UNIVERSIDADE FEDERAL RURAL DE PERNAMBUCO
CURSO DE BACHARELADO EM CIÊNCIA DA COMPUTAÇÃO

NOVA MEDIDA DE SIMILARIDADE ENTRE SENTENÇAS PARA ELIMINAÇÃO DE REDUNDÂNCIA EM SUMARIZAÇÃO MULTI-DOCUMENTO

LUCAS DORNELLES BARBOSA MAIA

RECIFE
2017

UNIVERSIDADE FEDERAL RURAL DE PERNAMBUCO

LUCAS DORNELLES BARBOSA MAIA

NOVA MEDIDA DE SIMILARIDADE ENTRE SENTENÇAS PARA ELIMINAÇÃO DE REDUNDÂNCIA EM SUMARIZAÇÃO MULTI-DOCUMENTO

Monografia apresentada ao Curso de Bacharelado em Ciência da Computação da Universidade Federal Rural de Pernambuco como requisito parcial para obtenção do título de Bacharel em Ciência da Computação.

MINISTÉRIO DA EDUCAÇÃO E DO DESPORTO
UNIVERSIDADE FEDERAL RURAL DE PERNAMBUCO (UFRPE)
BACHARELADO EM CIÊNCIA DA COMPUTAÇÃO

http://www.bcc.ufrpe.br

FICHA DE APROVAÇÃO DO TRABALHO DE CONCLUSÃO DE CURSO

Trabalho defendido, no dia 22 de agosto de 2017 às 15 horas, no Auditório do CEAGRI-02 - Sala 07, por Lucas Dornelles Barbosa Maia como requisito para conclusão do curso de Bacharelado em Ciência da Computação da Universidade Federal Rural de Pernambuco, intitulado **Nova Medida de Similaridade entre Sentenças para Eliminação de Redundância em Sumarização Multi-Documento**, orientado por Rafael Ferreira Leite de Mello e aprovado pela seguinte banca examinadora:

Rafael Ferreira Leite de Mello
DEINFO/UFRPE

Péricles Barbosa Cunha de Miranda
DEINFO/UFRPE

Rinaldo José de Lima
DEINFO/UFRPE

AGRADECIMENTOS

Agradeço primeiramente a Deus. Ao meu pai, Mariano que é a minha vida e permitiu que tudo isso fosse possível. A Sandra, minha madrasta que tenho como uma mãe e que sempre cuidou de mim. A minha namorada Larissa, pelo amor, e todo apoio que sempre esteve comigo em todos os momentos. A minha mãe Vilma, que mesmo estando muito longe, pode contribuir na minha formação fundamental e conselhos. Ao meu irmão Matheus por me aturar todos os dias. A minha avó pelo companheirismo nos momentos que eram possíveis.

Em especial agradeço ao meu orientador, Rafael Ferreira, pela oportunidade que me deu, paciência e acompanhamento deste trabalho.

Agradeço a Universidade Federal Rural de Pernambuco, a todos os docentes e funcionários por todo suporte dado nessa trajetória.

Agradeço também a todos os meus amigos da universidade que contribuiram diretamente no meu sucesso durante minha caminhada no curso.

RESUMO

Com a rápida popularização da Internet e a quantidade de informações que surgem a cada momento, particularmente as de documento de texto, a necessidade de recuperação dessas informações em tempo hábil economizando o máximo de recursos possíveis tornou-se imprescindível. Contudo, mesmo com a utilização de vários métodos de sumarização automática de texto em multi-documento, problemas como redundância que influenciam na perda de informatividade do sumário são evidentes. Uma solução para o problema de redundância é utilizar um algoritmo de agrupamento baseado em grafos. O algoritmo de agrupamento combina métricas estatísticas com tratamento linguístico nas suas arestas. Este trabalho propõe uma nova aresta para o algoritmo de agrupamento, sendo uma nova medida de similaridade entre sentenças para eliminação de redundância em sumarização multi-documento. As avaliações realizadas contra sistemas do DUC 2002, apresentaram que a nova medida de similaridade alcançou resultados muito melhores para métrica F-Measure.

Palavras-chave: *word embeddings*, *Word2vec*, similaridade.

ABSTRACT

With the rapid popularization of the Internet and the amount of information that comes up at every moment, particularly the text document, the need to recover this information in a timely manner saving the maximum possible resources has become indispensable. However, even with the use of multiple methods of automatic multi-document text summarization, problems such as redundancy that influence the loss of summary informationality are evident. One solution to the redundancy problem is to use a graphing-based clustering algorithm. The clustering algorithm combines statistical metrics with linguistic treatment on its edges. This paper proposes a new edge for the clustering algorithm, being a new measure of similarity between sentences for redundancy elimination in multi-document summarization. The evaluations performed against DUC 2002 systems showed that the new measure of similarity achieved much better results for F-Measure metrics.

Keywords: *word embeddings*, *Word2vec*, similarity.

LISTA DE FIGURAS

LISTA DE TABELAS

LISTA DE ABREVIATURAS, SIGLAS E SÍMBOLOS

TS	Text Summarization
TF	*Term Frequency*
IDF	*Inverse Document Frequency*
DUC	*Document Understanding Conferences*
gold	*Gold summary*
PLN	*Processamento de Linguagem Natural*

SUMÁRIO

1. INTRODUÇÃO

Com a rápida popularização da Internet e a quantidade de dados que surgem a cada momento, particularmente as de documento de texto, a necessidade de recuperação dessas informações em tempo hábil, economizando o máximo de recursos possíveis tornou-se imprescindível. Devido à grande massa de dados oriundas da Internet, verificou-se a inviabilidade da obtenção de informação relevante de forma ágil e precisa. Com isso a necessidade de criar métodos automáticos para a compreensão, indexação e classificação das informações de uma forma clara e concisa, para viabilizar aos usuários poupar tempo e recursos(FERREIRA,2013).

Técnicas de sumarização de texto provém uma solução para o problema das grandes quantidades de informações. O modo de produzir automaticamente uma versão sintetizada de um ou mais documentos é chamada de sumarização de texto(NENKOVA & McKeown,2012). Um resumo preciso, deve conseguir uma cobertura de várias partes do documento, a fim de que a possibilidade de redundância seja mínima. Os métodos de sumarização de texto podem ser classificados em sumarização extrativa e abstrativa(GUPTA,2010). O resumo extrativo consiste em selecionar partes relevantes do documento original, e conectar para produzir uma versão menor do texto. A relevância dessas sentenças, para serem escolhidas, leva em consideração, estatísticas e características linguísticas das sentenças. Já um resumo abstrativo tenta desenvolver uma compreensão dos conceitos fundamentais de um documento, para logo após explanar esses conceitos de forma clara e natural. O resumo abstrativo utiliza métodos linguísticos para examinar e interpretar as parcelas do texto e, logo depois, encontrar novos conceitos e expressões para reproduzi-lo melhor através da criação de um texto mais curto que passe a informação mais relevante do texto original.

As mesmas técnicas utilizadas em sistemas de sumarização de um único documento aplicam-se a documentos múltiplos. A sumarização automática multi-documento consiste na produção automática de um único sumário a partir de um grupo de textos sobre um mesmo tópico ou sobre tópicos relacionados a fim de se

recuperar a informação mais relevante. De acordo com (FERREIRA,2014) em uma coleção de textos sobre o mesmo assunto ou um único tópico(ou alguns tópicos), a probabilidade de encontrar sentenças semelhantes é significativamente maior do que o grau de redundância dentro de um único texto.

Para lidar com o problema da redundância podem ser usado algoritmos de agrupamento de sentenças (Cohn, Verma, & Pfleger, 2006). Baseado nisso, Ferreira et al.(2013) apresenta um algoritmo para converter o texto em um modelo de grafo contendo quatro tipos de relações entre sentenças: (i) similaridade estatística; (ii) similaridade semântica; (iii) correferência e (iv) relações de discurso. Através da representação de grafo foi aplicado para eliminar redundância(FERREIRA,2014).

Esta pesquisa propõe um novo método de similaridade entre sentenças baseada em *word embeddings* como uma nova dimensão para o algoritmo de agrupamento de sentenças proposto(FERREIRA,2014).Além disto, foi realizado uma avaliação detalhada de todas as combinações possíveis entre as arestas do grafo levando em conta as 4 relações originais, mais a proposta neste trabalho.

O foco dessa pesquisa é a sumarização de texto extrativa em multi-documento, pois tende a ser menos custoso e geralmente antecede o método abstrativo(Lloret & Palomar,2012).

Para avaliar a proposta foi utilizado o conjunto de dados do *Document Understanding Conference* 2002 contra os sistemas submetidos a essa conferência. Dois experimentos diferentes foram realizados seguindo as orientações do DUC 2002: Para cada coleção de documentos foram gerados resumos com 200 e 400 palavras.

1.1 JUSTIFICATIVA

O processo de sumarização extrativa de texto segundo (GUPTA,2010) pode ser dividido em duas etapas: 1) etapa de pré-processamento e 2) etapa de processamento.

