Universidade Federal Rural de Pernambuco
Departamento de Estatística e Informática
Programa de Pós-Graduação em Informática Aplicada

Anderson Pinheiro Cavalcanti

UMA MEDIDA DE SIMILARIDADE TEXTUAL PARA IDENTIFICAÇÃO DE PLÁGIO EM FÓRUNS EDUCACIONAIS

Dissertação de Mestrado

Recife
Janeiro de 2018

Universidade Federal Rural de Pernambuco
Departamento de Estatística e Informática
Pós-graduação em Informática Aplicada

Anderson Pinheiro Cavalcanti

UMA MEDIDA DE SIMILARIDADE TEXTUAL PARA IDENTIFICAÇÃO DE PLÁGIO EM FÓRUNS EDUCACIONAIS

Trabalho apresentado ao Programa de Pós-graduação em Informática Aplicada do Departamento de Estatística e Informática da Universidade Federal Rural de Pernambuco como requisito parcial para obtenção do grau de Mestre em Informática Aplicada.

Orientador: *Rafael Ferreira Leite de Mello*
Co-Orientador: *Péricles Barbosa Cunha de Miranda*

Recife
Janeiro de 2018

Dados Internacionais de Catalogação na Publicação (CIP)
Sistema Integrado de Bibliotecas da UFRPE
Biblioteca Central, Recife-PE, Brasil

C376m Cavalcanti, Anderson Pinheiro.
Uma medida de similaridade textual para identificação de plágio em fóruns educacionais / Anderson Pinheiro Cavalcanti. – 2018.
87 f.: il.

Orientador: Rafael Ferreira Leite de Mello.
Coorientador: Péricles Barbosa Cunha de Miranda.
Dissertação (Mestrado) – Universidade Federal Rural de Pernambuco, Programa de Pós-Graduação em Informática aplicada, Recife, BR-PE, 2018.
Inclui referências e apêndice(s).

1. Educação a distância 2. Fórum educacional 3. Plágio 4. Similaridade Semântica 5. Processamento de linguagem natural I. Mello, Rafael Ferreira Leite de, orient. II. Miranda, Péricles Barbosa Cunha de, coorient. III. Título

CDD 004

Dissertação de Mestrado apresentada por **Anderson Pinheiro Cavalcanti** ao programa de Pós-Graduação em Informática Aplicada do Departamento de Estatística e Informática da Universidade Federal Rural de Pernambuco, sob o título **Uma Medida de Similaridade Textual para Identificação de Plágio em Fóruns Educacionais**, orientada pelo **Prof. Rafael Ferreira Leite de Mello** e aprovada pela banca examinadora formada pelos professores:

Prof. Péricles Barbosa Cunha de Miranda
Departamento de Estatística e Informática/UFRPE

Prof. Rinaldo José de Lima
Departamento de Estatística e Informática/UFRPE

Prof. Frederico Luiz Gonçalves de Freitas
Centro de Informática/UFPE

Recife
Janeiro de 2018

Eu dedico esta dissertação aos meus familiares, amigos e professores que me deram o suporte necessário para chegar até aqui.

Agradecimentos

Agradeço a minha família que sempre me apoiou e motivou a seguir o caminho do estudo. Em especial a minha esposa que sempre está do meu lado me apoiando. A todos os colegas do mestrado que de uma forma ou de outra contribuíram com meu crescimento acadêmico. Agradeço imensamente ao meu orientador por toda ajuda, dedicação e paciência ao longo desses 2 anos de mestrado. Pessoas assim me motivam ainda mais para seguir a carreira acadêmica e poder ajudar outras pessoas. Agradeço também a FACEPE pelo apoio financeiro a este projeto de pesquisa.

Não tente se tornar uma pessoa de sucesso, mas sim uma pessoa de valor.

—ALBERT EINSTEIN

Resumo

Com o crescente uso da tecnologia como ferramenta de apoio educacional, o uso de Ambiente Virtual de Aprendizagem (AVA) tem aumentado nos últimos anos. Estes ambientes disponibilizam várias ferramentas para melhorar a interação entre professores e alunos, tais como fórum, blog, wiki, entre outras. Estas ferramentas possuem um grande potencial para gerar conteúdo, o que pode ser usado para auxiliar no processo de ensino-aprendizagem. Porém, devido a grande quantidade de interações entre os alunos e o professor, torna-se difícil para o professor avaliar e acompanhar todo o material que é disponibilizado pelos alunos.

Uma ferramenta que se destaca em relação à geração de conteúdo colaborativo é o fórum. Dentre as possíveis funcionalidades dos fóruns se destaca a questão da avaliação. Muitas disciplinas a distância utilizam a interação no fórum como forma de avaliação dos alunos. Contudo, devido a grande quantidade de dados postado na ferramenta, é difícil para o professor identificar problemas nas postagens, como por exemplo a detecção de plágio.

A base fundamental para a criação de sistemas automáticos de detecção de plágio é a criação de uma medida de similaridade que possa medir a relação existente entre dois textos. A similaridade entre textos é importante em diversas aplicações de Processamento de Linguagem Natural (PLN), como recuperação de informação, sumarização de texto, extração de informações e agrupamento de texto. Várias medidas de similaridade entre textos já foram criadas; entretanto, em geral, elas são dependentes de idioma. No caso do português, poucas medidas foram encontradas e a maioria utiliza apenas técnicas estatísticas, não levando em consideração aspectos semânticos dos textos. Além disso, existem trabalhos na literatura para identificação de plágio em atividades, artigos científicos ou trabalhos de conclusão de curso. No entanto, quando o contexto é fóruns educacionais a identificação de plágio se torna ainda mais difícil por causa principalmente do tamanho do texto e por não exigir uma linguagem formal.

Diante disso, este trabalho propõe uma medida que calcula a similaridade existente entre sentenças escritas em português levando em consideração a semântica dos textos. Esta medida foi avaliada na base da competição *Workshop* de Avaliação de Similaridade Semântica e Inferência Textual (ASSIN) 2016. A medida proposta alcançou resultados melhores que o primeiro colocado da competição atingindo 0,70 de correlação de Pearson e 0,47 de erro quadrático médio. Além desta avaliação, foi realizado um estudo de caso para avaliação de similaridade em postagens de fóruns educacionais em uma disciplina de Ciência da Computação. Os resultados foram avaliados pelos professores da disciplina que confirmaram a eficácia da ferramenta.

Palavras-chave: Educação a Distância, Fórum Educacional, Plágio, Similaridade Semântica, Mineração de Texto.

Abstract

With the increasing use of technology as an educational support tool, the use of *Virtual Learning Environment* (VLE) has increased in recent years. These environments provide several tools to improve the interaction between teachers and students, where some examples are: forum, blog, wiki, among others. These tools have great potential for generating content, which can be used to aid in the process of teaching learning. However, due to the great amount of interactions between the students and the teacher, it is difficult for the teacher to evaluate and follow up all the material that is made available by the students.

A tool that stands out in relation to the generation of collaborative content is the forum. Among the possible functionalities of the forums is the question of evaluation. Many distance disciplines use forum interaction as a form of student assessment. However, with the large amount of information posted on the tool, it often becomes impractical for the teacher to manually detect plagiarism in the responses.

The fundamental basis for the creation of automatic plagiarism detection systems is the creation of a measure of similarity that can measure the relationship between two texts. The similarity between texts is important in several *Natural Language Processing* (NLP) applications, such as retrieving information, summarizing text, extracting information, and grouping text. For example, in retrieval of information, the similarity measure is used to assign a classification score between a query and the obtained text. Various measures of similarity between texts can be found; however, in general, they are language dependent. In the case of Portuguese, few measures have been found and most use only statistical techniques, not taking into account semantic aspects of texts. In addition, there are papers in the literature to identify plagiarism in activities, scientific articles or course completion work. However, when context is educational forums the identification of plagiarism becomes even more difficult mainly because of the size of the text and by not requiring a formal language.

Therefore, this paper aims to propose a measure that calculates the similarity between sentences written in Portuguese taking into account the semantics of texts. This measure was evaluated on the basis of the ASSIN workshop 2016. The proposed measure achieved better results than the first place in the competition reaching 0.70 Pearson correlation and 0.47 mean squared error. In addition to this evaluation, a case study was carried out to evaluate similarity in postings of educational forums in a discipline of Computer Science. The results were evaluated by the teachers of the discipline who confirmed the effectiveness of the tool.

Keywords: Distance Education, Educational Forum, Plagiarism, Semantic Similarity, Text Mining.

Lista de Figuras

Lista de Tabelas

Lista de Acrônimos

Sumário

1 Introdução

A Educação a Distância (EAD) é uma modalidade de educação onde professores e alunos estão separados fisicamente no espaço e/ou no tempo e é efetivada através do uso das tecnologias de informação e comunicação, podendo ou não apresentar momentos presenciais (MORAN, 2009). A EAD está sendo cada vez mais utilizada na Educação Básica, Educação Superior e em cursos abertos. Segundo a Associação Brasileira de Educação a Distância (ABED)[1], em 2016, contabilizou-se 561.667 alunos que frequentam cursos regulamentados totalmente a distância, mostrando um aumento de 12,63% em relação ao ano de 2015.

A utilização dos Ambientes Virtuais de Aprendizagem (AVAs) para cursos EAD permitiu a uma parcela maior da população o acesso ao ensino superior (DILLENBOURG; SCHNEIDER; SYNTETA, 2002). Os AVAs possuem várias ferramentas que permitem ter uma grande interação entre professores e alunos, gerando um grande volume de dados nesses ambientes. Embora tenha sido uma boa solução para os alunos que não podem frequentar cursos presenciais, também é um problema para o professor fazer um acompanhamento adequado do desenvolvimento do aluno e dos trabalhos entregues, devido à grande quantidade de alunos, atividades e de turmas.

Dentre as ferramentas disponíveis nos AVAs os fóruns são os que promovem maior interação entre professores e alunos (BARROS; CARVALHO, 2011). Os alunos podem postar nos fóruns: respostas à perguntas levantadas pelo professor, dúvidas sobre atividades, comentários sobre o assunto da disciplina, levantamento de questões sobre o assunto, entre outros. Muitas disciplinas a distância utilizam a interação no fórum como forma de avaliação dos alunos. Contudo, com a grande quantidade de informação postada na ferramenta se torna um desafio para o professor avaliar as respostas manualmente, isso pode levar a diversos problemas, entre eles o plágio (OBERREUTER; VELÁSQUEZ, 2013).

O plágio acontece quando um aluno realiza cópia de ideias, conceitos ou frases de outro autor, sem dar os créditos ou citar o autor original. Dessa forma o aluno assume a autoria de algo que não é seu. De acordo com GARSCHAGEN (2006), basicamente existem três tipos de plágio:

- **Plágio integral**: a transcrição literal sem citação da fonte de um texto completo;

[1] http://abed.org.br/censoead2016/Censo_EAD_2016_portugues.pdf

- **Plágio parcial**: cópia de algumas frases ou parágrafos de diversas fontes diferentes, para dificultar a identificação;
- **Plágio conceitual**: apropriação de um ou vários conceitos, ou de uma teoria de outro autor.

No contexto dos AVAs LIU et al. (2007) considera o plágio de duas formas:

- **Externo**, quando um aluno copia um texto da internet ou utiliza um texto de outra pessoa sem citar o autor;
- **Interno**, quando os alunos copiam um texto que foi postado por outro aluno.

A disseminação da Internet permitiu que a população tivesse acesso a vários tipos de informações de forma rápida e fácil. Com isso, a prática de cópias de produções textuais pertencentes a outros autores tem se tornado comum em ambientes acadêmicos (ROCHA et al., 2012). Na EAD, essa prática é ainda mais frequente, pois o aluno precisa estar conectado a Internet para responder as atividades presentes no Ambiente Virtual de Aprendizagem (AVA). Um dos principais problemas em atividades EAD é a prática do plágio (SILVA, 2008).

Unindo o despreparo para escrever dos alunos com a falta de compromisso, acarreta diretamente na prática do plágio e também revela como o senso ético é pouco desenvolvido no meio acadêmico (SILVA, 2008). Nesse sentido, os principais desafios para os professores são: construir o entendimento ético de cada aluno e tentar diminuir a cópia textual de ideias e conceitos de outros autores sem citá-los.

Existem trabalhos na literatura para identificação de plágio em: atividades educacionais (FRANÇA; SOARES, 2012; PERTILE et al., 2010), artigos científicos (MASIC, 2012) e trabalhos de conclusão de curso (BARBASTEFANO; SOUZA, 2007). No entanto, quando o contexto é fóruns educacionais, a identificação de plágio se torna ainda mais difícil, devido principalmente ao tamanho do texto e por não exigir uma linguagem formal. Segundo ACHANANUPARP; HU; SHEN (2008), a variabilidade de expressão de linguagem natural faz com que seja difícil determinar sentenças semanticamente equivalentes.

A base fundamental para a criação de sistemas automáticos de detecção de plágio é a criação de uma medida de similaridade que possa mensurar a relação existente entre dois textos. Calcular a similaridade entre dois textos é um problema desafiador, pois, as frases podem representar significados diferentes, mesmo que utilizem as mesmas palavras, ou apenas uma pequena mudança na ordem das palavras no texto pode influenciar o significado da sentença (CHOUDHARY; BHATTACHARYYA, 2002).

Diante disso, este trabalho de dissertação tem por objetivo propor uma nova medida de similaridade textual para o português com o objetivo de detectar o plágio interno em fóruns educacionais. Esta medida extrai quatro características diferentes dos textos e em seguida utiliza regressão linear para determinar um valor entre 0 e 1, onde 0 significa que os textos não apresentam similaridade e 1 significa que os textos são exatamente iguais.

A medida foi avaliada na base disponibilizada pelo PROPOR[2] que possui 10.000 pares de sentenças com suas respectivas similaridades. A medida proposta alcançou melhores resultados do que todos os trabalhos relacionados. Além disso, foi desenvolvida uma ferramenta simples para simular um fórum educacional. Neste fórum foi aplicada a medida de similaridade criada, onde existia uma página que mostrava as similaridades entre as postagens dos alunos para o professor e também na página do fórum foi criado uma barra de originalidade, para que os alunos postagem conteúdos mais originais e não copiassem um do outro. O fórum foi utilizado em uma disciplina durante o semestre de 2017.1 e apresentou uma maior interatividade entre os alunos e uma maior diversificação nas postagens em relação a utilização de um fórum tradicional.

1.1 Objetivos

1.1.1 Objetivo Geral

O principal objetivo desta pesquisa é criar uma medida de similaridade textual para a língua portuguesa para identificação de postagens plagiadas em fóruns educacionais.

1.1.2 Objetivos Específicos

Para atingir o objetivo geral foram definidos os seguintes objetivos específicos:

- Analisar os métodos de similaridade textual existentes na literatura;
- Realizar uma revisão sistemática da literatura sobre o plágio em ambientes educacionais;
- Analisar técnicas de mineração de texto que podem auxiliar no cálculo da similaridade;
- Criar uma medida de similaridade textual para a língua Portuguesa;
- Avaliar a medida em uma base de dados de similaridade textual;
- Desenvolver um fórum web com a integração da medida criada;
- Validar o fórum criado.

1.2 Organização do trabalho

Este trabalho está dividido da seguinte forma: o Capítulo 1 apresenta uma breve introdução ao problema em questão e apresenta os objetivos do trabalho proposto. O Capítulo 2

[2] http://propor2016.di.fc.ul.pt/

apresenta os conceitos necessários para o entendimento desse trabalho: os fóruns educacionais, peça chave deste trabalho, as técnicas de Processamento de Linguagem Natural e alguns algoritmos de Aprendizagem de Máquina. O Capítulo 3 (CAVALCANTI et al., 2017) apresenta uma revisão da literatura sobre o plágio em ambientes virtuais de aprendizagem, mostrando dessa forma o estado da arte sobre o plágio nesses ambientes.

No Capítulo 4 são apresentados os trabalhos relacionados ao trabalho proposto, mostrando as técnicas que estão sendo aplicadas no plágio em ambientes virtuais de aprendizagem, seguido pelas medidas de similaridades presentes na literatura, principalmente para a língua portuguesa. O Capítulo 5 apresenta a proposta deste trabalho, que é a criação de uma medida de similaridade para sentenças escritas em português e a criação de uma ferramenta de fórum para aplicar a medida de similaridade criada.

O Capítulo 6 mostra os resultados obtidos. Inicialmente são apresentados os resultados da validação da medida de similaridade proposta. A medida foi avaliada em um banco de dados que possui sentenças similares, um valor de similaridade anotado e uma classe a qual pertence as sentenças. Em seguida são apresentados os resultados da aplicação da ferramenta de fórum educacional que foi desenvolvida em conjunto com a medida de similaridade proposta em uma turma de Computação no semestre de 2017.1.

Por fim, o Capítulo 7 apresenta as considerações finais, contribuições da pesquisa, artigos submetidos/aceitos, as limitações da pesquisa e os trabalhos futuros.

2

Fundamentação Teórica

2.1 Fóruns Educacionais

O fórum é uma ferramenta de comunicação onde vários usuários interagem de forma assíncrona. Desta forma é possível que o usuário faça a sua intervenção de forma mais organizada. No contexto educacional, o fórum abre várias possibilidades para professores e tutores interagirem de forma efetiva com a turma (ROLIM; FERREIRA; COSTA, 2016a).

O fórum tem uma característica importante, é nele que os alunos postam dúvidas, comentários sobre a disciplina, outras fontes de assunto, possíveis respostas para questões levantadas pelo professor, entre outros.

Outra característica importante é que o fórum é considerado uma das ferramentas que permite maior interatividade entre alunos e professores. Segundo o estudo promovido por BARROS; CARVALHO (2011) o fórum foi apontado por 69,2% dos alunos como a ferramenta mais interativa, outras ferramentas que também tem essa característica são: tarefa (41,0%), chat (38,5%) e questionário (20,5%).

Segundo FREITAS; SILVA (2009), o professor pode usar o fórum em diversos contextos, por exemplo:

(i) incentivar a criação de laços entres os alunos a partir da discussão de temas específicos da disciplina;

(ii) desenvolver a capacidade de debate crítico acerca de algum tema ou assunto;

(iii) dar uma resposta sobre dúvidas e comentários;

(iv) guiar os estudos dos alunos baseados nas suas postagens e;

(v) avaliar o aluno.

Dentre as possíveis funcionalidades dos fóruns se destaca a questão da avaliação. Muitas disciplinas a distância utilizam a interação no fórum como forma de avaliação dos alunos (ROLIM; FERREIRA; COSTA, 2016b). O ganho pedagógico causado pela utilização dos fóruns

educacionais é perceptível, mesmo que os alunos envolvidos no processo empreguem pouco tempo para utilizar esta ferramenta (CHENG et al., 2011).

2.2 Processamento de Linguagem Natural

Processamento de Linguagem Natural (PLN) é uma área de Ciência da Computação que estuda o desenvolvimento de programas de computador que analisam, reconhecem e/ou geram textos em linguagens humanas, ou linguagens naturais. O objetivo do PLN é fornecer aos computadores a capacidade de entender e compor textos. "Entender" um texto significa reconhecer o contexto, fazer análise sintática, semântica, léxica e morfológica, criar resumos, extrair informação, interpretar os sentidos, analisar sentimentos e até aprender conceitos com os textos processados (VIEIRA; LOPES, 2010).

Segundo PERNA; DELGADO; FINATTO (2010), o PLN não é uma tarefa trivial devido à rica ambiguidade da linguagem natural. Essa ambiguidade torna o PLN diferente do processamento das linguagens de programação de computador, as quais são formalmente definidas evitando, justamente, a ambiguidade. O PLN visa promover um nível mais alto de compreensão da linguagem natural através do uso de recursos computacionais, com o emprego de técnicas para o rápido processamento de texto (MACHADO et al., 2010).

Para LOULA (2011) embora a abordagem computacional tradicional veja a linguagem natural como uma mera sequência sintática de *tokens*, a linguagem natural pode e deve ser vista sob suas diversas faces, envolvendo não só aspectos sintáticos, mas também semânticos, pragmáticos, sociais, cognitivos, biológicos, semióticos, dentre outros.

As aplicações do PLN incluem tradução automática, reconhecimento automático de voz, geração automática de resumos, recuperação de informação, correção ortográfica e outras ferramentas que auxiliam a escrita.

Segundo BULEGON; MORO (2010), Processamento de Linguagem Natural envolve quatro etapas: análise morfológica, análise sintática, análise semântica e análise pragmática, que são realizadas nesta mesma ordem.

A análise morfológica é responsável por definir artigos, substantivos, verbos e adjetivos, os armazenando em um tipo de dicionário. Depois de construído o dicionário, a análise sintática faz uso dele procurando mostrar relacionamento entre as palavras e, em um segundo momento, verifica sujeito, predicado, complementos nominais e verbais, adjuntos e apostos. Na análise semântica, ocorre o encontro de termos ambíguos, de sufixos e afixos, ou seja, questões de significado associados aos morfemas componentes de uma palavra, o sentido real da frase ou palavra (SANTOS et al., 2015).

A seguir são apresentadas algumas técnicas de PLN que foram utilizadas neste trabalho de dissertação.

2.2.1 Pré-processamento

Em muitas aplicações de PLN é necessário utilizar técnicas de pré-processamento no texto antes de aplicá-lo a algum método computacional. Essas técnicas buscam filtrar as informações e convertê-las em uma representação compatível com os métodos computacionais utilizados. As subseções seguintes apresentam algumas técnicas de pré-processamento.

2.2.1.1 Tokenização

A tokenização é uma das primeiras fases do pré-processamento. Trata-se de identificar *tokens* que são as menores unidades de informação presentes no texto que possuem significado quando analisados de forma isolada. Desse modo, um *token* pode ser uma palavra, um número representado por um caractere numérico, um número de telefone, o nome de uma empresa formado pela combinação de uma ou mais palavras, um endereço da Web ou de e-mail e assim por diante BASTOS (2006).

