação ao *Query Quantized Ranking* e *Document Quantized Ranking* quando reduzimos a quantidade de *pivots*. Isso ocorre em quando retornamos 10%, 25%, 50% e 75%, sendo que quando se retorna 10% e 25% quando o *pruning* é de 50 *pivots* o *Quantized Ranking*, *Query Quantized Ranking* e o *Document Quantized Ranking* estão sobrepostos, enquanto já com 50% e 75% do *dataset* retornados, já vemos uma pequena diferença entre eles, tendo o *Quantized Ranking* um pouco superior.

Quando analisamos o *Fixed Ranking* em relação as técnicas *Quantized Ranking*, *Query Quantized Ranking* e o *Document Quantized Ranking* ele possui uma leve degradação maior podendo ser visto em todas as variações de documentos retornados na consulta sendo que com 75% podemos verificar a maior diferença entre eles, mas ainda sim tendo um resultado superior de quando não utilizado nenhuma técnica de quantificação, que pelos experimentos se mostra com uma degradação bem alta comparativamente com as técnicas de quantificação.

Comparando as analises do tempo de consulta e do *recall* vistas nas figuras 5.10 e 5.11, podemos verificar que que os cenários vistos na seção 4.4 demonstram uma que maior degradação e possivelmente consultas mais rápidas por causa de buscas menores ocorreriam entre o *Query Quantized Ranking* e *Document Quantized Ranking* em comparação ao *Quantized Ranking*, algo que apareceu no experimentos, de forma um pouco mais acentuado no tempo de consulta do que em relação ao *recall*. Outra ponto interessante de ser avaliado é desempenho em relação ao tempo de consulta do *Fixed Ranking* em relação as demais técnicas e com um pequeno aumento na degradação comparando as outras técnicas, coisa que não ocorreu quando se utilizou o *Permutation Based Indexing* sem nenhuma técnica de quantificação.

5.4 Experimento 3

No experimento 3 sera avaliado o *Permutation Based Indexing*, com as técnica *Min-Max*, que pertence ao conjunto de abordagens do *Locality Sensitive Hashing* e o *BM-25*, as técnicas que são utilizadas na etapa de busca heurística da detecção de plagio. Essa avaliação se fará em relação ao *Recall*, o tempo de consulta de cada uma das técnicas e a vazão. Esse experimento tem por fim demonstrar a viabilidade de se utilizar o *Permutation Based Indexing* na etapa de detecção de plágio.

Para esse experimento fixa-se o pré-processamento com a representação não *booleana*, sem utilização n-grama, utilizando o *TF-IDF*, com a quantidade miníma 2 documentos de forma idêntica ao do experimento 1 e 2 para as técnicas do *Permutation Based Indexing* e a técnica da família do *Locality Sensitive Hashing*. No

Permutation Based Indexing será utilizado *Jaccard* para a comparação par-a-par na técnicas de seleção de *pivot Farthest-first traversal* com *threshold* $\theta = 1$ e com 75 *pivots*, e também será utilizado o *Jaccard* na comparação da distancia do documento d e consulta q para cada *pivot* p, será utilizado o *pruning* de 50 *pivots* e as técnicas de quantificação *Quantized Ranking*, *Query Quantized Ranking*, *Document Quantized Ranking* e *Fixed Quantized Ranking*. Para as técnicas da família do *Locality Sensitive Hashing* será utilizado o *Min-Max*, com 50 permutações de forma a seguir os experimentos praticados nos trabalhos DUARTE (2017) e VIEIRA (2016). Para o *BM-25* será utilizado os parâmetros $k_1 = 1.2$ e $b = 0.75$ de forma recomendada no trabalho ROBERTSON *et al.* (2000).

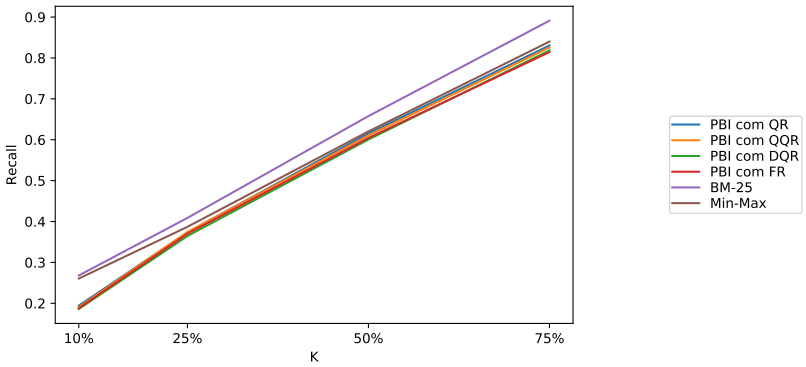

Figura 5.12 – Avaliação das técnicas em relação ao *Recall*

Analisando a figura 5.12, podemos verificar que as variações do *Permutation Based-Index* representadas pelas técnicas de quantificação *Quantized Ranking*, *Query Quantized Ranking*, *Document Quantized Ranking* e *Fixed Ranking* aparecem bem próximos, podendo ser consideradas sobrepostas, abaixo da técnica da família *Locality Sensitive Hashing*, o *Min-Max* quando retornamos 10% do *dataset* na consulta, a partir de 25% do *dataset* retornado na consulta, o *Min-Max* aparece sobreposto as técnicas do *Permutation Based Indexing* e a partir de 50% do *dataset* retornados na consulta, aparece abaixo do *BM-25*. A diferença entre as variações do *Permutation Based Indexing* e do *BM-25* se apresenta a maior diferença quando se retorna 75% do *dataset* na consulta, no qual a diferença chega 0.6 no *recall* e a menor se apresenta quando se retorna 25% aonde as curvas aparecem muito próximas uma da outra, podendo se considerar que estejam sobrepostas.

K	Minmax	BM25	QR	FR	RQR	LQR
1597	531.22	512.36	345.03	344.72	365.88	344.48
3991	526.39	494.99	356.22	350.90	352.22	362.34
7983	526.62	511.36	352.76	353.07	352.76	362.90
11975	529.11	508.30	352.92	351.98	364.29	356.78

Tabela 5.2 – Execução das diferentes técnicas em relação a vazão.

Quando analisamos o experimento 3 em relação ao tempo de consulta, é perceptível uma diferença significativa de todas as variações do *Permutation Based Indexing* em relação ao *BM-25* e ao *Min-Max* não importando quantos documento serão recuperados, na consulta, qual as variações do *Permutation Based Indexing* retornam a consulta em menor tempo.