Na etapa de pré-processamento o texto é uma representação estruturada do texto original. Em geral, observa-se (a) os limites das sentenças, ou seja, a presença do ponto no final da sentença; (b) "*Stop-Words*" quando isoladas não transmitem semântica e não agregam informações relevantes para o resumo, sendo assim, eliminadas; (c) "*Stemming*", cujo propósito é obter o radical de cada palavra, a fim de enfatizar sua semântica.

Na etapa de processamento, características que influenciam a relevância de sentenças extraídas do texto, sendo assim calculados e atribuídos pesos usando algum método de aprendizagem. A pontuação final de cada sentença é determinada usando uma equação característica de peso. O topo do ranking das sentenças são selecionados para o resumo final.

As mesmas técnicas utilizadas na sumarização automática de um único documento aplicam-se a multi-documento. Algumas questões como grau de redundância e diversidade de informação aumentam na sumarização de multi-documento. Em uma coleção de textos sobre um único tópico ou tópicos relacionados, a probabilidade de encontrar sentenças semelhantes é significativamente maior que o grau de redundância dentro de um único texto(FERREIRA,2014).

A partir do conhecimento de todas essas informações, foi identificado a necessidade de uma nova configuração de aresta para o algoritmo de grafo proposto por Ferreira et al. (2014) baseado no modelo de *word embeddings*.

1.2 OBJETIVOS

Esta seção contém os objetivos gerais e específicos que o trabalho visa atingir.

1.2.1 Geral

Propor uma nova medida de similaridade baseado em *word embeddings* para melhorar o algoritmo de agrupamento baseado em grafo aplicado a sumarização de texto multi-documento.

1.2.2 Específicos

Visando atingir o objetivo geral, alguns objetivos específicos são apresentados, entre eles:

1. Propor uma medida de similaridade baseada em *word embeddings*.
2. Estudo sobre diferentes combinações de arestas em grafos de texto.
3. Avaliar diferentes configurações para algoritmo de agrupamento.

1.3 ESTRUTURA DO TRABALHO

No primeiro capítulo deste trabalho foram apresentados a introdução, justificativas, objetivos do tema e metodologia adotada. No segundo capítulo, são relacionados assuntos reconhecidos como pré-requisito para o total entendimento do trabalho. Estes assuntos contemplam a fundamentação teórica. No terceiro capítulo são apontados os trabalhos relacionados ao tema escolhido. O quarto capítulo descreve a implementação da ferramenta proposta. No quinto capítulo são exibidos os experimentos realizados e resultados obtidos. O sexto capítulo trata sobre as conclusões e trabalhos futuros.

2. FUNDAMENTAÇÃO TEÓRICA

Este capítulo trata os temas necessários para a melhor compreensão dos tópicos abordados nesta pesquisa.

2.1 SUMARIZAÇÃO AUTOMÁTICA DE DOCUMENTOS

Para (NENKOVA & McKeown,2012) os sistemas de sumarização automáticos de texto precisam produzir um resumo conciso e fluente, transmitindo as informações-chave. Esses sumarizadores identificam as sentenças mais importantes da entrada, que pode ser um único ou um conjunto de documentos, para depois de processados, formarem um resumo. Além disso algumas tarefas independentes são realizadas praticamente por todos os sumarizadores: Criar uma representação intermediaria da entrada, pontuação de sentenças e seleção de sentenças. A representação intermediaria consiste na ideia de transformar a entrada em um modelo que possa formatar os dados de forma que depois seja possível aplicar os dois passos seguintes. O *TF-IDF* por exemplo pode prover as palavras e seus pesos correspondentes, sendo as mais ponderadas as palavras mais indicativas do tópico; Abordagens de cadeia léxica podem prover um dicionário de sinônimos como o Wordnet para encontrar tópicos ou conceitos de palavras semanticamente relacionadas e com seus respectivos pesos; Modelos baseados em grafos como o *LexRank* e o *TextRank* representam a entrada de texto através de vértices e arestas por exemplo.

A partir da entrada formatada em uma representação intermediaria, é atribuída uma pontuação a cada sentença indicando sua importância. E por fim o sumarizador automático tem de selecionar a melhor combinação de sentenças relevantes para formar o sumário final.

2.2 PRÉ-PROCESSAMENTO

Antes de serem aplicadas as técnicas de similaridade textual, alguns métodos de pré-processamento foram utilizados a fim de preparar melhor a entrada de texto. Segundo Ferreira et al. (2013) Dois aspectos foram usados, a análise estrutural e de texto. O primeiro aspecto consiste em dividir o texto, enquanto a análise de texto fornece a remoção de *stop words, POS tagging* e *lemmatization.* Os métodos podem ser vistos como:

1. Análise Estrutural
    - Tokenization: Faz a divisão do texto em palavras.
    - Sentence Splitter: Faz a divisão de parágrafos em sentenças.
    - Paragraph Splitter: Faz a divisão do texto em parágrafos.

    2. Análise de Texto
    - Stop Words: remove as palavras com um pequeno valor representativo para o documento, como artigos e pronomes.
    - POS Tagging: Associa a classificação morfológica para um texto em inglês.
    - Lemmatization: Mostra as formas verbais, como o infinitivo por exemplo e substantivos na forma singular.

2.3 TÉCNICAS DE SIMILARIDADE TEXTUAL

Esta seção apresenta as principais técnicas de similaridade textuais utilizadas no trabalho.

2.3.1 SIMILARIDADE ESTATÍSTICA

A similaridade entre sentenças mede o conteúdo de sobreposição entre pares de sentenças para criar as arestas. Caso o método exceda uma pontuação limite, selecionada pelo usuário, então a aresta entre o par de sentenças é criada. A

medida de similaridade do cosseno utilizado pelo Word2vec para cálculo das sentenças por exemplo, faz parte desse método de similaridade. Além disto outras medidas de similaridade fazem parte da similaridade estatística.

A centralidade, segundo Abuobieda et al. (2012), sempre que o vocabulário de uma sentença se sobrepõe com as outras sentenças em um documento, uma das sentenças vai expressar sobre a centralidade da sentença. Esta medida de similaridade pode ser calculada da seguinte forma:

$$Pontuação = \frac{Pc \cap POc}{Pc \cup POc}$$

Fonte: O autor.

Sendo, Pc as palavras-chave em c e POc as palavras-chave em outras sentenças.

A entropia, ou a entropia de informação, é uma medida de incerteza associada a uma variável aleatória e também quantifica informações em dados. Considerando um par de sentenças (Sa,Sb), e o número de ligação entre elas λ , podemos ter uma combinação da seguinte forma: $p = \frac{\lambda}{|Sa|} X \frac{\lambda}{|Sb|}$, obtendo p como o valor de uma variável aleatória no intervalo [0,1]. A função de entropia pode ser dada como:

$$E(X) = -p\,x\log_2(p) - (1-p)\,x\log_2(1-p)$$

Fonte: O autor.

Quando a incerteza é máxima, esta função atinge o valor máximo de 1,0 para p = 0.5. E se p está próximo de 0 ou 1, então significa que estamos com um grau de certeza elevado sobre resultado(JOAO,2007).

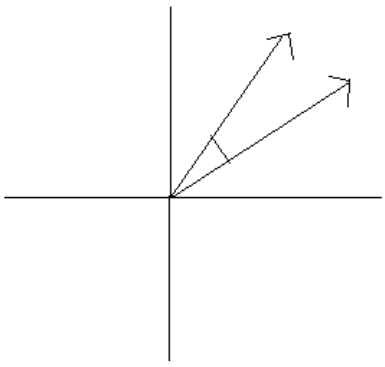
Coocorrência de palavras: Para Marino et al. (2006) a chance de dois termos de um texto aparecer ao lado um do outro em uma determinada ordem é chamado

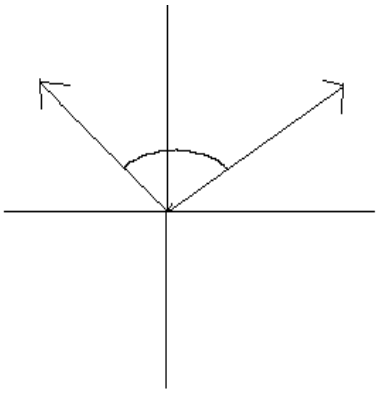
de coocorrência de palavras. A coocorrência de palavras é constituído de uma sequência de n itens de uma sequência de texto. Quanto maior for a pontuação das coocorrências das palavras, os termos mais frequentes aparecem em sequência.

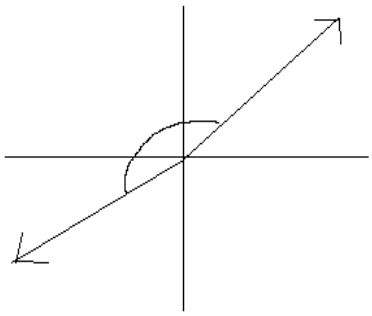
A similaridade do cosseno entre dois vetores é uma medida que calcula o cosseno do ângulo entre eles. Esta métrica é uma medida de orientação e não de magnitude, ou seja, a similaridade vai ser influenciada pelo ângulo das palavras em um espaço normalizado por exemplo e não pela frequência de vezes que ela aparece. A equação pode ser vista como:

$$\cos(\theta) = \frac{\vec{a} * \vec{b}}{\|\vec{a} * \vec{b}\|}$$

Onde a e b, são componentes do vetor a e b respectivamente.
Podemos observar como a pontuação da similaridade do cosseno é dada nas figuras abaixo:







Figuras 1: Similaridade do cosseno.
Fonte: O autor.