2.2.1.2 Stopwords

Stopwords é uma lista de termos pouco representativos para um documento, geralmente essa lista é composta por: preposições, artigos, advérbios, números, pronomes e pontuação (DAG et al., 2001). É importante eliminar estas palavras, pois elas podem não só prejudicar o desempenho computacional do processamento, como também distorcer os resultados obtidos.

2.2.1.3 Stemming

Em um texto, muitas vezes as variações morfológicas das palavras remetem a um mesmo significado semântico e em algumas tarefas como, por exemplo, buscas por palavras chave devem ser consideradas equivalentes (BASTOS, 2006). Segundo OLIVEIRA (2008), o processo de *stemming* é realizado para converter diferentes manifestações de uma palavra a uma forma básica denominada *stem*. As variações podem ocorrer em função de sufixos inseridos para indicar o plural, gênero, conjugação verbal. Basicamente *stemming* é a retirada de sufixos do radical.

2.2.1.4 Lematização

Lematização é uma técnica que transforma a palavra para o seu radical (ou lema). A lematização é o ato de representar as palavras através do seu masculino singular, adjetivos e substantivos e infinitivo (verbos). Um verbo pode ter várias conjugações de tempo, por exemplo, para o verbo andar existem as conjugações: ando, andei, andaram, andou, entre outras. Então para obter mais precisão na comparação entre as palavras, cada verbo é transformado para o seu radical NAU et al. (2017).

2.2.2 TF-IDF

O esquema *Term Frequency-Inverse Document Frequency* (TF-IDF) é uma abordagem clássica da área de PLN. Segundo SALTON; YANG (1973) TF-IDF é uma medida estatística destinada a medir o grau de importância de uma palavra para um conjunto de documentos. TF-IDF combina a frequência dos termos (TF) e a relevância do termo para uma coleção (IDF). Através da métrica TF-IDF a pontuação de um termo é feita levando-se em conta a sua frequência no documento e em todos os outros documentos da coleção de documentos em análise. A interpretação simplificada dessa estratégia é que quando uma palavra ocorre com elevada frequência na coleção, ela é considerada menos importante e quando ocorre frequentemente em poucos documentos, sua medida TF-IDF tende a ser maior e, consequentemente, pode representar um termo importante (SALTON, 1989). O esquema TF-IDF é calculado pela Equação 2.1.

$$TF - IDF_{(j,d)} = TF_{(j,d)} \times IDF_{(j)} \tag{2.1}$$

Onde $TF_{(j,d)}$ é a frequência da palavra j no documento d e o parâmetro $IDF_{(j)}$ (*Inverse Document Frequency*), associado à palavra j, é dado pela Equação 2.2, onde $DF_{(j)}$ (*Document Frequency*) é o número de documentos em que a palavra j ocorre ao menos uma vez e $|N|$ representa a cardinalidade da coleção de documentos.

$$IDF = \log\left(\frac{|N|}{DF_{(j)}}\right) \tag{2.2}$$

Um problema que pode ocorrer ao se representar documentos como vetores, é que se as variações morfológicas de uma palavra, quando utilizadas com um mesmo significado semântico, forem consideradas termos distintos, a dimensão do espaço vetorial pode se tornar muito grande. Para lidar com este problema, aplicam-se em geral técnicas como remoção de *stopwords* e redução dos *tokens* à suas raízes (*stemming*) (OLIVEIRA, 2008). A Figura 2.1 mostra a matriz TF-IDF, onde para cada palavra w existe um valor TF-IDF (x) para cada sentença s.

	$\mathbf{s_1}$	$\mathbf{s_2}$	$\mathbf{s_3}$	$\mathbf{\cdots}$	$\mathbf{s_n}$
$\mathbf{w_1}$	$x_{1,1}$	$x_{1,2}$	$x_{1,3}$	$\cdots$	$x_{1,n}$
$\mathbf{w_2}$	$x_{2,1}$	$x_{2,2}$	$x_{2,3}$	$\cdots$	$x_{2,n}$
$\mathbf{w_3}$	$x_{3,1}$	$x_{3,2}$	$x_{3,3}$	$\cdots$	$x_{3,n}$
$\mathbf{\vdots}$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
$\mathbf{w_m}$	$x_{m,1}$	$x_{m,2}$	$x_{m,3}$	$\cdots$	$x_{m,n}$

Figura 2.1: Matriz TF-IDF.

2.3 Aprendizagem de Máquina

A área de Aprendizagem de Máquina (AM) é uma área especializada no estudo e construção de sistemas que sejam capazes de aprender de forma automatizada a partir de dados (BRINK; RICHARDS; FETHEROLF, 2016). De acordo com SIMON (1983) aprendizado é qualquer mudança em um sistema que melhore o seu desempenho na segunda vez que ele repetir a mesma tarefa, ou outra tarefa da mesma população. Após efetuado o aprendizado, também denominado treinamento, um sistema pode ser utilizado para classificar ou estimar saídas para instâncias desconhecidas.

Os métodos para aprendizagem de máquina são divididos em duas categorias: supervisionado e não-supervisionado. A abordagem de aprendizado supervisionado consiste em utilizar uma série de exemplos (chamados de instâncias), já classificados, para induzir um modelo que seja capaz de classificar novas instâncias de forma precisa, com base no aprendizado obtido na fase de treinamento. Nessa abordagem tem-se a figura de um "professor externo", o qual apresenta um conhecimento do ambiente representado por conjuntos de exemplos na forma entrada- saída. Neste caso, o algoritmo de AM é treinado a partir de conjuntos de exemplos rotulados com o objetivo de aprender uma função desejada.

Já na abordagem de aprendizado não-supervisionado, o conjunto de dados utilizado não possui classificação, ou seja, a saída do conjunto de dados de treinamento não possui uma saída pré-definida para cada uma de suas instâncias. Esta é a abordagem indicada quando o objetivo do sistema não é construir um modelo de predição, e sim um modelo cuja função seja encontrar regularidades nos dados que possam vir a ser úteis (THEODORIDIS; KOUTROUMBAS, 2001).

2.3.1 Algoritmos de Classificação

Na literatura, há vários algoritmos para realizar a classificação automática de padrões. Neste trabalho foram utilizados os seguintes classificadores: *Support Vector Machine* (SVM), Naive Bayes, *Random Forest*, *Logistic* e redes neurais.

2.3.1.1 Support Vector Machine

As Máquinas de Vetores Suporte (*Support Vector Machines* - SVMs) constituem uma técnica embasada na Teoria de Aprendizado Estatístico (VAPNIK; VAPNIK, 1998). Dado um problema linearmente separável, o SVM encontra um hiperplano ótimo que separa os padrões em duas regiões, chamadas de positiva e negativa, onde cada região representa uma classe. O algoritmo define esse hiperplano com padrões que estão nas fronteiras de cada região, chamados vetores de suporte, esses exemplos são encontrados através de uma busca.

Segundo SMOLA et al. (2000), algumas das principais características das SVMs que tornam seu uso atrativo são:

(i) **Boa capacidade de generalização**: os classificadores gerados por uma SVM em geral alcançam bons resultados de generalização. A capacidade de generalização de um classificador é medida por sua eficiência na classificação de dados que não pertençam ao conjunto utilizado em seu treinamento.

(ii) **Robustez em grandes dimensões**: as SVMs são robustas diante de objetos de grandes dimensões.

(iii) **Teoria bem definida**: as SVMs possuem uma base teórica bem estabelecida dentro da Matemática e Estatística.

Segundo BURGES (1998), para efetuar classificações/reconhecimento de padrões o SVM constrói hiperplanos em um espaço multidimensional objetivando separar casos de diferentes classes. Quando o SVM separa os vetores das classes sem erro e com distância máxima para com os vetores mais próximos é considerado como separação ótima, como mostra a Figura 2.2 (VAPNIK, 2013).

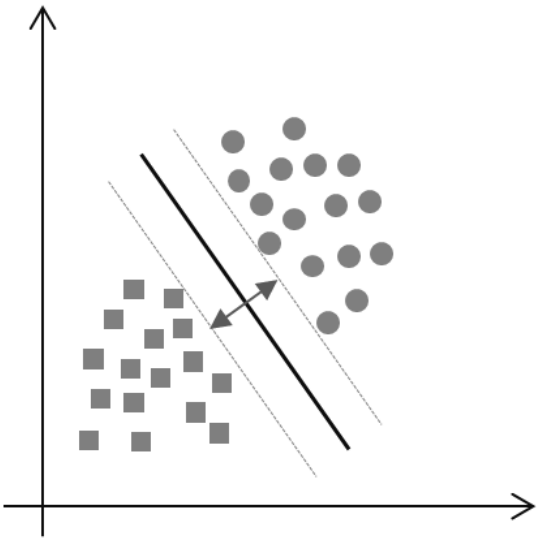

Figura 2.2: Exemplo de classificação com SVM.

Para problemas não-linearmente separáveis, o SVM faz um mapeamento dos dados em outro espaço de dimensão maior, através de uma função Φ, chamada de *Kernel*. Fazendo a escolha adequada de Φ, os dados podem ser separados por um SVM linear nesse novo espaço (HEARST et al., 1998).

Um *Kernel* K é uma função que recebe dois pontos x_i e x_j do espaço de entradas e computa o produto escalar desses dados no espaço de características (HAYKIN, 1994), conforme a Equação 2.3.

$$k(\mathbf{x}_i, \mathbf{x}_j) = \Phi(\mathbf{x}_i) \cdot \Phi(\mathbf{x}_j) \tag{2.3}$$

Segundo LORENA; CARVALHO (2007), é comum empregar a função *Kernel* sem conhecer o mapeamento Φ, que é gerado implicitamente. A utilidade dos *Kernels* está, portanto, na simplicidade de seu cálculo e em sua capacidade de representar espaços abstratos. Alguns dos *Kernels* mais utilizados na prática são os Polinomiais, os Gaussianos ou *Radial-Basis Function* (RBF) e os *Sigmoidais*, listados na Tabela 1. Cada um deles apresenta parâmetros que devem ser determinados pelo usuário, indicados também na tabela.

Tabela 2.1: Funções Kernels mais comuns.

Tipo de Kernel	**Função** $k(\mathbf{x}_i, \mathbf{x}_j)$	**Parâmetros**
Polinomial	$(\delta(\mathbf{x}_i \cdot \mathbf{x}_j) + \kappa)^d$	δ, κ e d
Gaussiano (RBF)	$\exp(-\sigma\|\|\mathbf{x}_i - \mathbf{x}_j\|\|^2)$	σ
Sigmoidal	$\tanh(\delta(\mathbf{x}_i \cdot \mathbf{x}_j) + \kappa)$	δ e κ

Há uma grande variedade de implementações de SVMs disponíveis para uso não comercial. Neste trabalho foi utilizado o LibSVM disponível no *Waikato Environment for Knowledge Analysis* (WEKA)[1]. O LibSVM possui biblioteca de funções para SVMs. Permite seleção de modelo por *cross-validation* e atribuir pesos aos dados para lidar com distribuições de dados desbalanceadas. Pode ser utilizado em problemas de classificação multiclasses e regressão.

2.3.1.2 Naive Bayes

Classificadores bayesianos tem ganhado popularidade ultimamente, mostrando terem ótima performance na classificação de texto (ROY; JOSHI; KRISHNAPURAM, 2004; SÁ, 2008). O classificador Naive Bayes é baseado no teorema de bayes e tem como principal característica assumir que todos os atributos dos exemplos são independentes um do outro, dado o contexto da classe (RISH, 2001). Esta é a suposição "ingênua"de Bayes. Embora esta suposição seja claramente falsa na maioria das tarefas do mundo real, Naive Bayes tem vários relatos na literatura sobre sua competitividade para com outros classificadores. DOMINGOS; PAZZANI (1997) mostraram teoricamente que a suposição de independência de palavras na maioria dos casos não prejudica a eficiência do classificador. O Naive Bayes calcula a probabilidade de um dado elemento pertencer a uma classe por meio da seguinte equação:

$$P(c|x_i) = \frac{P(x_i|c)P(c)}{P(x_i)} \tag{2.4}$$

onde $P(c|x_i)$ é a probabilidade a posteriori de um elemento pertencer a uma dada classe, $P(x_i)$ é a probabilidade de cada atributo, sem levar em consideração dependência, $P(x_i|c)$ a probabilidade de um elemento pertencer a uma dada classe e $P(c)$ a probabilidade original da classe.

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

2.3.1.3 Logistic Regression

Segundo DAYTON (1992), o método *Logistic Regression* ou Regressão Logística é um método estatístico responsável por produzir um modelo de predição de valores recebidos por uma variável categórica, em geral, binária. Quando a variável alvo possui duas classes, utiliza-se a regressão logística binária ou dicotômica. Segundo HOSMER JR; LEMESHOW; STURDIVANT (2013) nos casos de uma variável alvo com mais de duas classes, utiliza-se um modelo de regressão logística multinomial. A regressão logística é um recurso que nos permite estimar a probabilidade associada à ocorrência de determinado evento em face de um conjunto de variáveis explanatórias.

2.3.1.4 Random Forest

O classificador *Random Forest* ou Florestas Aleatórias consiste em um conjunto de árvores de decisão geradas dentro de um mesmo objeto. Esse método foi proposto por BREIMAN (2001) e consistem em um conjunto de árvores de decisão construídas no momento de treinamento do método. As árvores são construídas selecionando aleatoriamente alguns dos atributos contidos dentro do vetor de características. Cada objeto (conjunto de árvores) passa por um mecanismo de votação (*bagging*), que elege a classificação mais votada. A classificação encontra-se nos nós terminais das mesmas. A saída do classificador é dada pela classe que foi retornada como resposta pela maioria das árvores pertencentes à floresta.

2.3.1.5 Redes Neurais

Redes Neurais Artificiais são técnicas computacionais que apresentam um modelo matemático inspirado na estrutura neural de organismos inteligentes e que adquirem conhecimento através da experiência. Esse modelo é composto por um conjunto de neurônios, ou nós, que são interligados entre si, formando uma rede. Cada neurônio recebe entradas com um peso associado. A partir das entradas e de seus respectivos pesos, um somatório ponderado é realizado no núcleo do neurônio e com base em um limiar de ativação é verificado se a entrada será ou não propagada para neurônios das camadas adjacentes a camada atual.

Nessas redes, cada camada tem uma função específica. A camada de saída recebe os estímulos da camada intermediária e constrói o padrão que será a resposta. As camadas intermediárias funcionam como extratoras de características, seus pesos são uma codificação de características apresentadas nos padrões de entrada e permitem que a rede crie sua própria representação, mais rica e complexa, do problema. Neste trabalho foi utilizado o *MultiLayer Perceptron* (MLP) que consiste em um modelo clássico de rede neural (WANKHEDE, 2014).

Uma Rede Neural Artificial (RNA) do tipo MLP é constituída por um conjunto de nós fonte, os quais formam a camada de entrada da rede, uma ou mais camadas escondidas e uma camada de saída. Com exceção da camada de entrada, todas as outras camadas são constituídas por neurônios e, portanto, apresentam capacidade computacional. As RNAs MLPs têm sido

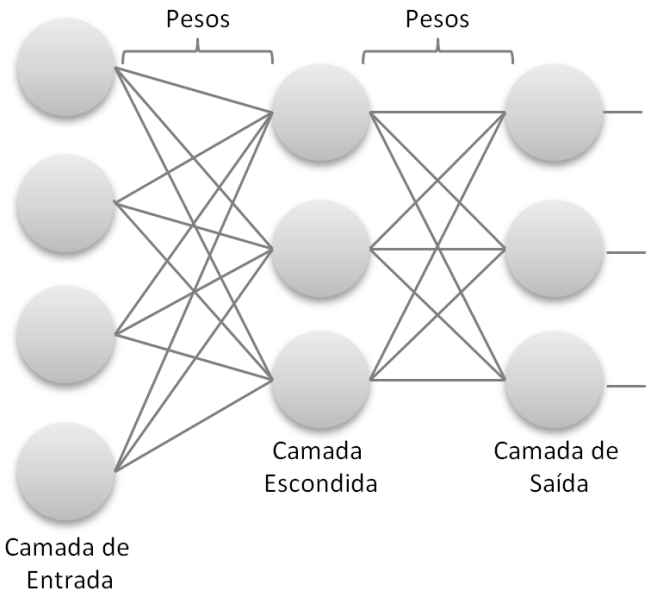

Figura 2.3: Estrutura de uma Rede Neural.

aplicadas na solução de diversos e difíceis problemas através da utilização de algoritmos. O algoritmo de treino muito utilizado é o algoritmo de retro-propagação do erro, conhecido na literatura como *Backpropagation*. O algoritmo *backpropagation* baseia-se na heurística do aprendizado por correção de erro (em que o erro é retro-propagado da camada de saída para as camadas intermediárias da RNA).

2.3.2 Avaliação do desempenho do Classificador

Validação cruzada (do inglês *Cross Validation*) é uma técnica estatística que particiona uma amostra dos dados em subconjuntos de tal modo que a análise é inicialmente executada em um único subconjunto, enquanto os outros subconjuntos são mantidos para treino (KOHAVI et al., 1995).

O K-*Fold Cross Validation* é um método de avaliação. Os documentos são aleatoriamente divididos em K partições mutuamente exclusivas ("*folds*") de tamanho aproximadamente igual a n/K, onde n é o tamanho do conjunto de documentos. Então, são realizados K experimentos, onde, em cada experimento, uma partição diferente é escolhida para o teste e as K-1 partições restantes são escolhidas para o treinamento. A medida de eficiência é a média das medidas de eficiência calculadas para cada uma das partições. Neste trabalho foi utilizado a taxa de acerto como medida de eficiência. A grande vantagem dessa técnica, é que todos os documentos são usados tanto para treinamento quanto para teste. Assim, a forma como a divisão do corpus foi feita influi menos no resultado final, qualquer documento será usado exatamente uma vez para teste e K-1 vezes para treinamento.

2.3.3 Algoritmo de Regressão Linear

A regressão consiste na execução de uma análise estatística para verificar a existência de uma relação funcional entre uma variável dependente com uma ou mais variáveis independentes. Em outras palavras consiste na obtenção de uma equação que tenta explicar a variação da variável dependente pela variação do(s) nível(is) da(s) variável(is) independente(s). Um tipo de algoritmo de regressão é a regressão linear, que recebe esse nome porque se considera que a relação da resposta às variáveis é uma função linear de alguns parâmetros (MATOS, 1995).

A regressão linear tem por objetivo encontrar uma relação do tipo:

$$Y = a + b_1X_1 + b_2X_2 + ... + b_pX_p, \tag{2.5}$$

onde $a, b_1, b_2, ..., b_p$ seriam os parâmetros da relação linear procurada, $X_1, X_2, ..., X_p$ o conjunto de variáveis independentes e Y a variável dependente.

2.4 Word Embeddings

Em linguagens naturais, uma palavra possui significado além de um simples conjunto de letras organizados em certa ordem. Uma série de conceitos e alusões existem por trás de cada palavra na mente humana, porém o mesmo não é valido para sistemas computacionais, que reconhecem uma palavra simplesmente como um conjunto de caracteres. Nos últimos anos, a representação de palavras por meio de vetores tem gerado resultados bastante promissores. Tais vetores de palavras são chamados de *word embeddings*. Estes vetores são treinados por modelos neurais em grandes corpora por meio de técnicas de aprendizado de máquina não supervisionado (BENGIO et al., 2003; MIKOLOV et al., 2013).

As *word embeddings* seguem o princípio das representações distribuídas, proposto relativamente há longo tempo, na segunda metade da década de 80 em WILLIAMS; HINTON (1986), onde as representações de palavras (semanticamente) similares estão próximas no espaço vetorial.

Um modelo de *embeddings* necessita apenas de um grande córpus de treinamento que possa modelar um contexto específico. A *embedding* de uma palavra representa o contexto no qual ela ocorre, capturando relações sintáticas e semânticas. O propósito de *word embeddings* é transformar as palavras em números, onde algoritmos de *Deep Learning* (DL) podem então ingerir e processar, para formular uma compreensão da linguagem natural.

Deep Learning ou Aprendizagem Profunda refere-se a uma RNA com a presença de muitas camadas. Pode-se considerar uma rede neural profunda quando esta possui mais de 100 camadas (DRAELOS et al., 2017). Nos últimos anos, a DL conseguiu resolver uma série de problemas de longa data na Inteligência Artificial, como reconhecimento de fala, tradução de texto, geração de sequência textual, classificação de imagem e geração de texto a partir da imagem (COELHO; SILVEIRA, 2017). O sucesso de DL também se deve a maior disponibilidade de um

grande córpus para treinamento e o baixo custo para utilização de GPUs (*Graphics Processing Unit*) (GAZZOLA, 2017).

MIKOLOV et al. (2013) propõe duas abordagens para modelagem de palavras, Skip-Ngram e *Continuous Bag of Words* (CBOW). CBOW utiliza uma sequência de n palavras para predizer a palavra no instante $n+1$. Skip-Ngram utiliza uma única palavra i para predizer uma janela j de palavras vizinhas. A Figura 2.4 mostra a diferença entre as duas abordagens.

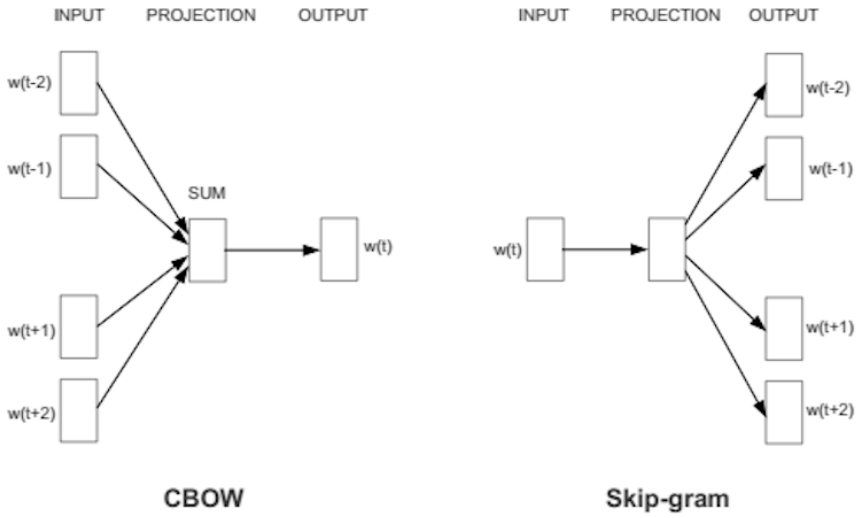

Figura 2.4: Diferença entre CBOW e *Skip-Ngram*.