As diferenças entre as técnicas do *Permutation Based Indexing* são significativas, no qual a variação do *Permutation Based Indexing* tem é a que tem menor desempenho entre as técnicas do *Permutation Based Indexing*, seguido Pelo *Query-Quantized Ranking* e *Document Quantized Ranking* sobrepostos e sendo $0,4$ segundos mais rápido por consulta. a A diferença para o *Fixed Ranking* sobe para $0,8$ segundos a diferença no tempo de consulta.

Já a diferença entre as variações da técnica *Permutation Based Indexing* e o *BM-25* é de $0,9$ segundos da variação da técnica *Permutation Based Indexing* utilizando o *Quantized Ranking* e de $1,7$ segundos, para a técnica *Permutation Based Indexing* utilizando o *Fixed Ranking*, a variação mais no tempo de consulta. Essa diferença representa que o *BM-25* é em torno de $2,5$ vezes mais lento que *Permutation Based Indexing* utilizando o *Fixed Ranking*. As diferença da variação *Permutation Based Indexing* utilizando o *Quantized Ranking* para o *Min-Max* é de aproximadamente $5,3$ segundos, isso representa que o *Min-Max* é 5.4 vezes mais lento que o *Permutation Based Indexing* utilizando o *Quantized Ranking*. Sendo que essa diferença aumenta para $5,7$ segundos se comparado com o *Permutation Based Indexing* utilizando o *Fixed Ranking*.

As variações da técnica do *Permutation Based Indexing* aparecem todas sobrepostas e bem abaixo na vazão em relação ao *BM-25* e o *Min-Max*, no qual o *Min-Max* aparece com uma leve vantagem sobre o *Min-Max*.

K	Minmax	BM25	QR	FR	LQR	RQR
1597	6.50	2.49	1.58	0.80	1.21	1.13
3991	6.52	2.48	1.56	0.81	1.20	1.18
7983	6.51	2.50	1.56	0.81	1.19	1.17
11975	6.48	2.50	1.59	0.80	1.18	1.18

Tabela 5.3 – Execução das diferentes técnicas em relação ao tempo de consulta.

Apesar de aparecer como uma grande desvantagem, era esperado que a taxa de documentos indexados por segundo para o *Permutation Based Indexing* fosse baixa, pois como visto no capitulo 3, a indexação tem se mais etapas mais custosas, pois nele deve-se selecionar documentos para representar o *dataset*, afim de reduzir o números de comparações na consulta, etapa na qual é muitas vezes executada.

De forma geral, no experimento 3 vemos que o *Permutation Based Indexing* teve um satisfatório desempenho quando avaliamos a qualidade do resultado gerado, avaliado pelo *recall*, muito bom desempenho em relação ao tempo de consulta, sendo bem mais eficiente que o *BM-25* e o *Min-Max* nesse quesito, e apesar e um desempenho mais fraco em termos da vazão comparados as outras técnicas, não é um problema que inviabiliza a utilização dela na busca Heurística, já que é uma etapa única sendo executada somente quando se inicia a utilização da mesma.

Capítulo 6

Conclusão

Neste trabalho, foi explorado o *Permutation Based Indexing* e suas variações aplicados na detecção de plágio extrínseco. Além disso foi apresentado as abordagens de quantificação *Document Quantized Ranking*, *Query Quantized Ranking* e *Fixed Quantized Ranking* com uma analise do comportamento de cada uma das abordagens em relação ao *pruning*. Após análise com cada uma das abordagens de quantificação, foi feito um estudo empírico com o experimento de Recuperação Heurística do Plágio extrínseco no qual separamos em 3 partes nomeadas de experimento 1, experimento 2 e experimento 3.

No experimento 1 vimos o comportamento do *Permutation Based Indexing* variando seus paremetros, como função de seleção de *pivot* e quantidade de *pivots*. No experimento 2 vimos de forma empírica as diferenças das abordagens *Quantized Ranking Document Quantized Ranking*, *Query Quantized Ranking*, *Fixed Quantized Ranking* e o *pruning* sem nenhuma técnica de quantificação. Sendo evidente a diferença de tempo aonde o *Quantized Ranking* demonstrou-se mais lento, algo esperado já que era ele o que possuía maior quantidade de informações para busca, e o mais rápido, o *Fixed Ranking* demonstrou tendo teoricamente uma grande quantidade de informação mais a forma na qual quantifica, faz com que seu desempenho em relação ao tempo seja melhor que os demais, diferentemente da relação ao *Recall* aonde sofre uma degradação maior que o *Document Quantized Ranking*, *Query Quantized Ranking* e o *Quantized Ranking*, sendo que o ultimo é o que possui melhor desempenho em relação ao *Recall*.

Já no experimento 3, comparamos o *Permutation Based Indexing* com as 4 diferentes abordagens de quantificação com as abordagens já utilizadas na detecção de plágio extrinseco, *BM-25* e o *Locality Sensitive Hashing* com a abordagem *Min-Max*. Em relação ao tempo de consulta ficou evidente o ganho do *Permutation Based Indexing* em relação ao *Min-Max* e o *BM-25*, já no tempo de indexação como já era esperado pela quantidade de pré-processamento o *Permutation Based Indexing* foi mais lento em relação aos demais, algo que não é tão importante visto que

a indexação ela feita uma única só vez. Por fim quando comparado em relação ao *Recall* vimos uma pequena perda em relação ao *BM-25* e boa parte do experimento o *Permutation Based Indexing* ficou sobreposto ao *Min-Max*.

Ao fim dos experimentos acreditamos que a utilização do *Permutation Based Indexing* é positiva na detecção de plágio extrinseco, assim como o uso das abordagens *Document Quantized Ranking*, *Query Quantized Ranking* e *Fixed Quantized Ranking* no *Permutation Based Indexing*.

6.1 Trabalhos Futuros

Como trabalhos futuros sugerimos a aplicação do *Permutation Based Indexing* em bases maiores e de forma paralelizada para a detecção de Plágio extrínseco, ainda na ideia de paralelização acreditamos ser um trabalho pertinente, a comparação neste cenário as abordagens *Quantized Ranking Document Quantized Ranking*, *Query Quantized Ranking* e *Fixed Quantized Ranking*, pois poderemos ter diferenças em relação ao tempo de consulta em implementações que utilizam distribuição em *GPU*.

Outro trabalho no qual apareceu como importante nesse contexto, é de utilização de outras formas de transformação para o espaço métrico, e a forma que isso altera na qualidade da consulta e na seleção de *pivots*. Isso é motivado pela importância dada na revisão literária numa boa escolha *pivot*, algo que não foi visto de forma evidente e que pode ser relacionada a forma que fizemos a transformação para o espaço métrico.

