VICTOR HUGO ANDRADE SOARES

COMBINAÇÕES DE SIMILARIDADE SEMÂNTICA E FREQUÊNCIA DE TERMOS PARA AGRUPAMENTO DE TEXTOS

Dissertação apresentada à Universidade Federal de Viçosa, como parte das exigências do Programa de Pós-Graduação em Ciência da Computação, para obtenção do título de *Magister Scientiae*.

VIÇOSA
MINAS GERAIS – BRASIL
2017

Ficha catalográfica preparada pela Biblioteca Central da Universidade Federal de Viçosa - Câmpus Viçosa

T
S676c
2017

Soares, Victor Hugo Andrade, 1991-
Combinações de similaridade semântica e frequência de termos para agrupamento de textos / Victor Hugo Andrade Soares. – Viçosa, MG, 2017.
xxv, 103f. : il. ; 29 cm.

Inclui apêndices.
Orientador: Murilo Coelho Naldi.
Dissertação (mestrado) - Universidade Federal de Viçosa.
Referências bibliográficas: f.77-85.

1. Algorítmos. 2. Semântica. 3. Web semântica. 4. Documentos eletrônicos. I. Universidade Federal de Viçosa. Departamento de Informática. Programa de Pós-graduação em Ciência da Computação. II. Título.

CDD 22 ed. 005.1

VICTOR HUGO ANDRADE SOARES

COMBINAÇÕES DE SIMILARIDADE SEMÂNTICA E FREQUÊNCIA DE TERMOS PARA AGRUPAMENTO DE TEXTOS

Dissertação apresentada à Universidade Federal de Viçosa, como parte das exigências do Programa de Pós-Graduação em Ciência da Computação, para obtenção do título de *Magister Scientiae*.

APROVADA: 20 de março de 2017.

Mauro Nacif Rocha

André Gustavo dos Santos

Murilo Coelho Naldi
Orientador

À minha família, amigos e professores, por lapidarem diariamente o meu ser.

"Stay hungry, stay foolish."
(Stewart Brand)

Agradecimentos

Agradeço ao meu orientador, Dr. Murilo Naldi, pelos inúmeros conselhos e ensinamentos transmitidos durante os últimos anos, que foram fundamentais para a construção deste trabalho e de minha formação.

Agradeço também ao meu coorientador, Dr. Ricardo Campello, pelo exemplo de dedicação e seriedade com a pesquisa científica.

Agradeço à Coordenação de Aperfeiçoamento de Pessoal de Nível Superior (CAPES) pelo essencial suporte financeiro, permitindo dedicar me integralmente a esse trabalho.

Agradeço à minha esposa, Dominique Ferreira, pelo imenso amor, paciência e motivação demonstrados durante a realização desse trabalho, mesmo nos momentos mais difíceis e em que o tempo para retribuir esse amor era escasso.

Agradeço aos meus pais, José Luiz e Marli, pelos bons exemplos de vida e incentivos ao aprendizado, e meu irmão Vinícius por me acompanhar e me fazer crescer durante toda minha vida.

Agradeço aos meus colegas de laboratório, Alba Assis, Aly Camilo, Charles Abreu, Daniel Louzada, Fabio Reinoso, Jonatas Chagas, Liliane Soares e Marco Pinto pelos incontáveis momentos de descontração no mercado, risadas jogando Imagem e Ação, e confraternizações regadas a feijão tropeiro.

Agradeço à Universidade Federal de Viçosa e aos professores pelo excelente suporte oferecido em toda minha formação acadêmica.

E finalmente, agradeço a Deus, que me capacita e sustenta a cada novo dia. A Ele toda honra, glória e louvor.

Sumário

Lista de Figuras

Lista de Tabelas

Lista de Algoritmos

Lista de Abreviaturas

AEs :	Algoritmos Evolutivos
ANOVA :	*Analysis of variance*
ARI :	*Ajusted Rand Index*
BOC :	*Bag of Concepts*
BOW :	*Bag of Words*
Cos+FGTM :	Combinação das medidas cosseno e FGTM
DF :	*Document Frequency*
EAC :	*Evolutionary Algorithm for Clustering*
EC :	Extração de Conhecimento
EI :	Extração de Informação
ELSDC :	*Ensemble Lexical-Semantic Document Clustering*
Euc+FGTM :	Combinação das medidas Euclidiana e FGTM
Euc+Wik :	Combinação das medidas Euclidiana e BOC da Wikipédia
F-EAC :	*Fast Evolutionary Algorithm for Clustering*
FGTM :	*Frequency Google Tri-grams Measure*
G-*means* :	*The Gaussian-means*
GTM :	*Google Tri-grams Measure*
KDD :	*Knowledge Discovery in Databases*
KDT :	*Knowledge Discovery in Text*
LDC :	*Lexical Document Clustering*
M3 :	*Multiple to Multiple Mapping Measure*
MG-*means* :	*Monitored Gaussian-means*
NMI :	*Normalized Mutual Information*

OMRK :	*Ordered Multiple runs of K-means*
SemEval :	*Semantic Evaluation*
SIGLEX :	*Special Interest Group on the Lexicon of the Association for Computational Linguistics*
SOC-PMI :	*Second Order Co-occurrence PMI*
SS :	Silhueta Simplificada
STS :	*Semantic Text Similarity*
TF :	*Term Frequency*
TF-IDF :	*Term Frequency Inverse Document Frequency*
Wik+FGTM :	Combinação da medida BOC da Wikipédia com a FGTM

Notação

$\lvert \cdot \rvert$:	Cardinalidade
$\lvert d_a \rvert \times \lvert d_b \rvert$:	Produto das normas de d_a e d_b
$\lvert G_i \cap G'_j \rvert$:	Número de instâncias em comum entre os grupos G_i e G'_j
$\lvert P \rvert$:	Tamanho da população P
A_i:	Conjunto de elementos selecionados da i-ésima linha da matriz M
$a(x_i)$:	Dissimilaridade entre o objeto x_i e seu grupo
α:	Nível de confiança
α_{ij}:	Célula da i-ésima linha e j-ésima coluna da matriz de similaridade entre termos
$avgVar$:	Média das variâncias
$b(x_i)$:	Menor dissimilaridade entre o objeto x_i e outros grupos
β:	Fuzzificador
c:	Quantidade de grupos no algoritmo c-means
C:	Conjunto de centróides
c_{ij}:	j-ésimo centróide filho do grupo G_i
C_f:	Frequência do uni-grama mais frequente do *corpus*
c_i:	Centróide do grupo G_i
C':	Número total de conceitos
C'':	Número de conceitos selecionados
$c(wd_i)$:	Frequência uni-grama da palavra wd_i no *corpus*
$c(wd_awd_iwd_b)$:	Frequência tri-gramas da sequência de palavras $wd_awd_iwd_b$
$classe(x_i)$:	Retorna a classe do objeto x_i
$cos(d_a, d_b)$:	Função de similaridade cosseno entre os documentos d_a e d_b

$d_a \cdot d_b$:	Produto escalar entre d_a e d_b
$D_E(d_a, d_b)$:	Função de distância Euclidiana entre os documentos d_a e d_b
d_i:	Vetor com as frequências dos termos no i-ésimo documento
DC_p:	Documento centróide gerado a partir do grupo p de documentos semente
δ:	Número de termos que ocorrem em dois documentos
$\delta Total$:	Somatório das menores frequências entre termos coincidentes em dois documentos
$dist(d_i, d_j)$:	Função de distância ou dissimilaridade entre o documento d_i e d_j
DS_p:	Conjunto de documentos semente obtidos a partir do grupo p de termos
$f_a(p)$:	Número de artigos Wikipédia em que a frase p é utilizada como âncora
$f_t(p)$:	Número de artigos Wikipédia em que a frase p aparece de qualquer forma
$f(\cdot)$:	Função de aptidão
$F(s_i)$:	Valor da função de distribuição acumulada do elemento s_i
$freq(t_i)$:	Frequência do termo t_i
G:	Quantidade de grupos Gaussianos
G_i:	i-ésimo grupo de uma partição
g_{max}:	Número máximo de gerações
H:	Matriz Identidade
I:	Número de iterações c-means
IDF_i:	*Inverse Document Frequency* do i-ésimo termo
K:	Número de grupos
K_{max}:	Valor máximo permitido para o número de grupos
K_{min}:	Valor mínimo permitido para o número de grupos
M:	Matriz de similaridade entre termos
M_i:	i-ésima linha da Matriz M
m_i:	Medóide do grupo G_i

