UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ
DEPARTAMENTO ACADÊMICO DE INFORMÁTICA
CURSO DE SISTEMAS DA INFORMAÇÃO

LUCAS FOLMANN LIMA

ANÁLISE SEMÂNTICA PARA INDICAÇÃO DE CURRÍCULOS SIMILARES

CURITIBA
2018

LUCAS FOLMANN LIMA

ANÁLISE SEMÂNTICA PARA INDICAÇÃO DE CURRÍCULOS SIMILARES

Proposta apresentada à disciplina de Trabalho de Conclusão de Curso do Bacharelado em Sistemas de Informação, da Universidade Tecnológica Federal do Paraná, como requisito parcial para obtenção do título de bacharel em Sistemas de Informação.

Orientadora: Profª. Ana Cristina Kochem Vendramin
Coorientadora: Profª. Anelise Munaretto Fonseca

CURITIBA
2018

Ministério da Educação
UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ
Câmpus Curitiba
Diretoria de Graduação e Educação Profissional
Departamento Acadêmico de Informática
Coordenação do Curso de Bacharelado em Sistemas de Informação

TERMO DE APROVAÇÃO

"ANÁLISE SEMÂNTICA PARA INDICAÇÃO DE CURRÍCULOS SIMILARES"

por

"LUCAS FOLMANN LIMA"

Este Trabalho de Conclusão de Curso foi apresentado no dia 26 de novembro de 2018 como requisito parcial à obtenção do grau de Bacharel em Sistemas de Informação na Universidade Tecnológica Federal do Paraná - UTFPR - Câmpus Curitiba. O aluno foi arguido pelos membros da Banca de Avaliação abaixo assinados. Após deliberação a Banca de Avaliação considerou o trabalho _____.

_____ **Ana Cristina Barreiras Kochem Vendramin** (Orientadora - UTFPR/Curitiba)	_____ **Anelise Munaretto Fonseca** (Coorientadora - UTFPR/Curitiba)
_____ **Thiago Henrique Silva** (Avaliador 1 – UTFPR/Curitiba)	_____ **<Profa. Leyza Baldo Dorini>** (Avaliadora 2 – UTFPR/Curitiba) (Professora Responsável pelo TCC – UTFPR/Curitiba)
_____ **<Prof. Leonelo Dell Anhol Almeida>** (Coordenador do curso de Bacharelado em Sistemas de Informação – UTFPR/Curitiba)	

"A Folha de Aprovação assinada encontra-se na Coordenação do Curso."

AGRADECIMENTOS

Agradeço primeiramente à minha mãe, Marise Folmann, que sempre me incentivou e me apoiou nos momentos mais difíceis, não apenas no desenvolvimento deste projeto, mas também em minha formação como um todo.

Agradeço às professoras Drª Ana Cristina Kochem Vendramin e Drª Anelise Munaretto Fonseca, pelo conhecimento compartilhado, pelo apoio e orientação em cada etapa do projeto, sempre com empenho e dedicação.

Um agradecimento especial para Rafaela Somavila Lima, pelo companheirismo, pela ajuda durante todos os momentos e por compreender minha ausência durante o tempo dedicado aos estudos.

Aos amigos que sempre estiveram comigo durante todos estes anos, em especial Alexandre Grocholski, Gabriel Klöckner e Priscila Mueller. As risadas e os momentos, que vocês compartilharam comigo nessa etapa tão desafiadora, também fizeram toda a diferença.

Agradeço aos demais professores da universidade que ministraram suas aulas com dedicação e contribuíram, de algum modo, para minha formação profissional.

A todos que, direta ou indiretamente, fizeram parte da minha formação, o meu eterno agradecimento.

“*Try not. Do... or do not. There is no try*.”

Yoda

RESUMO

FOLMANN, Lucas. Análise Semântica para Indicação de Currículos Similares. 2018. 54 f. Trabalho de Conclusão de Curso – Curso de Bacharelado em Sistemas de Informação, Universidade Tecnológica Federal do Paraná. Curitiba, 2018.

O conceito de similaridade é bastante complexo e vem sendo discutido no contexto da linguística, filosofia, informática, entre outros. Cada campo de estudo provê sua própria definição do que entende por similaridade, mas o que se mantém em cada uma delas é o uso da matemática. A forma mais comum de identificar a área de interesse e atuação de uma pessoa é através de seu currículo. Calcular a similaridade entre textos não estruturados, como é o caso de um currículo, é uma tarefa bem mais complexa do que somente uma comparação direta entre palavras. É necessário realizar uma comparação semântica do texto. O presente trabalho tem como intuito automatizar este processo de análise e identificação de acadêmicos com atuação em áreas similares, através da aplicação de análise semântica explícita em currículos extraídos da plataforma Lattes em formato XML.

Palavras-chave: similaridade, análise semântica explícita, comparação de currículos.

ABSTRACT

FOLMANN, Lucas. Semantic Analysis for Indication of Similar Resumes. 2018. 54 f. Course Conclusion Work - Bachelor's Degree in Information Systems, Federal Technological University of Paraná. Curitiba, 2018.

The concept of similarity is quite complex and has been discussed in the context of linguistics, philosophy, computer science, among others. Each field of study provides its own definition of what it means by similarity, but what remains in each of them is the use of mathematics. The most common way to identify a person's area of interest and performance is through their curriculum. Calculating the similarity between unstructured texts, such as a curriculum, is a much more complex task than just a direct comparison between words. It is necessary to make a semantic comparison of the text. The present work aims to automate this process of analysis and identification of academics working in similar areas, through the application of explicit semantic analysis in curricula extracted from the Lattes platform in XML format.

Keywords: similarity, explicit semantic analysis, curriculum correlation.

LISTA DE FIGURAS

LISTA DE SIGLAS

API	*Application Program Interface*
ABAP	*Advanced Business Application Programming*
BSD	*Berkeley Software Distribution*
CAPES	Coordenação de Aperfeiçoamento de Pessoal de Nível Superior
CSS	*Cascading Style Sheets*
CNPq	Conselho Nacional de Desenvolvimento Científico e Tecnológico
ESA	*Explicit Semantic Analysis*
GUI	*Graphical User Interface*
HTTP	*HyperText Transfer Protocol*
HTML	*HyperText Markup Language*
IDC	*International Data Corporation*
IDF	*Inverse Document Frequency*
IDE	*Integrated Development Environment*
IEEE	*IEEEXplore Digital Library*
JVM	*Java Virtual Machine*
LCS	*Longest Common Subsequence*
LDA	*Latent Dirichlet Allocation*
MAV	*Metric for Atomic Values*
MVC	*Metric for Complex Values*
PHP	*Hypertext Preprocessor*
SGBD	Sistema de Gerenciamento de Banco de Dados
SQL	*Structured Query Language*
TCC	Trabalho de Conclusão de Curso
TF	*Term Frequency*
TF-IDF	*Term Frequency - Inverse Document Frequency*
UML	*Unified Modeling Language*
URL	*Uniform Resource Locator*
UTFPR	Universidade Tecnológica Federal do Paraná
WNetSS	*WordNet-based Semantic Similarity*
XML	*eXtensible Markup Language*

SUMÁRIO

1 INTRODUÇÃO

Com o crescimento da comunidade acadêmica, tanto em membros quanto em áreas de pesquisa e publicações, torna-se complexo analisar e identificar profissionais com interesses semelhantes para sugerir colaborações. Desta forma, automatizar processos que façam essa análise é vantajoso, como por exemplo, extrair e analisar informações dos currículos de pessoas do meio acadêmico.

Quando acadêmicos desejam localizar colegas que possuem área de atuação ou interesses similares no campo da pesquisa, esse sempre se torna um esforço manual.

A forma mais comum de identificar a área de interesse e atuação de uma pessoa é através de seu currículo. No momento de localizar alguém que possua um currículo similar, muitos dados podem ser observados e levados em conta, como por exemplo: a autodescrição, áreas de atuação, produções bibliográficas, orientações em trabalhos, participações em bancas/congressos, entre outras, o que torna esse processo possivelmente demorado e trabalhoso.

No Brasil, o Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq) disponibiliza uma base de dados pública de currículos: a plataforma *Lattes* (Lattes, 2018). Pelo seu porte, a plataforma é uma ótima opção de fonte de dados para localizar currículos. Entretanto, não há na plataforma a possibilidade de extrair algumas informações mais avançadas, como, por exemplo, similaridade entre currículos.

Existem algumas iniciativas que abrangem as funcionalidades da plataforma *Lattes*, como por exemplo: uma rede social acadêmica de coautorias (Mena-Chalco et al, 2014); descrições sobre as informações presentes nos currículos (Digiampietri et al, 2012); ferramentas para a extração e/ou mineração dos dados da plataforma (Alves et al, 2011; Mena-Chalco et al, 2009) e uma proposta de uma ferramenta de sugestão de acadêmicos por coautoria em publicações (Colonetti, 2016). Existem também ferramentas que não utilizam a plataforma *Lattes*, mas que tentam agrupar acadêmicos, como por exemplo as redes sociais: *ResearchGate* (Ovadia, 2014) e Mendeley (Zaugg, 2011).

Nenhuma das ferramentas encontradas propõe um método de consulta automatizado e que leve em consideração a parte principal de um currículo: o texto não estruturado escrito pelo autor, onde ele se autodescreve.

Calcular a similaridade entre textos não estruturados é uma tarefa bem mais complexa do que somente comparação direta entre palavras. É necessário realizar uma comparação semântica do texto.

Na literatura pode-se encontrar algumas pesquisas que utilizam métricas de similaridade para calcular a semelhança semântica de palavras/texto, como por exemplo: Corley e Mihalcea (2005) que utilizam o *WordNet* (um banco de dados léxico) e Gabrilovich e Markovitch (2007) que propuseram a Análise Semântica Explícita (ESA), uma abordagem utilizando técnicas de inteligência artificial e a Wikipédia como base de conhecimento.

Após identificação de todas estas necessidades, foi desenvolvida uma aplicação que permite encontrar similaridade entre acadêmicos através do texto não estruturado de seus currículos. Para tal, é utilizada a Análise Semântica Explícita (ESA) adaptada para a linguagem portuguesa (*pt-BR*).

1.1 OBJETIVOS

Essa seção apresenta o objetivo geral e os objetivos específicos deste trabalho.

1.1.1 Objetivo Geral

O objetivo geral do presente trabalho é desenvolver uma aplicação para calcular a similaridade entre currículos e indicar acadêmicos que atuem em áreas semelhantes.

1.1.2 Objetivos Específicos

Os objetivos específicos deste trabalho são:

- Utilizar um algoritmo para calcular a similaridade semântica dos currículos coletados;
- Construir uma forma de visualização para demonstração dos resultados;
- Desenvolver uma aplicação para comparação curricular dos acadêmicos da UTFPR.

1.2 JUSTIFICATIVA DO PROBLEMA

Encontrar acadêmicos que atuem na mesma área de pesquisa pode ser um trabalho laborioso caso seja feito manualmente. Quando pesquisadores desejam encontrar outros que atuem na mesma área de pesquisa eles podem, por exemplo, conversar com conhecidos, falar com outros colegas da mesma instituição ou buscar pesquisadores que publicaram artigos na área desejada. O problema é que esse processo pode ser demorado.

O presente trabalho tem como intuito automatizar este processo de análise e identificação de acadêmicos com atuação em áreas similares, através da aplicação de métricas e algoritmos de similaridade em currículos extraídos da plataforma Lattes em formato XML.

> Pesquisas colaborativas são cada vez mais populares e importantes no ciclo acadêmico, porém não existem plataformas para recomendação desses potenciais colaboradores. (CHEN et al., 2011).

Foram encontradas algumas pesquisas parecidas com o tema deste trabalho: Chen et al. (2011) propõe o *CollabSeer*, um sistema que recomenda potenciais pesquisadores baseados na estrutura da rede de coautores, e Colonetti (2016) que também calcula a similaridade pela coautoria, comparando e agrupando profissionais (se um profissional X publicou junto com outro profissional Z e Y também publicou com Z, há uma possível similaridade de interesse entre X e Y). Porém, como os próprios autores comentaram, seria possível aprimorar os resultados caso as expertises dos acadêmicos fossem levadas em conta no cálculo da similaridade.

Até o presente momento, não foi encontrado na literatura nenhum trabalho que calcule a similaridade semântica em dados não-estruturados de currículos. Sendo que esta abordagem poderia até mesmo complementar à similaridade de coautoria, realizando indicações mais assertivas.

Este trabalho não está voltado somente a pesquisadores, mas a todo meio acadêmico. O estudante poderia se beneficiar desta pesquisa ao buscar professores para orientação, por exemplo. O professor poderia verificar a similaridade dos alunos com a área na qual ele atua.

1.3 ORGANIZAÇÃO DO DOCUMENTO

No próximo capítulo apresenta-se a fundamentação teórica, onde são descritos alguns conceitos e métricas referentes à similaridade. Posteriormente, no capítulo 3, é apresentado o estado da arte, onde os três algoritmos de similaridade semântica mais utilizados atualmente são abordados. No capítulo 4 são expostos os passos metodológicos que foram utilizados na elaboração deste trabalho. O capítulo 5 trata sobre o desenvolvimento da aplicação, seguido no capítulo 6 pelas telas e resultados. Por fim, no capítulo 7 se encontram as considerações finais.

2 FUNDAMENTAÇÃO TEÓRICA

O presente capítulo tem como objetivo apresentar alguns conceitos e métricas de similaridade, de forma a orientar no entendimento e interpretação das abordagens que serão expostas posteriormente.

2.1 CONCEITO DE SIMILARIDADE

Segundo Ali (2011), o conceito de similaridade é bastante complexo e vem sendo discutido no contexto da linguística, filosofia e informática. Ele tem sido um assunto de grande interesse na história da humanidade há muito tempo. Mesmo antes de computadores terem sido criados já existia interesse em encontrar similaridades.

Cada campo de estudo provê sua própria definição do que entende por similaridade. De acordo com Shepard (1962), em psicologia similaridade se refere à proximidade de duas representações mentais. Na música, a similaridade pode ser calculada entre dois ou mais fragmentos musicais. Já na geometria existe similaridade se ambos objetos possuírem a mesma forma.