A partir das novas formas de transformação para o espaço métrico, utilizar técnicas que ao invés de somente selecionar um *pivot*, criar um *pivot*, de forma que seja possivel que ele seja um melhor representante do *dataset*, avaliando o impacto que isso gerará nos resultados com grandes *dataset*, assim como pequenos *dataset*.

E por último, avaliar o β nas diferentes técnicas de quantificação, e quais suas consequências para cada técnica, já que o valor indicado para quantificação influência de forma direta os resultados.

Referências Bibliográficas

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

AMATO, G., ESULI, A., FALCHI, F., 2015, "A comparison of pivot selection techniques for permutation-based indexing", *Information Systems*, v. 52, pp. 176–188.

APOSTOLICO, A., BAEZA-YATES, R., MELUCCI, M., 2006, "Advances in information retrieval: An introduction to the special issue", *Inf. Syst.*, v. 31 (11), pp. 569–572. doi: 10.1016/j.is.2005.11.005.

ARYA, S., MOUNT, D. M., NETANYAHU, N. S., et al., 1998, "An optimal algorithm for approximate nearest neighbor searching fixed dimensions", *Journal of the ACM (JACM)*, v. 45, n. 6, pp. 891–923.

BAEZA-YATES, R., RIBEIRO-NETO, B., OTHERS, 1999, *Modern information retrieval*, v. 463. ACM press New York.

BARRÓN-CEDEÑO, A., 2010, "On the mono-and cross-language detection of text reuse and plagiarism". In: *Proceedings of the 33rd international ACM SIGIR conference on Research and development in information retrieval*, pp. 914–914. ACM.

BELLMAN, R., 1961, "Curse of dimensionality", *Adaptive control processes: a guided tour. Princeton, NJ.*

BRIN, S., 1995, "Near neighbor search in large metric spaces", .

BRODER, A. Z., 1997, "On the resemblance and containment of documents". In: *Proceedings. Compression and Complexity of SEQUENCES 1997 (Cat. No. 97TB100171)*, pp. 21–29. IEEE.

BUSTOS, B., NAVARRO, G., CHÁVEZ, E., 2001, "Pivot selection techniques for proximity searching in metric spaces". In: *SCCC 2001. 21st International Conference of the Chilean Computer Science Society*, pp. 33–40. IEEE.

BUTTLER, D., 2004, *A short survey of document structure similarity algorithms*. Relatório técnico, Lawrence Livermore National Lab.(LLNL), Livermore, CA (United States).

CALDWELL, C., 2010, "A ten-step model for academic integrity: A positive approach for business schools", *Journal of Business Ethics*, v. 92, n. 1, pp. 1–13.

CESKA, Z., TOMAN, M., JEZEK, K., 2008, "Multilingual plagiarism detection". In: *International Conference on Artificial Intelligence: Methodology, Systems, and Applications*, pp. 83–92. Springer.

CHÁVEZ, E., NAVARRO, G., 2000, "Measuring the dimensionality of general metric spaces", *Department of Computer Science, University of Chile, Tech. Rep. TR/DCC-00-1*.

CHÁVEZ, E., NAVARRO, G., 2001, "A probabilistic spell for the curse of dimensionality". In: *Workshop on Algorithm Engineering and Experimentation*, pp. 147–160. Springer.

CHÁVEZ, E., NAVARRO, G., 2005, "A compact space decomposition for effective metric indexing", *Pattern Recognition Letters*, v. 26, n. 9, pp. 1363–1376.

CHÁVEZ, E., NAVARRO, G., BAEZA-YATES, R., et al., 2001, "Searching in metric spaces", *ACM computing surveys (CSUR)*, v. 33, n. 3, pp. 273–321.

CHÁVEZ, E., FIGUEROA, K., NAVARRO, G., 2005, "Proximity Searching in High Dimensional Spaces with a Proximity Preserving Order", *MICAI '05: Advances in Artificial Intelligence*, pp. 405–414. ISSN: 03029743. `http://www.springerlink.com/index/p837532471h11218.pdf` .

CHAVEZ GONZALEZ, E., FIGUEROA, K., NAVARRO, G., et al., 2008, "Effective proximity retrieval by ordering permutations", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, v. 30, n. 9, pp. 1647–1658. ISSN: 01628828. doi: 10.1109/TPAMI.2007.70815.

CHEN, L., GAO, Y., ZHENG, B., et al., 2017, "Pivot-based metric indexing", *Proceedings of the VLDB Endowment*, v. 10, n. 10, pp. 1058–1069.

CHUM, O., PHILBIN, J., ZISSERMAN, A., et al., 2008, "Near duplicate image detection: min-hash and tf-idf weighting." In: *Bmvc*, v. 810, pp. 812–815.

CHÁVEZ, E., NAVARRO, G., 2008, "Proceedings - First International Workshop on Similarity Search and Applications, SISAP 2008: Preface", (01), pp. viii. doi: 10.1109/SISAP.2008.24.

COMAS, R., SUREDA, J., 2008, "Academic cyberplagiarism: tracing the causes to reach solutions", *Digithum*, , n. 10.

COVER, T., 1968, "Estimation by the nearest neighbor rule", *IEEE Transactions on Information Theory*, v. 14, n. 1, pp. 50–55.

CRASWELL, N., ZARAGOZA, H., ROBERTSON, S., 2005, "Microsoft cambridge at trec-14: Enterprise track", .

DIACONIS, P., 1988, "Group representations in probability and statistics", *Lecture Notes-Monograph Series*, v. 11, pp. i–192.

DO CARMO, F. B., 2018, *CONSIDERANDO O RUÍDO NO APRENDIZADO DE MODELOS PREDITIVOS ROBUSTOS PARA A FILTRAGEM COLABORATIVA*. Ph.D. Thesis, Universidade Federal do Rio de Janeiro.

DUARTE, F. R., 2017, *IDENTIFICANDO PLAGIO EXTERNO COM LOCALITY-SENSITIVE HASHING*. Ph.D. Thesis, Universidade Federal do Rio de Janeiro.

EHSAN, N., SHAKERY, A., 2016, "Candidate document retrieval for cross-lingual plagiarism detection using two-level proximity information", *Information Processing and Management*, v. 52, n. 6, pp. 1004–1017. ISSN: 03064573. doi: 10.1016/j.ipm.2016.04.006. `http://dx.doi.org/10.1016/j.ipm.2016.04.006` .

FAGIN, R., KUMAR, R., SIVAKUMAR, D., 2003, "Comparing top k lists", *SIAM Journal on discrete mathematics*, v. 17, n. 1, pp. 134–160.