Podemos observar que os vetores na mesma direção, com o ângulo próximo de 0 grau, a similaridade do cosseno para este ângulo é próximo de 1. Já com os vetores quase ortogonais, com o ângulo entre eles próximo a 90 graus, a similaridade do cosseno para este ângulo é próximo de 0. E com os vetores em posições opostas, com o ângulo entre eles próximo a 180 graus, a similaridade do cosseno para este ângulo é próximo de -1.

Assim podemos, por exemplo, medir a similaridade do cosseno de algumas palavras em relação a Suécia(*Sweden*) usando o *Word2vec*, em ordem de proximidade.

Word	Cosine distance
norway	0.760124
denmark	0.715460
finland	0.620022
switzerland	0.588132
belgium	0.585835
netherlands	0.574631
iceland	0.562368
estonia	0.547621
slovenia	0.531408

Figura 2: Exemplo similaridades Word2vec.

Fonte: DeepLearning4J,DL4J. Disponível em https://deeplearning4j.org/word2vec. Acesso em 30 de julho de 2017.

A Suécia é igual a Suécia, enquanto a Noruega(N*orway)* tem uma distância de cosseno de 0,76014 da Suécia, a mais alta de qualquer outro país, Figura 5.

2.3.2 SIMILARIDADE SEMÂNTICA

A similaridade semântica mede a semelhança semântica entre palavras em uma sentença. As principais etapas para obter o resultado da semelhança semântica das sentenças são:

1. As sentenças são representadas como um vetor de palavras. Apenas são mantidos os substantivos.

2. As pontuações de similaridade semântica para cada par de palavras entre duas frases é calculada.

3. Os resultados são combinados pela soma das pontuações.

4. Os resultados finais são normalizados. Retornando valores entre [0,1].

Essa medida de similaridade apenas é calculada se as instâncias de ambas as palavras aparecerem no *Wordnet*, caso contrário o valor de pontuação para o par é zero.

A métrica de *Path* calcula a relação semântica de sentido das palavras, contando o número de nós ao longo do caminho mais curto entre esse sentido, através da hierarquia para esta métrica do *Wordnet*. Quando maior o comprimento do caminho, menor será a relação entre essas palavras(WUBBEN,2009).

2.3.3 ANÁLISE DE CORREFERÊNCIA

A análise de correferência procura encontrar as menções no texto que se referem à mesma entidade do mundo real(CLARK,2008). Para Luo (2007) Uma referência de frase para uma entidade é chamada de menção. Um conjunto de menções referentes ao mesmo objeto físico pertence à mesma entidade. Por exemplo, na seguinte frase:

John disse que Mary era sua irmã.

Existem quatro menções: John, Mary, sua e irmã. John e sua pertencem à mesma entidade uma vez que se referem à mesma pessoa; Mary e irmã também se referem a uma outra pessoa. Além disso, John e Mary são nomeados menções, a palavra irmã é uma menção nominal e a palavra sua é uma menção pronominal. Quando encontrado uma relação de correferência, a aresta do grafo é construída.

2.3.4 RELAÇÕES DE DISCURSO

As relações de discurso podem ser descritas como uma coleção de frases que possuem alguma relação entre si. A tabela a seguir exibe um conjunto de relações de discurso:

Table 1
Contentful conjunctions used to illustrate coherence relations.

Cause–effect	because; and so
Violated expectation	although; but; while
Condition	if . . . (then); as long as; while
Similarity	and; (and) similarly
Contrast	by contrast; but
Temporal sequence	(and) then; first, second, . . . ; before; after; while
Attribution	according to . . . ; . . . said; claim that . . . ; maintain that . . . ; stated that . . .
Example	for example; for instance
Elaboration	also; furthermore; in addition; note (furthermore) that; (for, in, on, against, with, . . .) which; who; (for, in, on, against, with, . . .) whom
Generalization	in general

Figura 3: Relações de discurso baseadas em conjunções de conteúdo

Fonte: (WOLF,2005)

As relações de discurso apresentadas na Figura 1, apresentam conjuntos baseados em conjunções de conteúdo, ou seja, entidades que apresentam relações entre si(WOLF,2005).

2.4 WORD EMBEDDINGS

Segundo (LEVY,2014) a representação de palavras é fundamental para o PLN. A abordagem padrão de representação de palavras como símbolos discretos e distintos são insuficientes para muitas tarefas, e sofre com uma pobre generalização. Por exemplo, a representação simbólica das palavras “pizza” e “hambúrguer” são completamente independentes, mesmo que soubéssemos disso, a palavra “pizza” é um bom argumento para o verbo “comer”, mas não podemos inferir que o “hambúrguer” é também um bom argumento. Um paradigma muito comum para representações que buscam semelhanças semânticas e sintáticas entre as palavras é a distribuição da hipótese de Harris(1954), que afirma que as palavras em contextos semelhantes têm significados semelhantes. A partir disso muitos métodos foram explorados pela comunidade de PLN, e o mais recente proposto para representar palavras através de vetores densos que são derivados por vários métodos de treinamento inspirados por redes neurais são denominados

de “*word embeddings*” ou “neural *embeddings*”, que tem demonstrado um bom desempenho em uma variedade de tarefas. Ainda segundo (LEVY,2014) modelos baseados em *word embeddings* são fáceis de trabalhar porque permitem um calculo eficiente da similaridade entre as palavras por meio de operações de matrizes de baixa dimensão. A ferramenta *Word2vec* utilizado nesta pesquisa, é um tipo de word embedding e pode treinar e carregar corpus com bilhões de palavras com grandes dimensões.

O *Word2vec* é uma rede neural de duas camadas que são treinados para reconstruir contextos linguísticos de palavras. Sua entrada é um corpus de texto e sua saída é um conjunto de vetores de características. De modo que no espaço vetorial produzido, cada palavra única do corpus é atribuída a um vetor correspondente no espaço, de forma que as palavras que compartilham contextos em comum no corpus estão localizadas próximas uma das outras no espaço vetorial(MIKOLOV,2013).

O *Word2vec* treina as palavras do corpus de entrada de duas maneiras, usando o contexto para prever uma palavra-alvo (*CBOW*), ou usando uma palavra para prever um contexto de destino, que é chamado de *Skip-Gram*.

O modelo utilizado nesta pesquisa do *Word2vec* foi o *Skip-Gram*. Segundo Mikolov et al.(2013), por não envolver multiplicações de matriz densa, o treinamento do modelo se torna extremamente eficiente. Uma implementação otimizada de uma única máquina pode treinar mais de 100 bilhões de palavras em um dia.

2.4.1 SKIP-GRAM

O modelo usado *Skip-Gram*, introduzido por Mikolov et al. (2013), apresenta um método eficiente com uma alta qualidade de aprendizado para representação de vetores de palavras de grandes quantidades de dados de texto não estruturados. A arquitetura do modelo *Skip-gram*. O objetivo do treinamento do modelo é achar representações de palavras que sejam úteis para prever palavras em torno de uma sentença ou um documento.

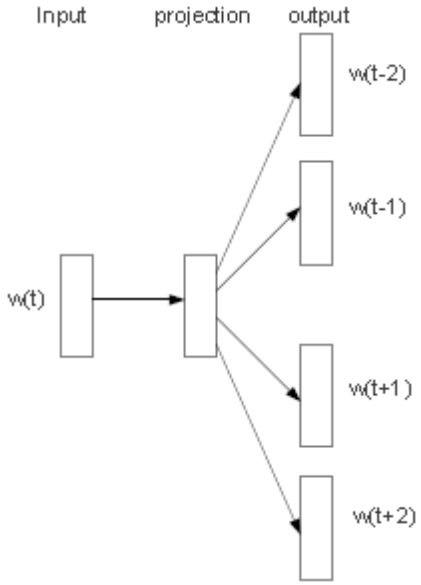


Figura 4: Arquitetura do modelo *Skip-Gram*.
Fonte : MIKOLOV(2013)

2.5 MODELO DE GRAFO PARA SUMARIZAÇÃO MULTI-DOCUMENTO

O modelo de grafo para sumarização de texto multi-documento proposto por Ferreira et al. (2013) e também utilizado para eliminar redundâncias em (FERREIRA,2014) possui seis passos para criar o agrupamento de texto. A entrada do algoritmo recebe um grafo e um arquivo de configuração. O grafo recebido como entrada é representado como vértices sendo sentenças e arestas como os métodos descritos na seção anterior desta pesquisa. No arquivo de configuração, algumas informações de orientação são definidas, como: o limite para medir a importância de um vértice, cálculo da pontuação do *TextRank,* escolha da aresta, tipo do grafo, idioma e domínio.

A segunda etapa calcula a pontuação do *TextRank* para cada vértice usando o método escolhido de aresta. A partir disto, são extraídas as palavras-chaves e determina um peso que se refere a importância das sentenças dentro do documento.

Na terceira etapa, a seleção do vértice principal é realizada, como sendo o de maior pontuação do *TextRank*.