Definições de similaridade são diferentes para cada área, mas o que se mantem em cada uma delas é o uso da matemática. De acordo com Ali (2011), após o início de duas novas áreas de estudo no século XX, Teoria da Informação e Ciência da Computação, o tópico similaridade não diminuiu, ao contrário, com o uso do computador tem sido mais fácil encontrar similaridade entre dois ou mais objetos. Ocorreu que com a introdução dessas grandes áreas foram criados algoritmos para desenvolver novos modos de calcular similaridade de maneira ainda mais simples, rápida e tão correta quanto possível.

Conforme mencionado por Lin (1998), o conceito de similaridade é intuitivo e deveria ser quase o mesmo para a maioria das pessoas.

As intuições apresentadas por Lin (1998) são descritas a seguir:

1) A similaridade entre dois objetos A e B está relacionada com os pontos em comum entre eles. Quanto mais eles têm em comum, maior a similaridade entre eles.
2) A similaridade entre dois objetos A e B está relacionada com suas diferenças. Quanto mais atípicos eles forem entre si, menor a similaridade entre eles.
3) A similaridade entre dois objetos A e B chega ao máximo quando ambos são idênticos entre si.

Para Gonçalves e Mello (2006), uma busca por similaridade envolve a comparação entre dois dados com a intenção de determinar um grau de similaridade entre os elementos, considerando um conjunto de atributos que determinam a similaridade entre eles.

Similaridade pode ser separada em categorias diferentes para especificar que tipo de similaridade é necessária, por exemplo: similaridade de imagem ou similaridade de som (Ali, 2011).

Uma das categorias de similaridade encontradas na literatura é a similaridade textual, onde, de acordo com Ali (2011), a ideia é analisar duas ou mais cadeias de caracteres e compará-las entre si para encontrar quão similares elas são. A similaridade textual pode ainda ser dividida em subcategorias como, por exemplo: **similaridade estrutural**, **similaridade de dados** e **similaridade semântica**.

Segundo Gonçalves e Mello (2006), a similaridade estrutural é amplamente utilizada na comparação de documentos semiestruturados (exemplo: *XML*). Nesta abordagem abstrai-se as informações contidas no documento e considera-se apenas a estrutura desses documentos (exemplo: *tags*).

Ainda de acordo com Gonçalves e Mello (2006), na similaridade de dados, além da estrutura do documento, as informações contidas no documento são utilizadas para estabelecer a similaridade.

De acordo com Harispe et al. (2015), a análise da similaridade semântica é realizada em um conjunto de palavras ou textos, onde a ideia de distância entre eles é baseada na semelhança de seu significado ou conteúdo semântico.

Há uma grande discussão entre os pesquisadores sobre quais algoritmos de similaridade utilizar para comparação de textos e em qual momento usá-los. O problema não consiste apenas em saber o quanto esses algoritmos se destacam, mas também o quão bom eles são para uma determinada tarefa. Alguns são bons o suficiente para executar uma pesquisa de similaridade em textos curtos, porém em textos longos apresentam resultados insatisfatórios. Alguns se preocupam com o tempo de execução, enquanto outros se preocupam mais com a precisão dos resultados.

A seguir serão apresentadas algumas métricas que podem ser utilizadas na comparação de textos. No capítulo 3 serão apresentados algoritmos de similaridade semântica que utilizam algumas destas métricas.

2.2 MÉTRICAS UTILIZADAS PARA SIMILARIDADE EM TEXTOS

De acordo com Gregorev et al. (2017), em computação, um texto é apenas uma série de caracteres sem estrutura particular que lhe seja imposta. Por isso, textos também são chamados de dados não estruturados. No entanto, para os seres humanos, textos certamente possuem uma estrutura, a qual usamos para entender o conteúdo.

Ainda segundo Gregorev et al. (2017), para um computador, a maneira mais simples de extrair a informação e entender o texto é chamada de *Bag of Words*: recebe-se um texto, divide-o em palavras individuais (chamadas *tokens*) e, em seguida, representa-se o texto como uma coleção não ordenada de *tokens*, juntamente com alguns pesos associados a cada *token*.

Por exemplo, recebe-se um documento, que consiste na frase "*usamos Java para Data Science porque gostamos de Java*". Essa frase pode ser representada da seguinte maneira:

$$(usamos, 1), (Java, 2), (para, 1), (Data, 1), (Science, 1), (porque, 1), (gostamos, 1), (de, 1)$$

Aqui, cada palavra é ponderada pelo número de vezes que ela ocorre no documento.

Com esta representação de documentos é possível realizar comparações entre dois ou mais documentos.

Por exemplo, utilizando outra frase: "*Java é um bom desenvolvimento empresarial*", pode-se representá-la da seguinte maneira:

$$(Java, 1), (\text{é}, 1), (um, 1), (bom, 1), (desenvolvimento, 1), (empresarial, 1)$$

É possível verificar que existe uma interseção entre esses dois documentos, o que, de acordo com Gregorev et al. (2017), demonstra que os documentos são semelhantes, e quanto maior a interseção, maior a semelhança.

Agora, pensando em palavras como dimensões em algum espaço vetorial e pesos como valores para essas dimensões, pode-se representar documentos como vetores (ver Tabela 1).

	usamos	Java	para	Data	Science	porque	gostamos	de	é	um	bom	desenvolvimento	empresarial
Doc1	1	2	1	1	1	1	1	1	0	0	0	0	0
Doc2	0	1	0	0	0	0	0	0	1	1	1	1	1

Tabela 1: Representação de um documento na forma de um vetor

Gregorev et al. (2017) mencionam que levando em conta essa representação vetorial, pode-se considerar o produto interno entre dois vetores como sendo uma medida de similaridade.

Se dois documentos têm muitas palavras em comum, o produto interno entre eles será alto e, se eles não compartilharem termos, o produto interno é zero. Para calcular esse produto interno pode-se utilizar a similaridade cosseno (que será apresentada na Seção 2.2.2).

No exemplo anterior foi utilizado o número de ocorrências como métrica. Essa métrica é conhecida como frequência do termo (TF). No entanto, algumas palavras são mais relevantes do que outras e a TF nem sempre captura isso. A seguir são apresentadas outras métricas que podem ser utilizadas para calcular similaridade textual.

2.2.1 Frequência do Termo – Frequência Inversa nos Documentos (TF-IDF)

Segundo Leskovec et al. (2014), Frequência do Termo – Frequência Inversa nos Documentos (TF-IDF) é uma maneira de demonstrar a relevância das palavras que aparecem em um texto. No TF-IDF algumas palavras tornam-se mais importantes do que outras e afetam a pontuação de similaridade. Em outras palavras, o TF-IDF é um peso que torna algumas palavras mais relevantes do que outras nos textos em que estão sendo comparadas.

De acordo com Gonçalves e Mello (2006), o valor TF-IDF aumenta proporcionalmente ao número de vezes que uma palavra aparece no documento, mas é compensado pela frequência da palavra no corpo documental (conjunto de todos os documentos), o que ajuda a ajustar o fato de que algumas palavras em geral aparecem mais frequentemente.

Isto fica claro se for considerada uma comparação de texto sobre o exemplo: “banco Itaú”. Caso os textos sendo comparados contenham o termo “banco”, isto não deve contribuir muito para indicar sua semelhança com o conteúdo desejado, pois “banco” é um termo comum e de duplo sentido. Porém, se os dois documentos contêm o termo “Itaú”, isso pode ser um forte indicativo de semelhança.

De acordo com Ramos (2003), o TF-IDF é calculado pela multiplicação de dois coeficientes: frequência do termo (TF) e frequência inversa nos documentos (IDF).

Segundo Thanaki (2017), a frequência do termo (TF) indica a frequência de cada uma das palavras presentes no documento ou no conjunto de dados. Então, sua equação básica é dada da seguinte forma:

$$TF(t,d) = \frac{(Número\ de\ vezes\ que\ o\ termo\ t\ aparece\ em\ um\ documento\ d)}{(Número\ total\ de\ termos\ no\ documento\ d)} \quad (1)$$

A equação (1) possui muitas variações. A que é utilizada no algoritmo Análise Semântica Explicita (ESA) de Gabrilovich e Markovitch (2009) é a seguinte:

$$TF(t,d) = \begin{cases} 1 + log_{10}\big(count(t,d)\big), & se\ count(t,d) > 0 \\ 0, & se\ count(t,d) \leq 0 \end{cases} \quad (2)$$

Onde:

t é o termo ou palavra para o qual se está calculando o TF;

d é o documento que contém o termo t;

$count(t,d)$ é uma função que calcula a quantidade de vezes que t aparece no documento d.

Conforme descrito por Thanaki (2017), a frequência inversa nos documentos (IDF) pondera quanto a palavra é importante para o documento. Isso ocorre porque quando calculamos o TF, damos a mesma importância para cada palavra. Agora, se a palavra aparecer muitas vezes no conjunto completo de dados, seu valor TF será alto, mesmo ela não sendo importante para o documento. Por exemplo, se a palavra "uns" aparece no documento 100 vezes, isso não significa que ela seja mais importante do que palavras que aparecem com menos frequência.

Assim, é preciso definir algum peso que aumente a importância dos termos relevantes de baixa frequência, este peso é o IDF.

Gabrilovich e Markovitch (2009) utilizam a seguinte equação para o cálculo do IDF:

$$IDF(t,D) = log_{10}\left(\frac{\text{Número total de documentos}}{\text{Número de documentos com o termo t}}\right) \quad (3)$$

Onde:

t é o termo ou palavra para o qual se está calculando o IDF;

D é o conjunto de todos os documentos (corpo documental).

Conforme citado anteriormente, de acordo com Ramos (2003), o TF-IDF é calculado pela multiplicação dos coeficientes TF e IDF:

$$TF\text{-}IDF(t,d,D) = TF(t,d) * IDF\ (t,D) \tag{4}$$

Onde:

t é o termo para o qual se está calculando o TF-IDF;

d é o documento que contém o termo t;

D é o conjunto de todos os documentos (corpo documental);

$TF(t,d)$ e $IDF(t,D)$ são funções que serão detalhadas a seguir.

Conforme Leskovec et al. (2014), quanto maior for o coeficiente TF-IDF, maior a relevância da palavra para o texto no qual ela está sendo analisada.

2.2.2 Similaridade Cosseno

Dangeti (2017) define a similaridade cosseno com sendo uma métrica na qual o coeficiente de similaridade é dado pelo cosseno do ângulo entre dois vetores (ou dois documentos no espaço vetorial). A equação para o cálculo da similaridade cosseno de acordo com Baeza-Yates e Ribeiro-Neto (1999) é:

$$SimilaridadeCosseno\ (X,Y) = \cos(\Theta) = \frac{\sum_{i=1}^{n} X_i * Y_i}{\sqrt{\sum_{i=1}^{n}(X_i)^2} * \sqrt{\sum_{i=1}^{n}(Y_i)^2}} \tag{5}$$

Onde:

Θ é o ângulo entre os dois vetores em um espaço vetorial;

X é um vetor com os valores dos termos de um documento;

Y é um vetor com os valores dos termos de um documento;

X_i é um dos valores de X;

Y_i é um dos valores de Y.

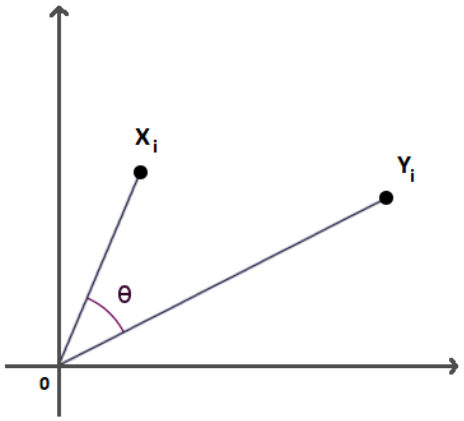

Figura 1. Similaridade Cosseno no Espaço Vetorial
Fonte: autoria própria.

A Figura 1 demonstra a similaridade cosseno aplicada ao elemento i de dois vetores (X e Y) em um espaço vetorial, resultando em Ɵ.

Conforme descrito por Dangeti (2017), a similaridade cosseno é uma métrica que demonstra através do ângulo como estão relacionados dois documentos no espaço vetorial. Se o ângulo entre os vetores do documento for 0, o resultado da similaridade será 1, indicando que ambos os documentos são idênticos. A menor similaridade possível é 0, indicando que os documentos são completamente diferentes.

Por exemplo, assumindo que V(x) = [2,1,0,2,0,1,1,1] e V(y) = [2,1,1,1,1,0,1,1] são os dois vetores, utilizando a equação (5) apresentada por Baeza-Yates e Ribeiro-Neto (1999), o cálculo da similaridade cosseno se daria por:

$$\sum_{i=1}^{n} x_i * y_i = \big((2*2)+(1*1)+(0*1)+(2*1)+(0*1)+(1*0)+(1*1)+(1*1)\big) = 9$$

$$\sqrt{\sum_{i=1}^{n}(x_i)^2} = \sqrt{(2^2+1^2+0^2+2^2+0^2+1^2+1^2+1^2)} = \sqrt{12} = 3.46$$

$$\sqrt{\sum_{i=1}^{n}(y_i)^2} = \sqrt{(2^2+1^2+1^2+1^2+1^2+0^2+1^2+1^2)} = \sqrt{10} = 3.16$$

$$\cos(\theta) = \frac{9}{3.46 * 3.16} = 0.823$$

Um valor de 0,823 indica uma semelhança muito alta entre os dois vetores, porque o maior valor possível é 1 (atingido quando o ângulo Ɵ for zero).

Para a Similaridade Semântica, os vetores de atributos X e Y são geralmente os valores TF-IDF dos documentos. Já a similaridade cosseno pode ser vista como um método de normalização do comprimento do documento durante a comparação.

2.2.3 Índice de Jaccard, Distância de Jaccard e Coeficiente de Tanimoto

As três métricas apresentadas a seguir são muito parecidas, pois elas derivam do índice de Jaccard.

O índice de Jaccard compara membros de dois conjuntos para ver quais destes membros são compartilhados e quais são distintos. Esta medida de similaridade pode variar em um intervalo de 0 a 1. Quanto mais próximo a 1, mais semelhantes são as duas populações (JACCARD, 1912).