FARAGÓ, A., LINDER, T., LUGOSI, G., 1993, "Fast nearest-neighbor search in dissimilarity spaces", *IEEE Transactions on Pattern Analysis & Machine Intelligence*, , n. 9, pp. 957–962.

FIGUEROA, K., FREDIKSSON, K., 2009, "Speeding Up Permutation Based Indexing with Indexing". In: *2009 Second International Workshop on Similarity Search and Applications*, pp. 107–114. doi: 10.1109/SISAP.2009.12.

FIGUEROA, K., FREDIKSSON, K., VON LÜCKEN, C., et al., 2015, "Fast large-scale multimedia indexing and searching". In: *2015 Latin American Computing Conference (CLEI)*, pp. 1–10. ISBN: 9781467368704. doi: 10.1109/SISAP.2009.12.

FISCHETTI1Ú, M., FISCHETTI, M., MONACI, M., 2014, "Proximity search heuristics for Mixed Integer Programs", *Proceedings of the Twenty-Sixth RAMP Symposium*.

GODDARD, R., RUDZKI, R., 2005, "Using an Electronic Text-Matching Tool (Turnitin) to Detect Plagiarism in a New Zealand University." *Journal of University Teaching and Learning Practice*, v. 2, n. 3, pp. 7.

HART, P., 1968, "The condensed nearest neighbor rule (Corresp.)", *IEEE transactions on information theory*, v. 14, n. 3, pp. 515–516.

HAYES, N., INTRONA, L. D., 2005, "Cultural values, plagiarism, and fairness: When plagiarism gets in the way of learning", *Ethics & Behavior*, v. 15, n. 3, pp. 213–231.

HELLMAN, M. E., 1970, "The nearest neighbor classification rule with a reject option", *IEEE Transactions on Systems Science and Cybernetics*, v. 6, n. 3, pp. 179–185.

HINNEBURG, A., AGGARWAL, C. C., KEIM, D. A., 2000, "What is the nearest neighbor in high dimensional spaces?" In: *26th Internat. Conference on Very Large Databases*, pp. 506–515.

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*, pp. 604–613. ACM.

JACCARD, P., 1908, "Nouvelles recherches sur la distribution florale", *Bull. Soc. Vaud. Sci. Nat.*, v. 44, pp. 223–270.

JI, J., LI, J., YAN, S., et al., 2012, "Super-bit locality-sensitive hashing". In: *Advances in Neural Information Processing Systems*, pp. 108–116.

JI, J., LI, J., YAN, S., et al., 2013a, "Min-max hash for jaccard similarity". In: *2013 IEEE 13th International Conference on Data Mining*, pp. 301–309. IEEE, a.

JI, J., LI, J., YAN, S., et al., 2013b, "Min-Max Hash for Jaccard Similarity". pp. 301–309, 12b. ISBN: 978-0-7695-5108-1. doi: 10.1109/ICDM.2013.119.

KENDALL, M. G., 1938, "A NEW MEASURE OF RANK CORRELATION", *Biometrika*, v. 30, n. 1-2 (06), pp. 81–93. ISSN: 0006-3444. doi: 10.1093/biomet/30.1-2.81. https://doi.org/10.1093/biomet/30.1-2.81 .

KENDALL, M. G., 1955, "Rank correlation methods", .

KRULIŠ, M., OSIPYAN, H., MARCHAND-MAILLET, S., 2015a, "Permutation based indexing for high dimensional data on GPU architectures". In: *2015 13th International Workshop on Content-Based Multimedia Indexing (CBMI)*, pp. 1–6, a. doi: 10.1109/CBMI.2015.7153619.

KRULIŠ, M., OSIPYAN, H., MARCHAND-MAILLET, S., 2015b, "Permutation based indexing for high dimensional data on GPU architectures". In: *2015 13th International Workshop on Content-Based Multimedia Indexing (CBMI)*, pp. 1–6. IEEE, b.

LAWLER, E. L., LENSTRA, J. K., RINNOOY KAN, A. H., et al., 1985, "The traveling salesman problem; a guided tour of combinatorial optimization", .

LOSE, G., 2011. "Plagiarism". .

LUHN, H. P., 1957, "A statistical approach to mechanized encoding and searching of literary information", *IBM Journal of research and development*, v. 1, n. 4, pp. 309–317.

LUISA MICÓ, M., ONCINA, J., 1994, "A new version of the nearest neighbour approximating and eliminating search algorithm (AESA) with linear processing time and memory requirements", *Pattern Recognition Letters*, v. 15, n. January, pp. 9–17.

LV, Y., ZHAI, C., 2011, "When documents are very long, BM25 fails!" In: *Proceedings of the 34th international ACM SIGIR conference on Research and development in Information Retrieval*, pp. 1103–1104. ACM.

MARIMONT, R. B., SHAPIRO, M. B., 1979, "Nearest Neighbour Searches and the Curse of Dimensionality", *IMA Journal of Applied Mathematics*, v. 24, n. 1 (08), pp. 59–70. ISSN: 0272-4960. doi: 10.1093/imamat/24.1.59. https://doi.proxy.ufrj.br/10.1093/imamat/24.1.59 .

MARTIN, B., 1994, "Plagiarism: a misplaced emphasis", *Journal of Information Ethics*, v. 3, n. 2, pp. 36–47.

MAURER, H. A., KAPPE, F., ZAKA, B., 2006, "Plagiarism-a survey." *J. UCS*, v. 12, n. 8, pp. 1050–1084.

MCCABE, D., 2005. "Research report of the center for academic integrity". .

MCCABE, D. L., TREVIÑO, L. K., BUTTERFIELD, K. D., 2001, "Cheating in academic institutions: A decade of research", *Ethics &Behavior*, v. 11, n. 3, pp. 219–232.

MENDEL, M., NAOR, A., 2006, "Ramsey partitions and proximity data structures". In: *2006 47th Annual IEEE Symposium on Foundations of Computer Science (FOCS'06)*, pp. 109–118. IEEE.

MERRIAM-WEBSTER ONLINE, 2009. "Merriam-Webster Online Dictionary". `http://www.merriam-webster.com` .

MOHAMED, H., MARCHAND-MAILLET, S., 2015, "Quantized ranking for permutation-based indexing", *Information Systems*, v. 52, pp. 163–175. ISSN: 03064379. doi: 10.1016/j.is.2015.01.009. `http://dx.doi.org/10.1016/j.is.2015.01.009` .

MOHAMED, H., OSIPYAN, H., MARCHAND-MAILLET, S., 2014a, "Multi-core (CPU and GPU) for permutation-based indexing". In: *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, v. 8821, pp. 277–288, a. ISBN: 9783319119878. doi: 10.1007/978-3-319-11988-5_26.

