Mario Caires Pereira

Estudo de Técnicas de Mineração de Texto Aplicadas na Classificação de Artigos Científicos

Projeto de Graduação em Computação submetido à Universidade Federal do ABC para a obtenção dos créditos na disciplina Projeto de Graduação em Computação III do curso de Ciência da Computação

Orientador: Prof Dr Thiago Ferreira Covões

Universidade Federal do ABC

28 de Abril de 2020

Resumo

Milhares de trabalhos científicos são desenvolvidos todos os anos em universidades públicas e privadas, entre eles: iniciações cientificas, trabalhos de conclusão de curso e teses de mestrado ou doutorado. O atual método para analisar tais trabalhos, é por meio de bancas avaliadoras, que na maioria das vezes é formada pelo orientador e dois (ou mais) docentes convidados pelo mesmo. Este projeto visa utilizar técnicas de mineração de textos aplicadas na classificação de trabalhos científicos, por meio da leituras de dados não estruturados presentes na Web, a partir dessa classificação, pretende-se identificar as palavras mais relevantes do trabalho. Com o intuito de, gerar uma banca coerente ao assunto abordado no artigo, através da indicação de docentes que atuam na área.

SUMÁRIO

CAPÍTULO 1

INTRODUÇÃO

Desde o surgimento dos sistemas computacionais, um dos principais objetivos de empresas e organizações têm sido o armazenamento de dados. Nas ultimas décadas, em virtude dos avanços tecnológicos, a capacidade desse armazenamento é cada vez maior. Devido a isto, no final da década de 80 surge o termo Mineração de Dados (MD), também conhecido como Descoberta do Conhecimento em Banco de Dados.

Mineração de Dados possui o objetivo de extrair informações e padrões por meio da análise de grandes quantidades de dados, que previamente eram incompreensíveis ou desconhecidos. É uma área interdisciplinar, que envolve banco de dados, inteligência artificial, aprendizado de máquinas, estatística, entre outras. Dentre as diferentes possibilidades de aplicações de MD, pode-se destacar: mercado financeiro, identificando segmentos de mercado; tomadas de decisão, filtrando informações relevantes; marketing, direcionando mensagens promocionais para um determinado público alvo [Camilo e Silva 2009].

A MD realiza o estudo de dados estruturados, ou seja, eles são organizados conforme a definição de uma rígida estrutura. Essa disposição geralmente é realizada via linhas e colunas, permitindo "etiquetar"os dados, como por exemplo: banco de dados, planilha eletrônicas, arquivo CSV. Porém em razão dos grandes avanços tecnológicos de hardware e software voltados para a Web, em especial as redes sociais, a criação de conteúdos de textos, áudio e

imagem aumentou significantemente. Esses conteúdos reapresentam dados não estruturados, isso porque não hà necessidade se preocupar com campos pré definidos, restrições e limites. O usuário pode mesclar tipo de dados, como texto e imagem, vídeo e áudio, ou seja, o oposto do que seria uma estrutura rígida de um dado estruturado.

Atualmente, mais de 80% do conteúdo digital gerado no mundo é do tipo não estruturado, gerando a necessidade do desenvolvimento técnicas capazes de transformar estes dados em dados estruturados. Por exemplo, no caso de textos este processo é conhecido como Mineração de Textos (MT) [Feldman e Sanger 2006].

Mineração de Textos, também conhecido como Descoberta de Conhecimento em Textos, utiliza técnicas de análise e extração de dados a partir de textos, frases ou apenas palavras. Envolve a aplicação de algoritmos computacionais que processam textos e identificam informações úteis e implícitas, que normalmente não poderiam ser recuperadas utilizando métodos tradicionais de consulta, pois a informação contida nestes textos não podem ser obtida de forma direta, uma vez que, em geral, estão armazenadas em formato não estruturado [Morais e Ambrósio 2007].

Há duas abordagens que podem ser utilizadas na mineração de textos, são elas a Análise Semântica e Análise Estatística. A primeira realiza a interpretação das palavras conforme um ser humano faria, ou seja, por meio do significado da palavra, o contexto na qual ela está inserida bem como conhecimentos morfológicos, sintáticos e semânticos. Na Análise Estatística, as palavras são valoradas mediante a frequência de sua aparição nos dados, não importando a sua contextualização. As abordagens podem ser aplicadas separadamente ou em conjunto, de acordo com a necessidade do problema [Junior 2007].

Entre as principais técnicas de Mineração de Textos, as tarefas de Agrupamento e Classificação recebem especial atenção na literatura [Aggarwal e Zhai 2012]. Agrupamento consiste em agrupar automaticamente os documentos em grupos de acordo com a sua similaridade. A Classificação, também chamada Categorização de Textos, é utilizada para classificar um conjunto de documentos em uma ou mais categorias (classes) pré-definidas [Passini 2012].

As aplicações da mineração de textos são variadas tanto nas áreas científicas quanto comerciais. Um grande exemplo é a utilização desta técnica na medicina, onde diariamente milhares de informações de textos são geradas (prontuários, registros hospitalares, receitas, fichas de pacientes), auxiliando médicos com diagnóstico de doenças e recomendação de tratamentos. Há um software chamado Medline, que utiliza dos conceitos apresentados, trabalhando como base de dados bibliográficos da Biblioteca Nacional de Medicina dos Estados Unidos [Pezzini 2017].

No meio de diversas aplicações, uma nova área de pesquisa está emergindo, a "Mineração de Dados Educacionais"("*Educational Data Mining*", ou EDM) que possui o objetivo de estudar dados coletados no âmbito educacional e responder questões como: métodos para melhorar a aprendizagem do estudante, desenvolver ambientes educacionais mais eficazes, identificar se um aluno está confuso ou desmotivado com o método de aprendizagem e assim realizar adequações para sanar essa deficiência [Baker, Isotani e Carvalho 2011].

Neste projeto a mineração de textos será aplicada no ambiente acadêmico, com a finalidade de melhorar o método de como bancas avaliadoras são formadas atualmente. Possibilitando uma banca coesa com o tema apresentado no trabalho e consequentemente uma avaliação mais enriquecedora.

CAPÍTULO 2

JUSTIFICATIVA

A ideia do projeto surgiu após a indagação de como seria possível aprimorar o método como bancas avaliadoras são formadas nas universidades, com o intuito de automatizar o processo e ao mesmo tempo, indicar docentes que de fato trabalhem na área abordada pelo trabalho. Isso porque, a função da banca não é apenas a de atribuir uma nota, mas também de propor melhorias para o desenvolvimento do trabalho. Essa contribuição é efetiva quando a mesa é composta por profissionais que dominem o tema do projeto.

Em regra o procedimento de composição de uma banca avaliadora é invariavelmente o mesmo: o orientador do trabalho é responsável por convidar docentes a sua escolha para compor a mesa avaliadora. Porém não é necessário que estes docentes possuam expertise na área de pesquisa apresentada no trabalho, nestes casos, a banca não possuirá uma colaboração positiva para o projeto, já que a análise será superficial, devido a falta de familiaridade com o assunto. Além disso, o processo é burocrático e o orientador por vezes não conhece as áreas de pesquisas em que outros pesquisadores da universidade estão trabalhando atualmente. Por tal razão, é possível que a escolha de membros da banca se torne por afinidade do que por experiência no assunto.

Com a aplicação de técnicas de mineração de textos, é possível identificar os docentes mais apropriados para qualificar o trabalho em questão de baseada em dados, evitando as-

sim, a seleção daqueles que não possuem proximidade ao tema abordado. Somado a isso, o processo seria automatizado, retirando a função do orientador de selecionar os docentes da mesa, impossibilitando que a decisão seja influenciada por critérios não técnicos.

CAPÍTULO 3

OBJETIVOS

Será desenvolvido um processo no qual, técnicas de mineração de textos serão aplicadas, afim de identificar os docentes que possuem maior afinidade com a área de pesquisa do trabalho sujeito a avaliação. Para tanto, serão utilizados o Currículo *Lattes*[1] dos docentes, com o intuito de identificar suas principais áreas de pesquisas.

Como objetivo secundário, visa-se realizar um agrupamento de docentes baseado em suas áreas de pesquisas. A ideia é semelhante ao objetivo principal, porém ao invés da leitura de um trabalho, serão analisados os currículos dos docentes e assim associar aqueles possuem maiores semelhanças. Os grupos serão de extrema utilidade, pois atualmente nos sites da UFABC as informações sobre os docentes e suas respectivas áreas de atuações, em sua maioria estão incorretas e/ou desatualizadas. Devido a isso, os discentes acabam enfrentando diversas dificuldades entre elas, a escolha de um professor orientador que atue na área de pesquisa do projeto ou identificar um docente apto a sanar dúvidas referentes a uma determinada área.