Um método amplamente utilizado no PLN para gerar *word embeddings* é o word2vec. O word2vec[2] contém o algoritmo de treinamento *Skip-gram*. Através do treinamento, o word2vec simplifica o processamento do contexto para o processamento vetorial no espaço vetorial K-dimensional. Portanto, podemos obter as representações vetoriais das palavras e a similaridade entre palavras pode ser calculada. O vetor de palavras pode ser considerado como um mapeamento de espaço de contexto para espaço de vetor, e pode representar fielmente as palavras.

Os resultados do treinamento do word2vec podem variar com base nos parâmetros definidos antes do treinamento. Alguns parâmetros importantes são apresentados a seguir.

- **Dimensão**: segundo MIKOLOV et al. (2013), a qualidade da incorporação de palavras aumenta com maior dimensionalidade, mas depois de chegar a algum ponto, o ganho marginal diminuirá. Normalmente a dimensão é definida entre 100 e 1.000.
- **Tamanho da janela**: o tamanho da janela determina quantas palavras vizinhas (antes e depois) da palavra atual serão incluídas como palavras do contexto da palavra atual. Esse valor geralmente é definido 10 para Skip-gram e 5 para CBOW.
- **Frequência das palavras**: é o número mínimo de vezes que uma palavra deve aparacer no corpus para ser incluída no treinamento.

[2] `https://code.google.com/archive/p/word2vec/`

- **Número de iterações**: é o número de vezes que a rede neural irá atualizar os seus coeficientes. Se forem definidas poucas iterações a rede pode não aprender tudo sobre os dados do treinamento. Se forem definidas muitas iterações, mais tempo levará para concluir o treinamento.

O word2vec recebe como entrada um grande córpus para treinamento e tem como saída um modelo com a representação vetorial de cada palavra do córpus (HARTMANN et al., 2017; MIKOLOV et al., 2013). A principal ideia dessas representações é medir a similaridade semântica entre as palavras. Segundo HARTMANN (2016), a vantagem de utilizar *word embeddings* é a baixa esparsidade de dados e a independência de recursos léxicos e semânticos.

3

O Plágio em Ambiente Educacional Virtual

Os AVAs possuem ferramentas que permitem ter uma grande interação entre professores e alunos. Entre elas, estão a ferramenta de envio e recebimento de atividades, produção textual, fórum de discussão, chats, blogs, entre outros. Essa grande interação é benéfica e necessária para a aprendizagem do aluno. Entretanto, se torna um desafio para o professor ter um acompanhamento adequado das atividades e informações geradas pelos alunos no AVA. Com isso, o plágio tem se tornado uma prática comum nesses ambientes.

Alguns trabalhos foram publicados tendo como finalidade um método para detecção de plágio externo. PERTILE et al. (2010) inicialmente apresentam uma modelagem de um agente detector de indícios de plágio, onde os autores apresentam um motor de busca que realiza buscas na internet a fim de encontrar documentos similares ao documento postado pelo aluno no AVA. ARENHARDT et al. (2012) apresentam um aprimoramento do método detector de indícios de plágio, onde, como diferenciais, os autores utilizam técnicas de processamento de linguagem natural e é feita uma análise do texto retornado pelo motor de busca em relação aos arquivos que já estão no repositório do Moodle.

Existem também várias ferramentas que auxiliam na detecção de plágio externo, entre elas: Turnitin, Plagiarism Detect, Farejador de Plágio, Plagius, Viper, entre outras. Entretanto, o método proposto por PERTILE et al. (2010) tem como diferencial a integração com os AVAs.

3.1 Planejamento da Revisão

Esta Revisão Sistemática da Literatura (RSL) está estruturada com base nas diretrizes originais propostas por KITCHENHAM (2004). O principal objetivo desta RSL foi analisar qual o cenário de publicações nacionais e internacionais cuja temática seja o plágio em ambientes educacionais virtuais, bem como identificar quais as principais técnicas que estão sendo utilizadas para detecção de plágio, dentro do período de 01 de Janeiro de 2007 a 31 de Dezembro de 2016. A partir disso, foram definidas as seguintes Questões de Pesquisa:

- Quais são os objetivos educacionais presentes nas publicações?

- Quais ferramentas educacionais estão sendo utilizadas?
- Quais técnicas de PLN estão sendo utilizadas para detecção de plágio?
- Quais ferramentas que estão sendo utilizadas na detecção de plágio?

Foram realizadas buscas nas páginas Google Scholar, ACM Digital Library e IEEExplore utilizando os termos de busca: "Fórum" + "Mineração de Texto" e "*Forum*" + "*Text Mining*. Após o resultado inicial foram definidas 56 referências (conferências e periódicos) das áreas de Computação e Educação, Inteligência Artificial (IA) e PLN que possibilitem responder as questões de pesquisa levantadas. Com isso, foi realizada uma busca manual nas 56 referências dentro do período definido (2007 a 2016). O Apêndice A mostra as 56 referências escolhidas separadas em três principais áreas: Educação (Tabela A.1), IA (Tabela A.2) e PLN (Tabela A.3).

A Tabela 3.1 apresenta os critérios de inclusão (CI) e exclusão (CE) utilizados durante esta RSL.

Tabela 3.1: Critérios de inclusão e exclusão

Critérios de Inclusão
(CI1) Artigos que apresentem técnicas, algoritmos ou ferramentas utilizadas para detecção de plágio. (CI2) Artigos que descrevem o problema do plágio ou relatam experiências com a detecção de plágio.
Critérios de Exclusão
(CE1) Não serão selecionadas publicações em que as palavras-chave da busca não apareçam no título, resumo e/ou palavras-chaves. (CE2) Não serão selecionadas publicações que envolvem detecção de plágio sem objetivos educacionais. (CE3) Não serão selecionadas publicações que não estejam no período definido (2007 a 2016).

3.2 Fases da Revisão

Com base nos critérios de inclusão e exclusão definidos (Tabela 3.1), foram definidas as seguintes fases para seleção das publicações:

(1) Leitura do título e/ou resumo da publicação e aplicação dos critérios de inclusão e exclusão;

(2) Leitura do texto completo de cada publicação selecionada na fase 1 aplicando os critérios de inclusão e exclusão.

A execução da primeira fase resultou em 54 artigos. Ao executar a segunda fase, 21 foram considerados relevantes para a pesquisa. A Tabela 3 mostra os artigos retornados durante as fases da seleção.

Tabela 3.2: Artigos retornados durante o processo de seleção

Fases da Seleção	Número de Artigos Selecionados
Fase 1	54
Fase 2	21

3.3 Síntese dos Resultados

Esta seção apresenta os resultados obtidos. Foram extraídos atributos dos 21 artigos selecionados a fim de melhor categorizar os resultados. A Figura 3.1 apresenta a distribuição dos artigos quanto ao tipo da fonte. A maioria dos artigos foram publicados em periódicos (62%).

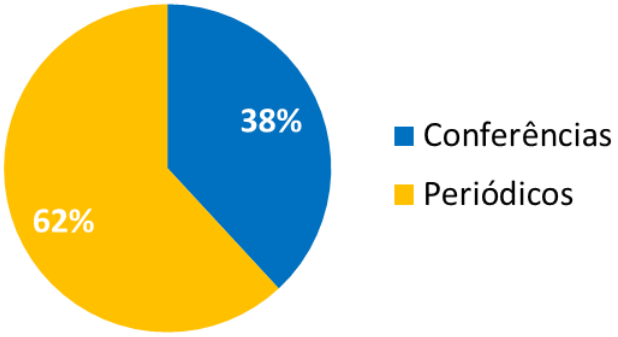

Figura 3.1: Distribuição dos artigos quanto ao tipo de fonte.

A Figura 3.2 mostra o gráfico com a frequência anual de publicação. O gráfico mostra uma média de 3 publicações por ano, com exceção dos anos de 2007 e 2014 que não tiveram publicações. A Figura 3.2 também mostra que o plágio é um tema que vem sendo abordado em várias pesquisas nos últimos 10 anos.

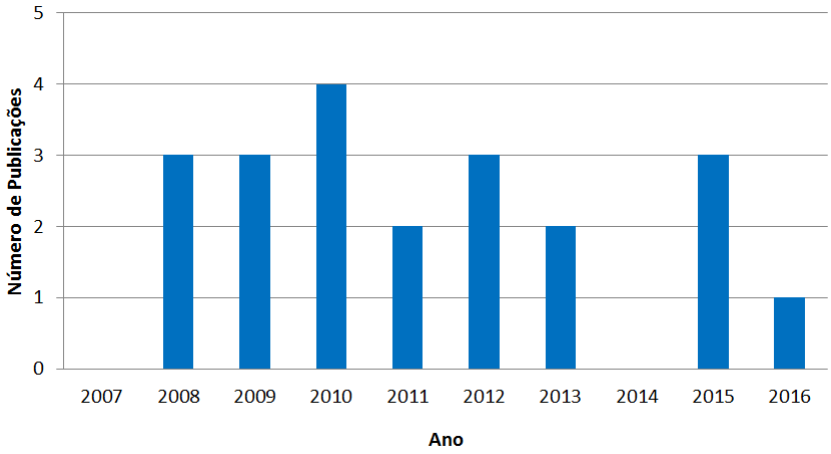

Figura 3.2: Frequência anual de publicações.

A Tabela 3.3 mostra a distribuição dos artigos por Instituição de Ensino Superior (IES). Um artigo pode ter autores de mais de uma instituição, por isso o total da Tabela 3.3 ultrapassou

o número total de artigos relevantes para a busca. Além disto, as publicações onde mais de um autor pertencia à mesma IES foram contabilizadas como sendo apenas uma publicação para a instituição. Das universidades brasileiras, a UFSM foi a que teve mais publicações com 5 artigos. Em "Outras" encontram-se as universidades brasileiras: UFRGS, UFPE, UFGD, UFCG, UNEB, UNIPAMPA e FSM com 1 artigo cada.

Tabela 3.3: Distribuição dos trabalhos por IES.

IES	Quantidade
UFSM	5
University of Aizu	2
SolBridge School of Business	2
American University of Nigeria	2
Outras	21
Total	**32**

A Tabela 3.4 detalha as conferências e periódicos. A conferência com mais publicações foi o Simpósio Brasileiro de Informática na Educação (SBIE) com 4 publicações. As outras conferências possuem 1 publicação cada. Em relação aos periódicos, se destacam o *Computers & Education* com 4 publicações e o *Journal of Educational Computing Research* com 2 publicações. Entre os periódicos que possuem 1 publicação estão: *International Journal on E-Learning*, *Expert Systems with Applications*, Revista Novas Tecnologias na Educação (RENOTE), Revista Brasileira de Educação, *Educational Technology & Society*, *Journal of Educational Technology Systems* e Revista Brasileira de Informática na Educação (RBIE).

Os artigos foram inicialmente classificados como: (i) **Aplicado**: artigos que propõe a utilização de técnicas, ferramentas ou algoritmos para solucionar o problema do plágio; (ii) **Teórico**: artigos que descrevem o problema de plágio relatando experiências ou mostrando o que poderia ser feito para resolver este problema. Dos 21 artigos, 13 foram classificados como aplicados (62%) e 8 como teóricos (38%). A Figura 3.3 mostra a distribuição dos artigos nestas duas categorias.

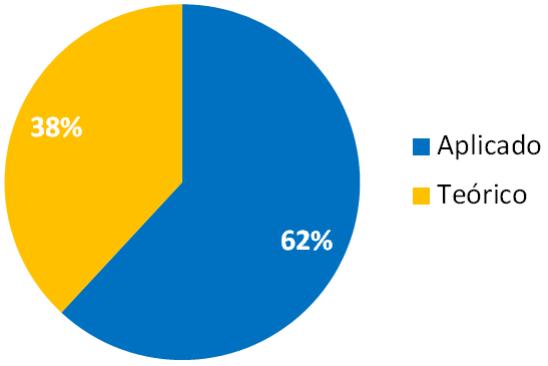

Figura 3.3: Distribuição de artigos nas categorias: Aplicado e Teórico.

Tabela 3.4: Detalhamento das conferências e periódicos.

Fonte		Quantidade
Conferências	Simpósio Brasileiro de Informática na Educação (SBIE)	4
	Workshop de Desafios da Computação aplicada à Educação (DesafiE)	1
	International Conference on Advanced Learning Technologies	1
	International Conference e-Learning	1
	Asia Pacific Conference on Educational Integrity	1
Periódicos	*Computers & Education*	4
	Journal of Educational Computing Research	2
	International Journal on E-Learning	1
	Expert Systems with Applications	1
	Revista Novas Tecnologias na Educação (RENOTE)	1
	Revista Brasileira de Educação	1
	Educational Technology & Society	1
	Journal of Educational Technology Systems	1
	Revista Brasileira de Informática na Educação (RBIE)	1
Total		**21**

Além da classificação em Aplicado e Teórico, os artigos também foram classificados em relação ao objetivo educacional (Tabela 3.5), conforme as seguintes categorias:

- **Método para detecção de plágio externo**: esses artigos têm por objetivo criar métodos ou ferramentas que detectem o plágio externo, ou seja, quando os alunos copiam de textos obtidos na Internet;
- **Método para detecção de plágio interno**: esses artigos têm por objetivo criar métodos ou ferramentas que detectem o plágio interno, ou seja, quando os alunos copiam de outros alunos que estão realizando a mesma atividade;
- **Método para detecção de plágio interno e externo**: nesse caso são artigos que englobam os dois objetivos acima juntos;
- **Análise do plágio na produção acadêmica**: esses artigos têm por objetivo descrever o problema de plágio ou relatar experiências de plágio na produção acadêmica;
- **Análise de ferramenta de detecção de plágio**: esses artigos têm por objetivo aplicar alguma ferramenta de detecção de plágio em ambientes virtuais de aprendizagem ou realizar comparações entre as ferramentas de plágio existentes na literatura;
- **Método para diminuir o comportamento de plágio**: esses artigos têm por objetivo aplicar alguma ferramenta ou método no intuito de diminuir o comportamento de plágio dos estudantes.

A Tabela 3.5 responde a primeira pergunta de pesquisa, "*Quais são os objetivos educacionais presentes nas publicações*?". Os objetivos "*Método para detecção de plágio externo*" e "*Análise de ferramenta de detecção de plágio*" contemplam mais da metade (57,14%) do total de artigos.

Tabela 3.5: Distribuição de artigos por categoria.

Categoria	Quantidade
Método para detecção de plágio externo	6
Método para detecção de plágio interno	1
Método para detecção de plágio interno e externo	1
Análise do plágio na produção acadêmica	4
Análise de ferramenta de detecção de plágio	6
Método para diminuir o comportamento de plágio	3
Total	**21**

Respondendo a segunda questão de pesquisa, "*Quais ferramentas educacionais estão sendo utilizadas?*", foi encontrada apenas uma ferramenta, o Moodle, estando presente em 7 das 21 publicações, ou seja, em 33% das publicações. Os trabalhos que não utilizaram o Moodle e foram classificados como "Aplicados", geralmente aplicavam métodos de detecção de plágio em bases de dados anotadas ou aplicavam ferramentas de plágio (Turnitin, Viper, etc) em trabalhos acadêmicos.

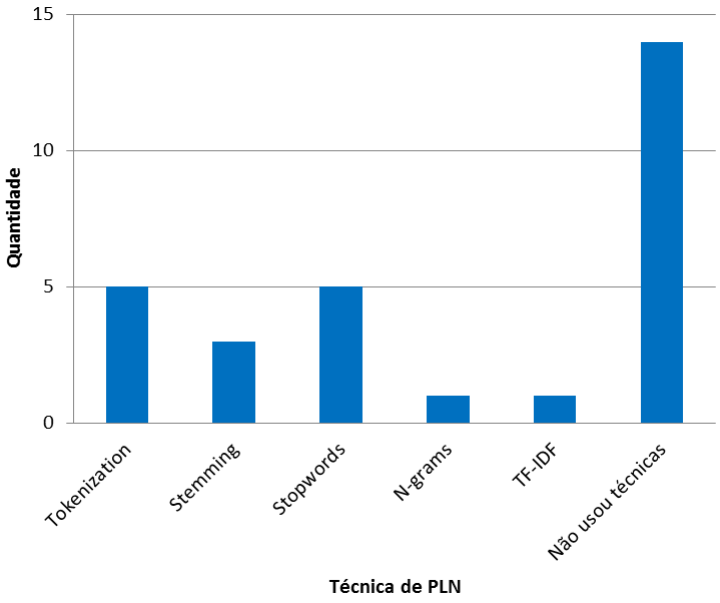

Figura 3.4: Técnicas de PLN utilizadas nos trabalhos.

Os métodos de detecção de plágio geralmente utilizam técnicas de PLN. Algumas delas são: *tokenization*, *stemming*, remoção de *stopwords*, entre outras. A Figura 3.4 mostra as técnicas de PLN utilizadas nas publicações. É importante salientar que uma publicação pode

conter mais de uma técnica. Em relação à terceira questão de pesquisa, "*Quais técnicas de PLN estão sendo utilizadas para detecção de plágio?*", a Figura 3.4 mostra as técnicas utilizadas e observa-se que as técnicas "*Tokenization*" e "*Stopwords*" são as mais utilizadas, estando presente em 23,81% das publicações cada. Entre as publicações que não utilizaram técnicas de PLN encontram-se todas as publicações teóricas.

A Figura 3.5 mostra a distribuição de artigos em relação à utilização de ferramentas. Do total de artigos 62% utilizaram alguma ferramenta e 38% não utilizaram ferramenta. Levando em consideração que 62% dos artigos foram classificados como "Aplicados", então podemos afirmar que todos os artigos "Aplicados" utilizaram alguma ferramenta.

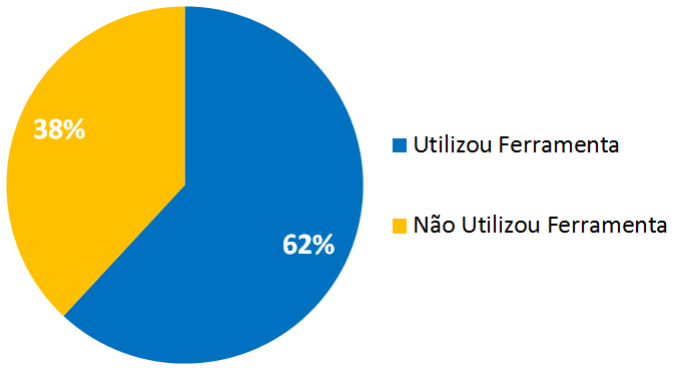

Figura 3.5: Distribuição de artigos quanto à utilização de ferramenta.

Tabela 3.6: Ferramentas utilizadas nos artigos.

Ferramenta	Nº de artigos	% de artigos
API Google Search	3	14%
Turnitin	3	14%
API de Busca da Bing	2	10%
Linguagem JAVA	2	10%
RapidMiner	2	10%
Lucene	1	5%
WCopyFind	1	5%
Microsoft Live Search	1	5%
Wordnet	1	5%
Glatt	1	5%
Linguagem C#.Net	1	5%
Linguagem PHP	1	5%
Miss Marple	1	5%
Farejador de Plágio	1	5%
Viper	1	5%
Plagius Detector	1	5%

Por fim, para responder a quarta questão de pesquisa: "*Quais ferramentas que estão sendo utilizadas na detecção de plágio?*", a Tabela 3.6 mostra todas as ferramentas utilizadas

nas publicações. As linguagens de programação estavam presentes em 19,04% dos trabalhos. Alguns trabalhos não informaram a linguagem utilizada. A API Google Search foi a ferramenta mais utilizada, presente em 3 publicações, seguida pela ferramenta Turnitin (uma ferramenta para detecção de plágio externo) com 3 publicações também. Vale ressaltar que as ferramentas API Google Search, API de Busca Bing e Microsoft Live Search são ferramentas que possuem o mesmo objetivo, mas utilizam motores de busca diferentes. Essas ferramentas estavam presentes em 28,57% dos trabalhos.

A Tabela 3.7 mostra informações de cada artigo selecionado. As informações são: título e autor, ano de publicação, objetivo educacional, técnicas de PLN ou ferramentas utilizadas e o número de citações obtido pelo Google Scholar (Dados coletados no mês de Novembro de 2017).

3.4 Discussão

Os AVAs possuem várias ferramentas para interação entre professores e alunos. Pesquisas foram desenvolvidas com o objetivo de evitar o plágio na ferramenta de envio e recebimento de atividades (PERTILE et al., 2010; ARENHARDT et al., 2012; GOMES; MEDINA, 2016; BATANE, 2010). Essas pesquisas detectam o plágio externo, ou seja, quando o aluno copia de fontes externas, como, por exemplo, livro, artigo de revista, monografias ou internet. Para detecção do plágio interno, ou seja, quando um aluno copia a tarefa de outro quando ambos estão realizando uma mesma tarefa, foram encontradas poucas pesquisas (BUTAKOV; SCHERBININ, 2009; ARENHARDT et al., 2012; OBERREUTER; VELÁSQUEZ, 2013).