Quando aplicado em documentos textuais, o coeficiente Jaccard propõe transformar cada texto em um conjunto de *tokens* (que podem ser palavras ou caracteres) e estabelecer a semelhança entre os mesmos através da razão entre o tamanho dos conjuntos de intersecção e união dos conjuntos de *tokens* (JACCARD, 1912).

O cálculo é simples, conforme Jaccard (1912), divide-se o tamanho da interseção dos conjuntos A e B pelo tamanho da união de A e B:

$$Índice Jaccard(A, B) = \frac{|A \cap B|}{|A \cup B|} \tag{6}$$

Onde:

A é um vetor de atributos (*tokens*);

B é um vetor de atributos (*tokens*).

Embora seja fácil de interpretar, é extremamente sensível, podendo apresentar resultados errados especialmente com amostras muito pequenas ou conjuntos de dados com observações faltantes.

A distância de Jaccard que, ao invés de semelhança, mede a dissimilaridade entre os conjuntos, pode ser encontrada subtraindo 1 do resultado do índice de Jaccard, conforme mostra a equação (JACCARD, 1912):

$$Distância Jaccard(A, B) = 1 - Índice Jaccard(A, B) = \frac{|A \cup B| - |A \cap B|}{|A \cup B|} \tag{7}$$

O coeficiente de *Tanimoto* é a junção do índice de *Jaccard* e da similaridade cosseno. Nele, também se pressupõe que cada conjunto é um vetor de atributos. Os atributos podem ou não ser binários. Entretanto, se todos forem binários, o método de Tanimoto se reduz ao método de Jaccard. O coeficiente de *Tanimoto* é encontrado a partir da seguinte equação (TANIMOTO, 1958):

$$Tanimoto(A,B) = \frac{A * B}{\left||A|\right|^2 + \left||B|\right|^2 - A * B} \quad (8)$$

Onde:

A é um vetor de atributos (*tokens*);

B é um vetor de atributos (*tokens*).

2.2.4 Distância Euclidiana ou Distância L2

Distância euclidiana também chamada de distância L2 é outra medida de proximidade no modelo de espaço vetorial.

A distância euclidiana é tão comum que quando se fala em “distância” quase sempre refere-se a essa métrica. Ela se diferencia das outras medidas de similaridade do modelo de espaço vetorial por não julgar os vetores através de ângulos, ao invés disso, a distância euclidiana é calculada diretamente entre as entradas vetoriais.

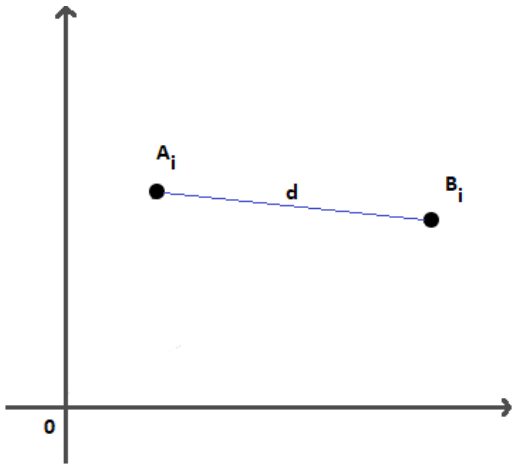

Figura 2. Distância Euclidiana no Espaço Vetorial
Fonte: Autoria própria.

A Figura 2 ajuda a visualizar o que ocorre na distância euclidiana. Ao invés de calcular-se a direção dos vetores (como é feita na Similaridade Cosseno), na Distância Euclidiana calcula-se a distância entre dois pontos representados no espaço vetorial.

A distância euclidiana examina a raiz das diferenças quadradas entre os pares coordenados dos vetores A e B, como mostra a equação (Deza, 2009):

$$Distância Euclidiana(A, B) = \sqrt{\sum_{i=1}^{n} (A_i - B_i)^2} \quad (9)$$

Onde:

A é um vetor de atributos;

B é um vetor de atributos (mesmo tamanho de A);

n é a quantidade de atributos de A (ou B, tanto faz).

2.2.5 Métricas baseadas na Distância de Edição

Dentre as diversas formas para comparar texto, uma que merece destaque é a Distância de Edição. Segundo Cormen et al. (2002) nesta métrica são realizadas operações sobre uma *string* (como inserção, remoção, troca e movimentação de caracteres) visando transformá-la na *string* com a qual ela está sendo comparada. O custo em termos de operações necessárias para tal, determina a semelhança entre elas.

Se existem pesos diferentes associados às operações (por exemplo, o custo de substituir um caractere é maior que inserir um novo caractere ou excluir) denomina-se a técnica de distância geral de edição. Caso os pesos sejam únicos para todas as operações, classifica-se a técnica como distância de edição simples ou simplesmente distância de edição.

De acordo com Navarro (2001), esta técnica tem sido adaptada para outros domínios além das *strings*, como árvores (distância de edição para árvores) ou subdomínios específicos da comparação entre *strings* (como a determinação da maior sequência de caracteres em comum entre duas *strings*).

Gonçalves e Mello (2006) citam que diversas métricas baseadas na distância de edição surgiram através de limitações impostas sobre as operações permitidas na transformação de uma *string* na outra.

Dentre elas tem-se: *Levensthein* (LEVENSTHEIN, 1966), que permite inserções, exclusões e substituições; *Hamming* (HAMMING, 1950), que permite apenas substituições (usada para detectar palavras escritas de forma errada); *Episode* (DAS et al, 1997), que permite apenas inserções (usada em pesquisas); e por último temos L*ongest Common Subsequence* (LCS) (HIRSCHBERG, 1977), que permite apenas inserções e exclusões, utilizadas para determinar a maior sequência de caracteres comum entre as *strings* comparadas.

De acordo com Naumman e Herchel (2010), Monge-Elkan é uma métrica híbrida, onde propõe-se um esquema de reconhecimento recursivo, que consiste em dividir as *strings* comparadas em *tokens* e para cada dupla de *tokens* aplicar outra métrica de Distância de Edição.

Ao final deve-se ponderar estas n-comparações em um único valor que será a semelhança entre os elementos, conforme equação (Monge e Elkan, 1996):

$$MongeElkan(s,t) = \frac{1}{K}\sum_{i=1}^{K} \max_{j=1}^{L} DistEdição(Ai, Bj) \qquad (10)$$

Onde:

s é uma *string*; t é uma *string*;

A é o conjunto de *tokens* da *string* s; B é o conjunto de *tokens* da *string* t ;

K é o número de *tokens* contidos em s; L é o número de *tokens* contidos em t ;

i é um contador utilizado para percorrer o conjunto A;

j é um contador utilizado para percorrer o conjunto B;

$DistEdição$ é uma função que calcula a similaridade entre dois *tokens* e retorna um valor numérico entre 0.0 e 1.0.

2.3 STOPWORDS

De acordo com Leskovec et al. (2014), palavras como "um", ”uns”, “a”, "o", "e" e "de" são exemplos de palavras que aparecem com muita frequência em textos. Para os seres humanos, elas têm relevância para entender o que a frase está dizendo, mas quando se trata de processamento de texto, essas palavras apenas atrapalham. Elas ocupam espaço, tornam o processo mais lento e até interferem negativamente nos resultados, uma vez que afetam o peso da ponderação de palavras (que é levado em consideração no algoritmo de similaridade).

Palavras como estas têm pouco valor de informação e são chamadas de *stopwords*. Ao filtrar essas palavras antes de executar a parte de processamento nos dados, o tempo de execução irá diminuir, menos memória será utilizada e a similaridade resultante será mais precisa (LESKOVEC et al., 2014).

Não há uma lista precisa de *stopwords*. Até existem algumas listas disponíveis na *internet*, por exemplo: Alopes (2018). Porém, como cada programa terá sua própria lista de palavras irrelevantes, a lista é controlada por entrada humana e não automatizada. Todos os algoritmos de similaridade semântica encontrados até o momento removem as *stopwords* para acelerar o processamento.

3 ESTADO DA ARTE

De acordo com Harispe et al. (2015), a análise da similaridade semântica é realizada em um conjunto de documentos ou termos, onde a ideia de distância entre eles é baseada na semelhança de seu significado ou conteúdo semântico.

> “Trata-se de ferramentas matemáticas usadas para estimar a força da relação semântica entre unidades de linguagem, conceitos ou instâncias, através de uma descrição numérica obtida de acordo com a comparação de informações que suportam seu significado ou descrevendo sua natureza.” (HARISPE et al., 2015).

Segundo Feng (2017), o termo similaridade semântica geralmente é confundido com a relação semântica. A relação semântica inclui qualquer relação entre dois termos, enquanto a similaridade semântica inclui apenas as relações do tipo "é semelhante". Ballatore et al. (2014) cita um exemplo sobre esta definição: "carro" é semelhante a "ônibus" (similaridade semântica), mas também está relacionado à "estrada" e à "condução" (relação semântica).

Segundo Budanistky (2001), tanto a similaridade semântica, quanto a distância semântica e a relação semântica, em essência, significam: "Quanto o termo A tem a ver com o termo B?". A resposta desta pergunta geralmente é um número entre -1 e 1, ou entre 0 e 1, onde 1 demonstra que o termo A e o termo B são totalmente similares (equivalentes).

Aouicha et al. (2016) afirmam que a similaridade semântica, computacionalmente, pode ser estimada através da definição de uma similaridade topológica, utilizando ontologias para definir a distância entre termos/conceitos. Por exemplo, uma métrica simples para comparação de conceitos ordenados em um conjunto parcialmente ordenado e representados como nós de um gráfico acíclico dirigido, seria o caminho mais curto que liga os dois nós conceituais.

Além disso, Aouicha et al. (2016) citam que, com base em análises de texto, a relação semântica entre unidades de linguagem (por exemplo, palavras, frases) também pode ser estimada usando meios estatísticos, como um modelo de espaço vetorial, para correlacionar palavras e contextos textuais em um conjunto de documentos.

3.1 WORDNET

Várias ferramentas podem ser utilizadas para medir a semelhança semântica entre conceitos como a WNetSS API (Aouicha et al., 2016). Essa API Java manipula uma grande variedade de medidas de similaridade semântica com base no recurso semântico do WordNet.

Miller et al. (1990) definem WordNet como sendo um banco de dados lexical. Segundo eles, o WordNet agrupa palavras em conjuntos de sinônimos chamados *synsets*. Com isso, ele é capaz de fornecer definições curtas e exemplos de utilização de cada palavra, além de registrar uma série de relações entre esses conjuntos de sinônimos e seus membros.

O WordNet pode ser visto como uma combinação de um dicionário comum e um dicionário de sinônimos. Embora seja acessível aos usuários através de um navegador da *web*, seu principal uso é na análise automatizada de texto e nas aplicações que utilizam inteligência artificial.

O banco de dados e as ferramentas de *software* foram lançados sob um estilo *Berkeley Software Distribution* (BSD) e estão disponíveis gratuitamente para *download* no site do WordNet. Tanto os dados lexicográficos (arquivos de lexicógrafo) quanto o compilador (chamado de *grind*) estão disponíveis.

> “Dados dois segmentos de texto de entrada, queremos obter automaticamente uma pontuação que indique sua similaridade no nível semântico, indo além dos métodos simples de correspondência lexical tradicionalmente utilizados para essa tarefa. Embora reconheçamos o fato de que uma métrica abrangente de similaridade semântica de texto deve levar em conta as relações entre palavras, bem como o papel desempenhado pelas várias entidades envolvidas nas interações descritas por cada um dos dois textos, tomamos um primeiro corte grosseiro neste problema para tentar modelar a semelhança semântica de textos como uma função da similaridade semântica das palavras componentes. Fazemos isso, combinando métricas de semelhança palavra a palavra e modelos de linguagem (WordNet) em uma fórmula, que é um indicador potencialmente bom da similaridade semântica dos dois textos de entrada.” (CORLEY; MIHALCEA, 2005, p.2).

3.2 ANÁLISE SEMÂNTICA EXPLÍCITA (ESA)

Uma abordagem um pouco diferente, foi desenvolvida por Gabrilovich e Markovitch (2007). Eles propuseram a Análise Semântica Explícita (ESA), um método que representa o significado de textos em um “espaço de alta dimensionalidade de conceitos derivados do Wikipédia”, como eles mesmo chamaram.

Gabrilovich e Markovitch dizem que optaram por utilizar o Wikipédia porque atualmente ele é o maior repositório de conhecimento aberto da *web*, além de ser um sistema *online* gratuito e utilizar inteligência coletiva. O Wikipédia está disponível em dezenas de idiomas, enquanto sua versão em inglês é a maior de todas, com 400 milhões de palavras em mais de um milhão de artigos (em comparação com 44 milhões de palavras em 65 mil artigos na *Encyclopedia Britannica*).