[1] A Plataforma Lattes é uma plataforma virtual criada e mantida pelo Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq), pela qual integra as bases de dados de currículos, grupos de pesquisa e instituições, em um único sistema de informações, das áreas de Ciência e Tecnologia, atuando no Brasil.

CAPÍTULO 4

METODOLOGIA

Dado um trabalho científico como parâmetro de entrada, o processo realizará a leitura de todas as palavras presentes no texto, armazenando-as para posterior consulta. Feito isso, serão acessados os currículos *Lattes* dos docentes, identificando as palavras presentes em sua formação acadêmica, área de atuação, projetos de pesquisa, prêmios recebidos, produções realizadas, bancas e orientações. As palavras encontradas na *Plataforma Lattes* serão comparadas com as do trabalho, identificando aquelas que são semelhantes. Docentes que obtiverem maior correspondência de palavras, consequentemente são aqueles que atuam em áreas relacionadas à apresentada no trabalho.

Sabendo que o processo de Mineração de Textos é representado por cinco grandes etapas: identificação do problema, pré-processamento, extração de padrões, pós processamento e utilização do conhecimento, as próximas secções possuem a finalidade de identificar essas etapas no problema que será estudado [Rezende, Marcacini e Moura 2011]

4.1 Identificação do Problema

Utilizar técnicas de mineração de textos na recomendação dos docentes mais apropriados para comporem a banca avaliadora de um determinado trabalho. Para isso, serão utilizado

as informações presentes na *Plataforma Lattes* dos docentes, tais como seus projetos de pesquisas, orientações, trabalhos realizados, entre outras.

4.2 Pré-Processamento

O desafio inicial é verificar que parte da *Plataforma Lattes* contém os dados que deverão ser analisadas pelo processo. Posteriormente, identificar um método ou ferramenta que realize a leitura desse conteúdo e disponibilize para uso.

Para que o processo realizar as comparações entre as palavras, é de extrema importância que o texto armazenado esteja normalizado. Assim, é preciso aplicar algumas técnicas para "limpar"o texto, tais como *stopwords* e *stemming*, eliminando redundâncias e/ou variações morfológicas. Após a normalização do texto, preposições, plurais, letras maiúsculas e acentuações não irão interferir na análise do processo.

4.3 Extração de Padrões

Aqui deverá ser aplicado uma técnica ou modelo matemático para identificar entre as palavras armazenadas, aquelas que serão *key words* (palavras chave) e assim, serem utilizadas nesta etapa. Para cada palavra chave do texto de entrada, será realizada uma comparação entre as palavras chave retiradas da *Plataforma Lattes* dos docentes. Isso posto, os docentes que obtiverem um maior número de correlação, são aqueles que possuem maior familiaridade com o assunto abordado no trabalho e portanto os mais indicados para preencherem a banca. Visando o objetivo secundário, o processo será similar, porém as palavras chave comparadas serão somente aquelas identificadas na *Plataforma Lattes*, e o objetivo da correlação será o agrupamento daqueles que atuam na mesma área.

4.4 Pós Processamento

Tendo em vista a informação extraída, a formação de bancas coerentes com o tema abordado será realizada de maneira simples, pois bastará selecionar os docentes que foram identificados com maior familiaridade ao assunto. Além disso, as informações dos docentes são coletadas de uma página *web*, elas estarão atualizadas de acordo com as publicações e participações em pesquisas de cada um.

CAPÍTULO 5

FUNDAMENTAÇÃO TEÓRICA

Neste Capítulo serão discutidos conceitos e técnicas referentes a Mineração de Textos (MT). As etapas da MT apresentadas no Capítulo 4 serão utilizadas para identificar em qual momento do processo cada técnica é aplicada.

5.1 Pré-Processamento

O objetivo dessa etapa consiste em transformar o conjunto de documentos em uma base mais limpa, na qual o trabalho de representação, processamento dos dados e a consequente interpretação destes, possam ser realizadas de maneira mais rápida e eficiente [Passini 2012]. Segue a descrição de algumas das técnicas usualmente utilizadas na preparação dos dados.

5.1.1 Identificação de Termos

5.1.1.1 Identificação de Termos Simples

É aplicado um analisador léxico para identificar as palavras presentes no documento, eliminando símbolos e caracteres indesejados, tais como hífen e vírgula. Nesta etapa os termos podem ser convertidos para letras minúsculas ou maiúsculas e tabulações convertidas a es-

paços simples, adequando os temos de acordo com a objetivo da análise [Morais e Ambrósio 2007].

5.1.1.2 Identificação de Termos Compostos

Há diversos termos que possuem diferentes significados quando descritos por meio da utilização de duas ou mais palavras adjacentes, são conhecidos como *Word-phrase formation*, podemos citar como exemplo o termo "Inteligência Artificial". Uma maneira de reconhecer estas palavras, consiste em identificar os termos que co-ocorrem com a maior frequência no documento, e posteriormente validar ou não as expressões [Morais e Ambrósio 2007].

5.1.2 *Stopwords*

No processo de análise de textos, é necessário identificar palavras que não demonstram relevância, possibilitando assim a sua remoção. Pode-se citar como exemplo os artigos, preposições, pronomes, advérbios e outras classes de palavras auxiliares. Estes termos formam a maior parte dos textos da língua portuguesa, não agregando valor ao entendimento do texto analisado [Passini 2012].

As *stopwords* formam um "dicionário negativo", também conhecido como *stoplist*. Assim, ao realizar a análise de um texto, as palavras encontradas no dicionário são identificadas como *stopwords*, resultando na remoção dos termos [Morais e Ambrósio 2007].

5.1.3 *Stemming*

Stemming é uma técnica de redução de termos a um radical comum, a partir da análise das características gramaticais dos elementos, como grau, número, gênero e desinência. Tem o objetivo de retirar os sufixos e prefixos das palavras, e encontrar a sua forma primitiva. Assim, as palavras no plural ou derivadas são reduzidas a um radical único, simplificando a representação dos termos envolvidos no documento [Passini 2012].

Dois erros típicos que costumam ocorrer durante o processo de *stemming* são *overstemming* e *understemming*. *Overstemming* ocorre quando não só o sufixo, mas também parte

do radical é retirado da palavra. Já *understemming* ocorre quando o sufixo não é removido, ou é apenas removido parcialmente [Uber 2004].

Há diversos algoritmos de *Stemming* desenvolvidos, porém eles devem ser projetados para o processamento de um idioma em específico. Como neste projeto serão analisados textos na língua portuguesa, destacam-se três algoritmos: a versão para português do algoritmo de PORTER [Portuguese stemming algorithm], o Removedor de Sufixo da língua Portuguesa (RSLP), proposto por Orengo e Huyck [Orengo e Huyck 2001] e o algoritmo STEMBR, proposto por Alvares [Alvares, Garcia e Ferraz 2005] [Passini 2012].

Foram realizados estudos para identificar a técnica mais eficiente, comparando o desempenho dos três algoritmos citados anteriormente. O RSLP foi considerado o mais eficiente devido a menor taxa de erros de *overstemming* e *understemming* [Orengo e Huyck 2001].

5.2 Extração de Padrões

Essa é a principal etapa do processo de Mineração de Textos, nela ocorre a busca efetiva por conhecimentos inovadores e úteis a partir dos dados textuais. A aplicação dos algoritmos, fundamentados em técnicas que procuram, segundo determinados paradigmas, visa explorar os dados de forma a produzir modelos de conhecimento.

5.2.1 Keywords

As palavras mais frequentes em um texto (com exceção das *stopwords*) geralmente possuem um maior significado para o entendimento do assunto abordado no documento. Há duas maneiras de calcular a relevância destas palavras, a primeira delas é por meio da frequência que ela aparece no texto ou por meio da sua posição sintática.

Neste projeto será utilizado a análise baseada na frequência, mediante a atribuição de um *peso* para cada palavra. Há diversas maneiras de realizar o cálculo deste peso, a seguir serão descritos três métodos: frequência absoluta, frequência relativa e frequência inversa de documentos [Morais e Ambrósio 2007].

5.2.1.1 Frequência Absoluta

É a técnica mais simples de se calcular o peso de uma palavra. Basta contabilizar a quantidade de vezes que o termo aparece no documento. Porém, por não levar em conta o tamanho do documento, palavras pouco frequentes em um texto pequeno, podem ter o mesmo peso que palavras muito frequentes em grandes documentos [Morais e Ambrósio 2007].