A base fundamental para a criação de sistemas automáticos de detecção de plágio é a criação de medidas de similaridade entre textos. Nos últimos anos foram propostos vários métodos para similaridade entre sentenças (ACHANANUPARP; HU; SHEN, 2008). Entretanto, uma dificuldade presente na área é a necessidade de cada idioma ter uma técnica de similaridade própria. Além disso, no contexto do português, os recursos de PLN ainda são escassos e não atingiram a acurácia de outros idiomas como o inglês.

Embora o plágio esteja sendo abordado em várias pesquisas nos últimos anos, ainda existem muitas tendências futuras de pesquisa, como por exemplo: (i) verificar o estilo da escrita presentes nos documentos enviados pelos alunos; (ii) detecção de plágio interno e externo nos fóruns de discussão (muitas disciplinas utilizam o fórum como forma de avaliação); (iii) integração de algoritmos de mineração de texto e aprendizagem de máquina aos Ambientes Virtuais de Aprendizagem. Dessa forma, pesquisas nessas áreas podem ajudar a diminuir ou até eliminar a cópia de ideias, conceitos ou trabalhos de outros autores e ajudar na construção do entendimento ético de cada aluno.

Dessa forma, este trabalho de dissertação de mestrado tem por objetivo criar uma medida de similaridade entre sentenças da língua portuguesa para detectar o plágio interno em fóruns educacionais. O capítulo seguinte mostra os trabalhos relacionados a este.

Tabela 3.7: Detalhamento dos artigos selecionados.

Título e Autor	Ano	Objetivo	Técnicas e Ferramentas	Nº Citações
Turning to Turnitin to Fight Plagiarism among University Students (BATANE, 2010)	2010	Análise de Ferramenta de Detecção de Plágio	Turnitin	101
The toolbox for local and global plagiarism detection (BUTAKOV; SCHERBININ, 2009)	2010	Método para detecção de plágio interno e externo	*Tokenization* Microsoft Live Search	81
Entre o plágio e a autoria: qual o papel da universidade? (SILVA, 2008)	2008	Análise do Plágio na Produção Acadêmica	Moodle	70
Automatic student plagiarism detection: future perspectives (MOZGOVOY; KAKKONEN; COSMA, 2010)	2010	Análise do Plágio na Produção Acadêmica	Indefinido	48
Text mining applied to plagiarism detection: The use of words for detecting deviations in the writing style (OBERREUTER; VELÁSQUEZ, 2013)	2013	Método para Detectar Plágio Interno	*N-grams*	40
University student online plagiarism (WANG, 2008)	2008	Análise do Plágio na Produção Acadêmica	Indefinido	34
Hermetic and web plagiarism detection systems for student essays—an evaluation of the state-of-the-art (KAKKONEN; MOZGOVOY, 2010)	2010	Análise de Ferramenta de Detecção de Plágio	Indefinido	33
			Glatt	

Continua na próxima página

Tabela 3.7 – *Continuação da página anterior*

Título e Autor	**Ano**	**Objetivo**	**Técnicas e Ferramentas**	**Nº Citações**
Design and usability testing of a learning and plagiarism avoidance tutorial system for paraphrasing and citing in English: A case study (LIU; LO; WANG, 2013)	2013	Análise de Ferramenta de Detecção de Plágio	Turnitin DWright	18
On the number of search queries required for Internet plagiarism detection (BUTAKOV; SHCHERBININ, 2009)	2009	Método para Detectar Plágio Externo	Indefinido	15
Digital plagiarism: An experimental study of the effect of instructional goals and copy-and-paste affordance (KAUFFMAN; YOUNG, 2015)	2015	Método para Diminuir o Comportamento de Plágio	WCopyFind	15
The effectiveness of plagiarism detection software as a learning tool in academic writing education (STAPPENBELT; ROWLES, 2010)	2010	Análise de Ferramenta de Detecção de Plágio	Turnitin	12
Plagiarism Detection: The Tool And The Case Study (SCHERBININ; BUTAKOV, 2008)	2008	Método para Detectar Plágio Externo	Moodle *Tokenization* Linguagem PHP	3
Desenvolvimento e Aplicação de um Método para Detecção de Indícios de Plágio (PERTILE et al., 2011)	2011	Método para Detectar Plágio Externo	Moodle *Tokenization* *Stopwords* API Google Search	3

Continua na próxima página

Tabela 3.7 – *Continuação da página anterior*

Título e Autor	Ano	Objetivo	Técnicas e Ferramentas	Nº Citações
How automated feedback through text mining changes plagiaristic behavior in online assignments (AKÇAPINAR, 2015)	2015	Método para Diminuir o Comportamento de Plágio	*Stopwords* Linguagem C#.NET RapidMiner API de Busca Bing	3
Miss Marple–Proposta de Desenvolvimento de Ferramenta de Detecção de Indícios de Plágio com base no Método DIP–Detector de Indícios de Plágio (ARENHARDT et al., 2012)	2012	Método para Detecção de Plágio Externo	Moodle *Stemming* *Stopwords* Lucene API Google Search Linguagem JAVA	2
Agente Integrado a Plataforma MLE-Moodle para Detecção Automática de Indícios de Plágio (PERTILE et al., 2010)	2010	Método para Detecção de Plágio Externo	Moodle	0
Detecção Automática de Plágio em Ambiente Educacional Virtual (ROCHA et al., 2012)	2012	Análise do Plágio na Produção Acadêmica	Moodle	0

Continua na próxima página

Tabela 3.7 – *Continuação da página anterior*

Título e Autor	**Ano**	**Objetivo**	**Técnicas e Ferramentas**	**Nº Citações**
			API de Busca Bing	
Parallel Miss Marple: Threads e Java RMI Aplicados à Verificação de Indícios de Plágio (GOMES; MEDINA, 2016)	2016	Método para Detectar Plágio Externo	Moodle *Tokenization* *Stemming* *Stopwords* API Google Search	0
Análise Comparativa Teórico-Prática entre Softwares de Detecção de Plágio (NUNES et al., 2012)	2012	Análise de Ferramenta de Detecção de Plágio	Farejador de Plágio Miss Marple Plagius Detector Viper	0
Using Computer Simulations and Games to Prevent Student Plagiarism (BRADLEY, 2015)	2015	Análise de Ferramenta de Detecção de Plágio	Indefinido	0
			Tokenization *Stopwords*	

Continua na próxima página

Tabela 3.7 – *Continuação da página anterior*

Título e Autor	**Ano**	**Objetivo**	**Técnicas e Ferramentas**	**Nº Citações**
Detecção e Avaliação de Cola em Provas Escolares Utilizando Mineração de Texto: um Estudo de Caso (CAVALCANTI et al., 2011)	2011	Método para Diminuir o Comportamento de Plágio	*Stemming* TF-IDF WordNet Linguagem JAVA RapidMiner	0

4

Trabalhos Relacionados

A base fundamental para a criação de sistemas automáticos de detecção de plágio é a criação de uma medida de similaridade que possa mensurar a relação existente entre dois textos. Calcular a similaridade entre duas sentenças é um problema desafiador, pois, as frases podem representar significados diferentes, mesmo que utilizem as mesmas palavras, ou apenas uma pequena mudança na ordem das palavras no texto pode influenciar o significado da sentença (CHOUDHARY; BHATTACHARYYA, 2002).

Existem muitas medidas para avaliar a semelhança entre sentenças escritas em inglês. Recentemente, o número de trabalhos que propõem uma medida para outra língua aumentou. Por exemplo, no SemEval 2017[1], a tarefa Similaridade Textual Semântica (STS) avaliou a capacidade dos sistemas em determinar o grau de similaridade semântica entre frases monolíngues e multilíngues em árabe, inglês e espanhol.

O *Workshop* de Avaliação de Similaridade Semântica e Inferência Textual (ASSIN) ocorrido na conferência PROPOR 2016[2] propôs as tarefas de STS e Reconhecimento de Implicação Textual (RIT) chamado em inglês de *Recognizing Textual Entailment* (RTE). Por causa disso, houve um aumento no número de artigos relacionados à similaridade entre sentenças em português. Consideramos o *workshop* ASSIN como o estado da arte para medidas de similaridade para o português, já que não foram encontrados outros trabalhos, para o português, anterior ao *workshop*. Os trabalhos listados a seguir foram alguns dos que foram apresentados no *workshop* ASSIN 2016.

FREIRE; PINHEIRO; FEITOSA (2016) propõem um framework chamado FlexSTS, o qual define diversos componentes a serem selecionados para o desenvolvimento de sistemas de STS. O *framework* FlexSTS foi instanciado em três sistemas: STS_MachineLearning, STS_HAL e STS_WNET_HAL. Em cada sistema foram realizadas algumas configurações com recursos de PLN. Essas configurações foram divididas em três etapas: (1) Análise Morfológica/Sintática, (2) Similaridade Semântica de Palavras e (3) Similaridade Semântica textual. O sistema STS_MachineLearning usou POS Tagger e Lematização na primeira etapa. Na segunda etapa

[1] http://alt.qcri.org/semeval2017/task1/
[2] http://propor2016.di.fc.ul.pt

foi utilizada aprendizagem de máquina com dois atributos: similaridade entre palavras pelo coeficiente DICE e similaridade entre palavras pela WordNet (um repositório de palavras onde substantivos, verbos, advérbios e adjetivos são organizados por uma variedade de relações semânticas). E na terceira etapa foi realizado um treinamento do algoritmo *ridge regression model* com o *dataset* disponibilizado no ASSIN. O sistema STS_HAL também utiliza na primeira etapa POS Tagger e Lematização. Na segunda etapa foram utilizados um algoritmo de alinhamento entre palavras e o modelo HAL+SVD, onde é usada a variação da técnica *Latent Semantic Analysis* (LSA) chamada *Hyperspace Analog to Language* (HAL) que constrói a matriz de coocorrência termo-termo e também foi aplicado a estratégia de *Singular Value Decomposition* (SVD), onde foram selecionados os k=300 maiores valores singulares. Na terceira etapa foram utilizados 3 algoritmos matemáticos de STS. O sistema STS_WORDNET_HAL utiliza a mesma configuração do sistema STS_HAL, com uma única diferença na segunda etapa, onde além do algoritmo de alinhamento entre palavras e o modelo HAL+SVD, foi utilizado a base de conhecimento WordNet. Um dos problemas deste trabalho foi a utilização da WordNet em inglês, onde eles traduziam as palavras do português para o inglês para poder utilizá-las. Este trabalho foi o quarto colocado na competição ASSIN.

ALVES; RODRIGUES; OLIVEIRA (2016) apresentam duas abordagens distintas à tarefa de STS para a língua portuguesa na competição ASSIN: uma primeira abordagem, apelidada de Reciclagem, baseada exclusivamente em heurísticas sob redes semânticas; e uma segunda abordagem, apelidada de ASAPP, baseada em aprendizagem automática supervisionada. Os autores extraíram características léxicas, sintáticas e semânticas das sentenças. Entre as características lexicais estão: Contagem de palavras e expressões consideradas negativas presentes em cada frase, contagem dos átomos em comum nas duas frases e contagem dos lemas em comum nas duas frases. Dentre as características morfo-sintáticas, foram contabilizadas as contagens de grupos nominais, verbais e preposicionais em cada uma das frases de cada par, e calculado o valor absoluto da diferença para cada tipo de grupo. Além disso, foi utilizado um algoritmo de *Named Entity Recognition* (NER). Foi calculada a similaridade semântica de cada par de frases, com base em heurísticas aplicadas sobre as redes semânticas utilizadas. As heurísticas aplicadas podem agrupar-se em três tipos: (i) semelhança entre as vizinhanças das palavras nas redes; (ii) baseadas na estrutura das redes de palavras; (iii) baseadas na presença e pertença em *synsets* difusos. Para RIT foram utilizados os seguintes algoritmos: Voto por maioria de 3 classificadores, Voto por maioria de 5 classificadores e Redução Automática de características. Para Similaridade Textual foram utilizados os algoritmos: Regressão Aditiva por *Boosting*, Esquema Múltiplo de Seleção e Processo Gaussiano Simples. Essa equipe obteve o segundo lugar na competição ASSIN. A abordagem Reciclagem não apresentou bons resultados e foi a quinta colocada na competição. Já a abordagem ASAPP foi a que obteve o segundo lugar na competição.

HARTMANN (2016) obteve o primeiro lugar na competição apresentando uma abordagem com *feature* clássica da classe *bag-of-words*, a TF-IDF; e uma *feature* emergente, capturada por meio de *word embeddings*. O autor utiliza TF-IDF para relacionar sentenças que comparti-

lham as mesmas palavras e *word embeddings* para capturar a sintaxe e semântica de uma palavra. Além disso, o autor utiliza o sistema word2vec[3] para a modelagem das *embeddings* das palavras. A *embedding* de uma palavra representa o contexto no qual ela ocorre, capturando relações sintáticas e semânticas. O uso das *embeddings* como *feature* é dado pela similaridade do cosseno entre as *embeddings* dos pares de sentenças. O valor da similaridade entre os dois vetores de *embeddings* é utilizado como uma *feature* para o sistema de regressão. O uso do TF-IDF como *feature* é dado pela distância do cosseno entre os vetores TF-IDF dos pares de sentenças. Esse valor foi utilizado como uma *feature* para o sistema de regressão.

BARBOSA et al. (2016) avaliaram métodos baseados no uso de vetores semânticos de palavras. Os autores também utilizaram o sistema word2vec para obter os vetores semânticos das palavras, a distância Euclidiana e do cosseno para obter um valor de similaridade entre as sentenças. Além disso, os autores utilizaram o classificador SVM e o *Support Vector Regression* (SVR), seu correspondente método para problemas de regressão. Esta equipe obteve o terceiro lugar na competição ASSIN.

No trabalho de SILVA et al. (2017) foi avaliada uma abordagem híbrida para similaridade semântica entre frases curtas. Os autores utilizam um conjunto de recursos linguísticos e probabilísticos, entre eles estão: Modelos de Espaço Vetorial, relações semânticas de aspectos como hiponímia, hiperonímia, sinonímia e autonímia, *Portuguese Unified Lexical Ontology* (PULO) (SIMOES; GUINOVART, 2014), Thesaurus para o português do Brasil (TeP) (MAZIERO et al., 2008), além das técnicas de *word embeddings*, TF-IDF e *Principal Component Analysis* (PCA). Os resultados obtidos se aproximaram do estado da arte, entretanto, os autores afirmam que a utilização de relações de hiperonímia e hiponímia, por si só, não apresentam informações suficientes para uma melhor avaliação de similaridade. Os autores ainda afirmam que a utilização destas relações como atributos auxiliaram na generalização dos termos das sentenças e consequentemente trouxe melhores resultados para as técnicas TF-IDF e *word embeddings*.

FEITOSA; PINHEIRO (2017) realizaram uma análise de medidas de similaridade semântica na tarefa de RIT. Os autores implementaram a solução do vencedor do *workshop* ASSIN para o português europeu, tendo como diferencial o uso das medidas da base léxico-semântica Wordnet. Os autores definiram 3 cenários para análise:

- Cenário 1 - Execução com as *features* sintáticas usadas em FIALHO et al. (2016); ´
- Cenário 2 - Execução com todas as *features* sintáticas e todas as *features* semânticas calculadas para toda variação de métrica de similaridade entre palavras;
- Cenario 3 - Execução com todas as sintáticas e uma *feature* semântica por vez.

Entretanto, a hipótese levantada pelos autores que a utilização de medidas semânticas na abordagem de FIALHO et al. (2016) aumentaria a performance não foi comprovada em nenhum

[3]https://code.google.com/archive/p/word2vec/

dos cenários propostos. O melhor resultado obtido pelos autores foi com a utilização da medida Path, mesmo assim esse resultado não ultrapassou o resultado obtido por FIALHO et al. (2016) no *workshop* ASSIN.

Além dos trabalhos para o Português, recentemente foi apresentada uma representação de sentença em três camadas para calcular a similaridade entre duas sentenças escritas em inglês (FERREIRA et al., 2016). Na camada chamada *Shallow metric*, os autores propõem uma matriz de similaridades para medir a similaridade entre as sentenças. O primeiro passo deste método é calcular as similaridades entre todas as palavras das sentenças e montar uma matriz, onde as linhas são as palavras de uma sentença e as colunas são as palavras da outra sentença. O segundo passo é identificar o maior valor de similaridade presente na matriz. No terceiro passo é removido o maior valor, ou seja, seria removida a linha e a coluna desse valor. O segundo e o terceiro passo são repetidos até que não existam mais valores de similaridades a serem calculados. O último passo é calcular a média entre os maiores valores de similaridades obtidos.

ZHAO; ZHU; LAN (2014) obtiveram a primeira colocação para a tarefa de similaridade sentencial da língua inglesa. Os autores extraíram algumas *features* das sentenças para obter a similaridade. Entre as *features* utilizadas, podemos citar: tamanho das sentenças, similaridade superficial (distância do cosseno), similaridade semântica, *ngrams* com base em córpus de referência, entre outras.

A Tabela 4.1 mostra as principais técnicas utilizadas nos trabalhos listados acima para a língua portuguesa.

Na literatura também foram encontrados trabalhos com objetivos de criar métodos para detecção de plágio externo, ou seja, quando o aluno copia de fontes externas ao AVA.

PERTILE et al. (2010) apresentam uma modelagem de um agente que realiza buscas na internet a fim de encontrar documentos similares ao documento postado pelo aluno no AVA. O agente atua como um detector de indícios de plágio em tarefas de produção textual submetidas pelos alunos em cursos ministrados no ambiente de aprendizagem virtual Moodle. O funcionamento do sistema obedece os seguintes passos: (i) Os trabalhos submetidos pelos alunos no módulo de envio de tarefas do Moodle serão armazenados em um banco de dados, onde a cada trabalho submetido o Agente entra em ação; (ii) O agente detector fará a busca em motores de busca na web por parágrafos similares ao do documento submetido; (iii) O professor será notificado via e-mail de tais indícios de plágio. Em seguida só serão criados relatórios dos documentos que foram notificados ao professor, com a porcentagem de originalidade do documento, complementado com o endereço virtual das fontes encontradas. Além disso, a parte do texto analisado que for considerado como indício de plágio será destacado na cor vermelha. O relatório é armazenado ao Banco de Dados para posterior análise do professor.

ARENHARDT et al. (2012) apresentam um aprimoramento do método detector de indícios de plágio. Os autores propõem o desenvolvimento de uma ferramenta de detecção de indícios de plágio textual, em arquivos com extensões .doc, .docx, .pdf e .rtf , utilizando técnicas de *stemming* (extração do radical das palavras e armazenamento em uma lista) que possibilita a

Tabela 4.1: Principais técnicas utilizadas pelos trabalhos citados.

Autor	Técnicas
(FREIRE; PINHEIRO; FEITOSA, 2016)	Coeficiente DICE WordNet HAL SVD Algoritmo *rigde regression model*
(ALVES; RODRIGUES; OLIVEIRA, 2016)	NER WordNet Métricas de similaridade, distância e contagens. PULO TeP.
(HARTMANN, 2016)	*Word Embeddings* TF-IDF
(BARBOSA et al., 2016)	*Word Embeddings* Métricas de distância.
(FIALHO et al., 2016)	*Soft* TF-IDF Métricas de similaridade. Sobreposição de *ngrams*.
(SILVA et al., 2017)	PCA *Word Embeddings* TF-IDF PULO TeP.
(FEITOSA; PINHEIRO, 2017)	WordNet Métricas de distância.

comparação de palavras sinônimas. Outra funcionalidade que os autores inseriram é a análise de referências cruzadas, ou seja, busca na Internet de documentos suspeitos/similares em relação ao original e com base na similaridade encontrada, então se fará o download dos documentos, que serão armazenados em diretório e este por fim formará um repositório de documentos suspeitos.

Na literatura não foi encontrado nenhum trabalho, para a língua portuguesa, cujo objetivo seja o plágio em fóruns educacionais. Desta forma, este trabalho de dissertação apresenta os seguintes diferenciais:

- Criação de uma medida de similaridade entre sentenças em Português;
- Criação de um Fórum Web que integre a medida de similaridade;
- Detecção do Plágio Interno no Fórum Educacional.

5 Proposta

Este trabalho tem por objetivo propor uma nova medida de similaridade entre sentenças em português para detecção de plágio interno em fóruns educacionais. As seções seguintes descrevem a medida de similaridade proposta, bem como a sua aplicação em um fórum educacional web desenvolvido.

5.1 Medida de Similaridade

A medida de similaridade proposta extrai 4 características das sentenças utilizando as medidas TF-IDF, Word2Vec e o método proposto por FERREIRA et al. (2016), que utiliza uma matriz com similaridades entre as palavras para calcular a similaridade entre as sentenças. As 4 características são: TF-IDF, Matriz de similaridades com word2vec, Matriz de similaridades binária e o tamanho da sentença. O valor de similaridade final é obtido pela combinação das 4 características, onde cada característica vai possuir um peso diferente. Cada característica extraída é apresentada a seguir.

5.1.1 TF-IDF

TF-IDF é uma abordagem clássica da área de PLN. Esta característica também foi utilizada por HARTMANN (2016), que obteve o primeiro lugar na competição ASSIN combinando o TF-IDF com *word embeddings*. TF-IDF combina a frequência dos termos (TF) e a relevância do termo para uma coleção (IDF). Desta forma, o esquema TF-IDF para similaridade entre sentenças é calculado conforme as Equações 5.1, 5.2 e 5.3.

$$TF = \left(\frac{\text{número de vezes em que um termo aparece em uma determinada sentença}}{\text{número total de termos presentes nas sentenças}}\right) \quad (5.1)$$

$$IDF = \log\left(\frac{\text{número total de sentenças}}{\text{quantidade de sentenças que apresentam determinado termo}}\right) \quad (5.2)$$

$$TF - IDF = TF \times IDF \tag{5.3}$$

Ao final é obtida uma matriz com tamanho sentenças x palavras e o valor TF-IDF de cada palavra para cada sentença. A medida TF-IDF pondera as palavras com base nas suas frequências dentro de um conjunto de dados. Termos que apresentam pouca frequência podem ser de grande importância para o conjunto de dados ao invés de termos muitos frequentes. A similaridade entre as sentenças é calculada pela distância do cosseno entre os vetores TF-IDF de cada sentença. O valor de similaridade obtido entre duas sentenças é usado como característica. A Tabela 5.1 mostra um exemplo de uma matriz TF-IDF.