$max_p w_{pj}$:	Maior frequência de um termo no documento j
MD_i:	Conjunto de medóides mais centrais do grupo DS_i
md_i:	i-ésimo medóide de um conjunto de medóides
Mean-TFIDF(t_j):	Pontuação Mean-TFIDF do termo t_j
$min(x, y)$:	Função que retorna o menor valor entre x e y
MO_1:	Operador de Mutação 1
MO_2:	Operador de Mutação 2
μ:	Média
$\mu(N1, N2)$:	Função que retorna a média entre $N1$ e $N2$
n:	Número de dimensões em um espaço ou gramas em uma sentença
N:	Número de documentos em um conjunto
N_i:	Número de documentos no i-ésimo grupo
$n1$:	Quantidade de tri-gramas que começam com wd_a e terminam com wd_b
$N1$:	Somatório dos tri-gramas que começam com wd_a e terminam com wd_b
$n2$:	Quantidade de tri-gramas que começam com wd_b e terminam com wd_a
$N2$:	Somatório dos tri-gramas que começam com wd_b e terminam com wd_a
p:	Frase candidata
p_{ij}:	Grau de associação do j-ésimo objeto ao i-ésimo grupo
π:	Partição
π_g:	Partição conhecida (ou "*gold standard*")
π_r:	Partição resultante
$pTotal$:	Somatório das frequências dos termos no documento P
$rel(t_i, t'_j)$:	Relacionamento semântico entre o i-ésimo termo de um documento e o j-ésimo termo de outro documento
$rTotal$:	Somatório das frequências dos termos no documento R
s_i:	i-ésimo elemento de uma lista de valores
$S(P, R)$:	Similaridade semântica entre os documentos P e R
$s(x_i)$:	Silhueta do objeto x_i

σ:	Desvio Padrão
$Sim(wd_a, wd_b)$:	Função de similaridade entre as palavras wd_a e wd_b
t:	Quantidade máxima de iterações do K-médias
$\vec{t_a}$:	Vetor a de termos
t_i:	i-ésimo termo de um documento
T:	Número total de termos em um conjunto
T':	Número de termos selecionados
T'':	Número de termos chave
$\bar{T}$:	Número médio de termos nos documentos de um conjunto
TC_p:	Termo-centróide do grupo p de termos
TF_{ij}:	*Term Frequency* do termo i no documento j
TFIDF_{ij}:	Pontuação TFIDF do termo t_j no documento d_i
U'_i:	Representação uni-dimensional do conjunto G_i
$U_{p(j),j}$:	Elemento da j-ésima coluna da matriz de agrupamento *fuzzy*
v_i:	i-ésimo protótipo do algoritmo C-means
VAR-TFIDF(t_j):	Pontuação VAR-TFIDF do termo t_j
$w_{i,a}$:	Peso do i-ésimo termo no documento d_a
w_i:	Peso do i-ésimo termo de um documento
wd_a:	Palavra a
X:	Conjunto de dados
x_i:	i-ésimo objeto de um grupo
y_i:	Número de elementos selecionados na i-ésima linha da matriz M

Resumo

SOARES, Victor Hugo Andrade, M.Sc., Universidade Federal de Viçosa, março de 2017. **Combinações de Similaridade Semântica e Frequência de Termos para Agrupamento de Textos.** Orientador: Murilo Coelho Naldi. Coorientador: Ricardo José Gabrielli Barreto Campello.

Um dos desafios ao se agrupar documentos é encontrar uma boa medida de similaridade para documentos de textos, que seja capaz de gerar grupos coesos. Algumas medidas são baseadas no clássico modelo *bag of words* e consideram apenas o vocabulário do documento. Com isso, documentos semanticamente similares podem ser atribuídos a diferentes grupos se eles não compartilham o mesmo vocabulário. Por essa razão, medidas de similaridade semântica que usam conhecimento externo, como um *corpus*, dicionários ou banco de palavras, têm sido propostas na literatura. Neste trabalho, a medida *Frequency Google Tri-grams Measures* (FGTM) é proposta para identificar similaridade entre documentos baseado nas frequências dos termos nos documentos e no *corpus* Google *n-grams*. A comparação entre as frequências de um termo em um dado par de documentos pode quantificar a importância daquele termo para o assunto dos documentos, assumindo que um termo possui maior relevância para um documento se ele ocorre mais vezes. Adicionalmente, as frequências dos termos dos documentos no *corpus* Google *n-grams* permitem estimar semanticamente suas similaridades. Adicionalmente, oito variantes de dois algoritmos de agrupamento são aplicadas a vários conjuntos de dados reais, com o objetivo de avaliar experimentalmente a qualidade dos grupos obtidos com a medida proposta e compará-la com outras medidas do estado da arte. Análises de complexidade computacional das medidas comparadas são apresentadas. Os resultados experimentais demonstram que a medida proposta melhora significativamente a qualidade dos agrupamentos de documentos, comprovado por testes estatísticos. Também é mostrado que, combinar resultados de agrupamento obtidos com *bag of words* e me-

dida semântica obtém melhores resultados que adotar uma medida individualmente. Para finalizar, é feito um estudo sobre heurísticas para estimar o número K de grupos em agrupamento de textos. Uma versão modificada da heurística *G-means* é proposta e comparada com heurísticas da literatura.

Abstract

SOARES, Victor Hugo Andrade Soares, M.Sc., Universidade Federal de Viçosa, March, 2017. **Combinations of Semantic and Term Frequency Similarities for Text Clustering.** Advisor: Murilo Coelho Naldi. Co-advisor: Ricardo José Gabrielli Barreto Campello.

One challenge for document clustering consists of finding a proper similarity measure for text documents, which enables the generation of cohesive groups. Some measures are based on the classic bag of words model and take into account the vocabulary of the documents solely. In doing so, semantically similar documents may reside in different clusters if they do not share the same vocabulary. For this reason, semantic similarity measures that use external knowledge, such as corpus, dictionaries, or word bases, have been proposed in the literature. In this paper, the Frequency Google Tri-grams Measure (FGTM) is proposed to assess similarity between documents based on the frequencies of terms in the compared documents and Google n-gram corpus. The comparison between the frequencies of a term in a given pair of documents can quantify the importance of that term to the documents' subjects, assuming that a term is relevant to a document if it occurs multiple times. Additionally, the frequencies of documents' terms in Google n-gram corpus allows to semantically estimate their similarity. Additionally, eight variants of two clustering algorithms are applied to several real data sets in order to experimentally evaluate the quality of the clusters obtained with the proposed measure and compare it with other state-of-the-art measures. Computational complexity analysis of the compared measures are provided. The experimental results demonstrate that the proposed measure improves significantly the quality of document clustering, based on statistical tests. Additionally, we show that combining clustering results obtained with bag of words and semantic measure give better results than adopting a single approach. Finally, a study involving heuristcs to identify the number K of clusters

in a document clustering is done. A modified version of the G-means heuristic is proposed and compared to other heuristcs from the literature.

1 INTRODUÇÃO

A evolução das tecnologias de informação provocou um grande aumento no volume de dados gerados diariamente. Além do volume e a velocidade em que os dados são gerados, a variedade desses dados é um dos fatores que dificulta o seu processamento e extração de conhecimento [Parikh & Tirkha, 2013]. Para que informações extraídas de bases de dados sejam úteis, é preciso que exista um processo de descoberta de conhecimento. Tal processo é conhecido como *Knowledge Discovery in Databases* (KDD) [Tan et al., 2005].

Extrair conhecimento de dados textuais é uma tarefa que exige técnicas específicas e complexas. A estruturação e processamento de textos deve considerar as características da linguagem utilizada, a fim de manter o conhecimento expresso no texto. Para isso, o processo de KDD foi adaptado para esse formato de dados, surgindo o KDT *(Knowledge Discovery in Text)* [Hotho et al., 2005].

Uma das tarefas mais trabalhosas no processo de KDT é o pré-processamento. Nessa tarefa ocorre a limpeza dos textos, onde são removidas figuras, tabelas, palavras não relevantes e etc. Com o pré-processamento pretende-se obter uma representação da coleção textual em formato estruturado, que preserve as principais características da coleção original [Feldman & Sanger, 2006]. Segundo Rezende et al. [2011], uma coleção de textos pode ter milhões de termos que, em partes, são redundantes e pouco informativos para a descoberta de conhecimento. Esse é um problema antigo que foi nomeado por Bellman [1961] como "maldição da dimensionalidade". Em [Beyer et al., 1999], o autor mostra que o uso de um elevado número de atributos gera alta dimensionalidade. Segundo os autores, manter uma baixa dimensionalidade dos dados faz com que a capacidade de discriminação do atributo seja mantida.