A quarta etapa utiliza o valor limite fornecido no arquivo de configuração pelo usuário e os resultados do *TextRank* para identificar os vértices lideres. Cada um vértices líderes cria um grupo.

Na quinta etapa, o caminho mais curto é calculado. Para cada vértice, o algoritmo calcula o caminho mais curto entre ele e cada vértice líder utilizando o algoritmo de Dijkstra.

A sexta etapa identifica o líder mais próximo, e na última etapa são removidos todos os caminhos que ligam um vértice a um líder, que são diferentes do líder mais próximo identificado no passo anterior.

A saída do algoritmo retorna n grafos, onde *n* é o número de vértices líderes, que representam os *clusters*.

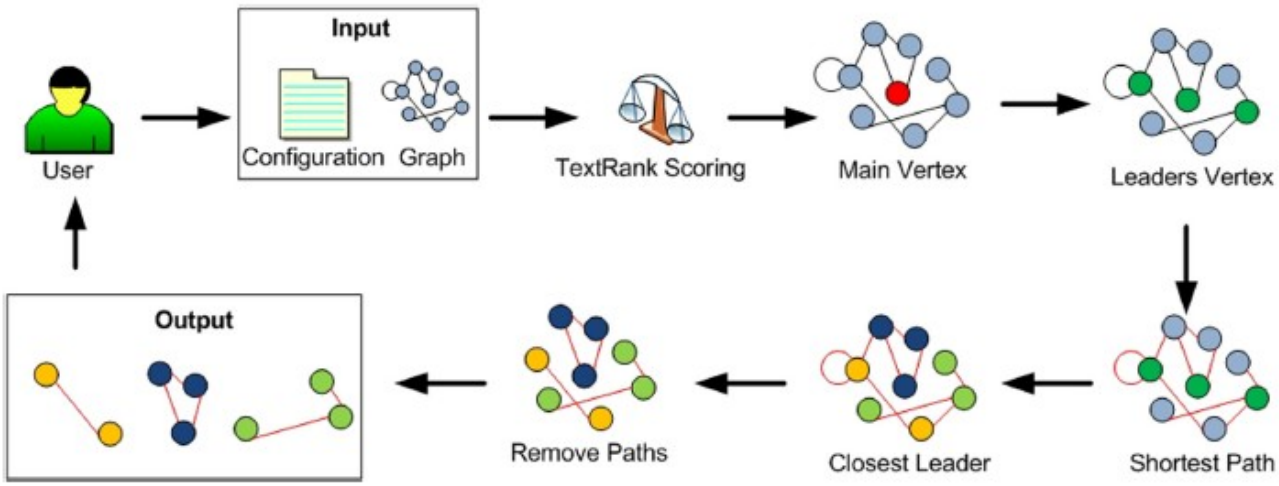


Figura 5: Fluxo das etapas do algoritmo de agrupamento

Fonte:(FERREIRA,2014).

3. TRABALHOS RELACIONADOS

O processamento de linguagem natural(PLN) consiste no desenvolvimento de modelos computacionais para a realização de tarefas que dependem de informações expressas em alguma linguagem natural.

Na área de PLN, novas abordagens de sistemas automáticos de sumarização de textos são estudados sob diferentes pontos de vista. Esta seção apresenta trabalhos relacionados a diferentes tipos de técnicas de sumarização para um único documento e multi-documento.

No documento de Regina Barzilay e Michael Elhadad(1999) um novo algoritmo foi proposto para calcular cadeias léxicas de um texto, fundindo várias fontes de conhecimento robustas como: *WordNet thesaurus*, *part-of-speech tagger*, *shallow parser* e um algoritmo de segmentação. A produção do resumo segue quatro etapas: o texto original é segmentado, cadeias léxicas são construídas, cadeias fortes são identificadas e as sentenças relevantes são extraídas.

A pesquisa de Rada Mihalcea e Hakan Ceylan(2007) concentrou-se em explorar a sumarização automática para livros. A maioria dos livros das coleções testadas tinham uma média de comprimento de 50.000 a 150.000 palavras, com um resumo de 2.000-6.000 palavras. A métrica de avaliação usada foi o *ROUGE,* incluindo o ROUGE-1,ROUGE-2 e o ROUGE-SU4 como as utilizadas nas amostras. A pesquisa utilizou abordagens existentes do estado da arte, fez reimplementações e combinações entre eles aplicados a livros. O trabalho ainda destaca dois pontos, o primeiro é que a maioria das pesquisas até agora tem se preocupado com o resumo de documentos curtos(Nesse contexto, a pesquisa tentou resolver essa lacuna, abordando o problema de sumarização de livros) e mostrou-se através dos resultados, que sistemas desenvolvidos para sumarização de documentos curtos não se saem bem quando aplicada a documentos de grande comprimento, tais como livros, e em vez disso pode ser alcançado um melhor resultado com um sistema que contabiliza o comprimento dos documentos.

O trabalho dos autores Sicui Wang,Weijang Li, Feng Wang, Hui Deng(2010) categoriza e descreve cinco técnicas de sumarização automática de textos: extração automática, compreensão baseada em sumarização automática, extração de informação, sumarização automática com base no discurso, e sumarização automática com base na facilidade de consulta. Através das pesquisas realizadas o intuito era realizar sínteses satisfatórias. No entanto, devido à flexibilidade da linguagem natural e à capacidade limitada de processamento do computador para linguagem natural, os resumos gerados por técnicas de sumarização automáticas existentes são incapazes de atender à necessidade dos usuários. A pesquisa tenta integrar duas categorias(sumarização automática com base na facilidade de consulta e com base no discurso) que foram descritas no trabalho a fim de extrair o resumo do artigo original.

A dissertação de GUPTA(2010) apresenta métodos de sumarização extrativa, focando principalmente na maneira que a distribuição dos pesos é realizada, visto que, características individuais são muito importantes quanto à qualidade do resumo final que é produzido. Um desafio destacado também na pesquisa é em relação ao domínio da sumarização de texto. A sumarização de texto ainda tem muita dependência para produção de resumos eficazes para domínios específicos. Fatores como o idioma, por exemplo, são citados para usuários específicos.

O artigo de Eliseo Reategui, Miriam Klemann e Mateus David Finco(2012) apresenta uma ferramenta de mineração de texto SOBEK, que é capaz de extrair grafos de textos e propõe seu uso para ajudar alunos a escrever resumos. A ideia baseia-se na utilização dos grafos como organizadores das palavras chaves relacionadas com o texto. A ferramenta *SOBEK* foi desenvolvida com um algoritmo de mineração baseado na distância *n-simple* de um grafo, ou seja, em que os nós representam os principais termos encontrados no texto, e as arestas representam informações de adjacência. O artigo foi capaz de produzir grafos que estavam próximos do que considerado importante sobre um texto lido pelos alunos, mas não perfeito demais para não lhes dar espaço para expressar suas ideias sobre as informações mais relevantes.

A pesquisa de Ani Nenkova e Kathleen McKeown(2012) foca principalmente na forma que os termos do texto vão ser capturados para a produção do resumo final. Uma representação baseada em tópicos deriva primeiro uma representação intermediaria do texto que capta os termos e marcam como importantes. Outra abordagem de representação de indicadores, o texto é representado por um conjunto de possíveis indicadores de importância que não visam a descoberta da atualidade. Quando esses indicadores são combinados, utilizam de técnicas de aprendizagem de máquina para marcação da relevância de cada sentenças e depois disso a produção do resumo final. O destaque fica para a medida KL divergência, como um método para sentenças de pontuação que incorporam diretamente uma intuição sobre as características de um bom resumo.

No trabalho de Elena Lloret e Manuel Palomar(2012) uma ferramenta de sumarização de texto chamada compêndio é capaz de gerar resumos para diferentes fins e podendo também lidar com uma ampla gama de domínios. Embora utilize um método de vinculação textual para detecção de redundâncias para gerar e avaliar resumos, esta técnica não foi empregada para lidar com problemas de redundância em sumarização de texto.

A pesquisa de Yang(2013) foca numa metodologia para investigar a sumarização automática de texto no contexto de aprendizagem móvel. O principal objetivo da pesquisa foi avaliar os resultados de aprendizagem associados à leitura de resumos de texto. O estudo realizado de tal forma que características como a motivação, interesse na aprendizagem, qualidade do ensino, inteligência, experiência e educação, não teria nenhuma influência significativa sobre os resultados. As questões da pesquisa investigaram, por exemplo, se o conteúdo contem informações suficientes para apoiar os alunos na obtenção de um nível suficiente de aprendizagem e qual a melhor taxa de compressão para resumos. Foram cuidadosamente selecionados participantes para esse experimento, evitando aqueles que já possuem conhecimento prévio no contexto de conteúdo de aprendizagem móvel. Uma aplicação prática em aprendizagem móvel foi projetada e usada para conduzir o experimento que comparou o texto completo com resumos automatizados. O sistema foi desenvolvido no trabalho anterior dos autores desta

pesquisa e foi treinado no padrão do *DUC 2006* e avaliado na ferramenta *ROUGE.* A pesquisa mostrou resultados satisfatórios para sumarização para apoiar a aprendizagem móvel, porém, ainda existem limitações para determinar corretamente as diferenças semânticas ou semelhanças em sentenças.