Tabela 5.1: Exemplo de Matriz TF-IDF.

	w_1	w_2	w_3	...	w_n
$Sentenca_1$	2.6	4.1	3.2	...	3.7
$Sentenca_2$	7.2	0	2.6	...	2.1
$Sentenca_3$	3.5	2.4	0	...	4.2
...	...	...	...	...	...
$Sentenca_m$	0	1.2	2.9	...	5.1

Para calcular a similaridade entre as sentenças 1 e 2 da Tabela 5.1, por exemplo, seria necessário calcular a distância do cosseno entre os vetores $[2.6, 4.1, 3.2, ..., 3.7]$ e $[7.2, 0, 2.6, ..., 2.1]$. Segundo HARTMANN (2016) a modelagem TF-IDF sofre com a esparsidade dos dados, por isso é necessário utilizar apenas os *stems* das palavras para a redução da matriz TF-IDF. Além disso, HARTMANN (2016) também afirma que frases curtas não necessariamente apresentam as mesmas palavras, por isso é necessário expandir o vocabulário das sentenças, buscando sinônimos para cada palavra no TeP (MAZIERO et al., 2008). Entretanto, HARTMANN (2016) limitou a expansão de sinônimos para palavras que possuem até 2 sinônimos no TeP, já que a expansão de sinônimos para todas as palavras faz com que os vetores TF-IDF das sentenças se tornem muito similares.

5.1.2 Matriz de Similaridades com Word2Vec

Nesta característica foi utilizado o word2vec para calcular a similaridade entre as palavras em conjunto com um método que usa uma matriz de similaridades proposta por (FERREIRA et al., 2016). A ideia é que a matriz de similaridades leva em consideração a ordem em que as palavras ocorrem, como defendido por FERREIRA et al. (2016), e o word2vec leva em consideração a semântica das palavras.

O sistema word2vec modela as *embeddings* das palavras, obtendo as representações vetoriais de cada palavra. Com a representação vetorial é possível calcular a similaridade entre as palavras. O treinamento do word2vec foi realizado usando a implementação original[1] sobre a

[1] `http://code.google.com/p/word2vec`

base da wikipedia [2]. O modelo foi obtido com os seguintes parâmetros básicos:

- dimensão: 250;
- janela: 10;
- frequência mínima de palavras: 5;
- número de iterações: 10.

Para chegar aos parâmetros definidos acima, foram realizados alguns testes, sendo estes parâmetros os que obtiveram resultados mais precisos. Contudo, não foi realizada nenhuma avaliação em relação a qualidade do modelo que foi gerado. Também não foi encontrado nenhum modelo disponível para utilização na literatura. Para obter o modelo treinado, foram necessárias cerca de 36 horas de treinamento utilizando uma máquina com 8GB de RAM e Processador Intel Core i7.

O Código B.1 presente no Apêndice B mostra como é realizado o treinamento do word2vec para um *córpus* de treinamento. Após o treinamento, o word2vec salva um arquivo com o modelo treinado. O modelo possui as representações vetoriais de cada palavra do *córpus* de treinamento. Dessa forma, é possível calcular a similaridade entre as palavras utilizando a distância do cosseno. O word2vec já possui um método chamado *similarity()* que calcula a similaridade entre as palavras. O Código B.2 em Java presente no Apêndice B mostra um exemplo da utilização deste método. Inicialmente é necessário ler o modelo e salvá-lo em uma variável do tipo Word2Vec. Depois utiliza-se o método *similarity()* para calcular a similaridade semântica entre duas palavras.

O método proposto por FERREIRA et al. (2016) utiliza uma matriz de similaridades entre palavras para obter a similaridade entre as sentenças. A similaridade entre as palavras é obtida utilizando o word2vec. A seguir são mostrados os passos para calcular a similaridade entre duas sentenças utilizando este método.

O primeiro passo é calcular a similaridade entre as palavras de duas sentenças. Seja $A = \{a_1, a_2, ..., a_n\}$ e $B = \{b_1, b_2, ..., b_m\}$ duas sentenças, onde a_n é uma palavra da sentença A, b_m é uma palavra da sentença B, n é o número de palavras da sentença A e m é o número de palavras da sentença B. Então é calculado o valor de similaridade entre cada palavra da sentença A com cada palavra da sentença B utilizando o sistema word2vec, que a partir do modelo criado calcula a similaridade entre duas palavras.

Por exemplo, sejam duas sentenças A e B, a sentença A com 5 palavras e a sentença B com 6 palavras. São calculadas as similaridade de todas as palavras da sentença A com todas as palavras da sentença B. A Tabela 5.2 mostra as similaridades obtidas. A sentença A possui as palavras a_1, a_2, a_3, a_4, a_5 e a sentença B possui as palavras $b_1, b_2, b_3, b_4, b_5, b_6$.

[2] https://dumps.wikimedia.org/ptwiki/20160920/

Tabela 5.2: Exemplo da matriz de similaridades.

	a_1	a_2	a_3	a_4	a_5
b_1	0.3	0.2	0.56	0.88	0.25
b_2	0.12	0.5	0.31	0.22	0.87
b_3	0.56	0.23	0.5	0.28	0.6
b_4	0.7	0.62	0.6	0.38	0.12
b_5	0.84	0.21	0.54	0.78	0.29
b_6	0.4	0.35	0.47	1.0	0.23

O segundo passo é identificar a maior similaridade presente na matriz. No exemplo anterior, a maior similaridade obtida foi entre as palavras a_4 e b_6 com valor 1.0 (Tabela 5.3), ou seja, as duas palavras são iguais.

Tabela 5.3: Maior valor presente na matriz de similaridades.

	a_1	a_2	a_3	a_4	a_5
b_1	0.3	0.2	0.56	0.88	0.25
b_2	0.12	0.5	0.31	0.22	0.87
b_3	0.56	0.23	0.5	0.28	0.6
b_4	0.7	0.62	0.6	0.38	0.12
b_5	0.84	0.21	0.54	0.78	0.29
b_6	0.4	0.35	0.47	**1.0**	0.23

O terceiro passo é remover a linha e a coluna onde se encontra a maior similaridade. No exemplo anterior seriam removidas a coluna da palavra a_4 e a linha da palavra b_6, como mostra a Tabela 5.4. Como a_4 é a mais similar com b_6 resta saber quais as palavras mais semelhantes entre as outras palavras das sentenças.

Tabela 5.4: Matriz de similaridades com remoção da linha b_6 e coluna a_4.

	a_1	a_2	a_3	a_4	a_5
b_1	0.3	0.2	0.56	0.88	0.25
b_2	0.12	0.5	0.31	0.22	0.87
b_3	0.56	0.23	0.5	0.28	0.6
b_4	0.7	0.62	0.6	0.38	0.12
b_5	0.84	0.21	0.54	0.78	0.29
b_6	0.4	0.35	0.47	1.0	0.23

Com a remoção da linha a_4 e b_6 é obtida a nova matriz como mostra a Tabela 5.5. O segundo e o terceiro passo são repetidos, identificando o maior valor da matriz (Tabela 5.6) e removendo a linha e a coluna de onde se encontra o maior valor (Tabela 5.7).

Então estes passos são repetidos até que sejam removidas ou todas as colunas ou todas as linhas da matriz, como mostram as Matrizes 5.8, 5.9 e 5.10.

Tabela 5.5: Nova matriz de similaridades com remoção da linha b_6 e coluna a_4.

	a_1	a_2	a_3	a_5
b_1	0.3	0.2	0.56	0.25
b_2	0.12	0.5	0.31	0.87
b_3	0.56	0.23	0.5	0.6
b_4	0.7	0.62	0.6	0.12
b_5	0.84	0.21	0.54	0.29

Tabela 5.6: Maior valor presente na matriz.

	a_1	a_2	a_3	a_5
b_1	0.3	0.2	0.56	0.25
b_2	0.12	0.5	0.31	**0.87**
b_3	0.56	0.23	0.5	0.6
b_4	0.7	0.62	0.6	0.12
b_5	0.84	0.21	0.54	0.29

Tabela 5.7: Remoção da linha b_2 e coluna a_5.

	a_1	a_2	a_3	a_5
b_1	0.3	0.2	0.56	0.25
b_2	0.12	0.5	0.31	0.87
b_3	0.56	0.23	0.5	0.6
b_4	0.7	0.62	0.6	0.12
b_5	0.84	0.21	0.54	0.29

Tabela 5.8: Remoção da linha b_5 e coluna a_1.

	a_1	a_2	a_3
b_1	0.3	0.2	0.56
b_3	0.56	0.23	0.5
b_4	0.7	0.62	0.6
b_5	0.84	0.21	0.54

Tabela 5.9: Remoção da linha b_4 e coluna a_2.

	a_2	a_3
b_1	0.2	0.56
b_3	0.23	0.5
b_4	0.62	0.6

O último passo é calcular a média entre os maiores valores de similaridades obtidos entre as sentenças, como mostra a Equação 5.4.

Tabela 5.10: Remoção da linha b_1 e coluna a_3.

	a_3
b_1	0.56
b_3	0.5

$$Similaridade(A,B) = \frac{\sum_{i=1}^{n} MaxSimilaridades(A,B)}{n} \tag{5.4}$$

Onde $MaxSimilaridades(A,B)$ possui as maiores similaridades obtidas pelos passos anteriores. O valor de similaridade entre a sentença A e a sentença B será a média das maiores similaridades presentes na matriz. Por exemplo, a similaridade entre as sentenças A e B apresentadas nos exemplos anteriores (Tabela 5.2 - Tabela 5.10) seria obtida pela Equação 5.5. O Algoritmo 1 mostra os passos para calcular esta métrica.

$$Similaridade(A,B) = \frac{1.0+0.87+0.84+0.62+0.56}{5} = \mathbf{0.654124} \tag{5.5}$$

Algoritmo 1: SIMILARIDADE ENTRE DUAS SENTENÇAS.

Entrada: Duas sentenças A e B
Saída: Similaridade entre as sentenças A e B

```
início
    para cada palavra a ∈ A faça
        para cada palavra b ∈ B faça
            Matriz[a,b] = similarity(a,b)
        fim
    fim
    para cada palavra p ∈ QuantidadePalavrasMenorSentenca faça
        vetorSimilaridades[p] = maxValor(Matriz)
        Matriz = removeLinhaMatriz(p)
        Matriz = removeColunaMatriz(p)
    fim
    ValorFinal = Media(vetorSimilaridades)
fim
retorna ValorFinal
```

Onde inicialmente a matriz de similaridades é preenchida calculando a similaridade de cada palavra da sentença A com cada palavra da sentença B através do método $similarity(a,b)$ que usa o word2vec para obter a similaridade entre as palavras. Depois três passos são repetidos até atingir a quantidade de palavras da menor sentença. O primeiro passo obtém o maior valor de similaridade da matriz e salva no vetor $vetorSimilaridades$. O segundo passo remove a linha onde se encontra o maior valor e o terceiro passo remove a coluna onde se encontra o maior valor. Depois é obtida a média aritmética dos maiores valores que foram salvos no vetor e então é obtido o valor final de similaridade entre as duas sentenças.

É importante enfatizar que o método proposto por FERREIRA et al. (2016) leva em consideração a ordem em que as palavras ocorrem. Por exemplo, se duas frases possuem as mesmas palavras mas em posições diferentes, o método da matriz conseguirá identificar essas palavras, pois ele sempre obtém os maiores valores da matriz.

5.1.3 Matriz de Similaridades Binária

A característica anterior obtém valores de similaridades altos. Ou seja, mesmo que as sentenças apresentem um valor de similaridade baixo, o método da matriz obtém um valor de similaridade alto. Dessa forma foi proposta uma abordagem binária para atingir os pares de sentenças que apresentem pouca semelhança. Este método também utiliza a matriz de similaridades proposta por FERREIRA et al. (2016), a diferença é que os valores de similaridade entre as palavras são obtidos pela Equação 5.6.

$$similaridade(a,b) = \begin{cases} 1, & \text{se as palavras são iguais,} \\ 0, & \text{se as palavras são diferentes.} \end{cases} \tag{5.6}$$

A Tabela 5.11 mostra um exemplo de uma matriz binária. Da mesma forma que a característica anterior, ao final é obtida a média das maiores similaridades entre as palavras para obter a similaridade entre as sentenças (Equação 5.4). Este valor de similaridade é usado como característica.

Tabela 5.11: Exemplo de matriz binária.

	a_1	a_2	a_3	a_4	a_5	a_6
b_1	1	0	0	1	0	0
b_2	0	0	0	0	0	0
b_3	0	1	0	0	0	0
b_4	0	0	0	0	0	1
b_5	0	0	0	0	0	0
b_6	0	0	0	0	0	0

5.1.4 Tamanho da Sentença

A última característica extraída, também utilizada por ZHAO; ZHU; LAN (2014) e BJERVA et al. (2014), foi o tamanho das sentenças. Para obter um valor que represente o tamanho das sentenças, é dividido o número de palavras da menor sentença pelo número de palavras da maior sentença, como mostra a Equação 5.7. O valor obtido é usado como característica.

$$TamanhoSentenca = \frac{\text{número de palavras da menor sentença}}{\text{número de palavras da maior sentença}} \tag{5.7}$$

5.1.5 Valor de Similaridade Final

O valor de similaridade final entre as sentenças é obtido utilizando o algoritmo de regressão linear presente no WEKA. Este algoritmo recebe como entrada as 4 características e tem como saída uma equação linear que combina as características. A Equação 5.8 mostra como o valor final é obtido.

$$\begin{aligned} ValorFinal \quad &= x_1 + (y_1 \times TF - IDF) + (y_2 \times MatrizWord2Vec) + \\ &(y_3 \times MatrizBinaria) + (y_4 \times TamanhoSentenca) \end{aligned} \tag{5.8}$$

As variáveis y_1, y_2, y_3 e y_4 são os coeficientes de regressão, onde cada um atribuirá um peso para as características e x_1 é o termo do erro aleatório. A regressão linear obtém os melhores coeficientes para cada característica com base no valor real (valor final) que queremos obter. É importante salientar que o algoritmo de regressão linear é aplicado em abordagens com aprendizado supervisionado, ou seja, quando conhecemos o valor real de similaridade entre os dois textos.

O Algoritmo 2 mostra como é obtida a similaridade entre dois textos. Para cada característica apresentada anteriormente é obtido um valor de similaridade ente 0 e 1. Ao final é obtido um valor final através do método *CombinaFeatures* que combina as características extraídas através de um algoritmo de regressão linear (Equação 5.8).

Algoritmo 2: SIMILARIDADE ENTRE DOIS TEXTOS

Entrada: t_1 e t_2
Saída: Similaridade entre t_1 e t_2
1 **início**
2 $\quad feature1 = CalculaTFIDF(t_1, t_2)$
3 $\quad feature2 = CalculaMatrizWord2Vec(t_1, t_2)$
4 $\quad feature3 = CalculaMatrizBinaria(t_1, t_2)$
5 $\quad feature4 = CalculaTamSentenca(t_1, t_2)$
6 $\quad ValorFinal = CombinaFeatures(feature1, feature2, feature3, feature4)$
7 **fim**
8 **retorna** *ValorFinal*

A Tabela 5.12 mostra a comparação das técnicas utilizadas pelo método proposto com o estado da arte.

O método proposto utiliza duas técnicas muito utilizadas em outros trabalhos, as *Word Embeddings* e o TF-IDF. Um dos motivos para utilizar *Word Embeddings* é que a *embedding* de uma palavra representa o contexto no qual ela ocorre, capturando relações sintáticas e semânticas. Além de apresentar bons resultados em alguns trabalhos (BARBOSA et al., 2016; HARTMANN, 2016). O TF-IDF é uma abordagem clássica de PLN que leva em consideração a frequência em que as palavras ocorrem e também apresentou bons resultados nos trabalhos de FIALHO

Tabela 5.12: Comparação do método proposto com o estado da arte.

Método	Técnicas
(FREIRE; PINHEIRO; FEITOSA, 2016)	Coeficiente DICE WordNet HAL SVD Algoritmo *rigde regression model*
(ALVES; RODRIGUES; OLIVEIRA, 2016)	NER WordNet Métricas de similaridade, distância e contagens. PULO TeP.
(HARTMANN, 2016)	*Word Embeddings* TF-IDF
(BARBOSA et al., 2016)	*Word Embeddings* Métricas de distância.
(FIALHO et al., 2016)	*Soft* TF-IDF Métricas de similaridade. Sobreposição de *ngrams*.
(SILVA et al., 2017)	PCA *Word Embeddings* TF-IDF PULO TeP.
(FEITOSA; PINHEIRO, 2017)	WordNet Métricas de distância.
Método Proposto	TF-IDF *Word Embeddings* Matriz de similaridades Matriz Binária Tamanho da sentença

et al. (2016) e HARTMANN (2016). A proposta da utilização da matriz de similaridades em conjunto com as *Word Embeddings* é que a matriz leva em consideração a ordem em que as palavras ocorrem e as *embeddings* das palavras levam em consideração o contexto na qual elas ocorrem. A Matriz Binária é uma abordagem léxica e tem como objetivo ponderar frases que apresentem as mesmas palavras. Já o tamanho das sentenças é uma medida que pondera frases que possuem a mesma quantidade de palavras.

6

Experimentos

Neste capítulo é descrito a metodologia utilizada, a validação da medida de similaridade proposta, um estudo de caso realizado em um Fórum Web e por fim a discussão.

6.1 Base de dados

A base de dados escolhida foi a base do workshop ASSIN[1]. Esta base foi criada com textos de notícias retirados do *Google News*[2], onde foram selecionadas sentenças similares de documentos diferentes utilizando modelos de espaço vetorial. Os pares de sentenças similares foram anotados por quatro juízes humanos. A base possui 10.000 (dez mil) pares de sentenças, 5.000 (cinco mil) em Português Brasileiro e 5.000 (cinco mil) em Português Europeu. Dos 5.000 pares de sentenças, 3.000 (três mil) são para treino e 2.000 (dois mil) são para teste. Cada par de sentenças possui um valor de similaridade semântica anotado que varia de 1 à 5. Segundo FONSECA et al. (2016) esse tipo de medida é inerentemente subjetivo, mas cada valor definido seguiu as diretrizes gerais para pontuação de cada juiz. Além disso, cada par de sentença possui uma classe a qual pertence:

- *Entaiment*: quando a primeira sentença implica a segunda;
- *Paraphrase*: quando há uma implicação mútua ou paráfrase;
- *None*: quando não há implicação textual.

A medida será avaliada para as tarefas de STS e RIT. A Figura 6.1 mostra um exemplo de pares de sentenças dessa base de dados que está estruturada em um arquivo *eXtensible Markup Language* (XML).

[1] `http://propor2016.di.fc.ul.pt/?page_id=381`
[2] `https://news.google.com/`

```
<pair entailment="None" id="69" similarity="1.5">
    <t>Mas a gente vive o futebol e sabe o porquê disso.</t>
    <h>Sabemos da qualidade que tem o Brasil, o berço do futebol.</h>
</pair>
<pair entailment="None" id="70" similarity="3.5">
    <t>O Brasil acabou superado pela Croácia por 3 a 1 na repescagem do Grupo Mundial da Copa Davis.</t>
    <h>O Brasil começou bem o confronto com a Croácia pela repescagem da Copa Davis.</h>
</pair>
<pair entailment="None" id="71" similarity="1.75">
    <t>No Catar, Raúl - já em fim de carreira - marcou 16 gols em 61 partidas.</t>
    <h>Foram 741 partidas e 323 gols durante esse período.</h>
</pair>
<pair entailment="Entailment" id="72" similarity="4.5">
    <t>Nós realmente iremos ver Peter Parker no ensino médio e pretendemos mostrar mais esse lado dele.</t>
    <h>Nós realmente vamos ver Peter Parker na escola e queremos mostrar mais esse lado dele.</h>
</pair>
<pair entailment="Paraphrase" id="73" similarity="5.0">
    <t>O goleiro Marcelo Grohe volta após servir à seleção brasileira nos amistosos com Costa Rica e Estados Unidos.</t>
    <h>Marcelo Grohe retorna após ser o goleiro titular do Brasil em amistosos contra os Estados Unidos e Costa Rica.</h>
</pair>
```

Figura 6.1: Exemplo de pares de sentenças da base de dados.

6.2 Medidas de Avaliação

As subseções seguintes detalham as medidas de avaliação usadas para avaliar a medida de similaridade proposta para a tarefa de STS e RIT, respectivamente.