Quando a análise é feita em textos, o modelo *Bag of Words* (BOW) é ampla-

mente utilizado. Neste modelo, cada termo (palavra) torna-se um atributo da base textual. Uma coleção de documentos é normalmente representada por uma matriz de documentos-termos, onde cada linha representa um documento da coleção e cada coluna representa um termo. As células da matriz são preenchidas, normalmente, com a frequência TF-IDF *(Term Frequency Inverse Document Frequency)* dos termos em cada documento. O método TF-IDF considera a frequência de termos simples (TF), inversamente ponderada pela frequência do termo nos documentos (DF) [Salton & Buckley, 1988].

O modelo BOW é baseado na ideia de que documentos relacionados possuem termos em comum, enquanto documentos não relacionados não compartilham nenhum termo. Porém, dois documentos com o mesmo significado (semântica) podem ser julgados como não relacionados, caso a sintaxe usada seja diferente. Como solução a isso, diversos métodos de análise semântica têm sido desenvolvidos. Em [Huang et al., 2008], o autor propõe o modelo *Bag of Concepts* (BOC), que utiliza conceitos obtidos da Wikipédia. A ideia é utilizar a base de artigos da Wikipédia a fim de identificar conceitos que aparecem em um documento. Um mesmo conceito pode ser obtido a partir de expressões ou termos diferentes. Desta forma, a identificação de relacionamento entre documentos não fica limitada à sintaxe utilizada. O documento passa a ser representado por um conjunto de conceitos.

Além do modelo BOC, existem outras formas de se comparar documentos semanticamente. Uma delas é com uso de medidas de similaridade semântica. Diferente das medidas de distância convencionais, como cosseno e euclidiana [Salton & McGill, 1986], que calculam a dissimilaridade entre documentos no espaço n-dimensional, as medidas semânticas utilizam a ocorrência ou frequência dos termos presentes em um determinado corpus, dicionário ou banco de palavras para estimar a similaridade entre documentos [Islam et al., 2012].

O *Google Tri-grams measure* (GTM) [Islam et al., 2012] é uma medida estatística que utiliza o corpus n-gramas do Google [Brants & Franz, 2006] a fim de encontrar similaridade semântica entre termos e documentos da língua inglesa. No cálculo da similaridade entre termos, a ideia principal é considerar todos os tri-gramas que começam e terminam com um dado par de termos e em seguida, normalizar a frequência média utilizando a frequência uni-grama de cada um deles e o uni-grama do termo mais frequente do corpus utilizado. Posteriormente, a similaridade entre documentos é mensurada computando os termos que aparecem em ambos documentos; e os pares de termos que possuem alta similaridade, de acordo com o cálculo de termo-similaridade.

Uma das técnicas de Mineração de Texto utilizada para extrair conhecimento

de um conjunto de documentos é o agrupamento, que tem como objetivo classificar os objetos de um conjunto de dados de maneira não supervisionada [Tan et al., 2005]. O agrupamento de dados busca encontrar grupos onde os documentos pertencentes a um mesmo grupo sejam altamente similares entre si e dissimilares aos documentos dos outros grupos [Pinheiro, 2008]. Isso faz com que a medida de similaridade utilizada seja diretamente responsável pela qualidade dos agrupamentos obtidos.

1.1 O problema e sua importância

A Extração de Informação (EI) [Hotho et al., 2005], ou Extração de Conhecimento (EC) [Rezende et al., 2011] é uma sub-área do KDT que busca extrair informações de interesse que, devido à quantidade e complexidade, não são observadas por um especialista [Rocha, 1999]. Para isso, métodos de agrupamento de textos podem ser utilizados para extrair padrões e organizar uma coleção de documentos de texto de maneira não supervisionada [Feldman & Sanger, 2006].

Em processamentos não supervisionados, é comum que documentos sejam agrupados juntos a outros documentos com assuntos dissimilares [Nourashrafeddin et al., 2014]. Isso ocorre por deficiências nas medidas de similaridade, que não conseguem contornar todos os problemas que envolvem o processamento de linguagem natural. Pode-se usar como exemplo palavras homônimas, que possuem mesma sintaxe, mas com sentidos diferentes. Por exemplo, a palavra "leve" pode ser um adjetivo, que caracteriza pouco peso, ou uma conjugação do verbo "levar". Outro problema comum são as palavras sinônimas como, "carro" e "automóvel", que possuem sintaxes diferentes mas o mesmo significado.

O foco dessa pesquisa foi investigar medidas de similaridade semântica para agrupamento de textos. Para isso, medidas de similaridade e algoritmos de agrupamento da literatura foram investigados e combinados, a fim de obter melhores resultados de agrupamento que o estado da arte. Foi desenvolvida uma nova medida de similaridade baseada em semântica e foram propostas diversas combinações de medidas e algoritmos de agrupamento. Para avaliar a qualidade da medida e combinações propostas, foram utilizados diversos conjuntos reais de textos e índices de validação. Também foram feitas análises da complexidade computacional dos algoritmos e medidas utilizados. Adicionalmente, foi realizado um estudo sobre heurísticas para encontrar o número ideal de grupos em tarefas de agrupamento de textos. Nesta etapa, foi feita uma análise dos conjuntos textuais e foram conduzidos experimentos utilizando heurísticas da literatura.

1.2 Objetivos

O objetivo geral deste trabalho é investigar e desenvolver uma nova medida de similaridade para tarefas de agrupamento de textos, capaz de obter melhores resultados que medidas do estado da arte.

Especificamente, pretende-se:

- Investigar e levantar os pontos positivos e negativos do estado da arte das medidas de similaridade textuais.
- Desenvolver uma nova medida híbrida, integrando os pontos fortes das medidas de similaridade semântica e das demais medidas investigadas.
- Aplicar a medida em algoritmos de agrupamento utilizando bases textuais diversificadas e conhecidas na literatura, a fim de comparar com trabalhos anteriores.

1.3 Organização da dissertação

Esta dissertação está organizada da seguinte forma: este capítulo introduz o problema e a importância de se trabalhar com agrupamento de textos, os objetivos da pesquisa e a estrutura do documento.

O Capítulo 2 faz um levantamento teórico sobre medidas de similaridade da literatura. Nele são apresentadas duas das principais métricas de distâncias, euclidiana e cosseno, além de três medidas de similaridade semântica, que fazem uso de um *corpus* para estimar a similaridade entre textos.

As técnicas de agrupamentos são abordadas no Capítulo 3. Nele é feita uma descrição de todo processo de agrupamento, desde a preparação dos dados até a avaliação dos resultados. São descritos algoritmos clássicos da literatura e algoritmos voltados somente para agrupamento de textos. Os principais índices de validação de agrupamento também são abordados nesse capítulo.

A principal contribuição deste trabalho está na criação de uma nova medida de similaridade para agrupamento de textos. Além disso, foram propostas combinações de medidas de similaridades em algoritmos de agrupamentos. Estas propostas são apresentadas no Capítulo 4. As análises de complexidade assintóticas das técnicas propostas também são apresentadas.

No Capítulo 5, são apresentados os experimentos utilizando a medida de similaridade proposta. O capítulo foi dividido em duas partes: em primeiro lugar,

são apresentados os conjuntos de dados utilizados e os resultados obtidos na tarefa de agrupamento desses conjuntos. Na segunda parte, são apresentados os resultados obtidos pela medida de similaridade proposta em tarefas de comparação de sentenças curtas.

O Capítulo 6 aborda um problema adicional ao investigado neste trabalho: o uso de heurísticas para identificação da quantidade de grupos em conjuntos textuais. Nesse capítulo são abordados os aspectos principais do problema e algumas heurísticas da literatura são apresentadas. Também é proposta uma nova versão da heurística *G-means*, que encontra grupos baseados em distribuição gaussiana. Além disso, é feito um estudo sobre os grupos dos conjuntos textuais quando utilizados em algoritmos de agrupamento baseados em K-médias. São apresentados experimentos, onde os resultados são analisados e discutidos.