O trabalho de Labeke(2013) apresenta um sistema chamado *OpenEssayist*, com o objetivo de fornecer uma solução de feedback interativo que produz um bom nível de apoio para estudantes universitários escreverem dissertações. O sistema encontra-se até a data de publicação do artigo em desenvolvimento e utiliza algoritmos de sumarização extrativa como principal técnica. A primeira versão do sistema concentrou-se na definição do mecanismo de análise das redações e integrar ao *OpenEssayist* que suporta apresentação, analise e elaboração de relatórios. Outras representações estão sendo projetadas, focando em listas simples de termos classificados(por meio de palavras e sentenças chaves) e soluções com grafos.

O foco do trabalho de FERREIRA(2013) está relacionado à qualidade dos métodos de sumarizações extrativas baseados na pontuação de sentenças. O documento explica e implementa estratégias de sumarização de textos encontrados na literatura nos últimos dez anos. No trabalho, 15 algoritmos de pontuação foram descritos e analisados. Foram selecionados os cinco melhores resultados obtidos com os diferentes conjuntos de teste: *Word Frequency*,*TF/IDF*, *Lexical Similarity* e *Sentence Lengt.* A estratégia *Text Rank Score* também foi escolhida por proporcionar bons resultados por dois dos três conjuntos de dados testados. A análise qualitativa utilizando ROUGE permitiu explanar alguns resultados interessantes: O *TF/IDF* considerávelmente o mais intensivo computacionalmente de todos os métodos testados. Os métodos de *Word Frequency* e *Sentence Length* proporcionam o melhor equilíbrio de desempenho em tempo de execução e em eleger sentenças relevantes. Estratégias para compor melhores resumos estão sendo atualmente investigadas.

A pesquisa de FERREIRA(2014) apresenta diferentes configurações de grafos para sumarização de texto multi-documento. Um novo algoritmo de

agrupamento identifica as sentenças relacionadas a diferentes tópicos abordados nos documentos a serem resumidos. Esse modelo de grafo é usado para representar o documento usando quatro relações diferentes entre as sentenças: (i) semelhança estatística; (ii) semelhança semântica; (iii) correferência; e (iv) relações de discurso. A ideia geral do algoritmo de agrupamento utilizado consiste em: (1) abrir todos os documentos de uma coleção de entrada e tratá-los como um único arquivo; (2) agrupar sentenças para encontrar sua relação com um tópico específico; (3) classifica sentenças para selecionar as com maior pontuação para compor cada grupo. Na maioria dos casos a correferência se sobressai entre as demais. O conjunto de dados mostrou que superou os demais sistemas concorrentes do DUC 2002.

O objetivo dos autores PadmaPriya, G. e K.Duraiswamy(2014) é usar um algoritmo de “deep learning” para melhorar a eficiência de um dos problemas da sumarização extrativa comum que é o das sentenças redundantes encontradas. O algoritmo utilizado foi o da Máquina Restrita de Boltzmann(RBM). Constituído por três camadas, uma de entrada, a camada do meio(“hidden”) e a camada de saída, os dados já preprocessados entram uniformemente para serem operados e gerar o resumo. A pesquisa teve um desempenho satisfatório para o problema de sumarização em multi-documentos, através da abordagem de pré-processamento adotada com características de pontuação das sentenças e o algoritmo *RBM*. Uma proposta futura seria considerar diferentes características e a adição de novas camadas “*hidden*” para o algorimo *RBM*.

Tendo em vista alguns dos problemas com a sumarização extrativa em multi-documento, principalmente por problemas de redundância e perda de informação, este trabalho pretende melhorar esses dois aspectos, propondo um novo método de sumarização de texto multi-documento com uma medida de similaridade baseada em word embeddings, e apresentar um estudo sobre diferentes combinações de métodos através de arestas em grafos de texto, e avaliar diferentes configurações para o algoritmo de agrupamento. A abordagem de grafo foi escolhida por ser uma ótima representação intermediaria para sumarização de textos e ter apresentado bons resultados na proposta de (FERREIRA,2014).

4. PROPOSTA

A proposta desta pesquisa é a inserção de uma nova medida de similaridade baseada em *word embeddings* com a utilização da ferramenta W*ord2vec* no modelo de grafo proposto por (FERREIRA,2014). O Word2vec agrupa os vetores de palavras semelhantes em um espaço vetorial, e detecta semelhanças matematicamente a fim de retornar as similaridades de palavras existentes no corpus que foi dado como entrada.

O algoritmo de agrupamento utilizado neste trabalho utiliza o *TextRank* para pontuar os vértices. Os algoritmos de ranqueamento baseados em grafos são essencialmente uma maneira de decidir a importância de um vértice dentro do grafo.

A ideia básica implementada por esse modelo de grafo é pela recomendação. Quando um vértice tem uma aresta ligando a outro vértice, uma recomendação é realizada para esse outro vértice. Quanto maior o número de recomendações lançados para um vértice, maior a importância dele(MIHALCEA,2004).

A entropia foi a medida de similaridade estatística utilizada neste trabalho por ter apresentado os melhores resultados na pesquisa de (FERREIRA,2014). Apesar de medidas de similaridade semântica como: *Resnik*;*Lin*; *Wu and Palmer*; *Path*; *Leacock and Chodorow* (FERREIRA,2013) terem sido utilizadas no artigo de agrupamento(FERREIRA,2014), neste trabalho foi utilizado a métrica de *Path* por ter sido a com melhor resultado.

4.1 INCLUSÃO DA NOVA MEDIDA DE SIMILARIDADE

A inserção da nova medida de similaridade no modelo de grafo proposto por (FERREIRA,2014) foi realizada através do ambiente de desenvolvimento eclipse e da linguagem de programação java. O novo método de similaridade é dividido em três etapas: Carregar modelo, Implementação do método de similaridade, Criação da nova aresta, Figura 6. O novo método será apresentado com mais detalhes nas próximas subseções.

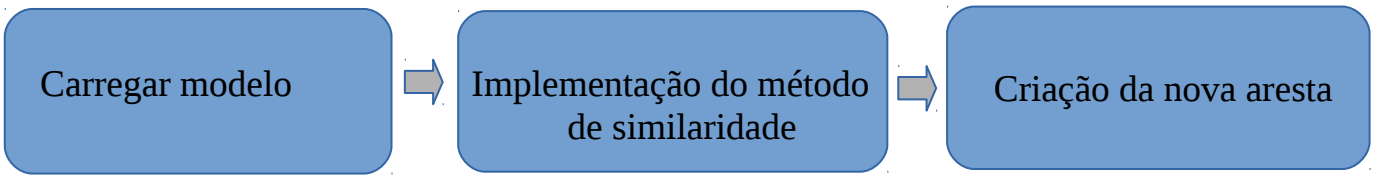


Figura 6: Etapas da proposta.

Fonte: O autor.

4.1.1 CARREGAR MODELO

O *Word2vec* permite que seu próprio modelo seja treinado a partir de um documento de texto. Para esta pesquisa, foi utilizado um modelo do Wikipédia pré treinado descrito na seção 4.1.3. Nessa etapa o modelo pré treinado é carregado em um objeto que foi instanciado do *Word2Vec*. O modelo pré treinado foi carregado utilizando os parâmetros padrões do Word2vec, descritos na seção 5.2 deste trabalho.

4.1.2 IMPLEMENTAÇÃO DO MÉTODO

Nesta etapa, o novo método que retorna a similaridade entre duas sentenças foi criado da seguinte forma:

1. Entrada com as sentenças.

2. Construção da matriz de similaridades do Cosseno para as palavras da primeira sentença em relação a segunda sentença.

3. Realizar uma iteração que captura o maior valor de similaridade para cada linha da matriz e armazenar esse valor. Em seguida reduzir a matriz retirando essa linha que foi percorrida. Realizar esse mesmo passo para cada iteração.

4. Quando não houver mais a possibilidade de redução da matriz, somar todos os maiores valores de similaridade obtidos, dividido pelo número de vezes que a matriz foi reduzida.

5. Retornar a similaridade para as duas sentenças.
   Um exemplo com duas sentenças é apresentado a seguir:

Sentenças	have	so	much	love	in	this	place
a	-0.05	0.21	0.046	0.01	0.01	-0.01	-0.02
place	0.28	-0.01	0.15	0.35	0.45	0.36	1.0
to	0.68	0.18	0.54	0.60	0.77	0.74	0.41
be	0.58	0.18	0.46	0.48	0.40	0.62	0.30
loved	0.37	-0.01	0.38	0.45	0.44	0.44	0.22

Figura 7: Matriz de similaridades do cosseno entre duas sentenças.
Fonte: O autor.