5.2.1.2 Frequência Relativa - TF

Segundo Santos [Wives 1999] esta é a técnica mais comum para a identificação do quanto uma determinada palavra é importante para um documento, de acordo com o número de ocorrências desta palavra no mesmo. Segue a fórmula da frequência relativa:

$$TF = \frac{F_{abs_X}}{N}, \tag{5.1}$$

F_{abs_x} é o número de vezes que palavra x aparece no documento e N o número total de palavras no documento.

5.2.1.3 Frequência Inversa de Documentos - IDF

Essa técnica leva em conta a quantidade de documentos nos quais um termo aparece, somado a frequência absoluta dos termos. Assim, as palavras que aparecem em poucos documentos têm sua importância aumentada, pois geralmente são as mais discriminantes [Morais e Ambrósio 2007]. A fórmula para o cálculo da frequência inversa (*inverse document frequency* - IDF) é:

$$IDF_{td} = log\left(\frac{Freq_{td}}{DocFreq_{td}}\right), \tag{5.2}$$

o grau de relação entre o termo t e o documento d é IDF_{td}, e a quantidade de vezes que o termo t aparece no documento d é representado por $Freq_{td}$. $DocFreq_{td}$ calcula o número de documento nos quais o termo t aparece na coleção.

5.2.1.4 Frequência de Termo Inverso da Frequência nos Documentos - TFIDF

Possui o objetivo de determinar o peso que um termo têm para descrever um documento específico dentre uma coleção, por meio da multiplicação da frequência relativa *TF* e a frequência inversa de documentos *IDF*, conforme apresentado na equação abaixo:

$$TFIDF = TF \cdot IDF, \tag{5.3}$$

TFIDF é o algoritmo de extração de palavras-chave mais utilizado quando há uma coleção de documentos disponível.

5.3 Considerações Finais

Há diversas maneiras de realizar a extração de padrões de um texto, porém a maioria delas são derivações do TFIDF, método que possui o melhor desempenho quando comparado à outros algoritmos. Com o objetivo de explorar aqueles que não utilizam o TFIDF como base de cálculo, foi selecionado neste trabalho o KeyGraph [Ohsawa, Benson e Yachida 1998], metodologia que considera os *clusters* de um grafo as principais ideias abordadas em um texto, e as palavras responsáveis por conectá-las como as *keywords*, conforme detalhado no Capítulo 6.

CAPÍTULO 6

KEYGRAPH

O método KeyGraph [Ohsawa, Benson e Yachida 1998] se baseia na ideia de que um documento é construído para conter pontos focais, e a correlação de palavras neste documento são responsáveis por expressá-los. Assim, o algoritmo constrói um grafo, no qual as palavras mais frequentes no texto tornam-se os vértices, e as arestas serão construídas caso a relação entre estes sejam forte. Os *clusters* formados no grafo são os principais conceitos abordados no documento [Ohsawa, Benson e Yachida 1998].

O algoritmo utilizará importantes conceitos, sendo eles:

- *Foundations*: são os *clusters* do grafo que representam os conceitos básicos do textos, construídos por meio da co-ocorrência dos termos no documento;
- *Columns*: é a relação entre os termos do documento com os *clusters*;
- *Roof*: Vértices do grafo que apresentam altos valores de *Columns*;
- Componente conexo: subconjunto de vértices de um grafo que pode-se chegar em qualquer um dos vértices a partir de outro vértice do subconjunto, i.e., são alcançáveis entre si.

Houve também a definição de alguns parâmetros:

- $N_{HF} = 30$
- $N_{HK} = 12$
- $N_{KW} = 12$

Para um melhor entendimento do passo a passo da metodologia, um pseudocódigo foi desenvolvido:

```
1 Normalizar Texto
2 Criar Grafo
3 Adicionar Arestas
4 Calcular Keys
5 Calcular Columns
6 Extrair Keywords
```

6.1 Normalizar Texto

Inicialmente o texto é submetido ao processo de *stemming* bem como a retirada de *stopwords*. Feito isso, cada sentença do texto será analisada separadamente. Para cada palavra presente na frase, ela será combinada à subsequente e contabilizada a quantidade de vezes que este termo conjugado ocorre nas sentenças do texto. Por exemplo, para uma sentença que possui palavras *a, b, c, d, e,* as seguintes combinações serão analisadas: (*a*), (*a,b*), (*a,b,c*), (*a,b,c,d*), (*b*), (*b,c*), (*b,c,d*),(*b,c,d,e*). Com o intuito de facilitar a compreensão das etapas, o texto do Capítulo 1 foi submetido ao algoritmo KeyGraph, apresentando em cada seção o resultado obtido.

6.2 Criar Grafo

Dentre todas as possíveis combinações, é selecionada aquela que possuir a maior frequência no texto (dando preferência pelo candidato com mais palavras). Este processo deverá

ser realizado para todas as sentenças do documento, armazenando os termos em uma lista ordenada (D_{terms}). Os termos mais frequentes são selecionados, seja HF o conjunto dos N_{HF} termos com maior frequência em D_{terms}, cria-se um grafo G com N_{HF} vértices correspondentes aos elementos de HF.

Ao analisar o exemplo proposto, definiu-se o valor 10 para o HF, resultando os seguintes termos: *form*, *educac*, *text*, *dad*, *estrut*, *ger*, *merc*, *palavr*, *desd* e *grup*, gerando o grafo da Figura 6.1.

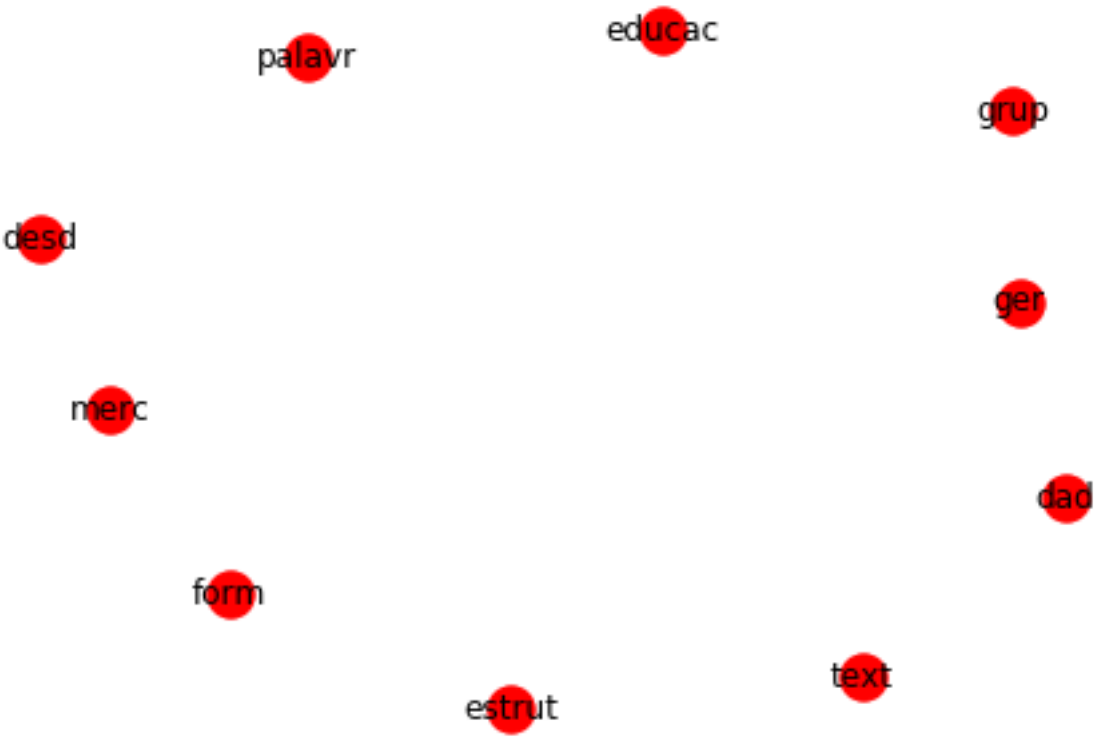

Figura 6.1: Grafo com *Dterms*

6.3 Adicionar Arestas

São adicionadas arestas entre os vértices de G que possuírem uma forte associação, definida como:

$$assoc(w_i, w_j) = \sum_{s \in D} \min(|w_i|_s, |w_j|_s), \tag{6.1}$$

a quantidade de vezes que o termo w_i ocorre em uma determinada sentença s, é representado por $|w_i|_s$.

Feito isso, são selecionados os N_{HF}-1 pares com maior valor de *assoc* e cria-se uma aresta conectando eles, este é o menor número de arestas necessárias para conectar todo o grafo. Caso o grafo tenha apenas um conceito básico ele deve formar um único componente conexo, caso contrário, a premissa é que teremos um componente conexo para cada conceito básico. Os pares com maiores valores de *assoc* no exemplo em análise foram (*text,dad*), (*text,palavr*) e (*dad,estruct*). A Figura 6.2 contém o grafo já com a adição das arestas.