Por fim, o Capítulo 7 trata das conclusões obtidas no desenvolvimento desse trabalho, além de propostas para trabalhos futuros.

2 MEDIDAS DE SIMILARIDADE

Um dos principais problemas de Mineração de Dados está em encontrar similaridade entre objetos [Rajaraman & Ullman, 2011]. Como os dados podem estar em diferentes formatos e escalas, definir uma medida de similaridade a ser utilizada não é uma tarefa trivial. No caso de medidas não ponderadas, elas consideram que todos os atributos contribuem para a similaridade na mesma proporção, portanto, o sucesso do uso de uma determinada medida de similaridade é dependente da preparação adequada dos dados [Sebastiani, 2002].

Este capítulo faz uma revisão das principais medidas de similaridade encontradas na literatura e que foram utilizadas neste trabalho. A Seção 2.1 faz uma pequena abordagem sobre métricas de distância, que calculam a similaridade entre objetos no espaço n-dimensional (onde n é a quantidade de atributos). A Seção 2.2 apresenta algumas medidas de similaridade semântica, que fazem uso de algum *corpus* para estimar similaridade entre documentos e termos com sintaxes diferentes. A Seção 2.3 trata das considerações finais deste capítulo.

2.1 Métricas de distância

Nesta seção é feita uma revisão sobre duas das principais e mais clássicas métricas de similaridade (dissimilaridade) da literatura, a distância euclidiana e similaridade cosseno. As métricas aqui apresentadas são utilizadas para atributos quantitativos, onde os dados são considerados pontos no espaço. Essas métricas também são amplamente empregadas em métodos que trabalham com documentos textuais.

2.1.1 Distância Euclidiana

A distância euclidiana é uma métrica padrão para problemas geométricos cuja ideia parte da métrica de Minkowski [Linden, 2009]. Para medir a distância entre dois documentos d_a e d_b, que são dois vetores de termos, a distância euclidiana é definida como:

$$D_E(d_a, d_b) = \sqrt{\sum_{i=1}^{T} (w_{i,a} - w_{i,b})^2} \tag{2.1}$$

onde T é a quantidade de termos diferentes contidos em ambos documentos. $w_{i,a}$ e $w_{i,b}$ são os pesos do i-ésimo termo nos documentos d_a e d_b, respectivamente. Na maioria dos casos, o peso é definido pela frequência em que o termo ocorre no documento.

Se usarmos como exemplo vetores no espaço bidimensional, a distância entre os vetores d_a e d_b é dada pelo módulo de $d_a - d_b$, conforme ilustrado na Figura 2.1.

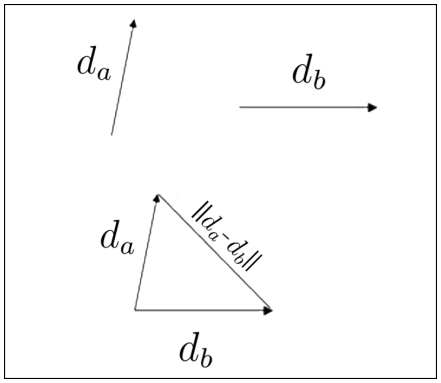

Figura 2.1: Exemplo de cálculo da distância Euclidiana entre dois vetores, d_a e d_b, no espaço bidimensional. O resultado é dado pelo módulo de $d_a - d_b$.

2.1.2 Similaridade Cosseno

A similaridade cosseno é uma das medidas de similaridade mais populares na Mineração de Texto [Feldman & Sanger, 2006]. Quando os documentos textuais são representados por vetores de termos, a similaridade entre eles pode ser obtida pela correlação entre os seus respectivos vetores. Quando essa correlação é quantificada como o cosseno do ângulo entre dois vetores, chamamos esse valor de similaridade cosseno. A Figura 2.2 ilustra como é obtido o ângulo entre dois vetores no espaço bidimensional.

Dado dois documentos textuais, d_a e d_b, a similaridade cosseno é definida como:

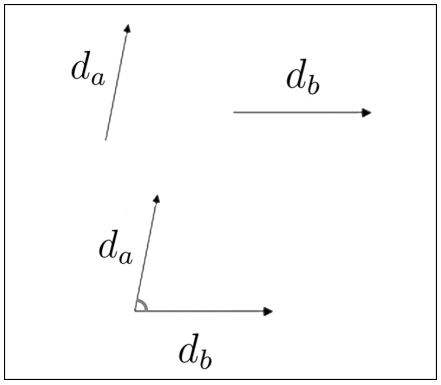

Figura 2.2: Exemplo de medição do ângulo entre dois vetores no espaço bidimensional. O cosseno do ângulo é utilizado como medida de similaridade entre os vetores.

$$cos(d_a, d_b) = \frac{d_a \cdot d_b}{|d_a| \times |d_b|} \tag{2.2}$$

onde cada dimensão dos documentos d_a e d_b representa um termo e seu respectivo peso. O peso de um termo em um documento é sempre não-negativo, com isso, a similaridade cosseno entre dois documentos é sempre não-negativa e delimitada entre [0,1] [Tan et al., 2005]. Entende-se $d_a \cdot d_b$ como o produto escalar de d_a com d_b. Já $|d_a| \times |d_b|$ é o produto das normas de d_a e d_b.

Quando a similaridade cosseno entre dois vetores (documentos) é próxima de 1, significa que os vetores formam um ângulo próximo de 0^o. Logo, os componentes dos vetores são semelhantes e, portanto, os documentos são semelhantes. Já quando a similaridade é próxima de 0, significa que os vetores formam um ângulo próximo de 90^o. Isso indica que os componentes dos vetores não são semelhantes e, portanto, os documentos também não são semelhantes.

2.2 Similaridade Semântica

As medidas de similaridade semântica são medidas estatísticas determinísticas. Medidas de similaridade estatísticas são amplamente usadas na literatura [Islam et al., 2012]. Essas medidas fazem uso de um *corpus* disponível na linguagem a ser analisada, que serve como parâmetro para estimar a similaridade entre dois termos ou documentos.

Diferente das métricas de distância, as medidas semânticas conseguem identificar similaridade entre sentenças, mesmo que não compartilhem nenhum termo. Por exemplo, ao se comparar as seguintes frases: *"We like dogs"* × *"I love puppies"*, a similaridade entre as mesmas, com uso das medidas euclidiana ou cosseno, é nula.

Isso porque as frases não compartilham termos. Nesses casos são aplicáveis as medidas semânticas de similaridade, pois, apesar de distintas, as sentenças são muito parecidas.

2.2.1 *Google Tri-grams Measure* (GTM)

A medida de similaridade GTM, utilizando *tri-grams*, foi proposta em [Islam et al., 2012] e utiliza as frequências dos termos disponíveis no *corpus* Google *n-grams* [Brants & Franz, 2006]. O Google disponibiliza em seu *corpus* as frequências de *1-gram*, ou seja, de um termo, até *5-grams*, que é a frequência em que cinco termos aparecem juntos. O *corpus* foi construído a partir de páginas públicas em inglês na Web, resultando em um conjunto com mais de 1 trilhão de símbolos, quase 14 milhões de *uni-grams* e mais de 1 bilhão de *5-grams*.

Ao se analisar o significado de uma palavra em uma frase, Kaplan [1955] afirma que a percepção de sentido com uma palavra de cada lado é mais eficaz do que com duas palavras antes e depois. Ele constatou que utilizar cinco palavras ou uma frase inteira, não é significantemente melhor ao uso de *tri-grams*, portanto, optou-se por utilizar *tri-grams*.

A medida GTM é dividida em duas fases: o cálculo da similaridade para cada par de termos, e o cálculo da similaridade entre documentos completos.