6.2.1 Medidas para STS

Para a análise dos resultados na avaliação de similaridade semântica foi utilizado o Coeficiente de Correlação de Pearson (CP) e o Erro Quadrático Médio (EQM) para medir o grau de correlação entre as similaridades obtidas e as similaridades presentes na base de dados. O CP varia de -1 a 1. O sinal indica direção positiva ou negativa do relacionamento e o valor sugere a força da relação entre as variáveis. Uma correlação de valor zero indica que não há relação linear entre as variáveis. Quanto mais próximo de 1 for o coeficiente, maior é o grau de dependência estatística entre as variáveis (FIGUEIREDO FILHO; JUNIOR, 2010). O CP (ρ) é calculado pela Equação 6.1.

$$\rho = \frac{\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^{n}(x_i - \bar{x})^2} \cdot \sqrt{\sum_{i=1}^{n}(y_i - \bar{y})^2}} = \frac{\text{cov}(X,Y)}{\sqrt{\text{var}(X) \cdot \text{var}(Y)}} \tag{6.1}$$

Onde X é o conjunto de valores de similaridade obtidos e Y é o conjunto de valores de similaridade desejados. O EQM é definido como sendo a soma das diferenças entre o valor estimado e o valor real dos dados, ponderados pelo número de termos (SENGIJPTA, 1995). O EQM é determinado somando os erros de previsão ao quadrado e dividindo pelo número de erros usados no cálculo. O EQM é calculado pela Equação 6.2.

$$\text{EQM} = \frac{\sum_{t=1}^{n} e_t^2}{n} \tag{6.2}$$

6.2.2 Medidas para RIT

Para a avaliação da tarefa de Reconhecimento de Implicação Textual (RIT), foram utilizadas as medidas F-*measure* e acurácia. A Acurácia mede a quantidade de instâncias que foram classificadas corretamente pelo classificador. A F-*measure* é uma combinação das medidas *Precision* e *Recall*. Estas medidas são calculadas da seguinte forma:

$$precision = \frac{VP}{(VP+FP)}, \tag{6.3}$$

$$recall = \frac{VP}{(VP+FN)}, \tag{6.4}$$

$$f\text{-}measure = 2 \times \left(\frac{precision \times recall}{precision + recall} \right). \tag{6.5}$$

Onde para calcular a *precision* e *recall* é utilizada como auxílio a matriz de confusão conhecida por tabelar os resultados obtidos da seguinte maneira: Verdadeiro Positivo (VP) – número de elementos positivos classificados como positivos; Verdadeiro Negativo (VN) – número de elementos positivos classificados como falsos; Falso Positivo (FP) – número de elementos falsos classificados como positivos e; Falso Negativo (FN) – número de elementos falsos classificados como falsos. Diante disso, CHEN; KUO; MERKEL (2004) definem a *f-measure* como uma medida ponderada da *precision* e *recall*. A Figura 6.2 mostra a matriz de confusão.

		Valor Verdadeiro *(Confirmado por Análise)*	
		Positivo	*Negativo*
Valor Previsto *(Predito pelo Teste)*	*Positivo*	**VP** Verdadeiro Positivo	**FP** Falso Positivo
	Negativo	**FN** Falso Negativo	**VN** Verdadeiro Negativo

Figura 6.2: Matriz de confusão.

6.3 Experimentos

Inicialmente foram realizados vários experimentos aplicando diferentes pré-processamentos para cada característica extraída. As técnicas foram aplicadas da seguinte forma:

- Nenhuma técnica;

- Remoção de *stopwords*;
- Lematização;
- Lematização com remoção de *stopwords*;
- *Stemming*;
- *Stemming* com remoção de *stopwords*.

Não foi utilizada a combinação: lematização com *stemming*, pois, lematização e *stemming* são técnicas que possuem objetivos semelhantes. Cada técnica foi aplicada nas sentenças da base de treino da competição ASSIN para cada característica extraída. A Tabela 6.1 mostra os resultados obtidos para o português brasileiro (PTBR) e para o português europeu (PTPT) utilizando a métrica TF-IDF.

Tabela 6.1: Resultados obtidos para a característica TF-IDF na base de treino.

	PTBR		**PTPT**	
Técnica	**CP**	**EQM**	**CP**	**EQM**
Nenhuma técnica	0,65	0,79	0,64	0,82
Remoção de *stopwords*	0,63	0,88	0,62	0,90
Lematização	0,64	0,87	0,59	1,17
Lematização com remoção de *stopwords*	0,62	0,91	0,57	1,33
Stemming	**0,67**	**0,63**	**0,66**	**0,64**
Stemming com remoção de *stopwords*	0,66	0,77	0,64	0,75

O melhor resultado foi obtido utilizando a técnica *stemming* nas sentenças, com 0,67 de CP e 0,63 de EQM para o PTBR, e 0,66 de CP e 0,64 de EQM para o PTPT. Este resultado se aproximou ao resultado obtido por HARTMANN (2016) que também utilizou TF-IDF com os *stems* das palavras na sua abordagem. Como a matriz TF-IDF sofre com a esparsidade dos dados, é necessário a utilização dos *stems* das palavras para reduzir o tamanho da matriz (HARTMANN, 2016). A Tabela 6.2 apresenta os resultados obtidos utilizando a métrica da matriz com Word2Vec.

Ao contrário da característica TF-IDF, esta obteve o melhor resultado sem aplicar nenhuma técnica de pré-processamento nas sentenças, com 0,65 de CP e 0,78 de EQM para o PTBR, e 0,61 de CP e 0,92 de EQM para o PTPT. Acreditamos que este método obteve melhores resultados sem aplicar nenhuma técnica de pré-processamento pelo fato dele não ter problemas com a esparsidade dos dados e também por levar em consideração a relação semântica entre cada palavra (*Word Embeddings*).

A Tabela 6.3 apresenta os resultados obtidos para a característica da matriz binária. O melhor resultado foi obtido utilizando *stemming* e remoção de *stopwords*, com 0,57 de CP e 1,12 de EQM para o PTBR e 0,61 de CP e 1,02 de EQM para o PTPT. Entretanto, este método

Tabela 6.2: Resultados obtidos para a característica Word2Vec na base de treino.

	PTBR		PTPT	
Técnica	**CP**	**EQM**	**CP**	**EQM**
Nenhuma técnica	**0,65**	**0,78**	**0,61**	**0,92**
Remoção de *stopwords*	0,56	1,35	0,51	1,36
Lematização	0,63	0,86	0,53	1,29
Lematização com remoção de *stopwords*	0,63	0,85	0,54	1,46
Stemming	0,64	0,89	0,60	0,94
Stemming com remoção de *stopwords*	0,55	1,69	0,50	1,88

Tabela 6.3: Resultados obtidos para a característica Matriz Binária na base de treino.

	PTBR		PTPT	
Técnica	**CP**	**EQM**	**CP**	**EQM**
Nenhuma técnica	0,49	2,14	0,56	1,60
Remoção de *stopwords*	0,55	1,54	0,60	1,12
Lematização	0,50	1,98	0,59	1,17
Lematização com remoção de *stopwords*	0,49	2,35	0,59	1,18
Stemming	0,51	1,56	0,57	1,29
Stemming com remoção de *stopwords*	**0,57**	**1,12**	**0,61**	**1,02**

obteve resultados abaixo dos outros métodos, apresentando um CP menor e um EQM elevado. Isto ocorre devido ao fato deste método não levar em consideração a semântica das palavras, verificando apenas as palavras que são lexicalmente iguais entre os pares de sentenças.

Para o tamanho da sentença não foi aplicada nenhuma técnica. Para esta característica não se fez necessário a análise com cada pré-processamento pois, se trata de uma simples medida estatística, onde basicamente são contabilizados os *tokens* de cada sentença e o valor de similaridade é obtido dividindo o número de *tokens* da menor sentença pelo número de *tokens* da maior sentença. Esta característica também foi utilizada nos trabalhos de ZHAO; ZHU; LAN (2014) e BJERVA et al. (2014).

Para cada métrica o melhor resultado foi obtido utilizando uma técnica de pré-processamento diferente. Isso mostra que análises desse tipo devem ser realizadas para se obter resultados mais precisos em aplicações de PLN. Com os resultados obtidos para a base de treino, a configuração para cada característica é apresentada na Tabela 6.4.

Tabela 6.4: Pré-processamento para cada característica.

Característica	**Pré-processamento**
TF-IDF	*Stemming*
Matriz com Word2Vec	Nenhuma técnica
Matriz binária	*Stemming* com remoção de *stopwords*
Tamanho da sentença	Nenhuma Técnica

6.4 Avaliação da Medida para STS

Para a base de teste foram realizados experimentos com cada característica individualmente e a combinação delas. Para isso foi utilizado o algoritmo de regressão linear do Weka, onde o algoritmo recebe como entrada as características e tem como saída uma equação que combina as características. A Tabela 6.5 mostra os resultados obtidos. Utilizamos MW para se referir a característica Matriz com Word2Vec, MB para se referir a característica Matriz Binária e TS para se referir ao Tamanho da sentença. Os intervalos de 0 a 1 foram convertidos para 1 a 5 e então foram calculados o CP e EQM das similaridades obtidas com as similaridades da base ASSIN.

Tabela 6.5: Resultados obtidos combinando as características.

	PTBR		**PTPT**	
Característica	**CP**	**EQM**	**CP**	**EQM**
TF-IDF	0,67	0,62	0,65	0,64
MW	0,64	0,88	0,68	1,50
MB	0,61	1,63	0,64	1,17
TS	0,10	2,18	-0,04	3,1
TF-IDF + MW	0,70	0,38	0,70	0,74
TF-IDF + MB	0,67	0,96	0,68	0,74
MW + MB	0,64	0,48	0,67	0,62
TF-IDF + MW + TS	0,66	1,21	0,66	1,21
TF-IDF + MB + TS	0,67	0,43	0,66	0,67
MW + MB + TS	0,65	0,40	0,67	0,92
TF-IDF + MW + MB	0,70	0,38	0,70	**0,57**
Todas as características	**0,71**	**0,37**	**0,71**	0,63

Como podemos perceber na Tabela 6.5, para o PTBR o uso de todas as características obteve o melhor resultado com 0,71 de CP e 0,37 de EQM. Já para o PTPT o uso de todas as características obteve o melhor CP com 0,71, entretanto o melhor EQM foi obtido usando as características: TF-IDF, Matriz com Word2Vec e Matriz Binária. Além disso, o tamanho da sentença não apresenta resultados significativos quando utilizado individualmente, mas quando combinado com as outras características produz resultados melhores. A Equação 6.6 foi a equação obtida com a combinação das quatro características. A equação obtida mostra que as características Tamanho da Sentença, Matriz com Word2Vec e TF-IDF são as mais importantes para medir a similaridade entre as sentenças por possuírem um peso maior.

$$\begin{aligned} Similaridade \quad &= -0.9581 + (0.4856 \times TF-IDF) + (0.5536 \times MatrizWord2Vec) + \\ &(0.0872 \times MatrizBinaria) + (0.6464 \times TamanhoSentenca) \end{aligned} \tag{6.6}$$

A Tabela 6.6 compara o melhor resultado obtido com o método proposto com os resultados obtidos pelas equipes da competição ASSIN 2016. A Tabela 6.6 mostra cada CP e EQM obtido com o nosso método e pelas equipes da competição.

Tabela 6.6: Comparação da medida proposta com as equipes do ASSIN para tarefa de STS.

	PTBR		**PTPT**		**TOTAL**	
Equipe/Método	**CP**	**EQM**	**CP**	**EQM**	**CP**	**EQM**
Medida Proposta	**0,71**	**0,37**	0,70	**0,57**	**0,70**	**0,47**
(HARTMANN, 2016)	0,70	0,38	0,70	0,66	0,68	0,52
(GONÇALO OLIVEIRA; ALVES; RODRIGUES, 2016)	0,59	1,31	0,54	1,10	0,54	1,23
(BARBOSA et al., 2016)	0,65	0,44	0,64	0,72	0,63	0,59
(ALVES; RODRIGUES; OLIVEIRA, 2016)	0,65	0,44	0,68	0,70	0,65	0,57
(FREIRE; PINHEIRO; FEITOSA, 2016)	0,62	0,47	0,64	0,72	0,62	0,59
(FIALHO et al., 2016)			**0,73**	0,61		

Como podemos perceber na Tabela 6.6 nossa medida obteve os melhores resultados para o PTBR e no resultado total (PTBR + PTPT). Para o PTPT nosso método obteve o CP abaixo da equipe L2F/INESC-ID, mas obteve o melhor EQM. A equipe L2F/INESC-ID obteve o primeiro lugar na competição para o PTPT, entretanto essa equipe não apresentou método para o PTBR. Dessa forma, o método proposto apresentou o melhor resultado no total com uma abordagem híbrida para o português brasileiro e o português europeu.

6.5 Avaliação da Medida para RIT

Para a tarefa de Reconhecimento de Implicação Textual, os experimentos foram conduzidos utilizando as medidas F-*measure* (F1), *Precision*, *Recall* e a Acurácia (porcentagem de instâncias classificadas corretamente) com os classificadores apresentados na Subseção 2.3.1. Os resultados são mostrados na Tabela 6.7.

Tabela 6.7: Resultados dos classificadores utilizados.

	PTBR				**PTPT**			
Classificador	***Acurácia***	**F1**	***Precision***	***Recall***	***Acurácia***	**F1**	***Precision***	***Recall***
SVM	84,85%	0,817	0,794	0,849	82,35%	0,805	0,803	0,824
Naive Bayes	83,90%	0,840	0,841	0,839	82,25%	0,823	0,827	0,823
MLP	85,35%	0,840	0,839	0,854	82,75%	0,819	0,829	0,828
Logistic	**85,50%**	**0,840**	**0,837**	**0,855**	**83,45%**	**0,827**	**0,826**	**0,835**
Random Forest	83,55%	0,825	0,819	0,836	81,10%	0,804	0,801	0,811

Tabela 6.8: Comparação da medida proposta com as equipes do ASSIN para tarefa de RIT.

	PTBR		PTPT		TOTAL	
Equipe/Método	**Acurácia**	**F1**	**Acurácia**	**F1**	**Acurácia**	**F1**
Medida Proposta	**85,50%**	**0,840**	83,45%	**0,827**	**84,175%**	**0,832**
(GONÇALO OLIVEIRA; ALVES; RODRIGUES, 2016)	79,05%	0,39	73,10%	0,43	75,58%	0,38
(BARBOSA et al., 2016)	81,65%	0,52	77,60%	0,61	79,62%	0,58
(ALVES; RODRIGUES; OLIVEIRA, 2016)	81,65%	0,47	78,90%	0,58	80,27%	0,54
(FIALHO et al., 2016)			**83,85%**	0,7		

Como podemos ver pela Tabela 6.7, o classificador *Logistic* obteve a melhor acurácia com 85,50% e 0,840 de F-*measure* para PTBR e 83,45% de acurácia e 0,827 de F-*measure* para o PTPT. A Tabela 6.7 compara o resultado obtido com o *Logistic* com os resultados das equipes da competição. É importante enfatizar que a competição ASSIN utilizou apenas as medidas F-*measure* e Acurácia para a tarefa de RIT.

Conforme apresentado na Tabela 6.8, a medida proposta obteve os melhores resultados, perdendo apenas na acurácia para a equipe L2F/INESC-ID. Os valores de F-*measure* obtidos foram expressivos chegando a atingir 61,53%, 18,14% e 43,44% melhores resultados em relação aos concorrentes para PTBR, PTPT e TOTAL, respectivamente. Em relação à Acurácia, a medida proposta atinge valores 4,71% e 4,86% maiores que os concorrentes para PTBR e TOTAL, respectivamente.

6.6 Estudo de caso em um Fórum Web

Com o objetivo de realizar uma avaliação da utilização da medida proposta em uma aplicação real, foi desenvolvida uma ferramenta de fórum com as mesmas funcionalidades de um fórum tradicional. A ferramenta foi desenvolvida utilizando as linguagens *HyperText Markup Language* (HTML), *Cascading Style Sheets* (CSS) com o *framework* Bootstrap, *Hypertext Preprocessor* (PHP) e o banco de dados MySQL. O objetivo da criação da ferramenta foi a integração com a medida de similaridade e a fácil aplicação em um ambiente real de aprendizagem em um tempo relativamente curto. Não utilizamos o Moodle, pois, seria necessário um estudo mais aprofundado de como desenvolver um módulo para integrar a medida de similaridade ao ambiente.

A ferramenta criada possui dois diferenciais, como apresentado nas Figuras 6.3 e 6.4:

- uma barra no canto superior esquerdo, mostrando o nível de originalidade, inversamente proporcional ao nível de similaridade entre as postagens;

- análise de estatísticas de similaridade entre postagens para o professor, onde para cada postagem aparece a postagem mais similar e o valor de similaridade entre as postagens.

É importante enfatizar que a página com as estatísticas de similaridade entre as postagens é apresentada apenas para o professor (administrador do fórum criado).

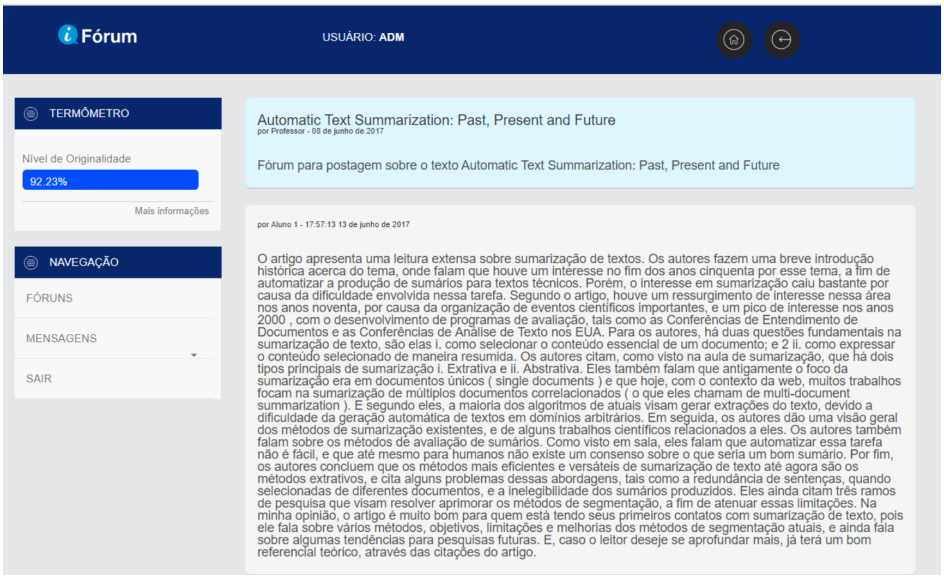

Figura 6.3: Página de um fórum com barra de originalidade no canto superior esquerdo.

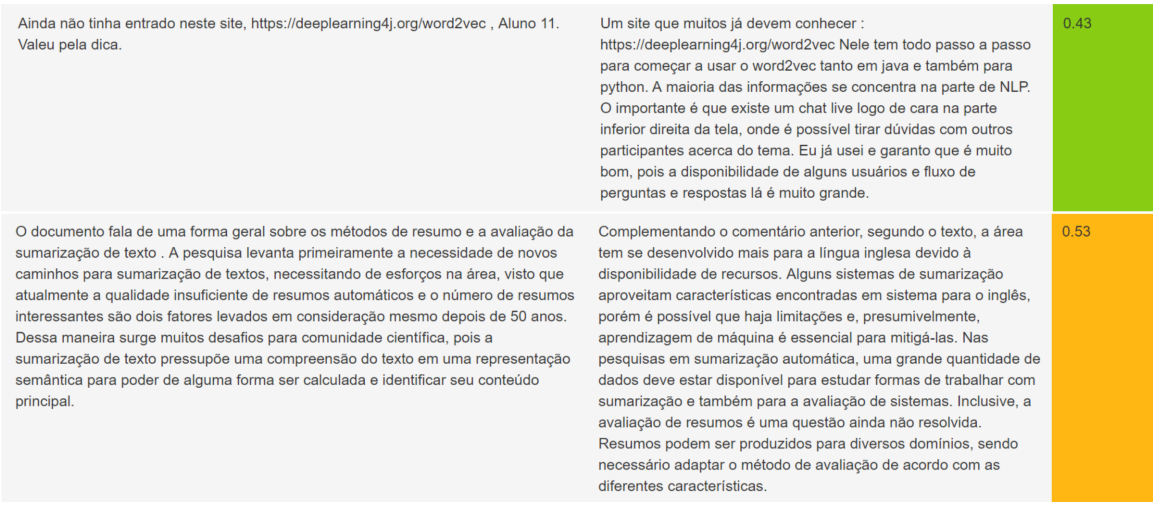

Figura 6.4: Página com as estatísticas de similaridade entre as postagens do fórum.

A Figura 6.5 mostra basicamente as funcionalidades que o professor tem acesso e as funcionalidades que o aluno tem acesso. O professor tem acesso ao gerenciamento dos fóruns, podendo criar, excluir, editar e visualizar os fóruns criados, além de ter acesso a página com as similaridades entre as postagens. As outras funcionalidades estão disponíveis para os alunos e o professor, onde na página do fórum eles podem visualizar os fóruns, cadastrar postagens, editar postagens e excluir postagens e na página de mensagens eles podem enviar e receber mensagens.

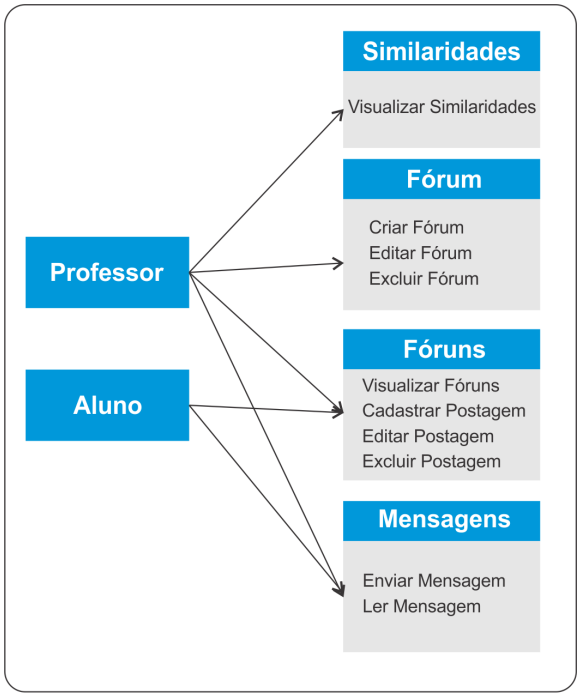

Figura 6.5: Funcionalidades para professor e estudante.

Além disso, foi construída uma *Web Service* em JAVA que calcula a similaridade entre as postagens. Para isso foi utilizada a plataforma de serviços da Amazon chamada *Amazon Web Services* (AWS). A Figura 6.6 mostra a arquitetura básica da ferramenta desenvolvida. A *Web Service* obtém do banco de dados as postagens do fórum e calcula todas as similaridades. As similaridades são mostradas em uma página para o administrador do fórum e são classificadas em 4 grupos:

(1) Azul (0 - 0,3);

(2) Verde (0,3 - 0,5);

(3) Amarelo (0,5 - 0,7);

(4) Vermelho (0,7 - 1,0).