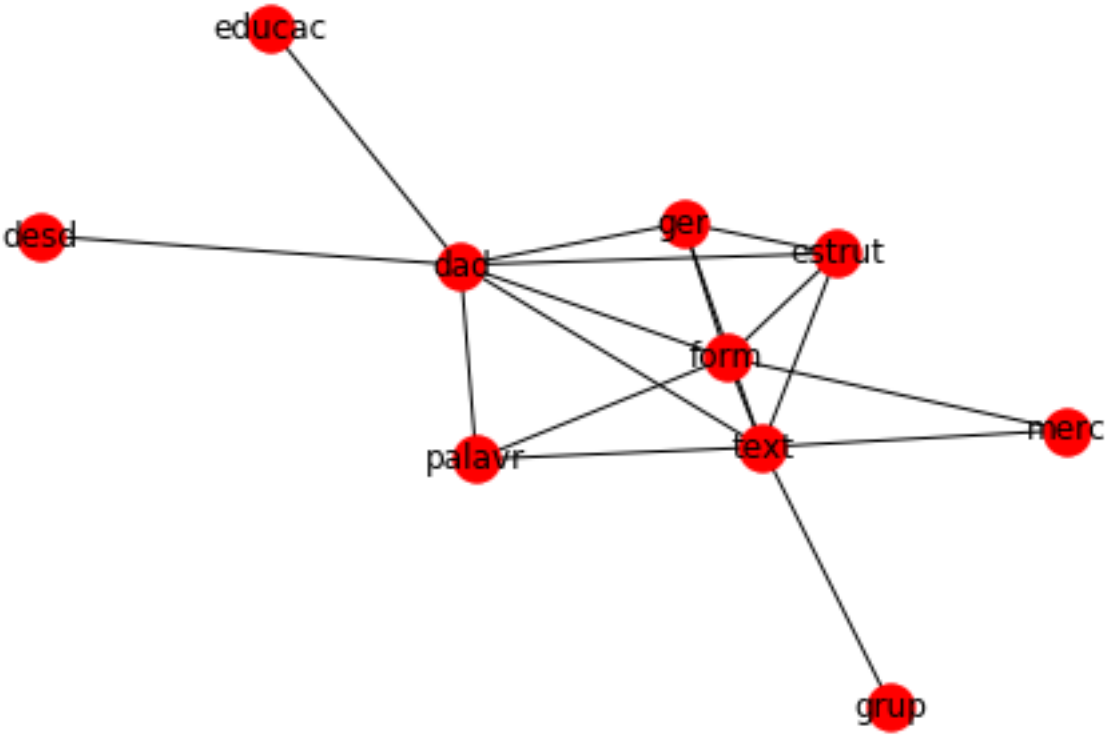

Figura 6.2: Grafo com Arestas

6.4 Calcular Keys

O objetivo é encontrar os termos que conectam os *clusters* do grafo, para isso é atribuído um valor de key_w para todos os termos w em D. Para tanto, serão utilizado duas funções auxiliares, são elas:

$$based(w,g) = \sum_{s \in D} |w|_s |g - w|_s, \tag{6.2}$$

$$neighbors(g) = \sum_{s \in D} \sum_{w \in s} |w|_s |g - w|_s, \tag{6.3}$$

$$|g - w|_s = \begin{cases} |g|_s - |w|_s, & \text{se } w \in g \\ |g|_s, & \text{se } w \notin g \end{cases}$$

based é responsável por contabilizar a quantidade de vezes que os termos em D_{terms} e os termos do *cluster g* ocorrem na mesma sentença no documento. *neighbors* realiza uma contagem similar, porém leva em consideração todos os termos do documento e não somente os contidos em D_{terms}. $|g - w|_s$ são os termos de g que são diferentes de w. Para para uma melhor ilustração das definições apresentadas, são listados os 5 termos com maiores *based* e seus respectivos valores (*form*, 35), (*text*, 31), (*dad*, 31), (*estrut*, 28) e (*educac*, 13).

A divisão dessas funções, resulta o key_w, que é a probabilidade do termo w aparecer, se todos os conceitos básicos (componentes conexos do grafo) forem considerados pelo autor. A fórmula da key_w é descrita como:

$$key(w) = 1 - \prod_{g \in G} \left(1 - \frac{based(w,g)}{neighbors(g)}\right) \tag{6.4}$$

Feito isso, são considerados os N_{HK} termos com maiores valores de $keys_w$. Os termos de N_{HK} são adicionados como vértices em G (vide Figura 6.3), caso eles já não se encontrem no mesmo. O cenário ideal seria cada termo de um *cluster* (*foundation*) em G estar conectado com todos os outros vértices do seu *cluster*, sem conexão com outros *cluster*. Alguns casos as *foundations* possuem uma leve relação, porém não o suficiente para separá-los.

O termo que apresentou o maior valor de *key* foi *dad* com o valor de 0.0955, o que faz todo o sentido, uma vez que ele é de suma importância para a compreensão dos tópicos do documento.

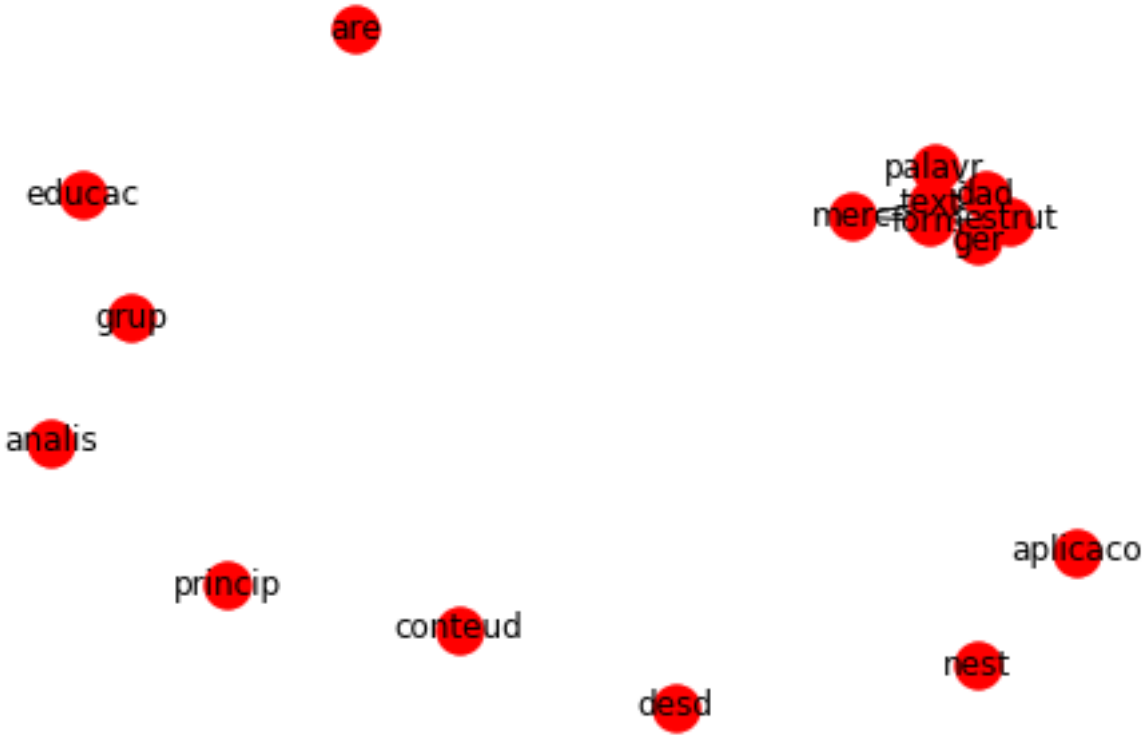

Figura 6.3: Grafo com N_{HK}

6.5 Calcular Columns

Para computar a força de uma *column* (ligação entre termos), vemos entre cada par de termos tal que w_i está em HK e w_j está em HF o valor de:

$$\text{column}(w_i, w_j) = \sum_{s \in D} \min(|w_i|_s, |w_j|_s), \tag{6.5}$$

selecionam-se os maiores valores de *column* conectando $|w_i|_s$ a dois ou mais grupos, e estas arestas são adicionadas ao grafo. Para cada aresta do grafo, é verificado se a mesma é de corte [1] para os vértices em que estão conectadas. Caso sejam de corte, essas arestas são removidas do grafo. O resultado da remoção dessas arestas no grafo de exemplo é apresentando na Figura 6.4.