2.2.1.1 Cálculo de termo-similaridade

Considerando dois termos, wd_a e wd_b, pretende-se encontrar a similaridade semântica entre eles. A frequência *uni-gram* de um termo wd_i no *corpus* do Google é dado por $c(wd_i)$. Já $c(wd_awd_jwd_b)$ é a frequência *tri-grams* de $wd_awd_jwd_b$. Se existem $n1$ *tri-grams* que começam com o termo wd_a e terminam com wd_b, consideramos que $N1 = \sum_{i=1}^{n1} c(wd_awd_iwd_b)$. Se existem $n2$ *tri-grams* que começam com o termo wd_b e terminam com wd_a, consideramos que $N2 = \sum_{i=1}^{n2} c(wd_bwd_iwd_a)$. Desta forma, definimos a função $\mu(N1, N2) = \frac{1}{2}(N1 + N2)$, que representa a média da frequência dos *tri-grams* que começam com wd_a e terminam com wd_b e vice-versa. Considerando que C_f seja a frequência do termo *uni-gram* mais frequente do *corpus*, e a função $min(x, y)$ retorna o menor valor entre x e y, a similaridade *tri-grams* entre wd_a e wd_b, $Sim(wd_a, wd_b) \in [0, 1]$ é definida pela Equação 2.3.

$$Sim(wd_a, wd_b) = \begin{cases} \frac{\log \frac{\mu(N1,N2)C_f^2}{c(wd_a)c(wd_b)min(c(wd_a),c(wd_b))}}{-2\times\log\frac{min(c(wd_a),c(wd_b))}{C_f}} & \text{se } \frac{\mu(N1,N2)C_f^2}{c(wd_a)c(wd_b)min(c(wd_a),c(wd_b))} > 1 \\ \frac{\log 1.01}{-2\times\log\frac{min(c(wd_a),c(wd_b))}{C_f}} & \text{se } \frac{\mu(N1,N2)C_f^2}{c(wd_a)c(wd_b)min(c(wd_a),c(wd_b))} \leq 1 \\ 0 & \text{se } \mu(N1,N2)C_f^2 = 0 \end{cases} \tag{2.3}$$

Seguindo o exemplo dado no início da Seção 2.2, se utilizamos a Equação 2.3 para estimar a similaridade entre as palavras *"dogs"* e *"puppies"*, teremos o seguinte processo:

- ***Uni-grams***
  - ***dogs*:** aparece 28.409.576 vezes no *corpus* (chamado $C(wd_a)$)
  - ***puppies*:** aparece 4.733.987 vezes no *corpus* (chamado $C(wd_b)$)
- ***Tri-grams***
  - Começa com *"dogs"* e termina com *"puppies"*: 259.825 (chamado $N1$)
    Exemplos: *dogs and puppies, dogs or puppies, dogs are puppies,* etc.
  - Começa com *"puppies"* e termina com *"dogs"*: 65.665 (chamado $N2$)
    Exemplos: *puppies to dogs, puppies PUN*[1] *dogs, puppies all dogs,* etc.
- ***Uni-gram* mais frequente do *corpus***
  - ***the***: 23.135.851.162 vezes (chamado C_f)

Com as consultas feitas ao *corpus*, verifica-se em qual das condições da Equação 2.3 as frequências resultam. Uma vez que:

$$\frac{\mu(259.825, 65.665) \cdot 23.135.851.162^2}{28.409.576 \cdot 4.733.987 \cdot 4.733.987} = 136.823,2 > 1$$

aplica-se a primeira opção, que resulta na seguinte operação:

$$\frac{\log 136.823,2}{-2 \times \log \frac{4.733.987}{23.135.851.162}} = 0,696$$

ou seja, $Sim(dogs, puppies) = 0,696$.

Como é possível perceber, embora as sintaxes dos termos em comparação sejam completamente diferentes, a proximidade e frequência com que os termos aparecem

[1]PUN pode ser qualquer sinal de pontuação. Ex.:(. , ; : ? !).

no *corpus* permite estimar uma similaridade entre os mesmos. Uma vez estimada a similaridade entre os termos, é possível calcular a similaridade entre frases ou documentos inteiros. Esse cálculo será abordado na próxima seção.

2.2.1.2 Cálculo de documento-similaridade

Para esta medida de similaridade, as frequências dos termos não são consideradas, apenas a ocorrência. O cálculo da similaridade consiste nos cinco passos a seguir:

- **Passo 1:** Tem-se como entrada dois documentos de textos pré-processados, $P = \{p_1, p_2, ..., p_m\}$ e $R = \{r_1, r_2, ..., r_n\}$, onde m e n são as quantidades de termos e $n \geq m$. Caso contrário, troque P com R.

- **Passo 2:** Conta-se o número de termos que ocorrem em ambos documentos (dado por δ). Ou seja, $\delta = \delta + 1$ para cada $p = r$, $\forall p \in P, \quad \forall r \in R$. Ao fim da contagem, remove-se os δ termos de P e R, assim, $P = \{p_1, p_2, ..., p_m - \delta\}$ e $R = \{r_1, r_2, ..., r_n - \delta\}$. Se $m - \delta = 0$ pule para o passo 5.

- **Passo 3:** Constrói-se uma matriz de similaridade semântica $(\alpha_{ij})_{(m-\delta)\times(n-\delta)}$ utilizando o seguinte processo: $\alpha_{ij} \leftarrow Sim(p_i, r_j)$ (utilizando a Equação 2.3). Chamaremos essa matriz de M.

- **Passo 4:** Utilizando as notações μ para a média e σ para o desvio padrão, considera-se que cada uma das $(m - \delta)$ linhas da matriz M é um conjunto de $(n - \delta)$ elementos, calcula-se μ e σ de cada linha e encontra-se os elementos maiores que $(\mu + \sigma)$ da respectiva linha. Se existem y_i elementos na linha i que atendem a este requisito, chamaremos este conjunto com y_i elementos de A_i. Formalmente, A_i pode ser definido da seguinte forma:

$$A_i = \{\alpha_{ij} : \alpha_{ij} \in M_i, \quad \alpha_{ij} > (\mu(M_i) + \sigma(M_i))\}$$

  A média dos y_i elementos do conjunto A_i é $\mu(A_i)$. O somatório de todas as $(m - \delta)$ linhas da matriz M é $\sum_{i=1}^{m-\delta} \mu(A_i)$.

- **Passo 5:** A similaridade entre os documentos P e R é dada pela média harmônica da soma de δ com o somatório obtido no passo 4. Desta forma, $S(P, R) \in [0, 1]$.

$$S(P, R) = \frac{(\delta + \sum_{i=1}^{m-\delta} \mu(A_i)) \times (m + n)}{2mn} \tag{2.4}$$

A ideia do Passo 4 é utilizar apenas os pares de palavras que possuem alta similaridade. Desta forma, a similaridade entre duas sentenças é estimada pelos termos em comum e termos com alta similaridade entre si.

2.2.2 Similaridade baseada em conceitos da Wikipédia

O método *Bag of Concepts* (BOC) foi proposto em [Huang et al., 2008] baseado na ideia do clássico *Bag of Words* (BOW), mas utilizando informações extraídas da base de dados da Wikipédia[2]. O BOC segue três passos principais:

1. Identificar frases candidatas em um dado documento.
2. Mapear as frases em artigos (ou tópicos) da Wikipédia.
3. Selecionar os conceitos mais importantes relacionados aos artigos.

A saída é uma lista de conceitos representando os tópicos extraídos do documento. Cada conceito é ponderado pela frequência em que o tópico ocorre no documento. A Figura 2.3 ilustra um exemplo de obtenção de conceitos a partir de tópicos da Wikipédia identificados em um texto. As palavras destacadas em negrito e sublinhadas no texto são tópicos da Wikipédia. As palavras e frases nos balões à direita na Figura 2.3 são os conceitos relacionados aos tópicos. Para os 8 tópicos da Wikipédia encontrados no texto da Figura 2.3, foram extraídos 11 conceitos relacionados. Esses conceitos formam o BOC do texto.

Para definir se uma frase é candidata a ser utilizada como fonte para extração de conceito na Wikipédia, é aplicada a seguinte equação:

$$\frac{f_a(p)}{f_a(p) + f_t(p)} \tag{2.5}$$

onde p é a frase candidata, $f_a(p)$ é o número de artigos da Wikipédia em que a frase é utilizada como âncora, e $f_t(p)$ é o número de artigos em que a frase aparece de qualquer forma. Cada link na Wikipédia está associado a um texto âncora, que pode ser considerado como um descritor do seu artigo alvo. Os textos de âncora têm um grande valor semântico: fornecem nomes alternativos, variações morfológicas e frases relacionadas para os artigos de destino. As âncoras também codificam a polissemia, porque a mesma âncora pode ligar a diferentes artigos, dependendo do contexto em que é encontrada.