MOHAMED, H., OSIPYAN, H., MARCHAND-MAILLET, S., 2014b, "Multi-core (CPU and GPU) for permutation-based indexing". In: *International Conference on Similarity Search and Applications*, pp. 277–288. Springer, b.

PARK, C., 2003, "In other (people's) words: Plagiarism by university students–literature and lessons", *Assessment & evaluation in higher education*, v. 28, n. 5, pp. 471–488.

PARKER, A., HAMBLEN, J. O., 1989, "Computer algorithms for plagiarism detection", *IEEE Transactions on Education*, v. 32, n. 2, pp. 94–99.

PATELLA, M., CIACCIA, P., 2009, "Approximate similarity search: A multi-faceted problem", *Journal of Discrete Algorithms*, v. 7, n. 1, pp. 36–48. ISSN: 15708667. doi: 10.1016/j.jda.2008.09.014. `http://dx.doi.org/10.1016/j.jda.2008.09.014` .

PENNYCOOK, A., 1996, "Borrowing others' words: Text, ownership, memory, and plagiarism", *TESOL quarterly*, v. 30, n. 2, pp. 201–230.

PERRY, L., 2001, "Why do students plagiarise". In: *Norfolk: Dominion University Factory Workshop*.

POTTHAST, M., STEIN, B., EISELT, A., et al., 2009, "P.: Overview of the 1st International Competition on Plagiarism Detection". In: *In: SEPLN 2009 Workshop on Uncovering Plagiarism, Authorship, and Social Software Misuse (PAN 09), CEUR-WS.org*, pp. 1–9.

POTTHAST, M., BARRÓN-CEDEÑO, A., STEIN, B., et al., 2011a, "Cross-language plagiarism detection", *Language Resources and Evaluation*, v. 45, n. 1, pp. 45–62.

POTTHAST, M., EISELT, A., BARRÓN-CEDEÑO, A., et al., 2011b, "Overview of the 3rd International Competition on Plagiarism Detection". In: Petras, V., Forner, P., Clough, P. (Eds.), *Working Notes Papers of the CLEF 2011 Evaluation Labs*, sep.b. ISBN: 978-88-904810-1-7. `http://www.clef-initiative.eu/publication/working-notes` .

POTTHAST, M., EISELT, A., BARRÓN CEDEÑO, L. A., et al., 2011c, "Overview of the 3rd international competition on plagiarism detection". In: *CEUR workshop proceedings*, v. 1177. CEUR Workshop Proceedings, c.

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.

ROBERTSON, S., WALKER, S., JONES, S., et al., 2000, "Okapi at trec-3", *Proceedings of the Text Retrieval Conference (TREC)*, (01).

ROBERTSON, S., ZARAGOZA, H., OTHERS, 2009, "The probabilistic relevance framework: BM25 and beyond", *Foundations and Trends® in Information Retrieval*, v. 3, n. 4, pp. 333–389.

ROBERTSON, S. E., 1997, "Overview of the okapi projects", *Journal of documentation*, v. 53, n. 1, pp. 3–7.

ROBERTSON, S. E., JONES, K. S., 1976, "Relevance weighting of search terms", *Journal of the American Society for Information science*, v. 27, n. 3, pp. 129–146.

ROBERTSON, S. E., WALKER, S., JONES, S., et al., 1993, "Okapi at TREC-2." In: *TREC*, pp. 21–34.

ROBERTSON, S. E., WALKER, S., JONES, S., et al., 1995, "Okapi at TREC-3", *Nist Special Publication Sp*, v. 109, pp. 109.

ROSENKRANTZ, D. J., STEARNS, R. E., LEWIS, II, P. M., 1977, "An analysis of several heuristics for the traveling salesman problem", *SIAM journal on computing*, v. 6, n. 3, pp. 563–581.

ROUSSEEUW, P. J., KAUFMAN, L., 1990, "Finding groups in data", *Hoboken: Wiley Online Library*.

SANDERSON, M., 2010, "Christopher D. Manning, Prabhakar Raghavan, Hinrich Schütze, Introduction to Information Retrieval, Cambridge University Press. 2008. ISBN-13 978-0-521-86571-5, xxi+ 482 pages." *Natural Language Engineering*, v. 16, n. 1, pp. 100–103.

SANTINI, S., JAIN, R., 1998, "Beyond query by example". In: *1998 IEEE Second Workshop on Multimedia Signal Processing (Cat. No. 98EX175)*, pp. 3–8. IEEE.

SPARCK JONES, K., 1972, "Some thesauric history". In: *Aslib Proceedings*, v. 24, pp. 400–411. MCB UP Ltd.

STAMATATOS, E., 2009, "Intrinsic plagiarism detection using character n-gram profiles", *threshold*, v. 2, n. 1,500.

STEVENSON, A., 2010, *Oxford Dictionary of English*. Oxford University Press. ISBN: 9780199571123. https://www.oxfordreference.com/view/10.1093/acref/9780199571123.001.0001/acref-9780199571123 .

STUDY.COM, 2017. "Merriam-Webster Online Dictionary". http://study.com/academy/lesson/what-is-paraphrasing-definition-examples-quiz.html .

TAYLOR, F. K., 1965, "Cryptomnesia and plagiarism", *The British Journal of Psychiatry*, v. 111, n. 480, pp. 1111–1118.

VANI, K., GUPTA, D., 2014, "Using K-means cluster based techniques in external plagiarism detection". In: *Contemporary computing and informatics (IC3I), 2014 international conference on*, pp. 1268–1273. IEEE.

VIEIRA, D. C., 2016, *Abordagens de Técnicas de LSH Aplicadas ao Problema de Similaridade de Documentos*. Ph.D. Thesis, Universidade Federal do Rio de Janeiro.

VON LUCKEN, C., JARMILA, L., BRITEZ, G., 2015, "Face recognition through a novel indexing method based on permutations", *2015 Latin American Computing Conference (CLEI)*, pp. 1–10. doi: 10.1109/CLEI.2015.7360043. http://ieeexplore.ieee.org/document/7360043/ .

VON LÜCKEN, C., JARMILA, L., BRÍTEZ, G., 2015, "Face recognition through a novel indexing method based on permutations". In: *2015 Latin American Computing Conference (CLEI)*, pp. 1–10. IEEE.

WAN, X., YANG, J., XIAO, J., 2008, "Towards a unified approach to document similarity search using manifold-ranking of blocks", *Information Processing & Management*, v. 44, n. 3, pp. 1032–1048.