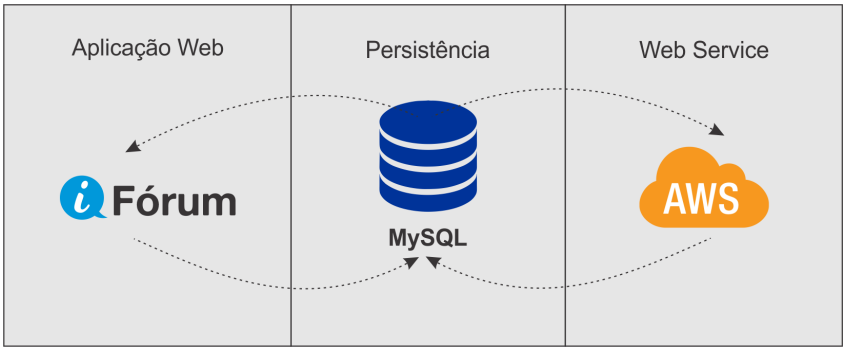

Figura 6.6: Arquitetura básica da aplicação.

A Figura 6.7 mostra um exemplo desta página. As cores azul e verde significam que as postagens não apresentam muita semelhança. As cores amarela e vermelha significam que as postagens estão em um nível de similaridade maior, o que pode significar o plágio, principalmente a vermelha.

VISUALIZAR

Nº	Postagem	Postagem similar	Similaridade
1	Lógica proposicional é um assunto bem interessante. Além de ser útil para computação, existem muitas questões de concursos públicos que abordam esse conteúdo. Eu queria solicitar uma lista de exercícios.	Lógica proposicional é um assunto bem interessante. Além de ser útil para computação, existem muitas questões de concursos públicos que abordam esse conteúdo. Eu queria solicitar uma lista de exercícios.	1.0
2	Eu gostei bastante dos exemplos utilizados em aula pelo professor para explicar consequências lógicas. Facilitou demais a entender que nem sempre uma consequência é bidirecional. Eu tentei entender pelo livro, mas não consegui, mas graças aos exemplos do professor consegui entender.	Eu gostei bastante dos exemplos utilizados em aula pelo professor para explicar consequências lógicas. Facilitou demais a entender que nem sempre uma consequência é bidirecional. Eu tentei entender pelo livro, mas não consegui, mas graças aos exemplos do professor consegui entender.	1.0
3	Lógica proposicional é um assunto bem interessante. Além de ser útil para computação, existem muitas questões de concursos públicos que abordam esse conteúdo. Eu queria solicitar uma lista de exercícios.	Lógica proposicional é um assunto bem interessante. Além de ser útil para computação, existem muitas questões de concursos públicos que abordam esse conteúdo. Eu queria solicitar uma lista de exercícios.	0.99
4	Eu gostei bastante dos exemplos utilizados em aula pelo professor para explicar consequências lógicas. Facilitou demais a entender que nem sempre uma consequência é bidirecional. Eu tentei entender pelo livro, mas não consegui, mas graças aos exemplos do professor consegui entender.	Eu gostei bastante dos exemplos utilizados em aula pelo professor para explicar consequências lógicas. Facilitou demais a entender que nem sempre uma consequência é bidirecional. Eu tentei entender pelo livro, mas não consegui, mas graças aos exemplos do professor consegui entender.	0.99
5	Estou bastante empolgado com proposições e operadores lógicos, achei um assunto muito interessante e bem fácil de entender do jeito que está sendo abordado.	Acredito que não vamos ter muita dificuldade nesse assunto. Gostei da abordagem do professor no assunto de proposições e operadores lógicos, vai ser bem mais tranquilo entender os conceitos agora e a lista de exercícios também vai ajudar bastante.	0.57
6	Consequências lógicas foi um assunto bem fácil de entender, com toda a base que o professor já nos deu sobre o assunto foi bem tranquilo de acompanhar e entender.	Acredito que não vamos ter muita dificuldade nesse assunto. Gostei da abordagem do professor no assunto de proposições e operadores lógicos, vai ser bem mais tranquilo entender os conceitos agora e a lista de exercícios também vai ajudar bastante.	0.48
7	Gostei de como o professor abordou e explicou lógica proposicional, o ritmo da explicação e os exemplos usados me ajudaram muito a entender. Espero que os próximos assuntos sejam abordados da mesma maneira.	Tabela-verdade me pareceu ser um assunto bem fácil de aprender, acho que consegui entender uma grande parte do assunto na primeira aula por causa dos exemplos mostrados e da abordagem do professor.	0.45

Figura 6.7: Exemplo da classificação de similaridade por cores.

O Algoritmo 3 mostra como são obtidas as similaridades pela *Web Service*. Inicialmente são obtidas as postagens do banco, em seguida para cada postagem p_1 é calculada a similaridade com todas as outras postagens do banco, e então é obtida a postagem mais similar com p_1 e o seu respectivo valor de similaridade. A saída do algoritmo é um vetor contendo uma postagem x, uma postagem y mais similar a postagem x e o valor de similaridade entre x e y.

Para avaliar a eficiência da abordagem, realizou-se um experimento em uma turma de Tópicos Avançados em Inteligência Artificial que possuía 12 alunos. Foi proposta a utilização de fórum para discussão de artigos científicos relacionados ao tema da disciplina, os alunos teriam uma semana para realizar as postagens. Em um primeiro momento, foi utilizado um fórum tradicional, utilizando o Moodle, onde houve apenas 10 postagens nesse fórum.

Algoritmo 3: SIMILARIDADES

Entrada: Postagens
Saída: Postagem, Postagem mais similar e valor de similaridade

```
1  início
2      P = Postagens
3      para cada postagem p1 ∈ P faça
4          MaiorSim = Double.MinValue
5          para cada postagem p2 ∈ (P − p1) faça
6              similaridade ← SIMILARIDADE(p1, p2)
7              se similaridade > MaiorSim então
8                  MaiorSim ← similaridade
9                  postagemSimilar ← p2
10             fim
11         fim
12         S ← p1, postagemSimilar, MaiorSim
13     fim
14 fim
15 retorna S
```

Na atividade seguinte foi passado outro artigo para a mesma turma, mas foi utilizada a ferramenta proposta, com estatísticas para professores e alunos Os alunos teriam também uma semana para realizar as postagens. Nesse fórum o número de postagem subiu para 30. A Figura 6.8 mostra o comparativo de postagens das duas atividades.

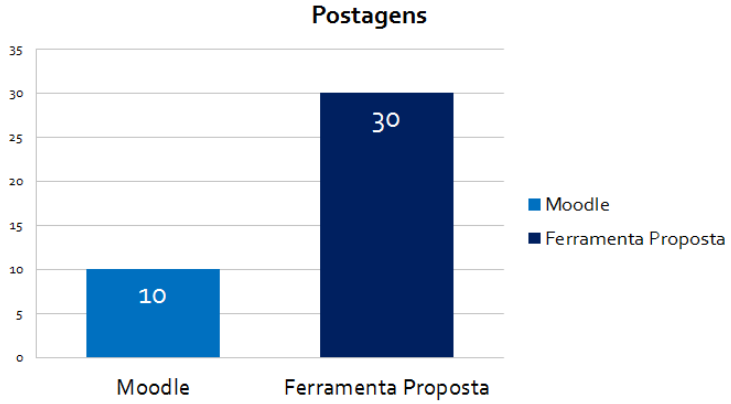

Figura 6.8: Comparativo de postagens entre a 1ª e a 2ª Atividade.

Além do acréscimo do número de postagens, na primeira semana a média de similaridade entre as postagens foi de 0,28 e com um desvio padrão de 0,13; enquanto que na segunda semana (ferramenta de fórum com a medida proposta) a média foi 0,27 e o desvio padrão 0,09. Em outras palavras, o nível de similaridade, consequentemente de plágio, foi menor com a utilização da ferramenta proposta, mesmo tendo três vezes mais postagens no fórum, e em geral as postagens tiveram um nível de similaridade mais próxima (menor desvio padrão). A Figura 6.9 mostra o comparativo da média e desvio padrão das similaridades entre as postagens das duas atividades.

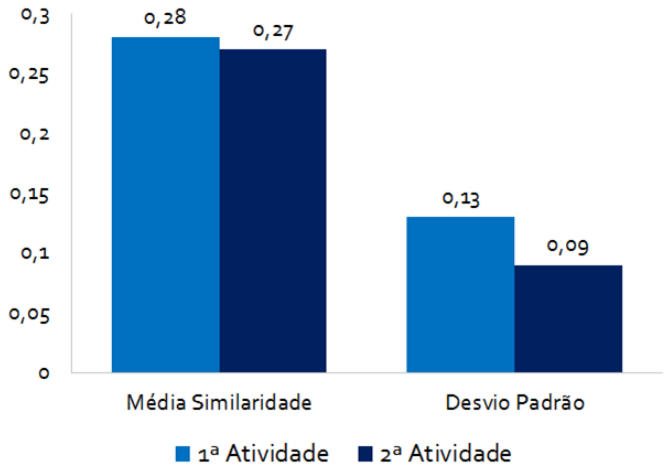

Figura 6.9: Média e desvio padrão das similaridades entre as postagens das duas atividades.

Por fim, também foi relevante para o professor o fato de poder contar com as estatísticas relacionadas às postagens similares. Esse fator foi levado em consideração para a atribuição final da nota dos alunos nos fóruns.

6.7 Conclusão

A similaridade entre sentenças torna-se importante em várias aplicações de PLN como sumarização de texto, extração de informação e agrupamento de texto. Neste trabalho foi apresentada uma nova abordagem para calcular a similaridade textual para a língua portuguesa. Essa abordagem extrai 4 características dos textos, sendo elas: TF-IDF, matriz de similaridades com word2vec, matriz binária e o tamanho do texto. O TF-IDF é uma abordagem clássica da área de PLN que combina a frequência dos termos de uma coleção (TF) e a relevância do termo para uma coleção (IDF). Esta técnica foi utilizada em vários trabalhos para calcular a similaridade entre sentenças. Podemos destacar os trabalhos de FIALHO et al. (2016) que obteve o primeiro lugar na tarefa de STS para o português europeu e o trabalho de HARTMANN (2016) que obteve o primeiro lugar na tarefa de STS para o português brasileiro. Neste trabalho o TF-IDF obteve o melhor resultado em comparação com as outras características. Outra técnica muito utilizada na literatura para similaridade entre sentenças são as *word embeddings*. Entretanto, ao invés de utilizar a soma das *embeddings*, como é realizado em outros trabalhos (HARTMANN, 2016; SILVA et al., 2017), neste trabalho foi utilizada em conjunto com a matriz de similaridades proposta por FERREIRA et al. (2016). A matriz binária foi uma característica proposta neste trabalho, que sozinha não apresenta bons resultados, mas quando combinada com as outras características demonstra uma melhoria. O tamanho do texto é uma característica utilizada em trabalhos da língua inglesa (ZHAO; ZHU; LAN, 2014; BJERVA et al., 2014). Os trabalhos para o português não utilizaram esta característica. Da mesma forma que a matriz binária, o tamanho

do texto apresenta uma melhoria apenas quando combinado com as outras características.

É importante salientar que a medida proposta extrai apenas quatro características das sentenças utilizando medidas estatísticas e uma medida semântica e apresenta resultados melhores que outras abordagens que utilizam várias características ou medidas mais complexas de serem utilizadas.

Para as duas tarefas do *workshop* ASSIN a medida proposta chegou a atingir os melhores resultados em relação aos resultados obtidos pelos participantes do *workshop*. Para a tarefa de Similaridade Textual Semântica foi utilizado um algoritmo de regressão linear para obter um valor de similaridade combinando as características extraídas. Para a tarefa de Reconhecimento de Implicação Textual foram analisados 5 classificadores, onde o *Logistic* obteve o melhor resultado em comparação com os outros.

Além disso, foi criada uma ferramenta de fórum que utiliza a medida proposta para calcular a similaridade entre as postagens. A ferramenta motiva os alunos a postarem no fórum com postagens mais originais, apresentando uma barra de originalidade na parte superior esquerda do fórum, e também apresenta para o professor uma página com as estatísticas de similaridade entre as postagens do fórum. A página com as estatísticas de similaridade entre as postagens é de bastante relevância para o professor, pois, além de verificar se existe plágio nas postagens dos alunos, ele pode utilizar as estatísticas como forma de avaliação dos alunos. Por exemplo, o aluno que tem uma média de similaridade baixa pode receber uma nota melhor, por postar um conteúdo mais original, do que o aluno que tem uma média de similaridade alta, mostrando que a postagem dele foi baseada na resposta de outro aluno. Os resultados mostraram que a ferramenta de fórum proposta obteve mais postagens com uma média de similaridade e desvio padrão menor em comparação com a utilização de um fórum tradicional.

7

Considerações Finais

7.1 Contribuições

Este trabalho de dissertação teve como objetivo tratar o problema de plágio em fóruns educacionais. Para isso, uma medida de similaridade entre sentenças para a língua portuguesa foi proposta. A medida foi aplicada em um fórum educacional para calcular a similaridade entre postagens do fórum, apresentando uma página com as similaridades entre todas as postagens do fórum e também mostrando uma barra de originalidade para os alunos.

Como contribuições deste trabalho destacamos: (i) Criação de uma medida de similaridade que extrai apenas quatro características das sentenças utilizando recursos estatísticos e semânticos, (ii) Avaliação de diferentes tipos de pré-processamentos nas sentenças para obter o melhor resultado, (iii) Criação de uma ferramenta de fórum Web, (iv) Integração da medida de similaridade ao fórum e (v) Revisão Sistemática da Literatura sobre o plágio em ambientes educacionais.

A medida de similaridade foi utilizada neste trabalho para o contexto de plágio em fóruns educacionais, entretanto, a medida pode ser utilizada para outras finalidades com textos curtos. A etapa de análise de diferentes pré-processamentos foi de grande importância, pois permitiu obter melhorias significativas para cada característica individualmente. O fórum criado é uma ferramenta simples sem muitas funcionalidades, mas que apresenta informações importantes para o professor sobre as postagens dos alunos, além de estimular os alunos a postarem conteúdos mais originais, evitando que os alunos copiem um do outro ou realize paráfrase nas respostas. Por fim, a revisão sistemática da literatura traz para a comunidade científica o estado da arte sobre o plágio em ambientes educacionais, mostrando as principais ferramentas e técnicas utilizadas para detectar o plágio, seja ele externo ou interno.

7.2 Artigos submetidos/aceitos

Abaixo são mostrados alguns artigos submetidos/aceitos em conferências e periódicos, frutos deste trabalho de dissertação.

(1) *Statistical and Semantic Features to Measure Sentence Similarity in Portuguese* (CAVALCANTI et al., 2017) (Aceito).

(2) Uma Nova Abordagem para Detecção de Plágio em Ambientes Educacionais (CAVALCANTI et al., 2017) (Aceito).

(3) O Plágio em Ambiente Educacional Virtual - Uma Revisão da Literatura (CAVALCANTI et al., 2017) (Aceito).

(4) *A Survey on Text Mining in Online Education* (FERREIRA et al., 2017) (Submetido).

7.3 Limitações da Pesquisa

Este trabalho de dissertação apresentou bons resultados para a medida de similaridade e para o fórum que foi aplicado em uma disciplina. Entretanto, este projeto também possui algumas limitações, dentre elas destacamos:

- A medida proposta, se aplicada em um grande conjunto de sentenças, apresenta um elevado tempo computacional;
- A medida também foi avaliada na similaridade entre textos maiores, como por exemplo redações, e além do elevado tempo computacional para calcular a similaridade entre os textos, a medida não apresentou bons resultados.

7.4 Trabalhos Futuros

Como trabalhos futuros pretende-se avaliar a medida de similaridade com outras abordagens de *Word Embeddings*, como por exemplo o GloVe e o Wang2Vec (LING et al., 2015; HARTMANN et al., 2017), além de analisar a medida com a utilização de recursos sintáticos das sentenças. Também pretende-se disponibilizar a medida como *Application Programming Interface* (API) para que outras pessoas possam utilizar. Outro trabalho futuro é aplicar a medida proposta para outros idiomas, como por exemplo o inglês.

Para a área educacional, pretende-se utilizar algum método de detecção de plágio externo em conjunto com a abordagem proposta, e assim detectar os dois tipos de plágio em fóruns educacionais. Além disso, pretende-se integrar a medida de similaridade ao Moodle, já que essa ferramenta é a mais utilizada na EAD (DIONÍSIO et al., 2017; CAVALCANTI et al., 2017).

Referências

ACHANANUPARP, P.; HU, X.; SHEN, X. The evaluation of sentence similarity measures. **Data warehousing and knowledge discovery**, [S.l.], p.305–316, 2008.

AKÇAPINAR, G. How automated feedback through text mining changes plagiaristic behavior in online assignments. **Computers & Education**, [S.l.], v.87, p.123–130, 2015.

ALVES, A. O.; RODRIGUES, R.; OLIVEIRA, H. G. ASAPP: alinhamento semântico automático de palavras aplicado ao português. **Linguamática**, [S.l.], v.8, n.2, p.43–58, 2016.

ARENHARDT, C. P. B. et al. Miss Marple–Proposta de Desenvolvimento de Ferramenta de Detecção de Indícios de Plágio com base no Método DIP–Detector de Indícios de Plágio. In: BRAZILIAN SYMPOSIUM ON COMPUTERS IN EDUCATION (SIMPÓSIO BRASILEIRO DE INFORMÁTICA NA EDUCAÇÃO-SBIE). **Anais...** [S.l.: s.n.], 2012. v.23, n.1.

BARBASTEFANO, R. G.; SOUZA, C. G. Plágio em trabalhos acadêmicos: uma pesquisa com alunos de graduação. **Anais do 27º Encontro Nacional de Engenharia de Produção**, [S.l.], p.8–11, 2007.

BARBOSA, L. et al. Blue Man Group at ASSIN: using distributed representations for semantic similarity and entailment recognition. **LINGUAMATICA**, [S.l.], v.8, n.2, p.15–22, 2016.

BARROS, M. d. G.; CARVALHO, A. B. G. As concepções de interatividade nos ambientes virtuais de aprendizagem. **Campina Grande: EDUEPB**, [S.l.], 2011.

BASTOS, V. M. Ambiente de descoberta de conhecimento na web para a língua portuguesa. **Monograph (Doctored), Federal University of Rio de Janeiro, Rio de Janeiro**, [S.l.], 2006.

BATANE, T. Turning to Turnitin to fight plagiarism among university students. **Journal of Educational Technology & Society**, [S.l.], v.13, n.2, p.1, 2010.

BENGIO, Y. et al. A neural probabilistic language model. **Journal of machine learning research**, [S.l.], v.3, n.Feb, p.1137–1155, 2003.

BJERVA, J. et al. The Meaning Factory: formal semantics for recognizing textual entailment and determining semantic similarity. In: SEMEVAL@ COLING. **Anais...** [S.l.: s.n.], 2014. p.642–646.

BRADLEY, E. G. Using Computer Simulations and Games to Prevent Student Plagiarism. **Journal of Educational Technology Systems**, [S.l.], v.44, n.2, p.240–252, 2015.

BREIMAN, L. Random forests. **Machine learning**, [S.l.], v.45, n.1, p.5–32, 2001.

BRINK, H.; RICHARDS, J.; FETHEROLF, M. **Real-world machine learning**. [S.l.]: Manning Publications Co., 2016.

BULEGON, H.; MORO, C. M. C. Mineração de texto e o processamento de linguagem natural em sumários de alta hospitalar. **Journal of Health Informatics**, [S.l.], v.2, n.2, 2010.

BURGES, C. J. A tutorial on support vector machines for pattern recognition. **Data mining and knowledge discovery**, [S.l.], v.2, n.2, p.121–167, 1998.

BUTAKOV, S.; SCHERBININ, V. The toolbox for local and global plagiarism detection. **Computers & Education**, [S.l.], v.52, n.4, p.781–788, 2009.

BUTAKOV, S.; SHCHERBININ, V. On the number of search queries required for Internet plagiarism detection. In: ADVANCED LEARNING TECHNOLOGIES, 2009. ICALT 2009. NINTH IEEE INTERNATIONAL CONFERENCE ON. **Anais...** [S.l.: s.n.], 2009. p.482–483.

CAVALCANTI, A. et al. Uma Nova Abordagem para Detecção de Plágio em Ambientes Educacionais. In: BRAZILIAN SYMPOSIUM ON COMPUTERS IN EDUCATION (SIMPÓSIO BRASILEIRO DE INFORMÁTICA NA EDUCAÇÃO-SBIE). **Anais...** [S.l.: s.n.], 2017. v.28, n.1, p.1177.

CAVALCANTI, A. P. et al. O Plágio em Ambiente Educacional Virtual: uma revisão da literatura. **RENOTE**, [S.l.], v.15, n.2, 2017.

CAVALCANTI, A. P. et al. Statistical and Semantic Features to Measure Sentence Similarity in Portuguese. In: BRAZILIAN CONFERENCE ON INTELLIGENT SYSTEMS (BRACIS), 2017. **Anais...** [S.l.: s.n.], 2017. p.342–347.

CAVALCANTI, E. R. et al. Detecção e Avaliação de Cola em Provas Escolares Utilizando Mineração de Texto: um estudo de caso. **Brazilian Journal of Computers in Education**, [S.l.], v.19, n.02, p.56, 2011.

CHEN, T. Y.; KUO, F.-C.; MERKEL, R. On the statistical properties of the f-measure. In: QUALITY SOFTWARE, 2004. QSIC 2004. PROCEEDINGS. FOURTH INTERNATIONAL CONFERENCE ON. **Anais...** [S.l.: s.n.], 2004. p.146–153.