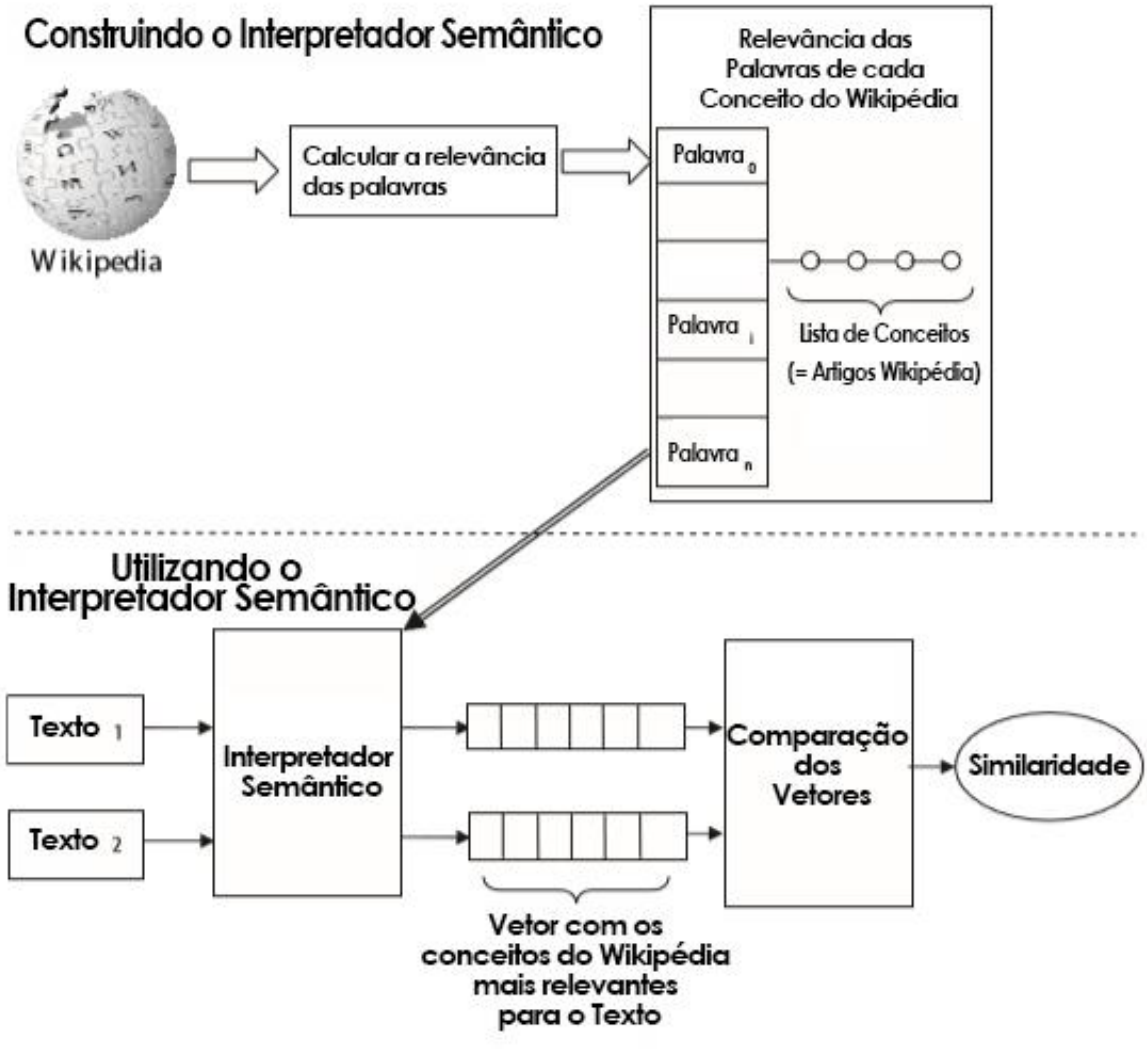

Figura 3. Interpretador Semântico
Fonte: Adaptado de Gabrilovich e Markovitch (2007)

A Figura 3 demonstra como o ESA está estruturado e como Gabrilovich e Markovitch arquitetaram o interpretador semântico. Foram utilizadas técnicas de aprendizado de máquina para construir um interpretador que mapeia fragmentos de texto, escritos em linguagem natural, para uma sequência ponderada de conceitos do Wikipédia. Esses conceitos são ordenados de acordo com suas relevâncias para o texto de entrada, ou seja, as palavras com maior coeficiente TF-IDF aparecerão por primeiro.

Os textos de entrada são dados na mesma forma que os artigos de Wikipédia, ou seja, como texto simples. Portanto, pode-se utilizar métricas convencionais de classificação de texto (ex.: TF-IDF) para classificar os conceitos representados por esses artigos de acordo com sua relevância para o fragmento de texto fornecido.

De acordo com Gabrilovich e Markovitch (2007), é essa observação fundamental que permite o uso da enciclopédia diretamente, sem a necessidade de uma compreensão profunda do conhecimento pré-catalogado.

A escolha dos artigos do Wikipédia como conceitos é bastante natural, pois cada artigo está possui um título e um texto que o descreve.

No interpretador semântico, cada conceito é representado como um vetor de palavras que ocorrem no texto do artigo correspondente. As entradas desses vetores são atribuídas com pesos usando o esquema TF-IDF (Salton e McGill, 1983). Esses pesos quantificam a força da associação entre palavras e conceitos. Para acelerar a interpretação semântica, é construído um índice invertido, que mapeia cada palavra em uma lista de conceitos nos quais ela aparece. O índice invertido também é utilizado para descartar associações insignificantes entre palavras e conceitos removendo os conceitos cujos pesos são muito baixos para uma determinada palavra.

> “Dado um fragmento de texto, primeiro ele é representado como um vetor, usando o esquema TF-IDF. O interpretador semântico itera sobre as palavras de texto, recupera as entradas correspondentes do índice invertido e as mescla em um vetor ponderado de conceitos que representa o texto fornecido.” (GABRILOVICH; MARKOVITCH, p.2, 2007)

	Texto 1: “Equipamento”	Texto 2: “Investidor”
1	Ferramenta	Investimento
2	Equipamento Digital	Investidor anjo
3	Tecnologia e Equipamento Militar	Comerciante de ações
4	Acampamento	Fundo mútuo
5	Veículo de engenharia	Margem (finanças)
6	Armamento	Teoria moderna de portfólios
7	Fabricante de Equipamentos	Investimento de capital
8	Exército Francês	Câmbio de fundo comercial
9	Equipamento de Teste Eletrônico	Fundo de conversão
10	Equipamento de Medição de Distância	Esquema Ponzi

Tabela 2. Vetor de conceitos - Exemplo
Fonte: Adaptado de Gabrilovich e Markovitch (2007)

A Tabela 2 ilustra a abordagem de Gabrilovich e Markovitch (2007). Nesta tabela são mostrados os dez conceitos da Wikipédia de maior relevância para os textos de entrada “Equipamento” e “Investidor”. Os conceitos que mais se relacionam ao texto de entrada, ou seja, tem um TF-IDF maior, aparecem em primeiro no vetor.

Conforme citado por Gabrilovich e Markovitch (2007), após obter os vetores de conceitos dos textos de entrada, é possível gerar a similaridade das duas entradas através da comparação entre os dois vetores de conceitos. Esta comparação é feita utilizando a métrica Similaridade Cosseno, apresentada na Seção 2.2.2. O resultado da comparação será um coeficiente de similaridade entre 0 e 1, sendo 1 a similaridade máxima.

Em seu trabalho, Gabrilovich e Markovitch (2009) mostram através de cálculos que a ESA é melhor que outros algoritmos como: WordNet; Roget’s Thesaurus; LSA e WikiRelate. Enquanto o WordNet teve uma correlação de 33 a 35% com a interpretação de texto feita por humanos, o ESA-Wikipédia chegou a 75%.

3.3 ALOCAÇÃO LATENTE DE DIRICHLET (LDA)

De acordo com Blei et al. (2003), na aprendizagem de máquina e no processamento de linguagem natural, um modelo de tópico é um tipo de modelo estatístico para descobrir os "tópicos" abstratos que ocorrem em uma coleção de documentos.

Segundo Blei (2012), Alocação Latente de Dirichlet (LDA) é um modelo estatístico generativo que utiliza abordagem bayesiana para aprender a estrutura latente de temas que compreendem cada um dos documentos.

Ainda de acordo com Blei (2012), a intuição por trás da LDA é que os documentos exibem múltiplos tópicos. Por exemplo, se as observações forem palavras coletadas em documentos textuais, o modelo LDA pressupõe que cada documento é uma mistura de um pequeno número de tópicos e que cada palavra é atribuível a um dos tópicos do documento. *Stopwords* são descartadas pois elas não apresentam conteúdo significativo para o tópico.

Um tópico não é definido nem semanticamente nem epistemologicamente. É identificado com base na detecção automática da probabilidade de coocorrência do termo. Uma palavra lexical pode ocorrer em vários tópicos com uma probabilidade diferente, no entanto, com um conjunto típico diferente de palavras vizinhas em cada tópico. Cada documento é considerado caracterizado por um conjunto particular de tópicos.

> “O algoritmo não tem informações sobre esses assuntos e os artigos não estão rotulados com tópicos ou palavras-chave. As distribuições interpretáveis dos tópicos surgem pelo cálculo da estrutura oculta que provavelmente gerou a coleção observada de documentos” – (BLEI, 2012, p.3)

Conforme mencionado por Blei (2012), uma das características mais marcantes do modelo LDA é o mínimo de intervenção humana necessária para sua aplicação. LDA é capaz de descobrir temas subjacentes em documentos e estabelecer *links* entre documentos mesmo não tendo nenhuma informação anterior sobre estes temas.

3.4 RESUMO DO ESTADO DA ARTE

No presente capítulo foi apresentado o estado da arte de similaridade textual aplicado à computação, descrevendo o conceito de similaridade e as principais métricas e algoritmos utilizados.

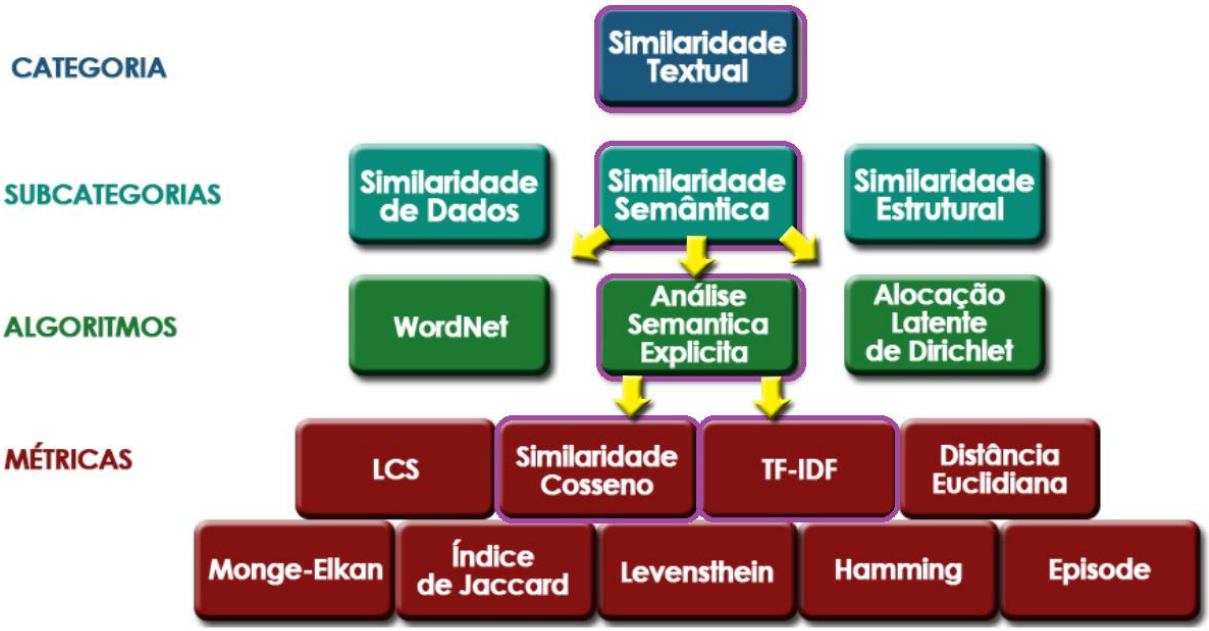

Figura 4. Principais Algoritmos e Métricas de Similaridade Semântica
Fonte: Autoria própria.

Na Figura 4 é possível visualizar as principais métricas e algoritmos apresentados neste capítulo. Os algoritmos apresentados são referentes à subcategoria Similaridade Semântica. Já as métricas podem ser utilizadas pelas demais subcategorias de Similaridade Textual.

Para alcançar o objetivo proposto neste trabalho, fez-se necessário utilizar um algoritmo que conseguisse calcular similaridade semântica entre currículos. Dentre os principais algoritmos de similaridade semântica, foi escolhido Análise Semântica Explicita (ESA) de Gabrilovich e Markovitch (2007), por este ter uma maior correlação com a interpretação de texto humana do que os demais algoritmos.

4 METODOLOGIA

Neste capítulo apresenta-se a metodologia que foi empregada a fim de alcançar os objetivos do trabalho.

4.1 ESTRUTURAÇÃO DA PESQUISA

Segundo Miguel (2007), pode-se considerar a pesquisa como um planejamento estratégico, em que uma abordagem metodológica compreende diversos níveis de abrangência e profundidade. Algumas decisões são estratégicas, tais como a escolha da abordagem e do método, e outras são mais relacionadas à ordem tática e operacional do trabalho, como por exemplo: procedimentos de conclusão da pesquisa (REINEHR, 2008).

Sendo assim, foi definido uma estrutura de pesquisa a ser seguida para cumprir com os objetivos iniciais, como pode ser visto na Figura 5. Basicamente esta estrutura é composta por quatro etapas: Planejamento Inicial, Fase Exploratória, Desenvolvimento, Avaliação e Conclusão, sendo que estas podem possuir uma ou mais sub-etapas que são ligadas entre si através de setas que indicam a direção do fluxo e a ação decorrente.

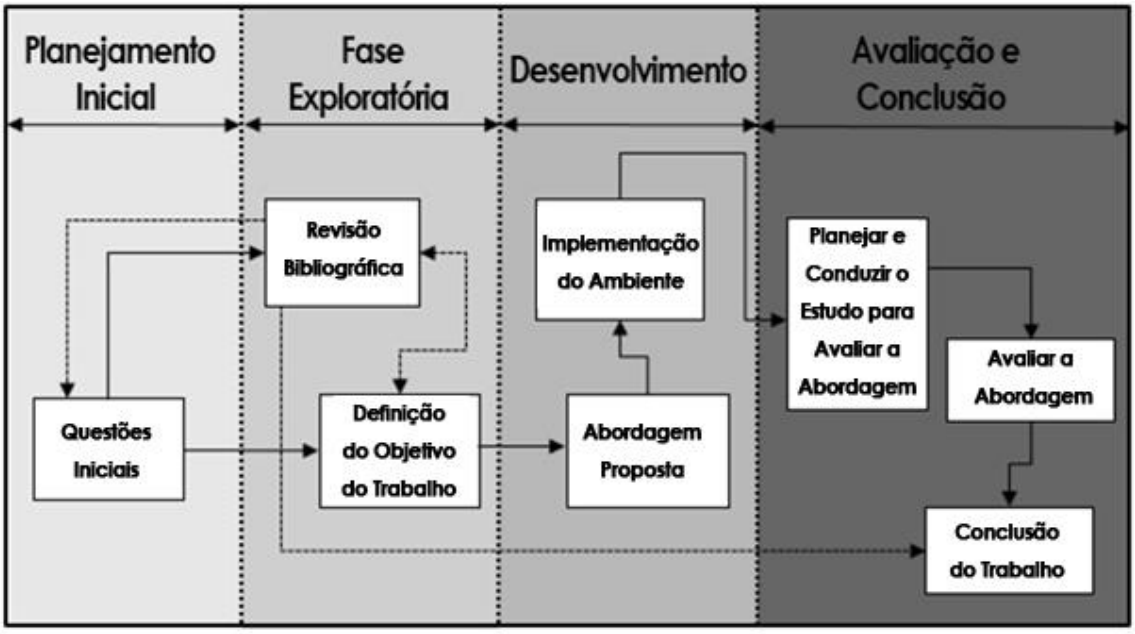

Figura 5. Estrutura da Pesquisa
Fonte: Autoria própria.