WEBER, R., SCHEK, H.-J., BLOTT, S., 1998, "A quantitative analysis and performance study for similarity-search methods in high-dimensional spaces". In: *VLDB*, v. 98, pp. 194–205.

WEBER-WULFF, D., 2010, "Test cases for plagiarism detection software". In: *Proceedings of the 4th International Plagiarism Conference*.

WILBUR, W. J., SIROTKIN, K., 1992, "The automatic identification of stop words", *Journal of information science*, v. 18, n. 1, pp. 45–55.

YIANILOS, P. N., 1993, "Data structures and algorithms for nearest neighbor search in general metric spaces". In: *Soda*, v. 93, pp. 311–21.

ZEZULA, P., SAVINO, P., AMATO, G., et al., 1998, "Approximate similarity retrieval with M-trees", *The VLDB Journal*, v. 7, n. 4, pp. 275–293.

ZHANG, H., CHOW, T. W., 2011, "A coarse-to-fine framework to efficiently thwart plagiarism", *Pattern Recognition*, v. 44, n. 2, pp. 471–487.

ZU EISSEN, S. M., STEIN, B., 2006, "Intrinsic plagiarism detection". In: *European Conference on Information Retrieval*, pp. 565–569. Springer.

Apêndice A

Resultados Completos da Seção de Experimentos

Resultados dos experimentos apresentados no capítulo 5 das diversas Variações do *Permutation Based Index*. Aqui os resultados estão apresentados de forma estruturada em tabela, agrupados em relação a todos os experimentos. Assim como Apresentados no capítulo 5 a Indexação é mostrada pela quantidade de documentos indexados por segundo, a Consulta é medida por Segundo, K representa o tamanho do filtro do *Dataset*, a coluna *Pivots* representa a quantidade *Pivots* utilizados, a coluna *Pruning* representa a quantidade de *Pivots* utilizados após o *pruning* na coluna Quantização representa a técnica de quantização na qual S/P representa quando não existe a possibilidade de *pruning*, QR é igual a *Quantized Ranking*, LQR representa *Left Quantized Ranking*, RQR representa *Right Quantized Ranking*, FR representa *Fixed Quantized Ranking* e – representa quando não foi utilizada nenhuma forma de quantificação.

	K	Recall	Indexação	Consulta
BM-25	1597	0.26 +- 0.27	512.36	2.49
	3991	0.40 +- 0.31	494.99	2.48
	7983	0.66 +- 0.30	511.36	2.50
	11975	0.89 +- 0.17	508.30	2.50
Min-Max	1597	0.26 +- 0.35	531.22	6.50
	3991	0.39 +- 0.38	526.39	6.52
	7983	0.62 +- 0.35	526.62	6.51
	11975	0.84 +- 0.26	529.11	6.48

	Pivots	K	Recall	Indexação	Consulta	Pruning	Quantização	θ
Pivoted Space Incremental Selection	10	1597	0.15 +- 0.24	1.75	0.84	10	S/P	1
	25	1597	0.16 +- 0.26	1.74	1.14	25	S/P	1
	50	1597	0.16 +- 0.27	1.73	1.27	50	S/P	1
	75	1597	0.17 +- 0.27	1.74	1.47	75	S/P	1
	100	1597	0.18 +- 0.27	1.72	1.68	100	S/P	1
	100	1597	0.18 +- 0.27	1.76	1.59	100	S/P	10
	100	1597	0.18 +- 0.27	1.79	1.57	100	S/P	50
	100	1597	0.18 +- 0.27	2.23	1.58	100	S/P	100
	100	1597	0.19 +- 0.28	12.84	1.58	100	S/P	150
	100	1597	0.18 +- 0.27	79.48	1.08	100	S/P	160
	100	1597	0.08 +- 0.16	83.23	0.50	100	S/P	170
	10	3991	0.31 +- 0.32	1.75	0.88	10	S/P	1
	25	3991	0.32 +- 0.32	1.74	1.13	25	S/P	1
	50	3991	0.33 +- 0.33	1.74	1.27	50	S/P	1
	75	3991	0.33 +- 0.32	1.72	1.47	75	S/P	1
	100	3991	0.34 +- 0.32	1.72	1.67	100	S/P	1
	100	3991	0.34 +- 0.32	1.75	1.57	100	S/P	10
	100	3991	0.34 +- 0.32	1.76	1.57	100	S/P	50
	100	3991	0.35 +- 0.32	2.33	1.57	100	S/P	100
	100	3991	0.36 +- 0.33	13.72	1.56	100	S/P	150
	100	3991	0.33 +- 0.33	78.61	1.07	100	S/P	160
	100	3991	0.23 +- 0.27	82.57	0.50	100	S/P	170
	10	7983	0.57 +- 0.32	1.74	0.88	10	S/P	1
	10	7983	0.57 +- 0.33	11.97	0.82	100	S/P	150
	25	7983	0.57 +- 0.32	1.74	1.12	25	S/P	1
	25	7983	0.58 +- 0.33	15.18	1.08	100	S/P	150
	50	7983	0.58 +- 0.33	1.73	1.27	50	S/P	1
	50	7983	0.59 +- 0.32	14.72	1.21	100	S/P	150
	75	7983	0.58 +- 0.33	1.71	1.48	75	S/P	1
	75	7983	0.59 +- 0.32	14.17	1.42	100	S/P	150
	100	7983	0.59 +- 0.33	1.72	1.68	100	S/P	1
	100	7983	0.58 +- 0.33	1.77	1.60	100	S/P	10
	100	7983	0.58 +- 0.33	1.79	1.61	100	S/P	50
	100	7983	0.58 +- 0.33	2.24	1.61	100	S/P	100
	100	7983	0.60 +- 0.32	13.68	1.60	100	S/P	150
	100	7983	0.59 +- 0.31	78.97	1.08	100	S/P	160
	100	7983	0.43 +- 0.32	82.52	0.50	100	S/P	170