- Maiores valores = 1 + 0.77 + 0.62 + 0.45 + 0.21 = 3,05.
- Resultado da soma dos valores/reduções realizadas = 3,05/5 = 0,61.
- Similaridade entre as sentenças: **0,61.**

4.1.3 CRIAÇÃO DA NOVA ARESTA

Na criação da nova aresta, a nova medida de similaridade é calculada para as sentenças do DUC 2002. Se esse valor for maior ou igual que a similaridade mínima definida por parâmetro do método criado do *Word2vec*(**0,35**), então o grafo adiciona essa aresta(sentença) no modelo. O limiar (**0,35**) foi utilizado, pois no artigo do algoritmo de (FERREIRA,2014) foi realizado um estudo das similaridades mínimas

para cada aresta e este foi o melhor limiar avaliado. Abaixo podemos ver a figura que ilustra a criação da nova aresta:

```
procedimento criarNovaArestaWord2Vec( double similaridadeMinima, Lista<String> metodos){

Inicio

    double pesoDaAresta = 0;


    se(listaDeMetodos não contém(metodos) então
        retorna falso;
    fimse

    criar uma instância do método word2vec chamada similaridade.

    repetir
        para  i <- 0  até i < listaVertices1 faça
            para j <- i+1 até j < listaVestices2 faça
                se ((metodos) contém ("word2vec")) então
                    pesoDaAresta = similaridade(listaVertices1.get(i).pegaSentenca(),listaVestices2.get(j).pegaSentenca)
                fimse

                se (pesoDaAresta >= similaridadeMinima)então
                    adiciona a nova aresta no grafo
                    fimse

    retorna true
}
```

Figura 8: Criação da aresta.

Fonte: O autor.

4.2 SUMARIZAÇÃO

Após o agrupamento dos vértices que contém as sentenças que vão fazer parte do sumário final, foi realizada a sumarização usando o algoritmo *TextRank*. As sentenças são ordenadas de acordo com a sua pontuação do *TextRank.* O *TexRank* extrai as palavras-chave de um documento de texto e também determina o peso(relevância) das sentenças dentro de todo o documento(FERREIRA,2014).

5. EXPERIMENTO E RESULTADOS

Este capítulo apresenta a descrição e o detalhamento da base de dados e métricas de avaliação utilizadas bem como todos os resultados obtidos durante o desenvolvimento da ferramenta. Todos os testes realizados são discutidos e sumarizados em tabelas, facilitando a visualização do progresso nas etapas da implementação.

5.1 METODOLOGIA DE AVALIAÇÃO

Os resultados obtidos foram feitos da seguinte forma:

- Executar o algoritmo de agrupamento baseado no modelo de grafo proposto por (FERREIRA,2014).
- Inserir o novo método do *Word2vec* aos outros quatro métodos do modelo, e utilizá-los em diversas combinações de arestas diferentes.
- Avaliar a informatividade dos resumos gerados pelo sistema de 200 e 400 de tamanho de palavras e os sumários *gold* do *DUC* 2002 com a ferramenta ROUGE.
- Comparar todas as possíveis combinações de arestas no algoritmo de agrupamento.
- Comparar os melhores resultados obtidos com os sistemas propostos no *DUC* 2002.

5.2 PARÂMETROS UTILIZADOS NO WORD2VEC

Neste documento, os parâmetros utilizados no *Word2vec* foram os padrões. Segue todos parâmetros utilizados por *default* nesta pesquisa. O *Word2vec* converte os dados de entrada(sentenças) em strings, e configura a rede através de alguns parâmetros:

- *BatchSize*: É a quantidade de palavras que você processa ao mesmo tempo.
- *MinWordFrequency*: É o mínimo de vezes que uma palavra deve aparecer no corpus. Todas as palavras abaixo deste limite serão removidas antes do treinamento do modelo.
- *LayerSize*: Especifica o número de características no vetor de palavras. Isso é igual ao número de dimensões no espaço vetorial.
- *Seed*: Este método define a semente para gerar números aleatórios
- *WindowSize*: Define o tamanho da janela de contexto.
- *Iterate*: Esse método é usado para alimentar o *SentenceIterator*, que contém o corpus de treinamento, em vetores de parágrafos.
- *TokenizerFactory*: Define qual o *TokenizerFactory* vai ser utilizado para a “tokenização” de strings durante o treinamento.
- *LearningRate*: Define o valor inicial da taxa de aprendizado para o treinamento do modelo.

Tabela 1: Parâmetros Word2vec.

batchSize	100
elementsLearningAlgorithm	null
epochs	1
hugeModelExpected	false
iterations	10
layersSize	300
learningRate	0.025
learningRateDecayWords	0
minLearningRate	1.0E-4
minWordFrequency	5
negative	0.0
sampling	0.0
scavengerActivationThreshold	2000000
scavengerRetentionDelay	3
seed	0
sequenceLearningAlgorithm	null

stop	STOP
stopList	[]
unk	UNK
useAdaGrad	false
variableWindows	null
vocabSize	217971
window	5

Fonte: O autor.

5.3 DATASET PARA TREINAMENTO DO WORD2VEC

O arquivo de texto que foi utilizado como *dataset* para o treinamento do modelo no *Word2vec* contém aproximadamente 2.750 GB de tamanho. O modelo contém milhares de textos de diversos assuntos do Wikipédia e está no idioma inglês. O link que pode ser encontrado vários corpora de modelos para treinamento e pré treinados é: https://code.google.com/archive/p/word2vec/.

5.4 BASE DE DADOS

A base de dados *DUC* 2002, que serviu para gerar os resumos do novo método proposto, além dos métodos já existentes no modelo de grafo proposto por (FERREIRA,2014), foi de uma conferência de compreensão de documentos[1].

A base de dados contém 59 diretórios com 567 documentos, onde cada diretório é referente a um conjunto de textos de tópicos relacionados. Para o primeiro diretório *d061j* por exemplo, há textos sobre a chegada de um furacão e suas consequências em várias regiões. Esses textos de cada diretório foram usados como entrada do algoritmo de agrupamento proposto por (FERREIRA,2014). A partir dessa entrada, várias configurações de arestas foram utilizadas para gerar um sumário para cada diretório.

Depois de gerado um resumo para cada diretório, dos 59 existentes, cada um deles foi avaliado com os dois sumários referência *gold* do *DUC* 2002. A cada dois

[1]www-nlpir.nist.gov/projects/duc/guidelines/2002.html

diretórios em sequência no diretório raiz, os sumários *gold* tratavam de tópicos relacionados. O primeiro sumário *gold* foi composto da primeira pasta de cada referência(d061jb,d062ja...) e o segundo sumário *gold* da segunda pasta de cada referência(d061ji,d062jg…). Quando existia apenas uma pasta, a mesma era utilizada nos dois sumários.

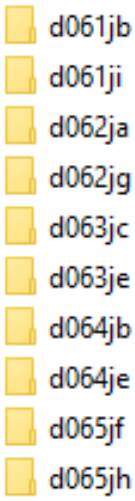


Figura 9: Sumários *gold.*

Fonte: O autor.

Em cada diretório dos sumários *gold* existem dois arquivos: 200e e 400e. O arquivo 200e é referente ao tamanho de 200 palavras, e o de 400e para 400 palavras. A avaliação foi realizada nestes dois tamanhos de sumários.

5.5 FERRAMENTA DE AVALIAÇÃO ROUGE

A ferramenta para análise de avaliações automatizadas de resumos será o *ROUGE*. Este avaliador mede a semelhança de conteúdo entre resumos desenvolvidos pelo sistema e os resumos "*gold*" referências correspondentes. A medida de avaliação utilizada foi a *ROUGE-N*, que é baseada em coocorrências estatísticas. Essa medida de avaliação estatística, amplamente utilizada em processamento de linguagem natural, se baseia na similaridade de *n-grams.* Para a

análise, as palavras são modeladas de modo que cada *n-gram* é composta de n palavras(LIN,2004).

Para avaliação de sistemas de sumarização, que normalmente selecionam as sentenças mais representativas na entrada para formar um resumo extrativo, as métricas de recuperação de informação como cobertura *e* precisão podem ser utilizadas. A cobertura avalia o número de sentenças selecionadas por humanos que também são identificadas pelo sistema, enquanto a precisão é a fração das sentenças identificadas pelo sistema que está correto(Nenkova,2006). Já o *F-Measure* pode ser calculado através da média ponderada entre a precisão e a cobertura.

Os resultados da próxima seção, mostram os valores gerados pela métrica *ROUGE-N* pois foram mais adotados nos trabalhos relacionados a esta pesquisa. Também considerando que para essa avaliação o N = 1, ou seja, palavras individuais em uma sentença. O resumo candidato gerado pelo sistema e o resumo referência da competição *DUC* de 2002 foram avaliados.

5.6 RESULTADOS OBTIDOS

A avaliação para os dois sumários ao mesmo tempo com a ferramenta *ROUGE* produziu uma tabela que vai ser apresentada nessa seção. A tabela 1 exibe os resultados de cada método separadamente para 200 de tamanho, e a tabela 2 mostra os resultados de cada método separadamente para 400 de tamanho.

Tabela 2: Métodos separados para o tamanho de 200.

Métodos	Average_R(200)	Average_P(200)	Average_F(200)
Coreferencia	0,62	0,26	0,35
Discourse	0,47	0,34	0,38
Semantic	0,54	0,28	0,37
Statical	**0,78**	0,13	0,23
Nova Aresta	0,58	**0,35**	**0,43**

Fonte: O autor.

Tabela 3: Métodos separados para o tamanho de 400.