CHENG, C. K. et al. Assessing the effectiveness of a voluntary online discussion forum on improving students? course performance. **Computers & Education**, [S.l.], v.56, n.1, p.253–261, 2011.

CHOUDHARY, B.; BHATTACHARYYA, P. Text clustering using semantics. In: INTERNATIONAL WORLD WIDE WEB CONFERENCE, 11. **Proceedings...** [S.l.: s.n.], 2002. p.1–4.

COELHO, O. B.; SILVEIRA, I. Deep Learning applied to Learning Analytics and Educational Data Mining: a systematic literature review. In: BRAZILIAN SYMPOSIUM ON COMPUTERS IN EDUCATION (SIMPÓSIO BRASILEIRO DE INFORMÁTICA NA EDUCAÇÃO-SBIE). **Anais...** [S.l.: s.n.], 2017. v.28, n.1, p.143.

DAG, J. N. och et al. Evaluating automated support for requirements similarity analysis in market-driven development. In: SEVENTH INTERNATIONAL WORKSHOP ON REQUIREMENTS ENGINEERING: FOUNDATION FOR SOFTWARE QUALITY (REFSQ'01). **Anais...** [S.l.: s.n.], 2001.

DAYTON, C. M. Logistic regression analysis. **Stat**, [S.l.], p.474–574, 1992.

DILLENBOURG, P.; SCHNEIDER, D.; SYNTETA, P. Virtual learning environments. In: HELLENIC CONFERENCE INFORMATION & COMMUNICATION TECHNOLOGIES IN EDUCATION, 3. **Anais...** [S.l.: s.n.], 2002. p.3–18.

DIONÍSIO, M. et al. Mineração de Texto Aplicada à Identificação de Colaboração em Fóruns Educacionais. In: BRAZILIAN SYMPOSIUM ON COMPUTERS IN EDUCATION (SIMPÓSIO BRASILEIRO DE INFORMÁTICA NA EDUCAÇÃO-SBIE). **Anais...** [S.l.: s.n.], 2017. v.28, n.1, p.1437.

DOMINGOS, P.; PAZZANI, M. On the optimality of the simple Bayesian classifier under zero-one loss. **Machine learning**, [S.l.], v.29, n.2, p.103–130, 1997.

DRAELOS, T. J. et al. Neurogenesis deep learning: extending deep networks to accommodate new classes. In: NEURAL NETWORKS (IJCNN), 2017 INTERNATIONAL JOINT CONFERENCE ON. **Anais...** [S.l.: s.n.], 2017. p.526–533.

FEITOSA, D.; PINHEIRO, V. Análise de Medidas de Similaridade Semântica na Tarefa de Reconhecimento de Implicação Textual. In: BRAZILIAN SYMPOSIUM IN INFORMATION AND HUMAN LANGUAGE TECHNOLOGY, 11. **Proceedings...** [S.l.: s.n.], 2017. p.161–170.

FERREIRA, R. et al. Assessing sentence similarity through lexical, syntactic and semantic analysis. **Computer Speech & Language**, [S.l.], v.39, p.1–28, 2016.

FERREIRA, R. et al. **A Survey on Text Mining in Online Education**. 2017.

FIALHO, P. et al. INESC-ID@ ASSIN: mediçao de similaridade semântica e reconhecimento de inferência textual. **Linguamática**, [S.l.], v.8, n.2, p.33–42, 2016.

FIGUEIREDO FILHO, D. B.; JUNIOR, J. A. S. Desvendando os Mistérios do Coeficiente de Correlação de Pearson (r). **Revista Política Hoje-ISSN: 0104-7094**, [S.l.], v.18, n.1, 2010.

FONSECA, E. R. et al. Visão geral da avaliação de similaridade semântica e inferência textual. **Linguamática**, [S.l.], v.8, n.2, p.3–13, 2016.

FRANÇA, A. B.; SOARES, J. M. Sistema de apoio a atividades de laboratório de programação com suporte ao balanceamento de carga e controle de plágio. In: BRAZILIAN SYMPOSIUM ON COMPUTERS IN EDUCATION (SIMPÓSIO BRASILEIRO DE INFORMÁTICA NA EDUCAÇÃO-SBIE). **Anais...** [S.l.: s.n.], 2012. v.23, n.1.

FREIRE, J.; PINHEIRO, V.; FEITOSA, D. **LEC_UNIFOR no ASSIN**: flexsts, um framework para similaridade semântica textual. 2016.

FREITAS, M.; SILVA, A. **Avaliação da Aprendizagem em ambientes de formação online**: aportes para uma abordagem hermenêutica. 2009. Tese (Doutorado em Ciência da Computação) — Tese (doutorado). UFBA: Faculdade de Educação, Salvador.

GARSCHAGEN, B. Universidade em tempos de plágio. **EAD-L [lista de discussão na internet]. Campinas: Unicamp/Centro de Computação**, [S.l.], 2006.

GAZZOLA, M. Um Método para Avaliação Automática da Qualidade de Recursos Educacionais Abertos usando Deep Learning. In: BRAZILIAN SYMPOSIUM ON COMPUTERS IN EDUCATION (SIMPÓSIO BRASILEIRO DE INFORMÁTICA NA EDUCAÇÃO-SBIE). **Anais...** [S.l.: s.n.], 2017. v.28, n.1, p.1477.

GOMES, R.; MEDINA, R. Parallel Miss Marple: threads e java rmi aplicados à verificação de indícios de plágio. In: WORKSHOP DE INFORMÁTICA NA ESCOLA. **Anais...** [S.l.: s.n.], 2016. v.22, n.1, p.369.

GONÇALO OLIVEIRA, H.; ALVES, A. O.; RODRIGUES, R. Reciclagem: exploring portuguese lexical knowledge-bases in the assin task. **Linguamática**, [S.l.], v.8, n.2, 2016.

HARTMANN, N. et al. Portuguese Word Embeddings: evaluating on word analogies and natural language tasks. **arXiv preprint arXiv:1708.06025**, [S.l.], 2017.

HARTMANN, N. S. Solo queue at ASSIN: combinando abordagens tradicionais e emergentes. **Linguamática**, [S.l.], v.8, n.2, p.59–64, 2016.

HAYKIN, S. **Neural networks**: a comprehensive foundation. [S.l.]: Prentice Hall PTR, 1994.

HEARST, M. A. et al. Support vector machines. **IEEE Intelligent Systems and their applications**, [S.l.], v.13, n.4, p.18–28, 1998.

HOSMER JR, D. W.; LEMESHOW, S.; STURDIVANT, R. X. **Applied logistic regression**. [S.l.]: John Wiley & Sons, 2013. v.398.

KAKKONEN, T.; MOZGOVOY, M. Hermetic and web plagiarism detection systems for student essays—an evaluation of the state-of-the-art. **Journal of Educational Computing Research**, [S.l.], v.42, n.2, p.135–159, 2010.

KAUFFMAN, Y.; YOUNG, M. F. Digital plagiarism: an experimental study of the effect of instructional goals and copy-and-paste affordance. **Computers & Education**, [S.l.], v.83, p.44–56, 2015.

KITCHENHAM, B. Procedures for performing systematic reviews. **Keele, UK, Keele University**, [S.l.], v.33, n.2004, p.1–26, 2004.

KOHAVI, R. et al. A study of cross-validation and bootstrap for accuracy estimation and model selection. In: IJCAI. **Anais...** [S.l.: s.n.], 1995. v.14, n.2, p.1137–1145.

LING, W. et al. Two/Too Simple Adaptations of Word2Vec for Syntax Problems. In: CONFERENCE OF THE NORTH AMERICAN CHAPTER OF THE ASSOCIATION FOR COMPUTATIONAL LINGUISTICS: HUMAN LANGUAGE TECHNOLOGIES, 2015. **Proceedings...** [S.l.: s.n.], 2015. p.1299–1304.

LIU, G.-Z.; LO, H.-Y.; WANG, H.-C. Design and usability testing of a learning and plagiarism avoidance tutorial system for paraphrasing and citing in English: a case study. **Computers & Education**, [S.l.], v.69, p.1–14, 2013.

LIU, Y.-T. et al. Extending Web search for online plagiarism detection. In: INFORMATION REUSE AND INTEGRATION, 2007. IRI 2007. IEEE INTERNATIONAL CONFERENCE ON. **Anais...** [S.l.: s.n.], 2007. p.164–169.

LORENA, A. C.; CARVALHO, A. C. de. Uma introdução às support vector machines. **Revista de Informática Teórica e Aplicada**, [S.l.], v.14, n.2, p.43–67, 2007.

LOULA, A. C. **Emergência de comunicação e representações em criaturas artificiais**. [S.l.]: Angelo Conrado Loula, 2011.

MACHADO, A. P. et al. Mineração de texto em Redes Sociais aplicada à Educação a Distância. **Colabor@-A Revista Digital da CVA-RICESU**, [S.l.], v.6, n.23, 2010.

MASIC, I. Plagiarism in scientific publishing. **Acta Informatica Medica**, [S.l.], v.20, n.4, p.208, 2012.

MATOS, M. A. Manual operacional para a regressão linear. **Faculdade de Engenharia da Universidade do Porto**, [S.l.], 1995.

MAZIERO, E. G. et al. A base de dados lexical e a interface web do TeP 2.0: thesaurus eletrônico para o português do brasil. In: COMPANION PROCEEDINGS OF THE XIV

BRAZILIAN SYMPOSIUM ON MULTIMEDIA AND THE WEB. **Anais...** [S.l.: s.n.], 2008. p.390–392.

MIKOLOV, T. et al. Efficient estimation of word representations in vector space. **arXiv preprint arXiv:1301.3781**, [S.l.], 2013.

MORAN, J. M. Modelos e avaliação do ensino superior a distância no Brasil. **Educação Temática Digital**, [S.l.], v.10, n.2, p.54, 2009.

MOZGOVOY, M.; KAKKONEN, T.; COSMA, G. Automatic student plagiarism detection: future perspectives. **Journal of Educational Computing Research**, [S.l.], v.43, n.4, p.511–531, 2010.

NAU, J. et al. Uma Ferramenta para Identificar Desvios de Linguagem na Língua Portuguesa. In: BRAZILIAN SYMPOSIUM IN INFORMATION AND HUMAN LANGUAGE TECHNOLOGY, 11. **Proceedings...** [S.l.: s.n.], 2017. p.12–16.

NUNES, F. B. et al. ANÁLISE COMPARATIVA TEÓRICO-PRÁTICA ENTRE SOFTWARES DE DETECÇÃO DE PLÁGIO. **RENOTE**, [S.l.], v.10, n.3, 2012.

OBERREUTER, G.; VELÁSQUEZ, J. D. Text mining applied to plagiarism detection: the use of words for detecting deviations in the writing style. **Expert Systems with Applications**, [S.l.], v.40, n.9, p.3756–3763, 2013.

OLIVEIRA, B. V. Uma Análise de Estratégias de Sumarização Automática. **81f, Dissertação (Mestrado em Engenharia Civil)-Universidade Federal do Rio de Janeiro. Disponível em:< http://www. dominiopublico. gov. br/download/texto/cp047453. pdf>. Acessado em**, [S.l.], v.16, 2008.

PERNA, C. L.; DELGADO, H. K.; FINATTO, M. J. **Linguagens Especializadas em Corpora**: modos de dizer e interfaces de pesquisa. [S.l.]: EDIPUCRS, 2010.

PERTILE, S. d. L. et al. Agente Integrado a Plataforma MLE-Moodle para Detecção Automática de Indícios de Plágio. In: BRAZILIAN SYMPOSIUM ON COMPUTERS IN EDUCATION (SIMPÓSIO BRASILEIRO DE INFORMÁTICA NA EDUCAÇÃO-SBIE). **Anais...** [S.l.: s.n.], 2010. v.1, n.1.

PERTILE, S. d. L. et al. **Desenvolvimento e Aplicacao de um Metodo para Deteccao de Indicios de Plagio**. [S.l.]: Universidade Federal de Santa Maria, 2011.

RISH, I. An empirical study of the naive Bayes classifier. In: IJCAI 2001 WORKSHOP ON EMPIRICAL METHODS IN ARTIFICIAL INTELLIGENCE. **Anais...** [S.l.: s.n.], 2001. v.3, n.22, p.41–46.

ROCHA, E. et al. Detecção Automática de Plágio em Ambiente Educacional Virtual. In: WORKSHOP DE DESAFIOS DA COMPUTAÇÃO APLICADA À EDUCAÇÃO. **Anais...** [S.l.: s.n.], 2012. p.120–127.

ROLIM, V.; FERREIRA, R.; COSTA, E. Identificação Automática de Dúvidas em Fóruns Educacionais. In: BRAZILIAN SYMPOSIUM ON COMPUTERS IN EDUCATION (SIMPÓSIO BRASILEIRO DE INFORMÁTICA NA EDUCAÇÃO-SBIE). **Anais...** [S.l.: s.n.], 2016. v.27, n.1, p.936.

ROLIM, V.; FERREIRA, R.; COSTA, E. Método Supervisionado para Identificação de Dúvidas em Fóruns Educacionais. In: WORKSHOPS DO CONGRESSO BRASILEIRO DE INFORMÁTICA NA EDUCAÇÃO. **Anais...** [S.l.: s.n.], 2016. v.5, n.1, p.102.

ROY, S.; JOSHI, S.; KRISHNAPURAM, R. Automatic categorization of web sites based on source types. In: ACM CONFERENCE ON HYPERTEXT AND HYPERMEDIA. **Proceedings...** [S.l.: s.n.], 2004. p.38–39.

SÁ, H. d. Seleção de características para classificação de texto. **Trabalho de Graduação–Universidade Federal de Pernambuco, Pernambuco**, [S.l.], 2008.

SALTON, G. Automatic text processing: the transformation, analysis, and retrieval of. **Reading: Addison-Wesley**, [S.l.], 1989.

SALTON, G.; YANG, C.-S. On the specification of term values in automatic indexing. **Journal of documentation**, [S.l.], v.29, n.4, p.351–372, 1973.

SANTOS, R. E. et al. TÉCNICAS DE PROCESSAMENTO DE LINGUAGEM NATURAL APLICADAS AO PROCESSO DE MINERAÇÃO DE TEXTOS: resultados preliminares de um mapeamento sistemático. **Revista de Sistemas e Computação-RSC**, [S.l.], v.4, n.2, 2015.

SCHERBININ, V.; BUTAKOV, S. Plagiarism Detection: the tool and the case study. In: LEARNING. **Anais...** [S.l.: s.n.], 2008. p.304–310.

SENGIJPTA, S. K. **Fundamentals of statistical signal processing**: estimation theory. [S.l.]: Taylor & Francis, 1995.

SILVA, A. et al. Avaliando a similaridade semântica entre frases curtas através de uma abordagem híbrida. In: BRAZILIAN SYMPOSIUM IN INFORMATION AND HUMAN LANGUAGE TECHNOLOGY, 11. **Proceedings...** [S.l.: s.n.], 2017. p.93–102.

SILVA, O. S. F. Entre o plágio e a autoria: qual o papel da universidade. **Revista Brasileira de Educação**, [S.l.], v.13, n.38, p.357–368, 2008.

SIMOES, A.; GUINOVART, X. G. Bootstrapping a Portuguese wordnet from Galician, Spanish and English wordnets. In: **Advances in Speech and Language Technologies for Iberian Languages**. [S.l.]: Springer, 2014. p.239–248.

SIMON, H. A. Why should machines learn? In: **Machine learning**. [S.l.]: Springer, 1983. p.25–37.

SMOLA, A. et al. Introduction to large margin classifiers. In: **Advances in large margin classifiers**. [S.l.]: MIT Press, 2000.

STAPPENBELT, B.; ROWLES, C. The effectiveness of plagiarism detection software as a learning tool in academic writing education. In: ASIA PACIFIC CONFERENCE ON EDUCATIONAL INTEGRITY (4APCEI), 4. **Anais...** [S.l.: s.n.], 2010. p.29.

THEODORIDIS, S.; KOUTROUMBAS, K. Pattern recognition and neural networks. **Machine Learning and Its Applications**, [S.l.], p.169–195, 2001.

VAPNIK, V. **The nature of statistical learning theory**. [S.l.]: Springer science & business media, 2013.

VAPNIK, V. N.; VAPNIK, V. **Statistical learning theory**. [S.l.]: Wiley New York, 1998. v.1.

VIEIRA, R.; LOPES, L. PROCESSAMENTO DE LINGUAGEM NATURAL E O TRATAMENTO COMPUTACIONAL DE LINGUAGENS CIENTÍFICAS. **EM CORPORA**, [S.l.], p.183, 2010.

WANG, Y.-W. University student online plagiarism. **International Journal on ELearning**, [S.l.], v.7, n.4, p.743, 2008.

WANKHEDE, S. B. Analytical Study of Neural Network Techniques: som, mlp and classifier-a survey. **IOSR J. Comput. Eng.(IOSR-JCE)**, [S.l.], v.16, n.3, p.86–92, 2014.

WILLIAMS, D.; HINTON, G. Learning representations by back-propagating errors. **Nature**, [S.l.], v.323, n.6088, p.533–538, 1986.

ZHAO, J.; ZHU, T.; LAN, M. ECNU: one stone two birds: ensemble of heterogenous measures for semantic relatedness and textual entailment. In: SEMEVAL@ COLING. **Anais...** [S.l.: s.n.], 2014. p.271–277.

Apêndice

A

Referências da Revisão

Tabela A.1: Lista de conferências e periódicos da área de Educação.

Área	Nome Conferência
Educação	International Conference on Computers in Education
Educação	Proceedings of Conference on advanced technology for education
Educação	International Conference on Computers and Advanced Technology in Education
Educação	IEEE International Conference on Advanced Learning Technologies
Educação	International Journal on E-Learning
Educação	Journal of Distance Education Technologies
Educação	Journal of Education and Information Technologies
Educação	Frontiers in education conference
Educação	Journal of Educational Computing Research
Educação	International Conference on e-Learning
Educação	Proceedings of the congress e-learning
Educação	Artificial Intelligence in Education
Educação	Intelligent Tutoring Systems
Educação	Adaptive Hypermedia
Educação	international Conference on Artificial Intelligence in Education
Educação	International Conference on Advances in Web-based Learning
Educação	Computer & Education Journal
Educação	Conference on Educational Data Mining
Educação	Conference on Open Learning and Distance Education
Educação	Journal of Educational Technology Systems
Educação	Journal of Interactive Learning Research
Educação	Conference on Information Technology for Application
Educação	Proceedings of the Web-based Education
Educação	Electronic Journal of e-Learning
Educação	Simpósio Brasileiro de Informática na Educação
Educação	Conferência Latino-Americana de Objetos e Tecnologias de Aprendizagem
Educação	Workshop de Desafios da Computação aplicada à Educação
Educação	Conferência Internacional sobre Informática na Educação
Educação	Revista Novas Tecnologias na Educação
Educação	Revista Brasileira de Informática na Educação

Tabela A.2: Lista de conferências e periódicos da área de Inteligência Artificial (AI).

Área	Nome Conferência
IA	AI Communications
IA	Journal of Artificial Intelligence Research
IA	Artificila inteligence
IA	Conference on Artificial Intelligence
IA	International Conference on Web Information Systems Engineering
IA	WWW Conference
IA	European conference on artificial intelligence
IA	Knowledge Discovery and Data Mining
IA	Computers in Human Behavior
IA	International Conference on Data Engineering
IA	International Conference on Machine Learning Applications
IA	Brazilian Conference on Intelligence Systems
IA	Expert Systems with application
IA	Artificial Intelligence Review
IA	International Joint Conference on Artificial Intelligence

Tabela A.3: Lista de conferências e periódicos da área de Processamento de Linguagem Natural (PLN).

Área	Nome Conferência
PLN	International Conference on the Computational Processing of Portuguese
PLN	Computer Speech and Language
PLN	North American Chapter of the Association for Computational Linguistics
PLN	Annual Meeting of the Association for Computational Linguistics
PLN	Interspeech
PLN	Conference on Computational Natural Language Learning
PLN	Special Interest Group on Discourse and Dialogue
PLN	International Conference on Natural Language Generation
PLN	Conference on Empirical Methods on Natural Language Processing
PLN	International Conference on Computational Linguistics
PLN	Conference on Research and Development in Information Retrieval
PLN	European Chapter of the Association for Computational Linguistics

B

Código Fonte do Word2Vec

```
1  import java.io.File;
2  import org.deeplearning4j.*;
3
4  public class TreinandoModelo {
5
6  public static void main(String[] args){
7
8  //DEFININDO CORPUS DE TREINAMENTO
9  SentenceIterator iter = new LineSentenceIterator(new File("corpus.
       txt"));
10 iter.setPreProcessor(new SentencePreProcessor() {
11 public String preProcess(String sentence) {
12 return sentence.toLowerCase();
13 }
14 });
15
16 TokenizerFactory t = new DefaultTokenizerFactory();
17 t.setTokenPreProcessor(new CommonPreprocessor());
18
19 Word2Vec vec = new Word2Vec.Builder()
20 .minWordFrequency(5)
21 .iterations(10)
22 .layerSize(250)
23 .seed(42)
24 .windowSize(10)
25 .iterate(iter)
26 .tokenizerFactory(t)
27 .build();
28
29 vec.fit();
```

```
30
31  //SALVANDO MODELO TREINADO
32  WordVectorSerializer.writeFullModel(vec, "Model.txt");
33  }
34  }
```

Listing B.1: Código do word2vec que realiza o treinamento.

```
1   import java.io.File;
2   import org.deeplearning4j.*;
3
4   public class LendoModelo {
5
6   public static void main(String[] args){
7
8   //LENDO MODELO TREINADO
9   Word2Vec vec = WordVectorSerializer.loadFullModel("Model.txt");
10
11  System.out.println("Similaridade entre as palavras: " + vec.
        similarity("palavra1", "palavra2"));
12  }
13  }
```

Listing B.2: Código do word2vec para calcular a similaridade entre palavras.