[2] `http://en.winkipedia.org`

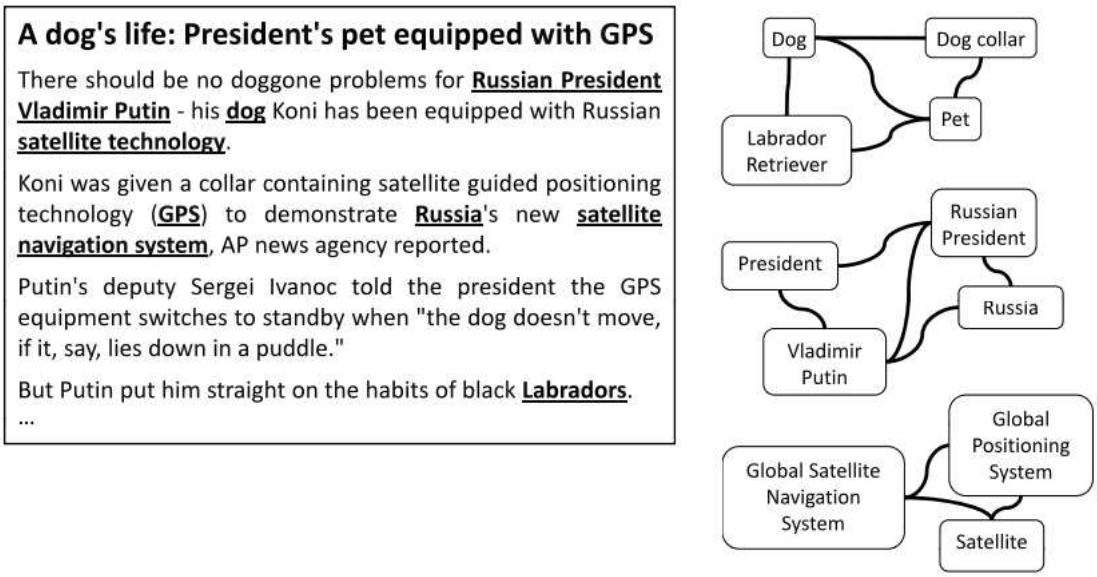

Figura 2.3: Exemplo de conceitos (à direita) extraídos a partir tópicos Wikipédia (em negrito e sublinhado) encontrados em um texto (à esquerda).
Fonte: Adaptado de [Milne & Witten, 2013].

Ao utilizar BOC, duas ou mais frases podem direcionar para um mesmo conceito. Isso permite acessar similaridade entre documentos, mesmo quando utilizam termos ou frases com diferentes sintaxes.

2.2.3 *Multiple to Multiple Mapping (M3) Measure*

A medida *Multiple to Multiple Mapping* (M3) é uma variação semântica para a similaridade cosseno [Wang, 2015]. Supõe-se que um documento d_a é representado pelo vetor de termos (t_1, t_2, t_3), e w_i é o peso (frequência) do termo t_i. Da mesma forma, um documento d_b é representado pelo vetor (t'_1, t'_2, t'_3) e w'_i é o peso do termo t'_i. Assim como na Equação 2.2, a similaridade cosseno entre esses dois documentos pode ser calculada conforme expresso na Equação 2.6.

$$\cos(d_a, d_b) = \frac{[w_1, w_2, w_3] \cdot \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} w'_1 \\ w'_2 \\ w'_3 \end{bmatrix}}{\sqrt{\sum_{i=1}^{3} w_i^2 \sum_{i=1}^{3} w_i'^2}} \tag{2.6}$$

Chamaremos a matriz identidade expressa na Equação 2.6 de H. Fica claro que a multiplicação entre os vetores e H resulta nos próprios vetores. Logo, a similaridade só considera os termos coincidentes em ambos os documentos. A ideia

da medida M3 é substituir a matriz H por uma matriz de relacionamento entre todos os termos. Desta forma, a similaridade M3 entre d_a e d_b é obtida pela seguinte equação:

$$\text{simM3}(\vec{d_a}, \vec{d_b}) = \frac{[w_1, w_2, w_3] \cdot \begin{bmatrix} 1 & rel(t_1, t'_2) & rel(t_1, t'_3) \\ rel(t_2, t'_1) & 1 & rel(t_2, t'_3) \\ rel(t_3, t'_1) & rel(t_3, t'_2) & 1 \end{bmatrix} \cdot \begin{bmatrix} w'_1 \\ w'_2 \\ w'_3 \end{bmatrix}}{\sqrt{\sum_{i=1}^{3} w_i^2 \sum_{i=1}^{3} w_i'^2}} \tag{2.7}$$

onde $rel(t_i, t'_j)$ é a similaridade entre i-ésimo termo do documento d_a e o j-ésimo termo do documento d_b. Desta forma, a medida de similaridade considera os relacionamentos entre todos os pares de termos nos documentos.

Em [Wang, 2015], a autora utiliza conceitos da Wikipédia para medir o relacionamento entre os termos, mas outras similaridade também podem ser utilizadas, como a obtida na Equação 2.3 da medida GTM, por exemplo.

2.3 Considerações finais

Neste capítulo foi feita uma revisão geral de importantes medidas de similaridade da literatura. As medidas aqui apresentadas serão utilizadas e comparadas no decorrer desta dissertação. Neste trabalho, também propomos a combinação de métricas de distâncias com medidas semânticas de similaridade para tarefas de agrupamento de textos.

3 TÉCNICAS DE AGRUPAMENTO

No Capítulo 2, foi feita uma revisão sobre algumas das principais medidas de similaridade encontradas na literatura. O objetivo deste capítulo é abordar as principais técnicas e algoritmos para agrupamento de documentos textuais, desde o pré-processamento até a avaliação dos resultados obtidos.

Segundo Rajaraman & Ullman [2011], agrupamento é o processo de examinar uma coleção de "pontos" e separá-los em grupos de acordo com alguma medida de similaridade. Técnicas de agrupamento devem ser capazes de encontrar grupos de objetos semelhantes de maneira não supervisionada, ou seja, sem utilizar conhecimento externo [Jain & Dubes, 1988].

Embora pareça simples, há subjetividade na definição do que realmente é um grupo [Tan et al., 2005]. Por exemplo, a Figura 3.1 mostra que existem diversas maneiras de agrupar um mesmo conjunto de dados, a Figura 3.1(a) ilustra o conjunto de dados original, a Figura 3.1(b) mostra o conjunto dividido em dois grupos distintos, e a Figura 3.1(c) divide o conjunto em 3 grupos.

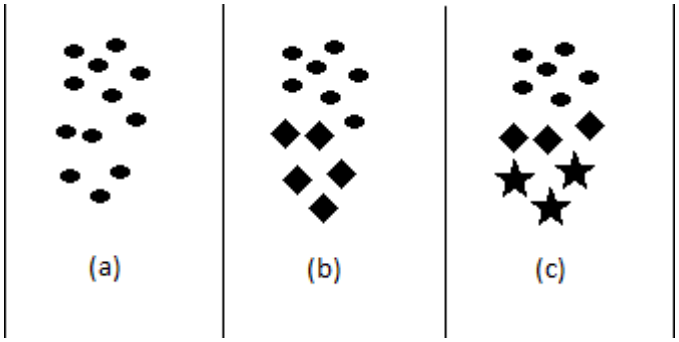

Figura 3.1: Formas diversas de se agrupar um mesmo conjunto de dados.
Fonte: Adaptado de [Tan et al., 2005].

Este capítulo é dividido da seguinte forma: na Seção 3.1, são abordados os

principais métodos para pré-processamento e estruturação dos dados textuais; a Seção 3.2 apresenta alguns métodos para seleção de atributos. Os principais e mais clássicos algoritmos de agrupamento particionais da literatura são apresentados na Seção 3.3; a Seção 3.4 aborda o problema específico de agrupamento de documentos textuais, onde alguns algoritmos do estado-da-arte são apresentados; os índices de validação, para mensurar a qualidade dos agrupamentos, são apresentados na Seção 3.5; por fim, a Seção 3.6 trata das considerações finais deste capítulo.

3.1 Preparação dos dados

Um dos problemas durante o processo de extração de conhecimento é a falta de padronização no armazenamento dos dados. Por isso, é necessário aplicar várias técnicas de pré-processamento para adequar os conjuntos, e por fim, utilizar algoritmos de Mineração de Dados [Faceli et al., 2011]. Uma das tarefas mais trabalhosas no processo de KDT é o pré-processamento. Nessa etapa, é necessário limpar os textos, onde ocorre a remoção de figuras, tabelas, palavras não relevantes, *stopwords*, etc. Segundo Aranha [2007], *stopwords* são termos comuns que ocorrem em todos os tipos de textos e não colaboram para a descoberta de conhecimento. Com o pré-processamento pretende-se obter uma representação da coleção textual em formato estruturado, que preserve as principais características da coleção original [Feldman & Sanger, 2006].