4.1.1 Planejamento inicial

A concepção do trabalho partiu de um problema real: a dificuldade para encontrar acadêmicos com áreas de atuação similares. Para resolver este problema, foi definida uma questão inicial: “Como comparar acadêmicos?”. Optou-se por utilizar o currículo dos acadêmicos para realizar essa comparação, pois a forma mais comum de identificar a área de interesse e atuação de uma pessoa é através de seu currículo.

Após definir o campo de pesquisa para o trabalho, algumas questões iniciais foram abordadas para assim dar início ao estudo, tais como: “Como calcular quão semelhante são os significados de duas palavras diferentes?”, “Qual o melhor algoritmo para calcular similaridade semântica em currículos?” e “Quais os prós e contras em utilizar análise semântica explicita para calcular similaridade semântica?” .

Entretanto, as questões iniciais levantadas servem apenas para traçar uma trajetória a ser seguida e contribuir para um direcionamento que possa enfatizar a pesquisa proposta. Após definir essas questões estabelecidas inicialmente, pode-se delimitar a área de estudo e o objetivo a ser tratado.

4.1.2 Fase Exploratória

Segundo Gil (2002), a fase exploratória tem como objetivo proporcionar maior familiaridade com o problema e torná-lo explícito ou a construir hipóteses. Desta forma, é necessário explorar o objetivo de estudo para conduzir o trabalho.

4.1.2.1 Revisão Bibliográfica

Segundo Mafra e Travassos (2006), o levantamento bibliográfico se faz necessário, sendo que a revisão de literatura é o meio pelo qual o pesquisador pode identificar o conhecimento científico existente em uma determinada área, de forma a planejar sua pesquisa, evitando a duplicação de esforços e a repetição de erros passados.

Para conseguir responder as questões elaboradas no planejamento inicial, foi necessário buscar artigos, os quais passaram por avaliações e interpretações. Aqueles que condiziam com os objetivos deste trabalho foram pesquisados em detalhes.

Os artigos encontrados foram selecionados devido à proximidade com o assunto da pesquisa e de acordo com um conjunto de palavras-chaves selecionadas previamente, que identificaram as áreas de pesquisa envolvidas neste trabalho, como por exemplo, similaridade textual, análise semântica explícita e comparação de currículos.

As dissertações e periódicos foram encontrados nos bancos de dados virtuais disponibilizados pela Coordenação de Aperfeiçoamento de Pessoal de Nível Superior (CAPES), IEEEXplore *Digital Library* (IEEE), ACM *Digital Library* e Google Acadêmico.

4.1.2.2 Definição do Objetivo do Trabalho

As questões iniciais delimitaram o escopo do trabalho, auxiliando a identificar os problemas a serem tratados. Além disso, a revisão bibliográfica demonstra uma visão geral do campo de estudo, que serviu como suporte para definir os objetivos deste trabalho, apresentados na Seção 1.1.

4.1.3 Desenvolvimento

Para atingir os objetivos propostos neste trabalho, o desenvolvimento foi dividido em duas etapas subsequentes e dependentes entre si, descritas na sequência.

4.1.3.1 Abordagem Proposta

O objetivo deste trabalho foi desenvolver uma aplicação, no qual fez-se necessário propor uma arquitetura para implementá-la. De acordo com vários pesquisadores como por exemplo, Gabrilovich e Markovitch (2007), Cimiano et al. (2009) e Stefanesco et al. (2014), atualmente a abordagem ESA é a que apresenta os melhores resultados dentre os algoritmos de similaridade semântica, ou seja, é a abordagem que tem uma maior correlação com a interpretação de texto feita por humanos, por isso ela foi escolhida para ser utilizada neste trabalho.

4.1.3.2 Implementação do Ambiente

Para atingir o objetivo principal deste trabalho, foi desenvolvida uma aplicação para comparação de acadêmicos através de seus currículos. Foram coletados 517 currículos de professores da UTFPR câmpus Curitiba, que foram comparados utilizando a abordagem ESA.

Como fonte de alimentação do algoritmo ESA, foi utilizada a versão em português do Wikipédia, que conta com 1.007.000 de artigos, bem menos do que a versão em inglês utilizada originalmente pelo algoritmo de Gabrilovich e Markovitch (2009), que possui aproximadamente 6.000.000.

Artigos com menos de 100 palavras foram removidos, assim como os artigos que continham apenas metadados. Após a filtragem dos artigos, sobraram aproximadamente 200.000 artigos.

Os artigos resultantes foram separados em vetores de palavras, onde após a remoção das *stopwords*, calculou-se o TF-IDF para cada uma das 5.437.498 palavras resultantes.

Com o TF-IDF calculado, foi possível buscar o vetor de conceitos dos 517 currículos e compará-los utilizando a Similaridade Cosseno. O resultado da comparação entre os currículos foi demonstrado através de tabelas e grafos.

O desenvolvimento da aplicação foi melhor detalhado no Capítulo 5.

4.1.4 Avaliação e Conclusão

A última fase da estrutura de pesquisa é a de avaliação e conclusão do trabalho. Esta fase foi dividida em três etapas. Sendo que as duas primeiras são referentes a avaliação da abordagem. Por fim, após avaliar a abordagem, a última etapa, de conclusão do trabalho, é executada.

4.1.4.1 Planejar e Conduzir o Estudo para Avaliar a Abordagem

O planejamento e a condução do experimento para avaliação da abordagem se deram inicialmente na escolha de quais dados do currículo utilizar.

Em um primeiro momento, foi utilizado somente o resumo do currículo para comparar os acadêmicos. Após uma avaliação feita de forma visual e pela experiência do autor, constatou-se que somente o texto não seria suficiente para obter-se resultados confiáveis. Optou-se por utilizar mais informações presentes no currículo, como por exemplo: participação em eventos e palestras, publicações, orientações, entre outras.

Para avaliar a nova abordagem selecionou-se ao acaso uma linha de pesquisa presente na UTFPR e todos os currículos dos acadêmicos atuantes nesta linha foram comparados.

Após isto, selecionou-se outra linha de pesquisa aleatoriamente e os currículos previamente comparados foram comparados novamente, agora com os acadêmicos da nova linha de pesquisa, esperando uma similaridade menor entre os acadêmicos de linhas de pesquisa diferentes.

4.1.4.2 Avaliar a Abordagem

Após o domínio ter sido definido, o experimento foi concebido e depois executado, tendo como base somente acadêmicos da UTFPR.

Além da comparação entre acadêmicos de linhas de pesquisas diferentes, também foi construído um *heatmap* de todos os acadêmicos coletados para poder avaliar os resultados obtidos pela abordagem escolhida.

4.1.4.3 Conclusões do Trabalho

Por fim, após o planejamento, execução e avaliação da abordagem, foi possível concluir o trabalho. Utilizando a análise dos dados da avaliação da abordagem, em conjunto com todo o referencial teórico obtido no decorrer da pesquisa, as conclusões finais do trabalho puderam ser expostas, apresentando elementos que indicam a efetividade da abordagem proposta.

5 DESENVOLVIMENTO

Este capítulo apresenta uma explicação sobre como se deu o desenvolvimento deste trabalho. Na primeira seção, discorre-se sobre como os currículos foram coletados e quais dados foram utilizados. Na sequência, é abordado o processo de desenvolvimento do algoritmo utilizado, mostrando principalmente como o interpretador semântico foi construído. Por fim, é demonstrado como a similaridade entre os currículos foi calculada. As telas e resultados são apresentados no Capítulo 6.

5.1 CURRÍCULOS LATTES

A Plataforma Lattes é uma plataforma virtual criada e mantida pelo 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.

Os dados de docentes e discentes, foram obtidos a partir de arquivos XMLs disponibilizados de forma gratuita pela plataforma Lattes.

Figura 6. Texto do Currículo Lattes
Fonte: Autoria própria.

A Figura 6 mostra um exemplo do texto contido no currículo, onde o próprio autor pode descrever sobre sua atuação acadêmica de maneira não estruturada. Entretanto, após alguns testes foi constatado que como o autor está livre para escrever qualquer coisa (ou nem escrever em alguns casos), utilizar somente o texto do currículo não seria suficiente. Fez-se necessário, a partir do XML, extrair mais informações julgadas como relevantes sobre o acadêmico.

Figura 7. Fragmento de um currículo no formato XML – Tree View
Fonte: Autoria própria.

A Figura 7 mostra parte de um currículo XML na visualização tipo árvore. Como é possível observar existem muitas *tags*, porém foram utilizadas apenas as seguintes informações para determinar a similaridade entre os acadêmicos:

- DADOS GERAIS: Currículo e área de atuação;
- PRODUÇÃO BIBLIOGRÁFICA: eventos, artigos, textos, jornais, revistas;
- PRODUÇÃO TÉCNICA: cursos ministrados;
- OUTRA PRODUÇÃO: orientações;
- DADOS COMPLEMENTARES: participações em cursos.

Após a coleta das informações relevantes, elas foram agrupadas juntamente com o texto do currículo, formando um texto maior e mais completo. Após isso, foram removidas as *stopwords* e os dados do acadêmico foram salvos no banco de dados.

	id int	nome character	instituicao character va	campus character	departamento character varyin	curriculo text	resumo text
1	8	Adalbert...	UTFPR	Curitiba	DACOC	professor associado - utfpr. possui grad...	análise ergonômica do trabalho em um frigorífico ...
2	376	Adao de...	UTFPR	Curitiba	DAINF	possui graduação em letras português ...	ultimamente tenho andado meio corcunda poem...
3	35	Adauto J...	UTFPR	Curitiba	DACOC	possui graduação em engenharia civil p...	básico de ensaios de materiais para construção c..
4	456	Admilso...	UTFPR	Curitiba	DAMEC	o prof. admilson teixeira franco é engen...	método de controle da perda espontânea de cal...
5	392	Adolfo G...	UTFPR	Curitiba	DAINF	é doutor em ciências da computação pe...	fazer revisar refazer an introduction to aspect-o..
6	94	Adriana ...	UTFPR	Curitiba	DADIN	possui graduação em desenho industri...	expotec 2000 o design de móveis e a redução da..
7	509	Adriano ...	UTFPR	Curitiba	DAMEC	possui graduação em engenharia mecâ...	mecânica técnica resistência dos materiais eleme...

Figura 8. Fragmento da Tabela de Acadêmicos
Fonte: Autoria própria.

A Figura 8 apresenta um fragmento da tabela “Acadêmicos”. Nesta tabela foram armazenados os dados dos acadêmicos, coletados da plataforma Lattes. Além do texto do currículo, os outros dados coletados (participação em eventos e palestras, publicações, orientações, etc.) foram agrupados em uma *string* chamada de resumo.

O escopo deste trabalho foi voltado apenas à comunidade acadêmica da Universidade Tecnológica Federal do Paraná (UTFPR). Em trabalhos futuros, poderá ser expandido para outras universidades, pois a Plataforma Lattes disponibiliza currículos de pesquisadores de várias partes do mundo.

5.2 INTERPRETADOR SEMÂNTICO

Conforme mencionado na Seção 3.2, o algoritmo ESA de Gabrilovich e Markovitch (2009) utiliza um interpretador semântico alimentado por artigos coletados do Wikipédia para poder calcular a similaridade semântica entre textos.

O interpretador semântico funciona como uma espécie de cérebro, pois é ele que vai diferenciar quais palavras são mais relevantes para que o computador possa entender o conteúdo semântico do currículo.

Na elaboração do interpretador algumas adaptações precisaram ser feitas para que o algoritmo funcionasse em currículos escritos em português.

No algoritmo original são apenas utilizados artigos em inglês. A versão inglesa do Wikipédia conta com mais de 5.734.000 artigos, muito mais do que a versão brasileira, que tem aproximadamente 1.007.000 artigos (Wikipédia, 2018). Mesmo com essa grande diferença, foi necessário utilizar os artigos do Wikipédia brasileiro, pois a maior parte dos currículos estão escritos em português.

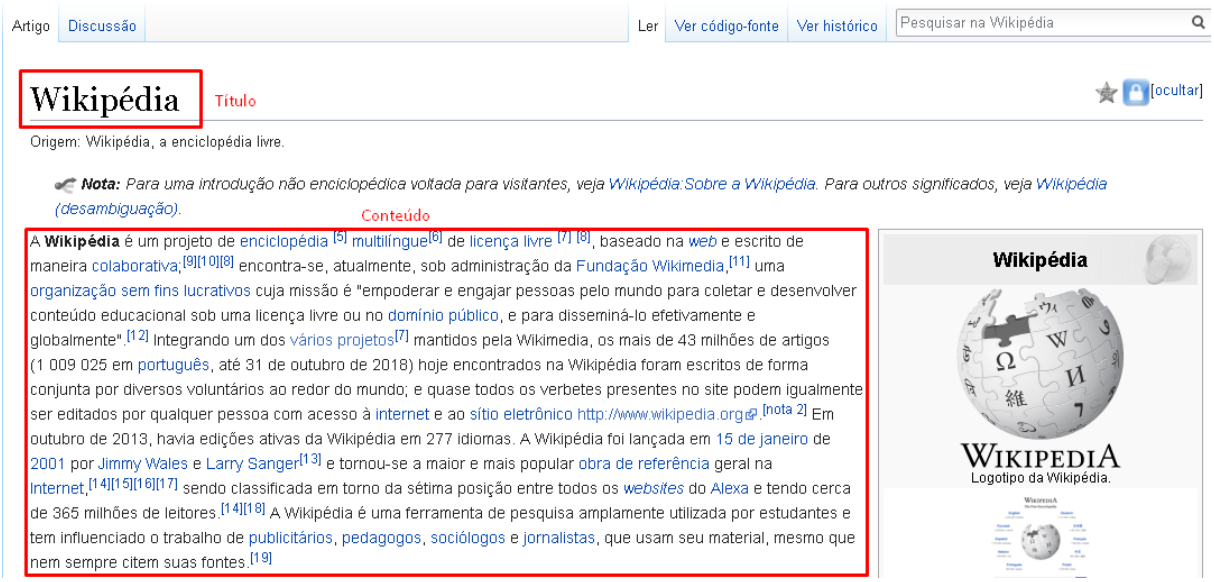

Figura 9. Exemplo de página do Wikipédia
Fonte: Autoria própria.