Continua na próxima página

Continua na Página Anterior

	Pivots	K	Recall	Indexação	Consulta	Pruning	Quantização	θ
Pivoted Space Incremental Selection	10	11975	0.79 +- 0.26	1.74	0.87	10	S/P	1
	10	11975	0.80 +- 0.25	15.57	0.82	100	S/P	150
	25	11975	0.80 +- 0.24	1.74	1.11	25	S/P	1
	25	11975	0.80 +- 0.25	15.31	1.10	100	S/P	150
	50	11975	0.80 +- 0.24	1.73	1.26	50	S/P	1
	50	11975	0.81 +- 0.25	14.66	1.22	100	S/P	150
	75	11975	0.81 +- 0.24	1.72	1.46	75	S/P	1
	75	11975	0.82 +- 0.24	14.25	1.42	100	S/P	150
	100	11975	0.82 +- 0.25	1.71	1.67	100	S/P	1
	100	11975	0.81 +- 0.25	1.76	1.61	100	S/P	10
	100	11975	0.81 +- 0.25	1.80	1.61	100	S/P	50
	100	11975	0.81 +- 0.25	2.22	1.61	100	S/P	100
	100	11975	0.82 +- 0.24	13.72	1.61	100	S/P	150
	100	11975	0.82 +- 0.24	80.32	1.10	100	S/P	160
	100	11975	0.74 +- 0.27	82.81	0.52	100	S/P	170
K-Medoids	10	1597	0.12 +- 0.21	5.89	0.77	10	S/P	–
	25	1597	0.14 +- 0.23	5.88	1.04	25	S/P	–
	50	1597	0.15 +- 0.24	5.84	1.17	50	S/P	–
	75	1597	0.17 +- 0.23	5.82	1.35	75	S/P	–
	100	1597	0.18 +- 0.26	5.80	1.57	100	S/P	–
	10	3991	0.29 +- 0.29	5.90	0.77	10	S/P	–
	25	3991	0.31 +- 0.31	5.89	1.03	25	S/P	–
	50	3991	0.33 +- 0.31	5.86	1.18	50	S/P	–
	75	3991	0.33 +- 0.31	5.84	1.36	75	S/P	–
	100	3991	0.34 +- 0.31	5.80	1.61	100	S/P	–
	10	7983	0.58 +- 0.32	5.90	0.78	100	S/P	–
	25	7983	0.59 +- 0.32	5.89	1.05	100	S/P	–
	50	7983	0.59 +- 0.31	5.87	1.19	100	S/P	–
	75	7983	0.58 +- 0.32	5.84	1.39	100	S/P	–
	100	7983	0.59 +- 0.32	5.82	1.59	100	S/P	–
	10	11975	0.83 +- 0.24	5.90	0.80	100	S/P	–
	25	11975	0.81 +- 0.24	5.88	1.08	100	S/P	–
	50	11975	0.82 +- 0.23	5.85	1.19	100	S/P	–
	75	11975	0.81 +- 0.24	5.82	1.38	100	S/P	–
	100	11975	0.82 +- 0.24	5.79	1.61	100	S/P	–

Continua na próxima página

Continua na Página Anterior

	Pivots	K	Recall	Indexação	Consulta	Pruning	Quantização	θ
Farthest-first traversal	10	1597	0.12 +- 0.21	1205.89	0.82	10	S/P	1
	25	1597	0.14 +- 0.25	888.48	1.08	25	S/P	1
	50	1597	0.15 +- 0.25	514.87	1.22	50	S/P	1
	75	1597	0.15 +- 0.25	356.07	1.39	75	S/P	1
	100	1597	0.17 +- 0.25	262.12	1.60	100	S/P	1
	100	1597	0.17 +- 0.25	264.08	1.61	100	S/P	10
	100	1597	0.17 +- 0.26	215.99	1.61	100	S/P	50
	100	1597	0.18 +- 0.26	150.65	1.61	100	S/P	100
	100	1597	0.19 +- 0.27	116.60	1.62	100	S/P	150
	10	3991	0.30 +- 0.30	1202.26	0.83	10	S/P	1
	25	3991	0.31 +- 0.31	881.61	1.07	25	S/P	1
	50	3991	0.32 +- 0.31	511.24	1.22	50	S/P	1
	75	3991	0.33 +- 0.31	355.27	1.39	75	S/P	1
	100	3991	0.34 +- 0.31	262.99	1.61	100	S/P	1
	100	3991	0.34 +- 0.31	262.77	1.61	100	S/P	10
	100	3991	0.34 +- 0.31	217.52	1.61	100	S/P	50
	100	3991	0.35 +- 0.32	148.29	1.61	100	S/P	100
	100	3991	0.36 +- 0.33	116.35	1.61	100	S/P	150
	100	7983	0.59 +- 0.32	260.46	1.57	100	S/P	1
	100	7983	0.59 +- 0.32	262.25	1.62	100	S/P	10
	100	7983	0.59 +- 0.32	218.06	1.61	100	S/P	50
	100	7983	0.60 +- 0.32	150.47	1.62	100	S/P	100
	100	7983	0.60 +- 0.32	116.71	1.62	100	S/P	150
	100	11975	0.81 +- 0.23	260.20	1.58	100	S/P	1
	100	11975	0.82 +- 0.23	263.60	1.62	100	S/P	10
	100	11975	0.82 +- 0.23	219.68	1.61	100	S/P	50
	100	11975	0.82 +- 0.24	150.86	1.62	100	S/P	100
	100	11975	0.83 +- 0.24	116.28	1.63	100	S/P	150
	75	1597	0.04 +- 0.12	259.87	0.32	5	–	1
	75	1597	0.08 +- 0.16	259.87	0.32	10	–	1
	75	1597	0.14 +- 0.23	260.86	0.47	25	–	1
	75	1597	0.17 +- 0.26	260.51	0.78	50	–	1
	75	1597	0.08 +- 0.15	263.94	0.57	5	RQR	1
	75	1597	0.10 +- 0.15	263.94	0.71	10	RQR	1
	75	1597	0.14 +- 0.15	264.78	0.96	25	RQR	1
	75	1597	0.17 +- 0.17	265.88	1.13	50	RQR	1