Os maiores valores de *column* do exercício, foram ('text', 14), ('dad', 9), ('form', 8) e ('estrut', 6). Para conseguir um melhor detalhamento, foi selecionado o *cluster* 'dad', 'text', 'pa-

[1] é uma aresta cuja remoção em um grafo, aumenta o número de componentes conectados deste

lavr', 'estrut', 'conteud', pois no mesmo será perceptível a remoção da aresta de corte entre *palavr* e *conteud*.

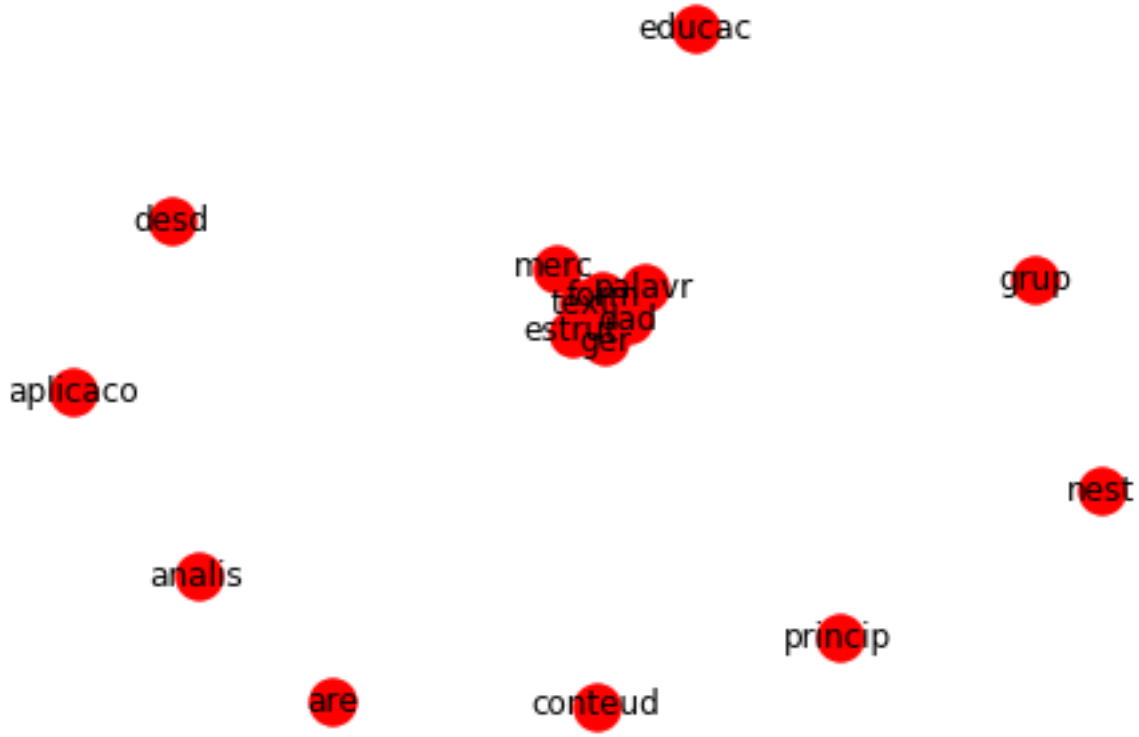

Figura 6.4: Grafo sem Aresta de Corte

6.6 Extrair Keywords

Os vértices do grafo são ordenados pela soma de valores de *column* que chegam neles. São selecionados os N_{HK} com os maiores valores de soma, resultando nas palavras chave. O N_{HK} do exemplo foi definido com o valor 4, isso devido ao tamanho do texto e a quantidade de tokens gerados. Assim, as *keywords* selecionadas foram *text*, *dad*, *estrut* e *analis*

6.7 Considerações Finais

O diferencial deste método é a extração de palavras chaves de documentos únicos, sem a necessidade de uma coleção na qual frequências de termos seriam calculados. Sua atribui-

ção de pesos é baseada na importância que os termos possuem para o entendimentos dos principais assuntos abordados no documento, por meio de cálculo da probabilidade de um termo conectar dois temas (*clusters* em g) e a probabilidade do termo estar no documento quando todos os temas ocorrem. Assim, as palavras extraídas são aquelas que possuem maior impacto para o entendimento dos assuntos do documento e/ou realizam a conexão entre dois temas abordados no texto.

CAPÍTULO 7

ANÁLISE DE RESULTADOS

7.1 Considerações Iniciais

Neste capítulo são descritos e analisados os experimentos realizados na avaliação de performance do método KeyGraph. Para tanto, foram selecionados dez docentes, sendo eles: Cristiane Sato, Denis Fantinato, Emilio Francesquini, Fabrício Olivetti, Guilherme Mota, Harlen Bargatelo, João Gois, Maycon Sambinelli, Raphael Camargo e Thiago Covões. Apesar de todos serem da área da Ciência da Computação, possuem linhas de pesquisas distintas. Harlen Bargatelo e João Gois ambos atuam no campo de Computação Gráfica, outro grupo que possui linha de pesquisa em comum são Guilherme Mota, Cristiane Sato e Maycon Sambinelli, voltados para Teoria dos Grafos e Matemática da Computação. Dois docentes que apesar de não trabalharem na mesma área, estão relacionadas são Emilio Francesquini com projetos em Computação Paralela e Sistemas Distribuídos e Raphael Camargo atuando com Neurociência Computacional. Por fim, Thiago Covões possui similaridade com Mineração de Dados enquanto Fabrício Olivetti e Denis Fantinato estudam a área de Inteligência Artificial.

Foi realizado a seleção de 10 artigos que discutissem diferentes áreas de pesquisas da Ciência da Computação. Os mesmos foram escolhidos dentre múltiplas conferências, sendo elas: Anais do Encontro Nacional de Inteligência Artificial e Computacional (ENIAC), Escola

Nacional de Alto Desempenho (ERAD), Simpósio Brasileiro de Telecomunicações e Processamento de Sinais (SBRT), Anais Estendidos da Conference on Graphics, Patterens and Images(SIBGRAPI) e Anais do Encontro de Teoria da Computação (ETC). Os nomes dos artigos e suas respectivas conferências são apresentados abaixo:

- "Sistema de medição e análise de qualidade de redes celulares móveis"(SBRT)
- "Um método para disciminação entre PSK e FSK uitlizando estatísticas de ordem superior"(SBRT)
- "Avaliação de desempenho de wavelet shrinkage pela esparsidade dos coeficientes"(SBRT)
- "Análise de Desempenho da Execução Remota de Método Aplicado ao Monitoramento de Animais com VANT"(ERAD)
- "Primitivas para aplicação de Transactional Boosting no STM Haskell"(ERAD)
- "Parametrização hierárquica de superfícies poligonais construídas com triangulação de Delaunay restrita"(SIBGRAPI)
- "Detecção de Desfolha de Soja Utilizando Redes Neurais Convolucionais"(SIBGRAPI)
- "Aspectos de complexidade parametrizada e problemas análogos em problemas de lista coloração de grafos e suas variações"(ETC)
- "Um esquema de aproximação para um problema de empacotamento com cenários"(ETC)
- "Reconhecimento de Grafos Dino de Precedência"(ETC)

7.2 Metodologia

No decorrer do capítulo é abordado constantemente dois conjuntos de textos. Para facilitar a distinção entre ele, adotou-se os termos texto-base e texto-avaliado. Sendo o texto-base, relativo aos docentes e texto-avaliado ao artigo que está sob análise. Vale ressaltar que todos os textos utilizados neste experimento receberam um tratamento, por meio do processo de normalização, remoção de acentos, *stemming* e retirada de *stopwords*.

Com objetivo de comparação e avaliação dos resultados, todos os textos-avaliado foram submetidos à duas metodologias: TFIDF e KeyGraph. Inicialmente será discorrido sobre o TFIDF e como são as etapas de tratamento de cada um dos conjuntos de textos. Em seguida, é apresentado o KeyGraph e como é realizado a sua integração com o TFIDF. Por fim, os resultados dos métodos são comparado por meio da similaridade de cossenos, e dispostos na forma de mapas de calor para uma melhor visualização.

Para a construção do texto-base, utilizou-se as dissertações dos docentes, ou seja, cada um possui respectivamente o seu trabalho de mestrado em seu texto-base. Feito isso, é computado a Frequência Relativa (TF) de todos os termos do texto-base. Em seguida, calcula-se o Inverso da Frequência do Documento (IDF), baseando-se na coleção de todos os textos-base. Por fim, para todos os termos multiplica-se o TF com o seu respectivo IDF, resultando no TFIDF. Da mesma forma o texto-avaliado é submetido aos procedimentos descritos, com a diferença que o IDF é o mesmo previamente calculado no texto-base, uma vez que o texto-avaliado não possui uma coleção. Caso haja um termo no texto-avaliado que não se encontra na coleção dos textos-base, é atribuído o valor 0 de IDF ao mesmo. Para um melhor entendimento foi desenvolvido um fluxograma das etapas na Figura 7.1.