Métodos	Average_R(400)	Average_P(400)	Average_F(400)
Coreferencia	0,65	0,37	0,46
Discourse	0,53	**0,45**	0,45
Semantic	0,57	0,41	0,47
Statical	**0,77**	0,25	0,37
Nova Aresta	0,62	0,44	**0,51**

Fonte: O autor.

Para cada configuração de aresta separada, podemos observar que o método novo inserido proposto nesta pesquisa se sobressaiu no geral entre os demais. A nova aresta apresentou resultados superiores nas métricas *Average_F* e *Average_P* para o tamanho de 200, e a métrica *Average_F* para o tamanho de 400. Apesar de a nova aresta ter o melhor *Average_F*, outros métodos tiveram melhores resultados para cobertura e precisão. A nova aresta ficou bem próximo do melhor resultado no tamanho de 400 em precisão que foi o do método *Discourse* com 0,45. O método de Correferência e o *Statical* apresentaram os melhores resultados em cobertura, mas os piores em precisão em relação aos demais para ambos os tamanhos.

Podemos observar a posição de cada método separado para ambos os tamanhos nas tabelas 4 e 5:

Tabela 4: Posição dos métodos separados para o tamanho de 200.

Posição	Average_R(200)	Average_P(200)	Average_F(200)
1º	Statical	Nova Aresta	Nova Aresta
2º	Coreferencia	Discourse	Discourse
3º	Nova Aresta	Semantic	Semantic
4º	Semantic	Coreferencia	Coreferencia
5º	Discourse	Statical	Statical

Fonte: O autor.

Tabela 5: Posição dos métodos separados para o tamanho de 400.

Posição	Average_R(400)	Average_P(400)	Average_F(400)
1º	Statical	Discourse	Nova Aresta
2º	Coreferencia	Nova Aresta	Semantic
3º	Nova Aresta	Semantic	Coreferencia
4º	Semantic	Coreferencia	Discourse
5º	Discourse	Statical	Statical

Fonte: O autor.

Outro objetivo proposto pela pesquisa é a avaliação de diferentes configurações de arestas no modelo de grafo. A tabela 2 e 3 mostram todas as combinações possíveis de arestas respectivamente para um tamanho de 200 e 400.

Tabela 6: Todas as combinações para o tamanho de 200.

Métodos	Average_R(200)	Average_P(200)	Average_F(200)
Coreferencia+Discourse	0,62	0,26	0,35
Semantic+Coreferencia	0,64	0,25	0,34
Semantic+Discourse	0,56	0,29	0,38
Statical+Coreferencia	0,80	0,14	0,23
Statical+Discourse	0,80	0,14	0,23
Statical+Semantic	0,80	0,14	0,23
Nova Aresta+Coreferencia	0,63	0,26	0,35
Nova Aresta+Discourse	0,48	**0,40**	**0,39**
Nova Aresta+Semantic	0,55	0,29	0,37
Nova Aresta+Statical	**0,80**	0,14	0,23
Semantic+Coreferencia+Discourse	0,65	0,25	0,34
Statical+Coreferencia+Discourse	0,80	0,14	0,23
Statical+Semantic+Coreferencia	0,79	0,14	0,24
Statical+Semantic+Discourse	0,79	0,14	0,23
Nova Aresta+Semantic+Coreferencia	0,64	0,25	0,34
Nova Aresta+Semantic+Discourse	0,55	0,29	0,38
Nova Aresta+Statical+Coreferencia	0,80	0,14	0,23
Nova Aresta+Statical+Discourse	0,80	0,14	0,23
Nova Aresta+Statical+Semantic	0,80	0,14	0,23
Statical+Semantic+Coreferencia+Discourse	0,80	0,14	0,23
Nova Aresta+Semantic+Coreferencia+Discourse	0,65	0,25	0,34
Nova Aresta+Statical+Coreferencia+Discourse	0,80	0,14	0,23
Nova Aresta+Statical+Semantic+Coreferencia	0,79	0,14	0,23
Nova Aresta+Statical+Semantic+Discourse	0,79	0,14	0,23
Todos	0,80	0,14	0,23

Fonte: O autor.

Tabela 7: Todas as combinações para o tamanho de 400.

Métodos	Average_R(400)	Average_P(400)	Average_F(400)
Coreferencia+Discourse	0,64	0,37	0,46
Semantic+Coreferencia	0,66	0,36	0,45
Semantic+Discourse	0,56	0,41	0,47
Statical+Coreferencia	0,78	0,25	0,37
Statical+Discourse	0,78	0,25	0,38
Statical+Semantic	0,78	0,25	0,37
Nova Aresta+Coreferencia	0,65	0,37	0,46
Nova Aresta+Discourse	0,54	**0,45**	0,46
Nova Aresta+Semantic	0,57	0,41	**0,47**
Nova Aresta+Statical	**0,79**	0,25	0,37
Semantic+Coreferencia+Discourse	0,66	0,36	0,46
Statical+Coreferencia+Discourse	0,77	0,25	0,37
Statical+Semantic+Coreferencia	0,78	0,25	0,38
Statical+Semantic+Discourse	0,77	0,25	0,37
Nova Aresta+Semantic+Coreferencia	0,66	0,36	0,45
Nova Aresta+Semantic+Discourse	0,56	0,41	0,47
Nova Aresta+Statical+Coreferencia	0,78	0,25	0,37
Nova Aresta+Statical+Discourse	0,78	0,25	0,38
Nova Aresta+Statical+Semantic	0,78	0,25	0,37
Statical+Semantic+Coreferencia+Discourse	0,78	0,25	0,37
Nova Aresta+Semantic+Coreferencia+Discourse	0,66	0,36	0,45
Nova Aresta+Statical+Coreferencia+Discourse	0,77	0,25	0,37
Nova Aresta+Statical+Semantic+Coreferencia	0,78	0,25	0,38
Nova Aresta+Statical+Semantic+Discourse	0,77	0,25	0,37
Todos	0,78	0,25	0,37

Fonte: O autor.

Podemos observar que nenhuma combinação alcançou o novo método inserido do nas métricas do *Average_F* para ambos os tamanhos anteriormente exibido nos métodos separados. Os métodos NovaAresta+*Statical* e NovaAresta+*Discourse* obtiveram os melhores resultados de cobertura e precisão respectivamente para ambos os tamanhos, sendo que o método NovaAresta+Discourse obteve os melhores resultados em precisão de toda avaliação. Os métodos NovaAresta+*Discourse* e NovaAresta+Semantic também obtiveram os melhores resultados para a métrica *Average*_F respectivamente para ambos os tamanhos na combinação de métodos. Os piores resultados para combinações de métodos para métricas de precisão em ambos os tamanhos não possuíam a nova aresta em suas combinações. Em geral as combinações com a

nova aresta tiveram os melhores resultados, mas o fato de aumentar o número de arestas não melhorou o resultado em relação ao melhor resultado separado da nova aresta. A combinação de arestas que não tiveram bons resultados separadamente, influenciou este fato.

Tabela 8: Posição dos métodos combinados para o tamanho de 200.

Posição	Average_R(200)	Average_P(200)	Average_F(200)
1º	Nova Aresta+Statical	Nova Aresta+Discourse	Nova Aresta+Discourse
2º	Nova Aresta+Statical+Discourse	Semantic+Discourse	Semantic+Discourse
3º	Statical+Coreferencia	Nova Aresta+Semantic	Nova Aresta+Semantic+Discourse
4º	Todos	Nova Aresta+Semantic+Discourse	Nova Aresta+Semantic
5º	Nova Aresta+Statical+Coreferencia+Discourse	Coreferencia+Discourse	Nova Aresta+Coreferencia

Fonte: O autor.

Tabela 9: Posição dos métodos combinados para o tamanho de 400.

Posição	Average_R(400)	Average_P(400)	Average_F(400)
1º	Nova Aresta+Statical	Nova Aresta+Discourse	Nova Aresta+Semantic
2º	Nova Aresta+Statical+Discourse	Nova Aresta+Semantic	Nova Aresta+Semantic+Discourse
3º	Todos	Nova Aresta+Semantic+Discourse	Nova Aresta+Coreferencia
4º	Nova Aresta+Statical+Coreferencia	Coreferencia+Discourse	Coreferencia+Discourse
5º	Nova Aresta+Statical+Semantic	Nova Aresta+Coreferencia	Nova Aresta+Discourse

Fonte: O autor.

Em seguida podemos ver os melhores resultados da métrica *Average_F* desta pesquisa em comparação com outros sistemas do DUC 2002 para ambos os tamanhos:

Tabela 10: Comparação contra os sistemas do DUC 2002 – resumo 200 palavras

Sistema	Average_F
Sistema 19	0,199
Sistema 24	0,193
Sistema 28	0,167
Sistema 20	0,144
Sistema 29	0,102
Sistema (Ferreira,2014)	0,3
Novo Sistema	**0,439**

Fonte: O autor.

Tabela 11: Comparação contra os sistemas do DUC 2002 – resumo 400 palavras

Sistema	Average_F
Sistema 19	0,24
Sistema 24	0,249
Sistema 28	0,241
Sistema 20	0,191
Sistema 29	0,179
Sistema (Ferreira,2014)	0,254
Novo Sistema	**0,518**

Fonte: O autor.