A Figura 9 mostra um exemplo de uma página do Wikipédia, onde é possível verificar que um artigo é composto de um título e um texto que explica sobre o que se trata o artigo.

	id integer	titulo character varying (255)	texto text
1	325133	Saint-Grégoire	Saint-Grégoire é uma comuna francesa na região administrativa da Bretanha, no departamento Ille...
2	325158	Eno	Eno (medicamento) Éno By Eno Teodoro Wanke Brian Eno Ricky Enø Jørgensen Categoria:Desambi...
3	325174	Marysville	Marysville (Califórnia) Marysville (Iowa) Marysville (Kansas) Marysville (Michigan) Marysville (Ohio) M...
4	325184	AOS	AOS pode ser uma sigla de: Austin Osman Spare Castlevania: Aria of Sorrow Categoria:Desambigu...
5	325200	Slater	Pessoas Christian Slater Localidades Slater-Marietta Slater (Iowa) Slater (Missouri) Slater (Wyomin...
6	325252	Marlow	Pessoas Charles Marlow Ellen Marlow Localidades Marlow (Mecklemburgo-Pomerânia Ocidental) Ma...
7	325285	B9	Vitamina B9 B9 (tamanho de papel) Iceberg B-9 Akaflieg Berlin B-9 Categoria:Desambiguação

Figura 10. Fragmento da Tabela de Artigos do Wikipédia
Fonte: Autoria própria.

Para montar o interpretador semântico utilizado pelo algoritmo, foi criado um *script* em Python que coleta o título e o conteúdo de todos os artigos em português do Wikipédia e os armazena no banco de dados. A Figura 10 apresenta um fragmento da tabela “Artigos”, onde estão armazenados todos os artigos coletados do Wikipédia.

Após a coleta dos artigos, o conteúdo de cada artigo foi tokenizado, ou seja, o texto foi transformado em uma lista de palavras, onde removeram-se as *stopwords*. Conforme citado na Seção 2.3, *stopwords* são palavras que aparecem com grande frequência, mas que possuem pouca relevância para o entendimento do texto.

Como os artigos baixados do Wikipédia estavam em português, foi necessário buscar uma lista de *stopwords* voltada para a linguagem portuguesa. Na *internet* é possível encontrar várias listas de *stopwords* em português, como por exemplo a apresentada por Alopes (2018).

Com o conteúdo de cada artigo transformado em listas de palavras, foi então calculado o TF-IDF (coeficiente de relevância de cada palavra). Para calcular o TF-IDF foi utilizada a equação (1) apresentada na Seção 2.2.1 onde primeiro é calculado o TF (a frequência com que a palavra aparece no artigo) e depois multiplicado pelo IDF (frequência com que a palavra aparece em todos os artigos).

Os resultados foram salvos no banco de dados, onde o título foi chamado de “conceito” e o conteúdo do artigo de “texto”.

	artigoid integer	artigo character varying (255)	palavra character varyir	tf numeric	idf numeric	tfidf numeric
1	652624	Banco de dados relacional	dados	0.2	5.52227830091981	1.10445566018396
2	652624	Banco de dados relacional	banco	0.133333333333333	6.57673514526505	0.876898019368674
3	464577	Administrador de banco de dados	dados	0.130434782608696	5.52227830091981	0.720297169685193
4	282241	Banco de dados	SGBD	0.0666666666666667	10.6445401522525	0.709636010150169
5	652624	Banco de dados relacional	modela	0.0666666666666667	10.511008759628	0.700733917308535
6	652624	Banco de dados relacional	percebidos	0.0666666666666667	10.0255009438463	0.668366729589755
7	652624	Banco de dados relacional	relacional	0.0666666666666667	9.81786157906807	0.654524105271205

Figura 11. Fragmento da Tabela de TF-IDFs
Fonte: Autoria própria.

A Figura 11 apresenta um fragmento da tabela “Palavras”. Esta tabela é equivalente ao Interpretador Semântico apresentado no algoritmo ESA. Nesta tabela foram salvos os valores TF-IDF de todas as palavras contidas nos artigos coletados.

5.3 CÁLCULO DA SIMILARIDADE

O algoritmo ESA foi implementado em Java, seguindo o artigo de Gabrilovich e Markovitch (2009), como descrito na Seção 3.2. Porém, conforme mencionado anteriormente, foi necessário adaptar o interpretador semântico e a lista de *stopwords* para a língua portuguesa.

O algoritmo recebe como dados de entrada duas *strings* (no caso deste trabalho, dois currículos), e retorna um coeficiente de similaridade como resultado. Este processo é explicado mais detalhadamente a seguir.

Com a base de dados pronta é possível aplicar o algoritmo ESA nos currículos coletados. Cada comparação entre dois currículos resulta em um coeficiente de similaridade que é armazenado no banco de dados, juntamente com os dados dos proprietários dos currículos comparados.

Para calcular o coeficiente de similaridade, primeiro as duas *strings* passam pelo mesmo processo de tokenização e remoção de *stopwords* feitas para a base de dados (interpretador semântico).

Após isto, para cada *token* resultante da lista de palavras, busca-se na base de dados (interpretador semântico) os dez artigos que possuem maior TF-IDF com o *token* informado, resultando em um vetor de conceitos (artigos). O mesmo conceito pode aparecer mais de uma vez no vetor, por isso é calculada também a frequência de cada conceito. Este processo é feito para os dois currículos sendo comparados, resultando em dois vetores de conceitos.

Para calcular a similaridade entre estes dois vetores, é utilizada a Similaridade Cosseno, apresentada na Seção 2.2.2. Na Similaridade Cosseno é necessário que os dois vetores sejam iguais. Então os conceitos presentes nos 2 vetores são agrupados em 2 novos vetores.

Exemplo:

Vetor 1: [(Java,7), (Banco de Dados,2), (Elétrica,1)].

Vetor 2: [(Testes,8), (Redes,2)].

Novos vetores:

Vetor 3 (1): [(Java,7), (Banco de Dados, 2), (Elétrica, 1), (Teste,0), (Redes,0)].

Vetor 4 (2): [(Java,0), (Banco de Dados, 0), (Elétrica, 0), (Teste,8), (Redes,2)].

Com os dois vetores gerados e ordenados igualmente, é aplicada então a similaridade cosseno baseada nas frequências dos conceitos presentes no vetor, que resultará em um coeficiente de similaridade. Conforme mencionado no Capítulo 2, este coeficiente está entre 0 ou 1, sendo que 1 significa que os textos são idênticos.

	pessoa1 [PK] character varying (255)	**pessoa2** [PK] character varying (255)	**coeficiente** double precision
1	Adalberto Matoski	Adao de Araujo	0.0947244545135556
2	Adalberto Matoski	Adauto José Miranda de Lima	0.716247586559074
3	Adalberto Matoski	Admilson Teixeira Franco	0.657065132076591
4	Adalberto Matoski	Adolfo Gustavo Serra Seca N...	0.390829599267701
5	Adalberto Matoski	Adriana da Costa Ferreira	0.524425389574246
6	Adalberto Matoski	Adriano Araujo de Lima	0.352601790009968
7	Adalberto Matoski	Adriano Perpétuo de Lara	0.371464238033191

Figura 12. Fragmento da Tabela de Similaridades
Fonte: Autoria própria.

A Figura 12 apresenta um fragmento da tabela “Similaridades”, onde foram armazenados os resultados das comparações entre os acadêmicos. Estes resultados são utilizados para montar o grafo de similaridade, que é apresentado no próximo capítulo.

6 TELAS E RESULTADOS

Neste capítulo estão presentes todos os fluxos e telas do aplicativo. Na Seção 6.1 é abordada a primeira tela do sistema. Na Seção 6.2 são detalhados os filtros apresentados na tela inicial. Na Seção 6.3 é exposto o principal fluxo do aplicativo: Calcular Similaridade. Na Seção 6.4 é exposto o fluxo alternativo de apresentação dos resultados: Gerar Grafo. Por fim, na Seção 6.5, é demonstrada a análise de alguns resultados obtidos.

6.1 TELA INICIAL

Na tela inicial, apresentada pela Figura 13, é possível ver as duas funcionalidades principais: Calcular Similaridade e Abrir Grafo. Além disso, a tela inicial possui alguns filtros, que auxiliam a realizar uma busca mais específica.

Figura 13. Tela Inicial
Fonte: Autoria própria.

6.2 FILTROS

Nesta aplicação é possível filtrar os currículos por universidade, campus, departamento, entre outros, conforme demonstrado na Figura 14:

Figura 14. Tela Inicial – Filtros
Fonte: Autoria própria.

Na aba “Acadêmico” é possível filtrar os resultados de duas maneiras: mostrando os resultados para todos os acadêmicos ou selecionando uma pessoa específica para ser o centro das comparações entre os acadêmicos.

Na aba “Departamentos” é possível restringir as comparações de currículos à um ou mais departamentos selecionados pelo usuário.

Na aba “Configurações” é possível estabelecer a quantidade máxima de resultados que serão apresentados e definir a similaridade mínima desejada, um número entre 0.0 e 1.0, sendo que 1.0 é a maior similaridade possível. Os acadêmicos que tiverem similaridade abaixo da mínima estabelecida não serão apresentados nos resultados.

6.3 CALCULAR SIMILARIDADE

Figura 15. Fluxo Principal: Calcular Similaridade
Fonte: Autoria própria.

A Figura 15 demonstra o fluxo principal da aplicação: Calcular Similaridade. Neste fluxo, o resultado é coletado do banco de dados, filtrado com os parâmetros selecionados pelo usuário e demonstrado através de uma tabela contendo o nome dos acadêmicos comparados e o coeficiente de similaridade entre eles.

6.4 GERAR GRAFO

Também é possível visualizar o resultado através de um grafo desenvolvido em HTML/CSS e utilizando a biblioteca D3 do *JavaScript,* onde os coeficientes são os pesos das arestas e os nós são os nomes dos acadêmicos que foram comparados.

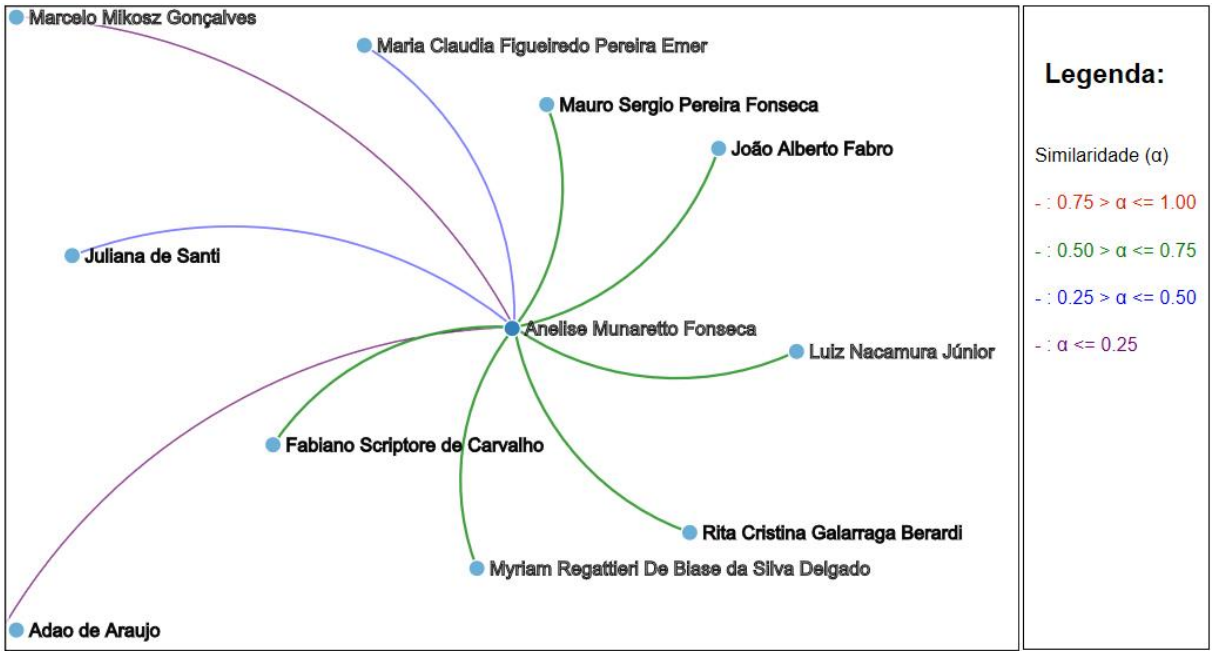

Figura 16. Grafo de Similaridade
Fonte: Autoria própria.

A Figura 16 demonstra um exemplo do grafo de similaridade. No centro está o acadêmico para o qual foram aplicados os filtros (Anelise Munaretto), selecionando o departamento DAINF, número máximo de resultados 10 e similaridade mínima 0.0. Como é possível observar, quanto mais distante, menor é a similaridade entre os acadêmicos. As cores das arestas também auxiliam na visualização da similaridade.

Além das comparações individuais (partindo apenas de um pesquisador específico) é possível também comparar todos os pesquisadores de um ou vários departamentos selecionando a opção “Mostrar Todos Acadêmicos”.

6.5 RESULTADOS

Para demonstrar os resultados foi feito um estudo de caso com pesquisadores de duas linhas de pesquisa escolhidas ao acaso. Além disso, foi feito também um *heatmap* com as similaridades de aproximadamente 500 professores da UTFPR - Campus Curitiba encontrados na plataforma Lattes.