Um dos modelos mais comuns na estruturação de documentos textuais é o *Bag of Words* (BOW). Neste modelo, cada documento é estruturado como um vetor, onde cada termo (palavra) é uma dimensão do espaço a ser considerado. Uma coleção de documentos é normalmente representada por uma matriz de documentos-termos, onde cada linha representa um documento (vetor de documento) da coleção, e cada coluna representa um termo (vetor de termo). As células da matriz são preenchidas, normalmente, com a frequência em que os termos aparecem nos documentos.

Os métodos de normalização de frequências são aplicados aos vetores de documentos e termos a fim de adequar e ponderar a importância dos termos de acordo com sua capacidade discriminativa. A normalização TF-IDF *(Term Frequency Inverse Document Frequency)* [Salton & Buckley, 1988] é uma das mais usadas da literatura [Aranha, 2007]. O método considera a frequência de termos simples (TF) inversamente ponderada pela frequência do termo nos documentos (DF). Ou seja, dado uma coleção com N documentos, onde w_{ij} é a frequência (contagem de ocorrências) do termo i no documento j, o termo-frequência TF_{ij} será:

$$TF_{ij} = \frac{w_{ij}}{max_p w_{pj}} \tag{3.1}$$

onde $max_p w_{pj}$ é a maior frequência de qualquer termo no mesmo documento j (excluindo as *stopwords*). Desta forma, o termo com maior frequência no documento j terá $TF = 1$.

Para calcular o IDF_i, suponha que o termo i ocorra em n_i dos N documentos da coleção. Desta forma, $IDF_i = \log_2(N/n_i)$. Por fim, o valor TF-IDF do termo i no documento j é dado por $TF_{ij} \times IDF_i$. Os termos com maiores pontuações TF-IDF são frequentemente os termos que melhor caracterizam o tópico do documento [Rajaraman & Ullman, 2011].

Outro processo da preparação dos dados textuais é a identificação de variações morfológicas e termos sinônimos. Um dos processos mais comuns para essa tarefa é a redução das palavras às suas raízes, utilizando processos de *stemming* [Rezende et al., 2011]. Em medidas de similaridade semântica, que fazem consultas a um *corpus*, técnicas de *stemming* não são empregadas [Nourashrafeddin, 2014]. Isso porque as consultas ao *corpus* são feitas a palavras completas, ou seja, as palavras presentes no *corpus* não estão em seu formato de *stem*. Para métricas de distâncias, *stemming* é amplamente aplicado [Aranha, 2007; Naldi et al., 2011; Rezende et al., 2011; Nourashrafeddin et al., 2014]. No Apêndice B é feito um estudo comparativo do uso de bases com e sem *stemming* para agrupamento de textos.

3.2 Seleção de atributos

O uso de um elevado número de atributos gera alta dimensionalidade [Beyer et al., 1999], o que inibe a capacidade discriminativa dos atributos. Métodos de seleção são utilizados para remover atributos menos discriminativos, evitando que isso aconteça. Nesta seção serão apresentados dois métodos conhecidos na literatura [Kogan et al., 2003; Tang et al., 2005; Nourashrafeddin et al., 2013] que são de interesse para este trabalho.

3.2.1 Mean-TFIDF

No modelo BOW, um documento é representado por um vetor de termos. De igual forma, se considerarmos as colunas, um termo é representado por um vetor de documentos, onde cada componente contém a frequência TF-IDF em que o termo aparece no documento. Na seleção Mean-TFIDF [Tang et al., 2005], para cada

termo t_j do conjunto de documento, sua média TF-IDF sobre todos os documentos é calculada a partir da seguinte equação:

$$\text{Mean-TFIDF}(t_j) = \frac{1}{N}\sum_{i=1}^{N}\text{TFIDF}_{ij} \tag{3.2}$$

onde N é a quantidade de documentos no conjunto e TFIDF_{ij} é a frequência TF-IDF do termo t_j no documento d_i. Quanto maior a pontuação Mean-TFIDF do termo, mais relevante é o termo e, portanto, maior a chance de ser preservado.

3.2.2 VAR-TFIDF

Diferente da seleção Mean-TFIDF, a VAR-TFIDF baseia-se na variância em que um determinado termo aparece em um conjunto de documentos [Kogan et al., 2003]. Para cada termo t_j do conjunto de dados, a pontuação VAR-TFIDF do termo sobre todos os documentos é medida usando a seguinte equação:

$$\text{VAR-TFIDF}(t_j) = \frac{1}{N-1}\sum_{i=1}^{N}(\text{TFIDF}_{ij} - \text{Mean-TFIDF}(t_j))^2 \tag{3.3}$$

onde $\text{Mean-TFIDF}(t_j)$ é obtido pela Equação 3.2.

Quando um termo possui alta frequência em um determinado número de documentos e frequência baixa ou nula nos demais, ele possui alta variância. A ideia é que os termos com maior variância sejam discriminantes para formação de grupos. Com os termos ordenados por pontuação VAR-TFIDF, basta selecionar um *threshold* para eliminar os termos menos relevantes. No Apêndice A, é apresentado um estudo para definir o *threshold* utilizado na parte experimental deste trabalho (Capítulo 5).

3.3 Algoritmos Clássicos

O problema de agrupamento de dados pode ser visto como um problema de otimização, onde o objetivo é minimizar o somatório das distâncias entre os objetos do mesmo grupo. Formalmente, a seguinte função objetivo pode ser considerada:

$$Minimizar \sum_{i=1}^{n}\sum_{j=1}^{n} dist(d_i, d_j)\cdot x_{ij} \tag{3.4}$$

- $dist(d_i, d_j)$ retorna a dissimilaridade do documento d_i ao documento d_j.

- x_{ij} $\begin{cases} 1 \text{ - se o documento } d_i \text{ pertence ao mesmo grupo de } d_j \\ 0 \text{ - caso contrário} \end{cases}$

Qualquer objeto pode ser atribuído a qualquer grupo, podendo haver inúmeras combinações de dados e diversas quantidades de grupos. Os algoritmos de agrupamento são heurísticas que buscam soluções sub-ótimas para esse problema.

Nesta seção são apresentados os mais clássicos algoritmos de agrupamento de dados da literatura. O K-médias e K-medóides são algoritmos particionais *hard*, ou seja, cada objeto pertence a apenas um grupo. Já o algoritmo *fuzzy c-means* é não exclusivo, ou seja, são geradas partições *fuzzy*, onde cada objeto possui graus de associação aos grupos. Nesse caso, pode-se desfuzzificar as partições *fuzzy* em partições *soft*, ou seja, onde os objetos pertencem a um grupo ou mais.

3.3.1 *K*-médias

O K-médias (ou *K-means*) é um algoritmo particional de agrupamento de dados baseado em protótipos [Larose, 2005]. O K-médias visa encontrar grupos por meio de centróides, que são formados pelas médias entre os objetos de um mesmo grupo. Medidas de similaridade são usadas para definir grupos, de forma que um determinado objeto é mais semelhante aos objetos de seu grupo e menos semelhante aos objetos de outros grupos no conjunto de dados.

O algoritmo K-médias possui características simples e conta com vários métodos de implementação disponíveis na literatura. O parâmetro K, que indica a priori a quantidade de grupos que se deseja obter no agrupamento, deve ser definido pelo usuário ou estimado[1]. No Algoritmo 1 são apresentados os passos para implementação do algoritmo K-médias clássico, onde o atributo K é um parâmetro de entrada definido pelo usuário. A Figura 3.2 ilustra um exemplo de execução do algoritmo K-médias para $K = 3$.

Em 2009, o K-médias foi considerado um dos dez algoritmos mais influentes na mineração de dados [Wu & Kumar, 2009]. Isso se dá pela simplicidade e escalabilidade, uma vez que o K-médias possui complexidade $O(t \cdot N \cdot K)$, onde t é a quantidade de iterações do K-médias, N é a quantidade de objetos e K é a quantidade de grupos [Wu et al., 2007]. Desta forma, a complexidade é linear para qualquer variável do problema.