O texto-avaliado é então submetido ao método do KeyGraph para a extração de suas palavras-chave. Na sequência é aplicado o mesmo procedimento do TFIDF, porém ao invés de utilizar o texto-avaliado é considerado os termos extraídos pelo KeyGraph, conforme apresentado na Figura 7.2.

A identificação do docente com maior afinidade ao tema retratado no artigo é dado por meio da função de similaridade de cossenos, no qual o vetor com os TFIDF do texto-avaliado é comparado com o vetor de TFIDF do docente. Para finalizar, ordenam-se os resultados, e os docentes que apresentarem maiores similaridade, possuem a maior relação com o assunto abordado no texto-avaliado.

Com o intuito de alcançar uma análise apurada, foram utilizados diferentes parâmetros entre as metodologias, seguem os cenários que foram realizados nos testes:

- TFIDFG: TFIDF - todas as palavras do texto-avaliado;
- TFIDF15: TFIDF - 15 palavras mais frequentes (TF) do texto-avaliado;

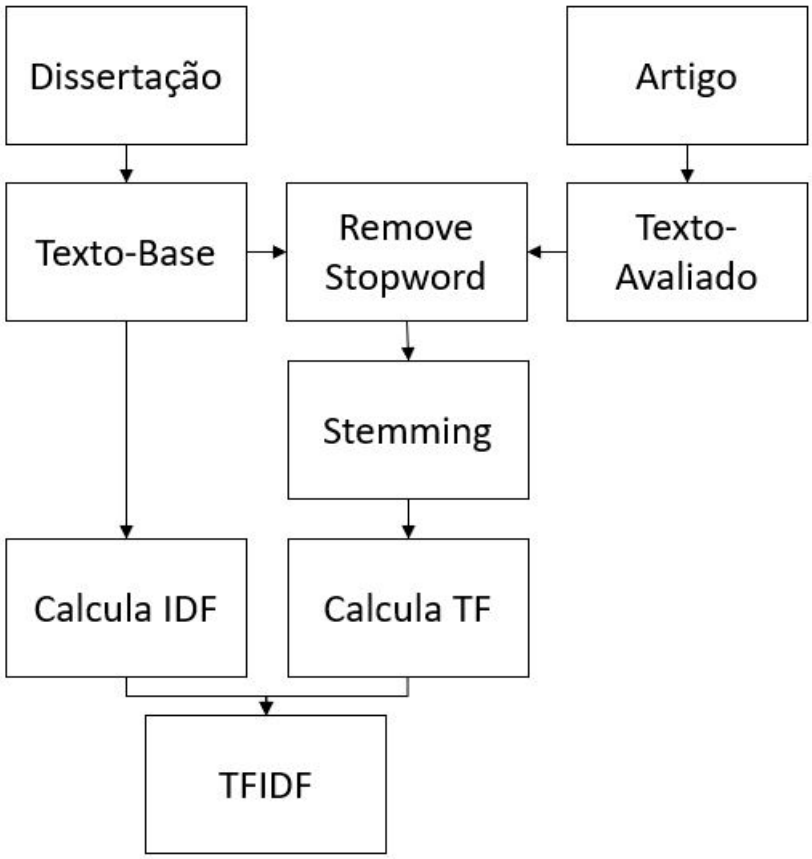

Figura 7.1: Fluxograma TFIDF

- TFIDF12: TFIDF - 12 palavras mais frequentes (TF) do texto-avaliado;
- KeyGraph30: Keygraph - $N_{HF} = 30$, $N_{HK} = 12$ e $N_{KW} = 12$;
- KeyGraph50: Keygraph - $N_{HF} = 50$, $N_{HK} = 15$ e $N_{KW} = 15$.

Para mensurar a eficiência de cada metodologia, utilizou-se de um cálculo que se baseia na posição em que o docente, que de fato possui similaridade com o artigo, foi classificado. Por exemplo, sabe-se que os artigo retirados da conferência SBRT, possui maior similaridade com o Denis Fantinato, uma vez que sua área de atuação é amplamente abordada na conferência. Dito isto, caso a metodologia classifique ele em primeiro nestes artigos, será atribuído o valor 1, caso ele fique na quinta posição, é atribuído o valor 5. Esse processo é realizado para todos os artigos, e os docentes que se relacionam ao mesmo. No final, essa nota é dividida pelo número de artigos, assim quanto mais próximo de 1 a metodologia ficar, mais assertiva ela performou. Seguem os resultados:

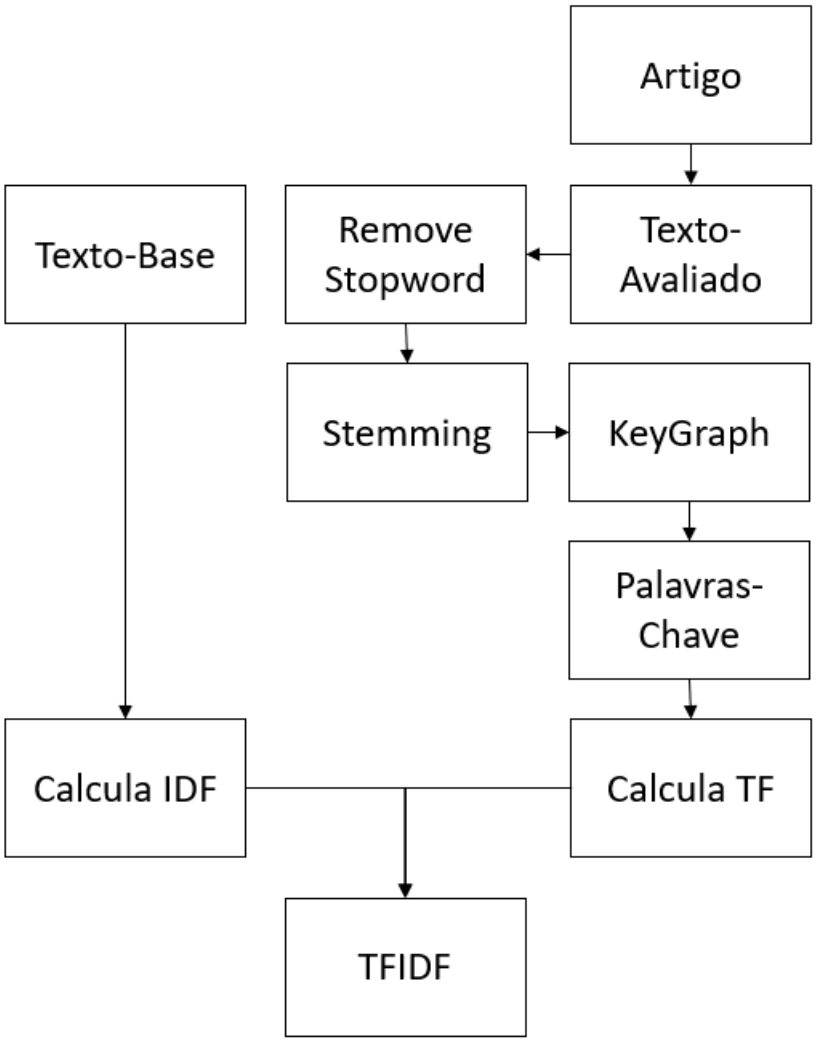

Figura 7.2: Fluxograma KeyGraph

- TFIDFG - nota 4;
- TFIDF15 - nota 3,7;
- TFIDF12 - nota 4,1;
- KeyGraph30 - nota 7;
- KeyGraph50 - nota 7.

Os resultados serão apresentados também no modelo de mapas de calor, nos quais as linhas do gráfico representam os docentes e as colunas os artigos. As cores mais escuras representam os docentes que possuem maior similaridade com o artigo em questão.