O resultado das tabelas anteriores, mostra que a nova medida de similaridade do trabalho conseguiu resultados de 46% e 101% melhor do que a abordagem anterior para resumos com 200 e 400 palavras respectivamente na métrica *F-measure.*

Podemos concluir que para a métrica Average_F devemos ressaltar que obteve mais que o dobro melhor em relação melhor resultado anterior. Por outro lado, os piores resultados para essas duas métricas era de combinações de arestas onde a nova aresta não fazia parte. Já na cobertura, a nova aresta ficou com resultados intermediários entre os demais.

6.**CONCLUSÕES E TRABALHOS FUTUROS**

Esta pesquisa apresentou uma nova medida de similaridade entre sentenças a partir de uma abordagem com *word embeddings* para reduzir redundância em sumarização multi-documento. O novo método consistiu do carregamento de um *dataset* contendo milhões de palavras no *Word2vec* e da utilização da medida de similaridade do cosseno. Para se ter uma percepção da utilização de uma abordagem de *word embeddings* com a ferramenta *Word2vec*, foi calculado duas sentenças que mudavam apenas uma palavra em sua estrutura, que obteve o resultado em torno de 78% de similaridade entre as sentenças.

Além disso foi utilizado a ferramenta de avaliação ROUGE para avaliar o novo método proposto e as diferentes configurações do modelo de grafo com as outras medidas proposto por (FERREIRA,2014). De acordo com a avaliação realizada, a nova aresta implementada foi extremamente positiva, pois consegue resultados muito melhores na métrica *F-measure* contra outros sistemas da competição DUC 2002. A nova aresta obteve 43% para um sumário de 200 palavras e 51% para um sumário de 400 palavras a mais do que o melhor sistema anterior(FERREIRA.2014). Outro ponto observado, é de que os piores resultados obtidos para as métricas de precisão e *F-Measure* não possuía o novo método de similaridade proposto.

Pensando em trabalhos futuros:

1. A utilização de um dataset na ordem de bilhões de palavras pode potencializar o aumento do valor da similaridade obtida entre duas sentenças. O treinamento de um *dataset* ou o carregamento de um *dataset* pré treinado dessa magnitude pode refinar o vetor de características das palavras.
2. Utilização de outras técnicas de *word embeddings* para o cálculo de similaridade, como o Glove(PENNINGTON,2014) por exemplo.
3. Propor outras medidas de similaridade que poderiam ser combinadas para o cálculo de similaridade entre essas sentenças, podendo obter resultados diferentes para análise.

4. Utilizar outros *datasets* ou até variações *n-gram* do ROUGE para avaliar os resultados. A presença de mais sumários referências poderiam melhorar também os resultados.

Por fim, esta pesquisa apresentou um novo método de similaridade entre sentenças que melhora a sumarização de texto-multidocumento e também pode ser usada para um único documento.

REFERÊNCIAS

ABUOBIEDA, Albaraa et al. Text summarization features selection method using pseudo genetic-based model. In:Information Retrieval & Knowledge Management (CAMP), 2012 International Conference on. IEEE, 2012. p. 193-197.

ADAMOPOULOS, Panagiotis. What makes a great MOOC? An interdisciplinary analysis of student retention in online courses. 2013.

Barzilay, Regina, and Michael Elhadad. "Using lexical chains for text summarization." *Advances in automatic text summarization* (1999): 111-121.

CLARK, Jonathan H.; GONZÁLEZ-BRENES, José P. Coreference resolution: Current trends and future directions. Language and Statistics II Literature Review, p. 1-14, 2008.

DILLENBOURG, Pierre; SCHNEIDER, Daniel; SYNTETA, Paraskevi. Virtual learning environments. In: 3rd Hellenic Conference" Information & Communication Technologies in Education". Kastaniotis Editions, Greece, 2002. p. 3-18.

FERREIRA, Rafael et al. Assessing sentence scoring techniques for extractive text summarization. Expert Systems with Applications. pp. 5755-5764, 2013.

FERREIRA, Rafael et al. A multi-document summarization system based on statistics and linguistic treatment. Expert Systems with Applications, v. 41, n. 13, p. 5780-5787, 2014.

FERREIRA, Rafael et al. A four dimension graph model for automatic text summarization. In: Proceedings of the 2013 IEEE/WIC/ACM International Joint Conferences on Web Intelligence (WI) and Intelligent Agent Technologies (IAT)-Volume 01. IEEE Computer Society, 2013. p. 389 396.

GUPTA, Vishal; LEHAL, Gurpreet Singh. A survey of text summarization extractive techniques. Journal of emerging technologies in web intelligence, v. 2, n. 3, p. 258-268, 2010.

Jackie CK Cheung, “Comparing Abstractive and Extractive Summarization of Evaluative Text: Controversiality and Content Selection”, B. Sc. (Hons.) Thesis in the Department of Computer Science of the Faculty of Science, University of British Columbia, 2008.

Jimmy Lin., “Summarization.”, Encyclopedia of Database Systems. Heidelberg, Germany: Springer-Verlag, 2009.

JOAO, Cordeiro; GAËL, Dias; PAVEL, Brazdil. New functions for unsupervised asymmetrical paraphrase detection.Journal of Software, v. 2, n. 4, p. 12-23, 2007.

Kunder, M. (2016). The size of the world wide web. Último acesso Julho, (2017).<www.worldwidewebsize.com/?>.

LEVY, Omer; GOLDBERG, Yoav. Dependency-Based Word Embeddings. In:ACL (2). 2014. p. 302-308.

LIN, Chin-Yew. Rouge: A package for automatic evaluation of summaries. In: Text summarization branches out: Proceedings of the ACL-04 workshop. 2004.

LLORET, Elena; PALOMAR, Manuel. Text summarisation in progress: a literature review. Artificial Intelligence Review, v. 37, n. 1, p. 1-41, 2012.

LUO, Xiaoqiang. Coreference or Not: A Twin Model for Coreference Resolution. In:HLT-NAACL. 2007. p. 73-80.

LUSTIGOVA, Zdena; NOVOTNA, Veronika. Advantages and Limits of Text Mining Software for Analysis of Students' Satisfaction in Online Education.

MARINO, José B. et al. N-gram-based machine translation.Computational Linguistics, v. 32, n. 4, p. 527-549, 2006.

MIHALCEA, Rada; TARAU, Paul. TextRank: Bringing Order into Text. In:EMNLP. 2004. p. 404-411.

MIHALCEA, Rada; CEYLAN, Hakan. Explorations in Automatic Book Summarization. In: EMNLP-CoNLL. 2007. p. 380-389.

MIKOLOV, Tomas et al. Distributed representations of words and phrases and their compositionality. In:Advances in neural information processing systems. 2013. p. 3111-3119.

NENKOVA, Ani. Summarization evaluation for text and speech: issues and approaches. In:Ninth International Conference on Spoken Language Processing. 2006.

NENKOVA, Ani; MCKEOWN, Kathleen. A survey of text summarization techniques. In: Mining text data. Springer US, 2012. p. 43-76.

PadmaPriya, G. and K. Duraiswamy, “An Approach for Text Summarization using Deep Learning Algorithm”. Journal of Computer Science 10(1): 1-9, 2014.

PEREIRA, Silvio. Processamento de Linguagem Natural.

PENNINGTON, Jeffrey; SOCHER, Richard; MANNING, Christopher D. Glove: Global vectors for word representation. In:EMNLP. 2014. p. 1532-1543.

REATEGUI, Eliseo; EPSTEIN, Daniel. Using text mining to support text summarization. In: Brazilian Symposium on Computers in Education (Simpósio Brasileiro de Informática na Educação-SBIE**)**. 2015. p. 1217.

REATEGUI, Eliseo; KLEMANN, Miriam; FINCO, Mateus David. Using a text mining tool to support text summarization. In: Advanced Learning Technologies (ICALT), 2012 IEEE 12th International Conference on. IEEE, 2012. p. 607-609.

Silva, Thales do N. Uma arquitetura para descoberta de conhecimento a partir de bases textuais. 2012. 78f. Trabalho de Conclusão de Curso em Universidade Federal de Santa Catarina, Araranguá, 2012.

VAN LABEKE, Nicolas et al. OpenEssayist: extractive summarisation and formative assessment of free-text essays. 2013.

WINOGRAD, Peter N. Strategic difficulties in summarizing texts. Reading Research Quarterly, p. 404-425, 1984.

WOLF, Florian; GIBSON, Edward. Representing discourse coherence: A corpus-based study.Computational Linguistics, v. 31, n. 2, p. 249-287, 2005.

WUBBEN, Sander; VAN DEN BOSCH, Antal. A semantic relatedness metric based on free link structure. In: Proceedings of the Eighth International Conference on Computational Semantics. Association for Computational Linguistics, 2009. p. 355-358.

YANG, Guangbing et al. The effectiveness of automatic text summarization in mobile learning contexts. Computers & Education, v. 68, p. 233-243, 2013.

ZIPITRIA, Iraide; ARRUARTE, Ana; ELORRIAGA, Jon Ander. LEA: A Summarization Web Environment Based on Human Instructors' Behaviour. In: Advanced Learning Technologies, 2008. ICALT'08. Eighth IEEE International Conference on. IEEE, 2008. p. 564-568.