Continua na próxima página

Continua na Página Anterior

	Pivots	K	Recall	Indexação	Consulta	Pruning	Quantização	θ
Farthest-first traversal	75	1597	0.06 +- 0.24	345.05	0.29	5	FR	1
	75	1597	0.10 +- 0.25	345.12	0.33	10	FR	1
	75	1597	0.14 +- 0.26	344.84	0.48	25	FR	1
	75	1597	0.16 +- 0.25	344.72	0.80	50	FR	1
	75	1597	0.10 +- 0.16	345.12	0.70	5	QR	1
	75	1597	0.12 +- 0.15	345.13	0.97	10	QR	1
	75	1597	0.15 +- 0.18	345.16	1.30	25	QR	1
	75	1597	0.17 +- 0.25	345.03	1.58	50	QR	1
	75	1597	0.08 +- 0.16	344.69	0.57	5	LQR	1
	75	1597	0.10 +- 0.16	345.14	0.77	10	LQR	1
	75	1597	0.14 +- 0.20	344.78	1.04	25	LQR	1
	75	1597	0.17 +- 0.26	344.48	1.21	50	LQR	1
	75	3991	0.29 +- 0.24	260.33	0.58	5	RQR	1
	75	3991	0.32 +- 0.24	260.33	0.73	10	RQR	1
	75	3991	0.34 +- 0.24	261.65	1.00	25	RQR	1
	75	3991	0.35 +- 0.27	262.34	1.18	50	RQR	1
	75	3991	0.34 +- 0.31	356.15	0.78	50	–	1
	75	3991	0.32 +- 0.31	353.15	0.47	25	–	1
	75	3991	0.25 +- 0.26	356.38	0.33	10	–	1
	75	3991	0.20 +- 0.21	351.75	0.30	5	–	1
	75	3991	0.35 +- 0.31	350.90	0.81	50	FR	1
	75	3991	0.33 +- 0.31	354.48	0.49	25	FR	1
	75	3991	0.30 +- 0.31	354.33	0.35	10	FR	1
	75	3991	0.28 +- 0.31	351.29	0.31	5	FR	1
	75	3991	0.35 +- 0.31	352.22	1.20	50	LQR	1
	75	3991	0.34 +- 0.32	355.04	1.02	25	LQR	1
	75	3991	0.32 +- 0.31	355.51	0.76	10	LQR	1
	75	3991	0.30 +- 0.31	351.44	0.57	5	LQR	1
	75	3991	0.36 +- 0.31	356.22	1.56	50	QR	1
	75	3991	0.34 +- 0.31	355.51	1.29	25	QR	1
	75	3991	0.33 +- 0.32	355.59	1.01	10	QR	1
	75	3991	0.31 +- 0.31	352.06	0.72	5	QR	1
	75	7983	0.57 +- 0.31	358.06	0.79	50	–	1
	75	7983	0.54 +- 0.32	352.45	0.47	25	–	1
	75	7983	0.45 +- 0.32	355.99	0.33	10	–	1
	75	7983	0.44 +- 0.32	352.92	0.29	5	–	1

Continua na próxima página

Continua na Página Anterior

	Pivots	K	Recall	Indexação	Consulta	Pruning	Quantização	θ
Farthest-first traversal	75	7983	0.59 +- 0.31	353.07	0.81	50	FR	1
	75	7983	0.58 +- 0.32	353.15	0.49	25	FR	1
	75	7983	0.56 +- 0.31	357.74	0.34	10	FR	1
	75	7983	0.53 +- 0.32	353.15	0.31	5	FR	1
	75	7983	0.60 +- 0.31	352.76	1.19	50	LQR	1
	75	7983	0.59 +- 0.31	353.23	1.02	25	LQR	1
	75	7983	0.57 +- 0.31	353.39	0.76	10	LQR	1
	75	7983	0.55 +- 0.31	352.37	0.57	5	LQR	1
	75	7983	0.61 +- 0.31	352.76	1.56	50	QR	1
	75	7983	0.60 +- 0.31	353.31	1.30	25	QR	1
	75	7983	0.58 +- 0.31	356.78	1.00	10	QR	1
	75	7983	0.57 +- 0.32	356.30	0.72	5	QR	1
	75	7983	0.56 +- 0.32	265.00	0.74	10	RQR	1
	75	7983	0.58 +- 0.32	263.86	1.00	25	RQR	1
	75	7983	0.59 +- 0.32	262.90	1.17	50	RQR	1
	75	7983	0.55 +- 0.32	344.63	0.52	5	RQR	1
	75	11975	0.74 +- 0.31	344.69	0.53	5	RQR	1
	75	11975	0.77 +- 0.30	261.69	0.73	10	RQR	1
	75	11975	0.79 +- 0.31	258.43	1.01	25	RQR	1
	75	11975	0.81 +- 0.27	264.29	1.18	50	RQR	1
	75	11975	0.68 +- 0.30	344.63	0.29	5	–	1
	75	11975	0.71 +- 0.29	344.60	0.32	10	–	1
	75	11975	0.77 +- 0.26	351.75	0.47	25	–	1
	75	11975	0.81 +- 0.23	351.98	0.79	50	–	1
	75	11975	0.72 +- 0.24	345.26	0.30	5	FR	1
	75	11975	0.76 +- 0.23	344.70	0.33	10	FR	1
	75	11975	0.79 +- 0.23	353.00	0.49	25	FR	1
	75	11975	0.81 +- 0.23	351.98	0.80	50	FR	1
	75	11975	0.79 +- 0.23	351.91	1.01	25	LQR	1
	75	11975	0.83 +- 0.23	356.78	1.18	50	LQR	1
	75	11975	0.77 +- 0.31	344.93	0.70	5	QR	1
	75	11975	0.79 +- 0.31	344.56	0.99	10	QR	1
	75	11975	0.81 +- 0.23	351.60	1.29	25	QR	1
	75	11975	0.83 +- 0.23	352.92	1.59	50	QR	1

Continua na próxima página

Continua na Página Anterior

	Pivots	K	Recall	Indexação	Consulta	Pruning	Quantização	θ
Randomized	10	1597	0.12 +- 0.23	1186.18	1.03	10	S/P	–
	25	1597	0.15 +- 0.25	779.21	1.36	25	S/P	–
	50	1597	0.15 +- 0.25	519.73	1.53	50	S/P	–
	75	1597	0.16 +- 0.26	339.05	1.77	75	S/P	–
	100	1597	0.17 +- 0.25	327.31	2.04	100	S/P	–
	10	3991	0.31 +- 0.31	1118.85	1.03	10	S/P	–
	25	3991	0.33 +- 0.31	850.61	1.36	25	S/P	–
	50	3991	0.34 +- 0.32	478.17	1.54	50	S/P	–
	75	3991	0.35 +- 0.32	387.81	1.76	75	S/P	–
	100	3991	0.35 +- 0.32	274.52	2.02	100	S/P	–
	10	7983	0.56 +- 0.32	1144.52	1.03	10	S/P	–
	25	7983	0.58 +- 0.31	775.80	1.38	25	S/P	–
	50	7983	0.58 +- 0.31	595.52	1.53	50	S/P	–
	75	7983	0.60 +- 0.31	221.11	2.03	75	S/P	–
	100	7983	0.60 +- 0.31	221.11	2.03	100	S/P	–
	10	11975	0.78 +- 0.25	1261.14	1.03	10	S/P	–
	25	11975	0.80 +- 0.25	894.45	1.37	25	S/P	–
	50	11975	0.81 +- 0.23	425.31	1.55	50	S/P	–
	75	11975	0.82 +- 0.23	342.40	1.75	75	S/P	–
	100	11975	0.82 +- 0.24	256.19	2.03	100	S/P	–