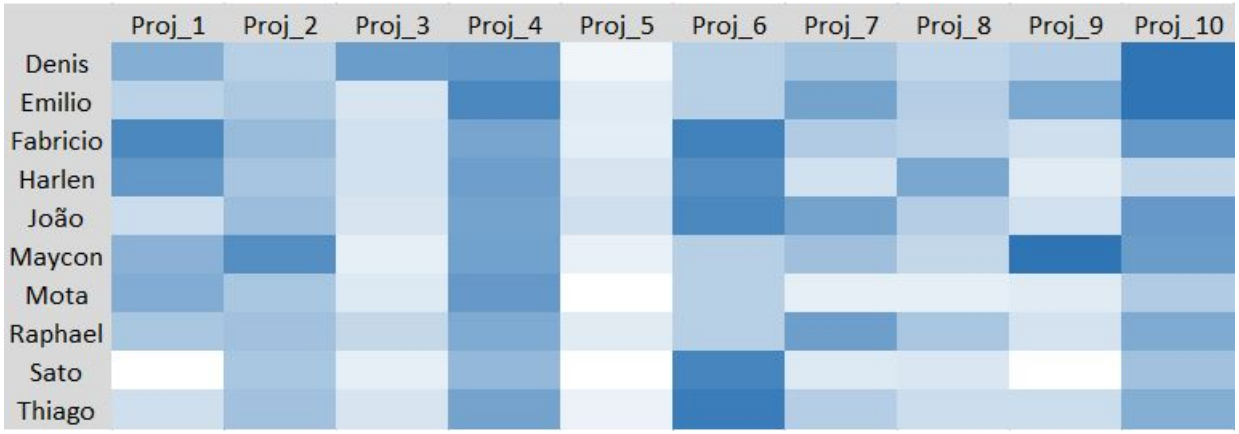

Figura 7.3: Mapa de Calor - Keygraph $N_{HF} = 30$, $N_{HK} = 12$ e $N_{KW} = 12$

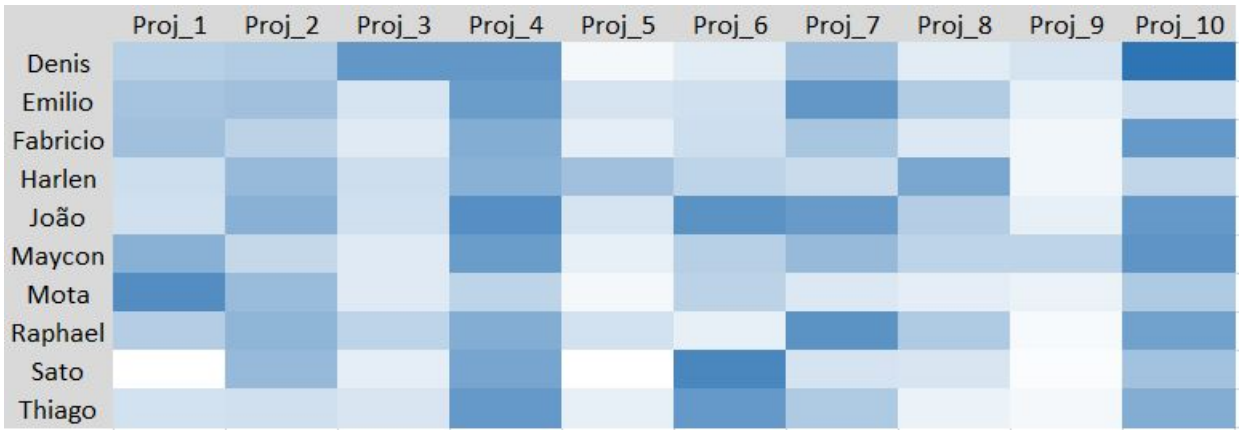

Figura 7.4: Mapa de Calor - Keygraph $N_{HF} = 50$, $N_{HK} = 15$ e $N_{KW} = 15$

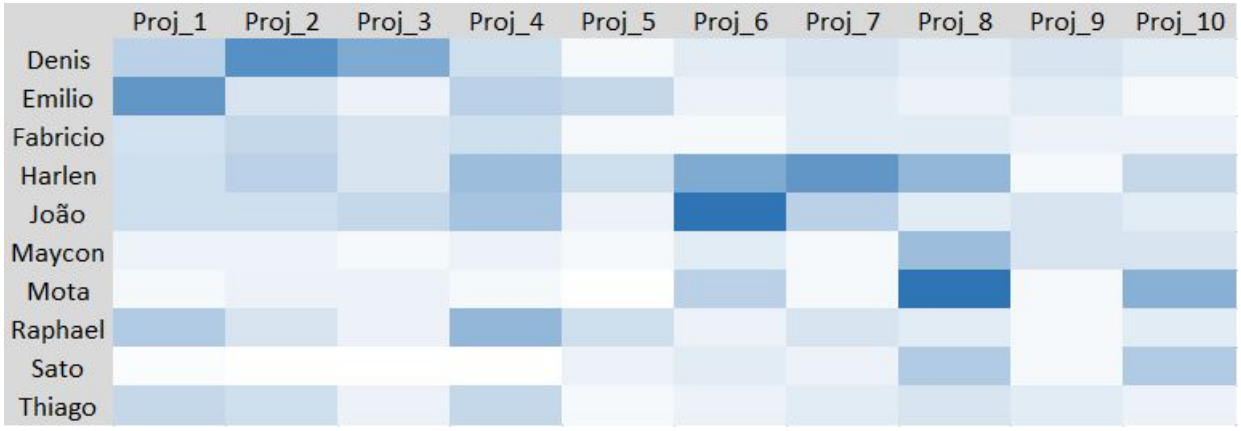

Figura 7.5: Mapa de Calor - TFIDF todas as palavras

Tanto visualmente por meio dos mapas de calor, quanto pelos cálculos fica evidente a superioridade do TFIDF sobre o KeyGraph. Esse resultado era esperado visto que o TFIDF é uma uma medida estatística muito popular, devido a sua confiabilidade em identificar a importância de uma palavra dentre uma coleção. Já o KeyGraph possui outra abordagem,

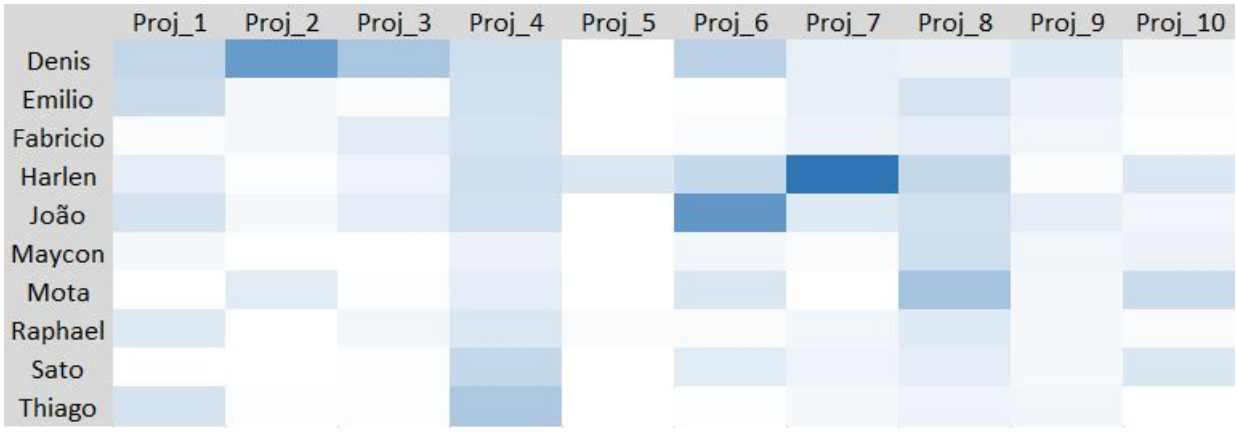

Figura 7.6: Mapa de Calor - TFIDF com 12 termos mais frequentes

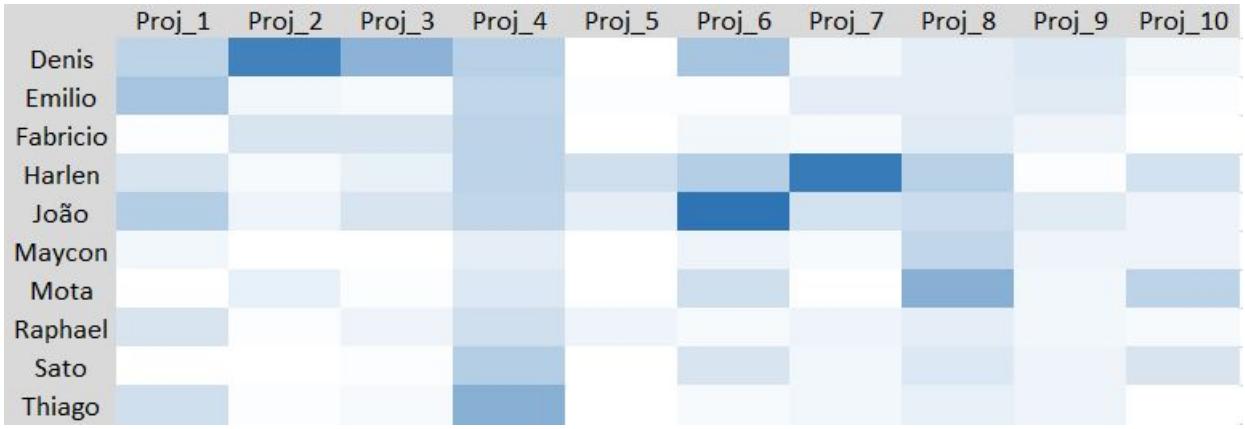

Figura 7.7: Mapa de Calor - TFIDF com 15 termos mais frequentes

que evita a necessidade de uma coleção de documentos para apontar palavras-chave, além de não levar em conta nenhum cálculo de frequência do termo.