6.5.1 Estudo de Caso

Como forma de testar e validar os resultados, foram escolhidas ao acaso duas linhas de pesquisa da UTFPR: Engenharia de Software e Visão Computacional e Reconhecimento de Padrões em Imagens.

Engenharia de Software	Visão Computacional
Adolfo Gustavo Serra Seca Neto	Bogdan Tomoyuki Nassu
Jean Marcelo Simão	Leyza Elmeri Baldo Dorini
Laudelino Cordeiro Bastos	Ricardo Dutra da Silva
Maria Cláudia Figueiredo Pereira Emer	Rodrigo Minetto
Paulo Cézar Stadzisz	Tânia Mezzadri Centeno

Tabela 3. Linhas de Pesquisa
Fonte: Autoria própria

A Tabela 3 mostra o nome dos acadêmicos de cada uma das linhas de pesquisa selecionadas. Os currículos desses pesquisadores foram comparados com o intuito de verificar se dentro de uma mesma linha de pesquisa as pessoas possuiriam um coeficiente de similaridade alto.

	pessoa1 character varying (255)	pessoa2 character varying (255)	coeficiente double precision
1	Jean Marcelo Simão	Paulo Cézar Stadzisz	0.764169340152122
2	Jean Marcelo Simão	Laudelino Cordeiro Bastos	0.653854864027384
3	Maria Claudia Figueiredo P...	Adolfo Gustavo Serra Seca...	0.621443519651316
4	Paulo Cézar Stadzisz	Maria Claudia Figueiredo P...	0.587852548547014
5	Jean Marcelo Simão	Adolfo Gustavo Serra Seca...	0.587167190709781
6	Adolfo Gustavo Serra Seca ...	Laudelino Cordeiro Bastos	0.583029031911177
7	Jean Marcelo Simão	Maria Claudia Figueiredo P...	0.574549109315809
8	Paulo Cézar Stadzisz	Adolfo Gustavo Serra Seca...	0.571839435386904
9	Paulo Cézar Stadzisz	Laudelino Cordeiro Bastos	0.547175344280837
10	Maria Claudia Figueiredo P...	Laudelino Cordeiro Bastos	0.529114726066164

Figura 17. Engenharia de Software – Comparação
Fonte: Autoria própria.

A Figura 17 mostra o resultado das comparações entre os pesquisadores de Engenharia de Software. Utilizando como referência a média geral de similaridades deste trabalho, que foi 0.27, a similaridade entre estes acadêmicos foi alta, chegando a uma máxima de 0.71.

O mesmo procedimento foi realizado para os pesquisadores de Visão Computacional e Reconhecimento de Padrões em Imagens.

	pessoa1 character varying (255)	**pessoa2** character varying (255)	**coeficiente** double precision
1	Leyza Elmeri Baldo Dorini	Tania Mezzadri Centeno	0.653930456349405
2	Bogdan Tomoyuki Nassu	Leyza Elmeri Baldo Dorini	0.641545553491404
3	Rodrigo Minetto	Ricardo Dutra da Silva	0.594232460755841
4	Bogdan Tomoyuki Nassu	Tania Mezzadri Centeno	0.530114102339978
5	Tania Mezzadri Centeno	Rodrigo Minetto	0.496660169985357
6	Bogdan Tomoyuki Nassu	Rodrigo Minetto	0.480503893393354
7	Leyza Elmeri Baldo Dorini	Rodrigo Minetto	0.470208798820323
8	Tania Mezzadri Centeno	Ricardo Dutra da Silva	0.432856901105097
9	Leyza Elmeri Baldo Dorini	Ricardo Dutra da Silva	0.402747806991851
10	Bogdan Tomoyuki Nassu	Ricardo Dutra da Silva	0.365491138112575

Figura 18. Visão Computacional – Comparação
Fonte: Autoria própria.

A Figura 18 mostra o resultado das comparações entre os pesquisadores de Visão Computacional e Reconhecimento de Padrões em Imagens. Como é possível notar, a similaridade entre eles foi menor do que entre os pesquisadores de Engenharia de Software, porém, ainda assim foi superior à média.

Para um último teste, foram comparados os acadêmicos das duas linhas de pesquisa.

	departamento1 text	departamento2 text	pessoa1 character varying (255)	pessoa2 character varying (255)	coeficiente double precision
1	SOFTWARE	SOFTWARE	Jean Marcelo Simão	Paulo Cézar Stadzisz	0.764169340152122
2	VISAO	VISAO	Leyza Elmeri Baldo Dorini	Tania Mezzadri Centeno	0.653930456349405
3	SOFTWARE	SOFTWARE	Jean Marcelo Simão	Laudelino Cordeiro Bas...	0.653854864027384
4	SOFTWARE	VISAO	Jean Marcelo Simão	Tania Mezzadri Centeno	0.65147418328044
5	VISAO	VISAO	Bogdan Tomoyuki Nassu	Leyza Elmeri Baldo Dorini	0.641545553491404
6	SOFTWARE	VISAO	Tania Mezzadri Centeno	Laudelino Cordeiro Bas...	0.632182910075786
7	SOFTWARE	VISAO	Jean Marcelo Simão	Leyza Elmeri Baldo Dorini	0.61165569180867
8	SOFTWARE	VISAO	Leyza Elmeri Baldo Dorini	Laudelino Cordeiro Bas...	0.606310916283116
9	SOFTWARE	VISAO	Paulo Cézar Stadzisz	Tania Mezzadri Centeno	0.600412443978301
10	VISAO	VISAO	Rodrigo Minetto	Ricardo Dutra da Silva	0.594232460755841
11	SOFTWARE	SOFTWARE	Paulo Cézar Stadzisz	Maria Claudia Figueired...	0.587852548547014
12	SOFTWARE	VISAO	Tania Mezzadri Centeno	Adolfo Gustavo Serra S...	0.587343900368078
13	SOFTWARE	SOFTWARE	Jean Marcelo Simão	Adolfo Gustavo Serra S...	0.587167190709781
14	SOFTWARE	SOFTWARE	Adolfo Gustavo Serra S...	Laudelino Cordeiro Bas...	0.583029031911177
15	SOFTWARE	SOFTWARE	Jean Marcelo Simão	Maria Claudia Figueired...	0.574549109315809

Figura 19. Engenharia de *Software* x Visão Computacional
Fonte: Autoria própria.

A Figura 19 mostra parte do resultado da comparação entre os acadêmicos das duas linhas de pesquisa selecionadas. Como esperado, a maior similaridade foi entre pesquisadores de uma mesma linha de pesquisa. Entretanto, é possível notar que a 4ª maior similaridade ocorreu entre acadêmicos de linhas diferentes.

Buscando por mais informações do motivo disso ter acontecido, foi constatado que os currículos dos dois pesquisadores possuíam várias coisas em comum, como por exemplo: pós-graduação em engenharia elétrica e informática industrial no CPGEI, computação aplicada no PPGCA, trabalhos que utilizavam inferência fuzzy, redes neurais, entre outros. Isto acabou gerando uma similaridade alta, mesmo que atualmente eles estejam fazendo pesquisas em áreas diferentes.

6.5.2 *Heatmap*

A seguir é apresentado um *heatmap*, uma representação gráfica de dados em que os valores individuais contidos em uma matriz são representados como cores (Wilkinson e Friendly, 2009).

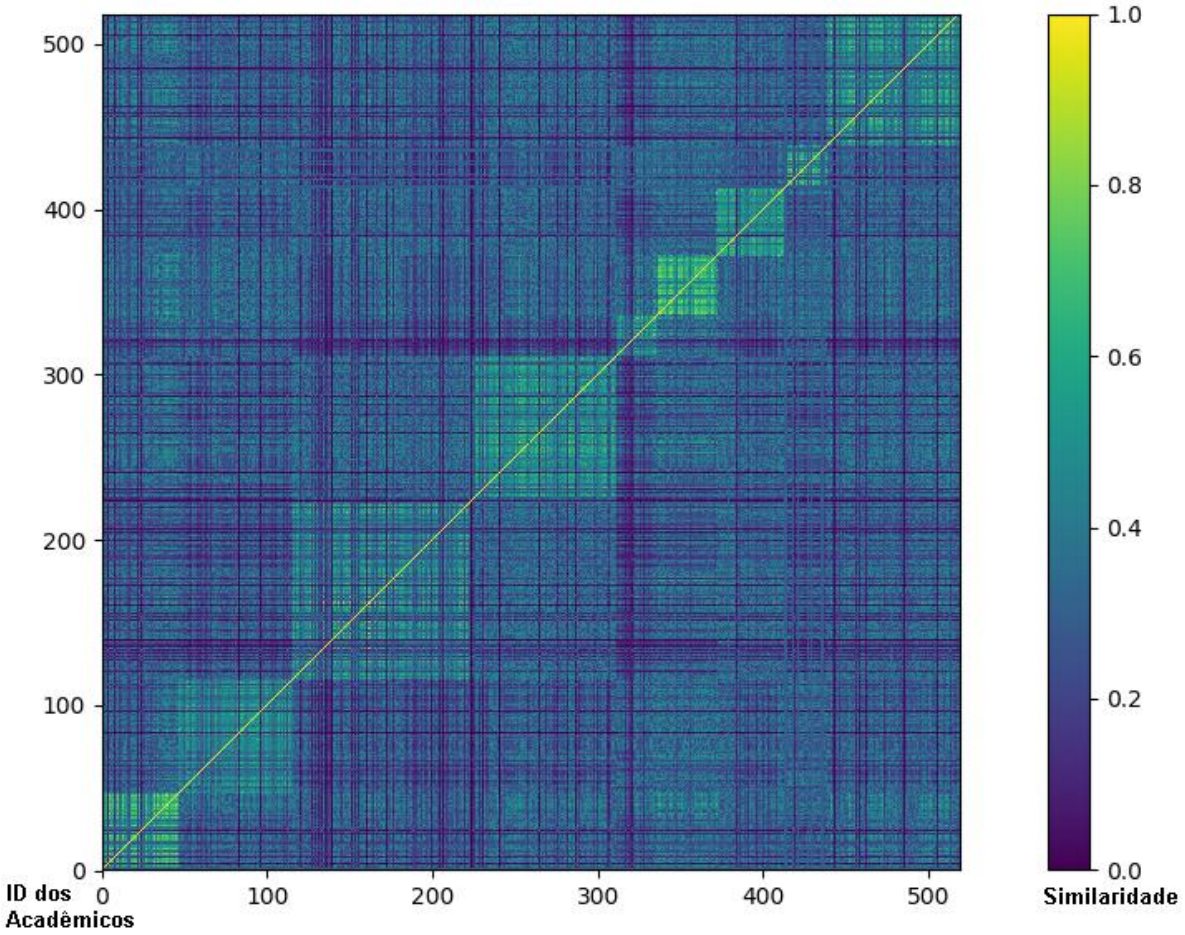

Figura 20. *Heatmap* dos Acadêmicos
Fonte: Autoria própria

Na Figura 20 é possível visualizar o *heatmap* feito a partir dos resultados das comparações de todos os currículos coletados. Nele pode-se verificar alguns pontos mais claros, onde a similaridade entre os acadêmicos é maior.

Para realizar este *heatmap*, aproximadamente 500 acadêmicos foram ordenados por departamento e cada um deles recebeu um ID. O eixo X e o eixo Y representam esses IDs dos acadêmicos, de forma espelhada. A interseção dos eixos X,Y representa o valor da similaridade entre os acadêmicos sendo comparados.

Por exemplo, comparando X[1] com Y[1] o resultado vai ser uma similaridade de 1.0, pois os dois pontos representam o mesmo acadêmico.

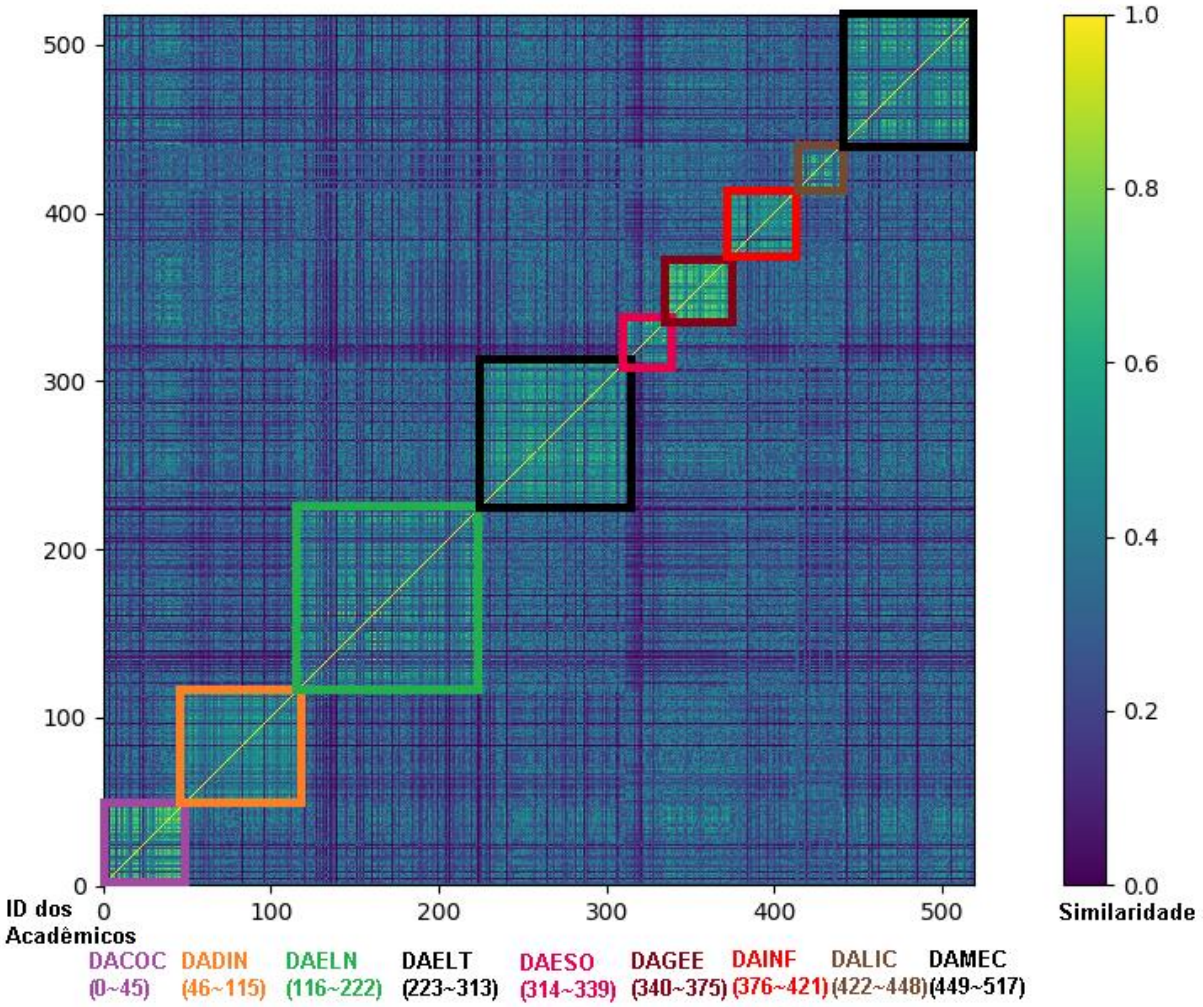

Figura 21. *Heatmap* dos Acadêmicos por Departamentos
Fonte: Autoria própria.