Pelo fato do KeyGraph extrair um número pequeno de palavras-chave, a metodologia enfrenta dificuldades em identificar temas específicos. Na grande maioria dos textos, as palavras-chave extraídas eram de âmbito geral da grande área da Ciência da Computação, tais como "dados", "algoritmo"e "sistema", ou seja, termos que possuem sempre muita ocorrência nos textos referentes a área de tecnologia. Assim, a metodologia não consegue apontar o docente que possui a maior relação com o texto, uma vez as palavras-chave são constantes na maioria dos textos-avaliado e textos-base. Isso resulta em similaridades parelhas, impossibilitando diferenciar docentes que possuem destaque em uma linha de pesquisa específica.

Vale ressaltar que os dois cenários no qual o KeyGraph foi exposto, de uma maneira geral, não houve diferença de desempenho. Visto que, em um número de artigos sua performance foi superior com o KeyGraph30, como no Proj1 no qual o docente foi recomendado na primeira posição enquanto no KeyGraph50 o mesmo caiu para a quinta posição. Por outro lado, no Proj2 haviam dois docentes diretamente relacionados ao artigo, com os parâmetros do KeyGraph50 eles foram ranqueados em primeiro e terceiro, em contrapartida eles passaram para terceiro e quarto no KeyGraph30. Em suma, uma variação nos parâmetros utilizados na metodologia não impactaram no resultado, isso porque a nota geral do algoritmo se manteve a mesma.

O mesmo vale para o TFIDF que apresentou leves variações com as mudanças nos parâmetros, porém mesmo nos cenários em que foram utilizados a mesma quantidade de termos do KeyGraph, TFIDF12 e TFIDF15, o desempenho se manteve constante e soberano quando comparado ao KeyGraph30 e KeyGraph50. Ressaltando a superioridade do TFIDF quando comparado a uma quantidade equivalente de termos, o que em teoria deveria representar a mesma adversidade enfrentado pelo KeyGraph em trabalhar com termos muito abrangentes, o TFIDF por meio de cálculos matemáticos ameniza drasticamente esta complicação.

CAPÍTULO 8

CONCLUSÃO E TRABALHOS FUTUROS

Trabalhar com textos em geral é uma tarefa árdua, no qual os mínimos detalhes impactam diretamente no resultado final. São diversas etapas para tratamento do texto, afim de possibilitar um manuseio mais "limpo". O KeyGraph é um exemplo disso, no qual a simples decisão de como selecionar uma sentença em um texto, reflete diretamente na saída do algoritmo. Uma grande dificuldade encontrada no projeto, foi a leitura dos textos presente nos PDF?s, uma vez que o modo como o documento foi salvo, alteram-se as configuração do arquivo, e no momento de extração deste texto, é necessário interpretar todas elas.

Além disso, por não haver muita documentação referente ao KeyGraph e não ser código aberto, foi necessário a implementação do método, baseando-se apenas no artigo (referenciar artigo). Muitas definições apresentadas não estavam claras, e foi necessário interpretar algumas delas, de forma que faria sentido para o projeto.

Inicialmente, a ideia era realizar a leitura do currículo Lattes de cada docente para a geração do texto-base, e foi realizado uma implementação para tal. Porém, grande parte de dados relevantes presentes nas página estavam na língua inglesa, o que impossibilitou a sua utilização porque o texto-base e o texto-avalido devem estar na mesma língua.

Mas apesar das dificuldades, o conhecimento adquirido durante o período na Universidade ajudou a sanar grande parte delas. Apesar de todas as matérias cursadas terem o seu

papel, algumas merecem um maior destaque, como Programação Orientada a Objetos, na qual eu tive o primeiro contato com a linguagem Python, utilizada nas implementações de todos os algoritmos aqui presentes, Análise de Algoritmo que foi essencial para a otimização do código, pois diversas comparações são realizadas no KeyGraph e além das matérias, a Iniciação Científica desenvolvida no meu segundo ano, construiu um pequeno conhecimento em escrita de trabalho cientifico.

Embora os resultados encontrados não serem o esperado, há possibilidades para aprimorar esta pesquisa. Acredito que o aumento da coleção dos docentes é um ponto de suma importância, tanto na questão de quantidade de documentos, como na de docentes. Com a adição de artigos publicados, orientações de mestrado/doutorado e projetos de pesquisa à coleção, a linha de pesquisa do docente se manterá sempre atualizada. Possibilitando uma maior eficiência do algoritmo, uma vez que ele estará trabalhando com dados atuais.

REFERÊNCIAS

[Aggarwal e Zhai 2012]AGGARWAL, C. C.; ZHAI, C. *Mining text data*. [S.l.]: Springer Science Business Media, 2012.

[Alvares, Garcia e Ferraz 2005]ALVARES, R. V.; GARCIA, A. C. B.; FERRAZ, I. Stembr: a stemming algorithm for the brazilian portuguese language. In: SPRINGER. *Portuguese Conference on Artificial Intelligence*. [S.l.], 2005. p. 693–701.

[Baker, Isotani e Carvalho 2011]BAKER, R.; ISOTANI, S.; CARVALHO, A. Mineraçao de dados educacionais: Oportunidades para o brasil. *Brazilian Journal of Computers in Education*, v. 19, n. 02, p. 03, 2011.

[Camilo e Silva 2009]CAMILO, C. O.; SILVA, J. C. da. *Mineraçãode Dados: Conceitos, Tarefas, Métodos e Ferramentas*. [S.l.], 2009.

[Feldman e Sanger 2006]FELDMAN, R.; SANGER, J. *Text Mining Handbook: Advanced Approaches in Analyzing Unstructured Data*. [S.l.: s.n.], 2006.

[Junior 2007]JUNIOR, J. R. C. Desenvolvimento de uma metodologia para mineração de textos. *Departamento de Engenharia Elétrica, Pontífica Universidade Católica do Rio de Janeiro*, 2007.

[Morais e Ambrósio 2007]MORAIS, E. A. M.; AMBRÓSIO, A. P. L. *Mineração de Textos*. [S.l.], 2007.

[Ohsawa, Benson e Yachida 1998]OHSAWA, Y.; BENSON, N. E.; YACHIDA, M. Keygraph: Automatic indexing by co-occurrence graph based on building construction metaphor. In: IEEE. *Proceedings IEEE International Forum on Research and Technology Advances in Digital Libraries-ADL'98-*. [S.l.], 1998. p. 12–18.

[Orengo e Huyck 2001]ORENGO, V. M.; HUYCK, C. A stemming algorithm for the portuguese language. In: IEEE. *Proceedings Eighth Symposium on String Processing and Information Retrieval*. [S.l.], 2001. p. 186–193.

[Passini 2012]PASSINI, M. L. C. *MINERAÇÃO DE TEXTOS PARA ORGANIZAÇÃO DE DOCUMENTOS EM CENTRAIS DE ATENDIMENTO*. Dissertação (Mestrado) — Universidade Federal do Rio de Janeiro, 2012.

[Pezzini 2017]PEZZINI, A. Mineração de textos: Conceito, processo e aplicações. *REAVI-Revista Eletrônica do Alto Vale do Itajaí*, v. 5, n. 8, p. 58–61, 2017.

[Portuguese stemming algorithm]PORTUGUESE stemming algorithm. http://snowball.tartarus.org/algorithms/portuguese/stemmer.html. Acessado: 09/05/2019.

[Rezende, Marcacini e Moura 2011]REZENDE, S. O.; MARCACINI, R. M.; MOURA, M. F. O uso da mineração de textos para extração e organização não supervisionada de conhecimento. *Revista de Sistemas de Informação da FSMA*, v. 7, p. 7–21, 2011.

[Uber 2004]UBER, J. L. Descoberta de conhecimento com o uso de text mining aplicada ao sac. Universidade Regional de Blumenau, 2004.

[Wives 1999]WIVES, L. K. Um estudo sobre agrupamento de documentos textuais em processamento de informações não estruturadas usando técnicas de"clustering". 1999.