Na Figura 21 foi representado o limite de cada departamento, o que facilitou observar que os acadêmicos possuem mais similaridade com outros acadêmicos de um mesmo departamento. Isso mostra um resultado positivo do algoritmo, pois já era esperado que os departamentos tivessem acadêmicos atuando em áreas similares, assim como também é esperado que alguns acadêmicos possam ter similaridade alta com pesquisadores de outros departamentos.

7 CONCLUSÃO

Este trabalho teve como objetivo implementar uma ferramenta que, ao receber uma série de currículos Lattes, em formato XML, extraia os dados e compare os currículos para indicação de similaridade entre os acadêmicos.

No primeiro momento do trabalho, foi utilizado somente o texto não estruturado dos currículos, contudo os resultados não foram satisfatórios devido à falta de padronização, no resumo cada autor pode se descrever da maneira que quiser, ou não se descrever em alguns casos. Com isso, além do texto não estruturado foram também utilizadas outras informações contidas no XML do currículo Lattes, como por exemplo: área de atuação, cursos ministrados, orientações, participação em cursos, e publicações (eventos, artigos, textos, jornais, revistas). Estas informações foram transformadas em uma *string* de palavras-chave e ajudaram a complementar o texto do currículo, tornando os resultados mais satisfatórios.

Após elaboração do *heatmap* foi possível notar que em geral as similaridades eram maiores para acadêmicos de um mesmo departamento, o que faz sentido, visto que a separação por departamentos em uma universidade serve justamente para agrupar acadêmicos de uma área de pesquisa similar. Contudo, alguns pesquisadores de áreas diferentes também tiveram similaridade alta. Nestes casos, foi necessária uma análise mais detalhada dos currículos para constatar que eles realmente possuíam atividades em comum.

Durante o desenvolvimento da aplicação foram encontradas algumas dificuldades, como por exemplo a exibição do grafo quando o conjunto de acadêmicos é muito grande, neste caso o grafo fica muito poluído pela grande quantidade de nós e arestas. Para contornar este problema, foi inserida uma opção de visualização dos dados em forma de tabela.

7.1 TRABALHOS FUTUROS

Durante a elaboração deste trabalho foram observadas várias oportunidades para trabalhos futuros. A aplicação proposta possui uma ampla capacidade de expansão. Alguns pontos observados:

- **Interpretador semântico -** Um ponto observado é que o interpretador semântico poderia ser feito de várias outras formas, por exemplo utilizando redes neurais de Kohonem para tentar classificar as palavras de uma forma mais assertiva.
- **Base de alimentação –** No algoritmo ESA original de Gabrilovich e Markovitch (2009) é utilizado como fonte de dados o Wikipédia, que é uma base de conhecimentos gerais. Porém, além do Wikipédia, poderia ser também utilizada uma outra fonte de dados mais específica (voltada para currículos, por exemplo).
- **Linguagem –** Neste trabalho foi utilizado somente o Wikipédia em português, porém, como os currículos podem ter dados em outras línguas, a base de conhecimento do interpretador semântico poderia ser aumentada com os dados do Wikipédia de outros idiomas.
- **Acadêmicos –** Este trabalho foi feito apenas com dados de acadêmicos da UTFPR, entretanto ele poderia ser expandido, já que a Plataforma Lattes contém dados de acadêmicos de várias partes do mundo.
- **Coautoria –** Além da similaridade semântica aplicada nos currículos, para indicar a similaridade entre os acadêmicos poderia ser utilizado também a similaridade por coautoria desenvolvida no trabalho de Colonetti (2016), onde para cada publicação feita em conjunto com outro acadêmico, a similaridade entre eles aumenta.
- **Aplicativo Móvel –** Neste primeiro momento foi apenas desenvolvida uma aplicação *desktop*, porém este trabalho poderia ser muito bem aproveitado por uma aplicação móvel. O banco de dados e o processamento do interpretador semântico permaneceriam iguais, sendo feitos no servidor. Somente as telas seriam modificadas para que pudessem rodar de forma responsiva em *smartphones* e navegadores *web*.

REFERÊNCIAS

ALI, Abdulla. Textual similarity. Technical University of Denmark, Informatics and Mathematical Modelling. Kongens Lyngby, 2011. IMM-BSc-2011-19

ALOPES. Github: Alopes - Portuguese stop words. Disponível em: <https://gist.github.com/alopes/5358189>. Acesso em: 12 de nov. de 2018.

ALVES, Alexandre D.; YANASSE, Horacio H.; SOMA, Nei Y. LattesMiner: a multilingual DSL for information extraction from lattes platform. In: Proceedings of the compilation of the co-located workshops on DSM'11, TMC'11, AGERE! 2011, AOOPES'11, NEAT'11, & VMIL'11. ACM, 2011. p. 85-92.

AOUICHA, Mohamed Ben; TAIEB, Mohamed Ali Hadj; HAMADOU, Abdelmajid Ben. SISR: System for integrating semantic relatedness and similarity measures. Soft Computing, 2016. p. 1-25.

BAEZA-YATES, Ricardo; RIBEIRO-NETO, Berthier. Modern Information Retrieval, v.1, 1999, p. 1-513.

BALLATORE, Andrea; BERTOLOTTO, Michela; WILSON, David C. An evaluative baseline for geo-semantic relatedness and similarity. GeoInformatica, v. 18, n. 4, 2014. p. 747-767

BLEI, David M.; NG, Andrew Y.; JORDAN, Michael I. Lafferty, John, ed. Latent Dirichlet Allocation. Journal of Machine Learning Research, 2003. 3 (4–5): pp. 993–1022. doi:10.1162/jmlr.2003.3.4-5.993

BLEI, David. Probabilistic topic models. Communications of the ACM, 2012, pp. 77-84. doi:10.1145/2133806.2133826

BUDANITSKY, Alexander; HIRST, Graeme. Semantic distance in WordNet: An experimental, application-oriented evaluation of five measures. In: Workshop on WordNet and other lexical resources. 2001. p. 2-2.

CIMIANO, P. et al. Explicit vs. la tent concept models for cross-language information re-trieval. In: Proc, of IJCAI, 2009

COLONETTI, Gabriela Bussolo. Agrupando pesquisadores por coautoria de publicações segundo currículo Lattes. Universidade Federal de Santa Catarina (UFSC). Florianópolis, 2016.

CORLEY, Courtney; MIHALCEA, Rada. Measuring the semantic similarity of texts. In: Proceedings of the ACL workshop on empirical modeling of semantic equivalence and entailment. Association for Computational Linguistics, 2005. p. 13-18.

CORMEN, Thomas H. et al. Algoritmos: teoria e prática. Editora Campus, v. 2, 2002. p. 296.

DANGETI, Pratap. Statistics for Machine Learning. Packt Publishing Ltd, 2017.

DAS, G., FLEISCHER, R., GASIENIEC, L., GUNOPULOS, D., & KÄRKKÄINEN, J. (1997, June). Episode matching. In Annual Symposium on Combinatorial Pattern Matching (pp. 12-27). Springer, Berlin, Heidelberg.

DEZA, Elena; DEZA, Michael Marie. Encyclopedia of Distances. v. 1, 2009, p. 94-604.

DIGIAMPIETRI, L. et al. Minerando e caracterizando dados de currıculos lattes. In: Brazilian Workshop on Social Network Analysis and Mining (BraSNAM). 2012.

FENG, Y. et al. "The state of the art in semantic relatedness: a framework for comparison". Knowledge Engineering Review, 2017. p. 1–30.

FIGUEIREDO, Daniel R. Introdução a redes complexas. Atualizações em Informática, p. 303-358, 2011.

GABRILOVICH, Evgeniy; MARKOVITCH, Shaul. Computing semantic relatedness using wikipedia-based explicit semantic analysis. In: IJcAI. 2007. p. 1606-1611.

GIL, Antonio Carlos. Como Elaborar Projetos de Pesquisa. Atlas, 2002, 4ª edição.

GONÇALVES, Rodrigo; MELLO, Ronaldo dos Santos. Similaridade entre documentos semi-estruturados. II ESCOLA REGIONAL DE BANCO DE DADOS, 2006.

GRIGOREV, Alexey; REESE, Richard M.; REESE, Jennifer L. Java: Data Science Made Easy. Published by Packt Publishing, 2017.

HAMMING, Richard W. Error detecting and error correcting codes. Bell System technical journal, v. 29, n. 2, p. 147-160, 1950.

HARISPE, Sébastien et al. Semantic similarity from natural language and ontology analysis. Synthesis Lectures on Human Language Technologies, v. 8, n. 1, 2015.p. 1-254.

HIRSCHBERG, Daniel S. Algorithms for the longest common subsequence problem. Journal of the ACM (JACM), v. 24, n. 4, p. 664-675, 1977.

JACCARD, Paul. The distribution of the flora in the alpine zone.1. New Phytologist, [s.l.], v. 11, n. 2, p.37-50, fev. 1912. Wiley. http://dx.doi.org/10.1111/j.1469-8137.1912.tb05611.x.

KOCH, N. Classification of model transformation techniques used in UMLbased Web engineering. Software, IET. v. 1, n. 3, p. 98-111, 2007.

LATTES. Plataforma Lattes. Disponível em: <http://lattes.cnpq.br>. Acesso em: 12 de nov. de 2018.

LEVENSHTEIN, Vladimir I. Binary codes capable of correcting deletions, insertions, and reversals. In: Soviet physics doklady. 1966. p. 707-710.

LESKOVEC, Jure; RAJARAMAN, Anand; ULLMAN, Jeffrey David. Mining of massive datasets. Cambridge university press, 2014.

LIN, Dekang. An Information-Theoretic Definition of Similarity. Em Proceedings of the Fifteenth International Conference on Machine Learning, 1998.

MAFRA, Sômulo Nogueira; TRAVASSOS, Guilherme Horta. Estudo de caso na engenharia de produção: estruturação e recomendações para sua condução. 2006. Relatório Técnico, RT-ES 687/06.

MENA-CHALCO, Jesús Pascual et al. Brazilian bibliometric coauthorship networks. Journal of the Association for Information Science and Technology, v. 65, n. 7, p. 1424-1445, 2014.

MENA-CHALCO, Jesús Pascual; JUNIOR, Cesar; MARCONDES, Roberto. ScriptLattes: an open-source knowledge extraction system from the Lattes platform. Journal of the Brazilian Computer Society, v. 15, n. 4, p. 31-39, 2009.

MIGUEL, Paulo Augusto Cauchick. Estudo de caso na engenharia de produção: estruturação e recomendações para sua condução. Em Produção, 2007, volume 17, páginas 216-229.

MONGE, Alvaro E.; ELKAN, Charles P. The Field-Matching Problem: Algorithm and Applications. Em: Proceedings of the second international Conference on Knowledge Discovery and Data Mining, 1996.

NAUMANN, Felix; HERSCHEL, Melanie. An introduction to duplicate detection. Synthesis Lectures on Data Management, v. 2, n. 1, p. 1-87, 2010.

NAVARRO, Gonzalo. A guided tour to approximate string matching. ACM Comput. Surv., 33(1):31–88, 2001.

OVADIA, Steven. ResearchGate and Academia.edu: Academic Social Networks in Behavioral & Social Sciences Librarian, 2014. Volume 33, terceira edição, páginas 165-169.

RAMOS, Juan et al. Using tf-idf to determine word relevance in document queries. In: Proceedings of the first instructional conference on machine learning. 2003. p. 133-142.

REINEHR, S. Reúso Sistematizado de Software e Linhas de Produto de Software No Setor Financeiro: Estudos De Caso No Brasil. Tese (Doutorado) - Escola Politécnica, Universidade de São Paulo (USP), São Paulo, 2008, 310p.

SALTON, Gerard; MICHAEL, J. McGill. Introduction to modern information retrieval, 1983.

SHEPARD, Roger N. The analysis of proximities: Multidimensional scaling with an unknown distance function. Psychometrika, 1962. Volume 27, páginas 125-140.

ȘTEFĂNESCU, Dan; BANJADE, Rajendra; RUS, Vasile. Latent semantic analysis models on wikipedia and tasa. In: Language Resources Evaluation Conference (LREC), 2014.

TANIMOTO, T.T. An elementary mathematical theory of classification and prediction. IBM Internal Report. (1958)

THANAKI, Jalaj. Python Natural Language Processing. Packt Publishing Ltd, 2017.
WILKINSON, Leland; FRIENDLY, Michael. The history of the cluster heat map. The American Statistician, v. 63, n. 2, p. 179-184, 2009.

WIKIPÉDIA: A Enciclopédia Livre. Disponível em: <https://www.wikipedia.org/> Acesso em: 6 de nov. de 2018.

ZAUGG, Holt; WEST, Richard E.; TATEISHI. Isaku; RANDALL, Daniel L. Mendeley: Creating communities of scholarly inquiry through research collaboration. TechTrends, 2011, edição 55, páginas 32-36.