[1] No Capítulo 6, são abordadas heurísticas baseadas em conceitos estatísticos e meta-heurísticas evolutivas para estimar a quantidade K de grupos.

Algoritmo 1: K-médias Clássico

Entrada: Conjunto de Dados, K

Saída: K grupos de dados

1: Escolhe K centróides iniciais aleatoriamente, definidos pelo usuário.
2: Calcula a distância de cada objeto aos centróides.
3: Atribui cada objeto a seu centróide mais próximo.
4: Atualiza a posição dos centróides para a médias dos objetos de cada grupo.
5: Repete passo 2, 3 e 4, até que nenhum centróide mude de posição.

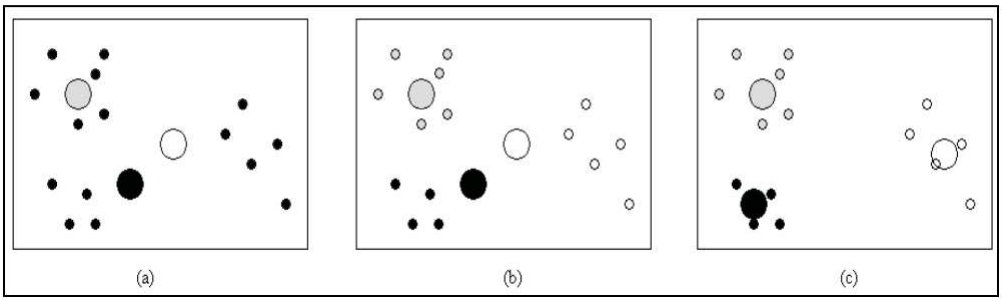

Figura 3.2: Exemplo de execução do algoritmo K-médias. (a) Foram criados três centróides aleatórios (círculos maiores). (b) Os elementos foram designados para os grupos cujos centróides lhe estão mais próximos (c) Os centróides foram recalculados. Os grupos já estão em sua forma final. Caso não estivessem, repetiríamos os passos (b) e (c) até que estivessem.

Fonte: Adaptado de [Linden, 2009].

3.3.2 *K*-medóides

O K-medóides é a versão relacional do algoritmo K-médias. Diferente do K-médias, o K-medóides dispões apenas de uma matriz de dissimilaridade entre os objetos do conjunto de dados, logo, não é possível determinar a dissimilaridade para objetos que não estão no conjunto, como é o caso do centróide. Nesse cenário, emprega-se uma técnica para encontrar o objeto mais central do grupo, chamado de medóide [Krishnapuram et al., 2001]. A tarefa de definir o medóide de um grupo consiste em minimizar o somatório das distâncias entre os objetos do mesmo grupo, a partir de um objeto fixo. Formalmente, dado um conjunto com N objetos e K sub-conjuntos, onde G_i é o i-ésimo sub-conjunto, para encontrar o medóide (dado por, m_i) de G_i, calcula-se:

$$m_i \leftarrow argmin_{\{i':i' \in G_i\}} \sum_{j=1}^{N} u_j * dist(j, i'), \quad u_j = \begin{cases} 1, \text{ se } \ j \in G_i \\ 0, \text{ outro caso} \end{cases}$$

Caso ocorra um empate, ou seja, mais de um objeto minimiza a soma das

distâncias em um mesmo grupo, escolhe-se um dos índices aleatoriamente [Horta, 2013] ou pode-se escolher o índice que maximize a distância para o vizinho mais próximo, pertencente a outro grupo. A atualização dos medóides no K-medóides possui complexidade N_k^2 para cada grupo, onde N_k é a quantidade de objetos no grupo k. Portanto, K-medóides tende a ser mais dispendioso computacionalmente que o K-médias [Hastie et al., 2009].

3.3.3 *Fuzzy c-means*

Quando um algoritmo *fuzzy* é aplicado a um conjunto de dados, o resultado é uma matriz *fuzzy*, de modo que:

$$\begin{cases} P = [p_{ij}]_{c \times N}, \\ p_{ij} \in [0, 1], \end{cases} \tag{3.5}$$

onde P é uma matriz de $c \times N$, onde c e N são as quantidades de grupos e objetos, respectivamente. O valor p_{ij} é o grau de associação do j-ésimo objeto ao i-ésimo grupo *fuzzy*. Desta forma, todo objeto de dados possui algum grau de associação a todos os grupos, onde esse grau pode também ser nulo.

O algoritmo *fuzzy c*-means [Dunn, 1973] é uma extensão *fuzzy* do algoritmo K-médias. Considerando que d_j é o j-ésimo documento do conjunto, a versão básica do algoritmo *fuzzy c*-means consiste nos seguintes passos:

1. Seleciona o número c de grupos *fuzzy*.
2. Seleciona c protótipos iniciais, $v_1, v_2, \cdots, v_c$.
3. Calcula a distância $||d_j - v_i||$ entre os documentos e protótipos.
4. Calcula os elementos da matriz de partição *fuzzy*:

$$p_{ij} = \left[\sum_{l=1}^{c} \left(\frac{\|d_j - v_i\|}{\|d_j - v_l\|} \right)^2 \right]^{-1} \tag{3.6}$$

5. Atualiza os centróides dos grupos:

$$v_i = \frac{\sum_{j=1}^{N} p_{ij}^2 d_j}{\sum_{j=1}^{N} p_{ij}^2} \tag{3.7}$$

6. Para se os centróides não mudarem de posição ou se atingir uma quantidade t de iterações. Caso contrário, volta no Passo 3.

A Equação 3.6 requer que $\|d_j - v_i\| > 0 \quad \forall j \in \{1, \cdots, N\}$ e $i \in \{1, \cdots, c\}$. Para qualquer j, se $\|d_j - v_i\| = 0$ para $i \in I \subseteq \{1, \cdots, c\}$, então defina p_{ij} de tal modo que: (a) $p_{ij} = 0$ para $i \in I$; e (b) $\sum_{i \in I} p_{ij} = 1$

3.4 Agrupamento de Textos

Nesta seção, são apresentados dois algoritmos para agrupamento de textos. Embora esses algoritmos utilizem ou sejam baseados nos algoritmos clássicos de agrupamento, eles possuem passos específicos para análise textual, como agrupamento de termos para identificação de assuntos. Isso os torna mais eficazes que o uso dos algoritmos clássicos isoladamente [Nourashrafeddin, 2014].

3.4.1 *Lexical Document Clustering* (LDC)

O algoritmo de agrupamento particional de documentos, *Lexical Document Clustering* (LDC) [Nourashrafeddin et al., 2013], é dividido em 3 fases principais: na primeira fase, a ideia está em encontrar *termos chave* que representam tópicos. A segunda fase tem a tarefa de selecionar documentos que melhor representam esses tópicos. A terceira e última fase consiste no agrupamento dos documentos. As fases serão detalhadas nas Seções 3.4.1.1, 3.4.1.2 e 3.4.1.3. Os passos do LDC são mostrados no Algoritmo 2 e descritos nas próximas seções.

Algoritmo 2: *Lexical Document Clustering* (LDC)

Entrada: uma matriz de documentos-termos, K
Saída: K grupos de documentos

1: Usa *fuzzy c-means* para gerar K grupos de termos
2: Remove termos não discriminativos dos grupos de termos.
3: **Para** cada grupo de termos selecionados **faça:**
4: Extrai documentos representativos (semente)
5: Calcula um documento centróide para os documentos semente
6: **fim Para**
7: **Para** cada documento da matriz **faça:**
8: Mede a distância para todos os documentos centróides
9: Atribui o documento ao centróide mais próximo
10: **fim Para**

3.4.1.1 Agrupamento de termos

O agrupamento de termos é uma tarefa do LDC que busca dividir os termos por tópicos. Quando um conjunto de documentos possui tópicos similares, é normal que algum termo pertença a mais de um tópico. Em razão disso, o LDC implementa a versão *fuzzy* do algoritmo *c-means*, utilizando um método de desfuzzificação para gerar partições *soft*. Isto é, dado como entrada uma matriz de documentos-termos, o *fuzzy c-means* retorna K grupos de termos, onde cada grupo contém somente os termos que, ao final da execução, possuem grau de filiação maior que $1/K$. Com isso, o mesmo objeto (termo) pode pertencer a mais de um grupo. A Figura 3.3 ilustra a importância de se utilizar métodos *fuzzy* em agrupamento de termos. A medida utilizada no *fuzzy c-means* é a similaridade cosseno.

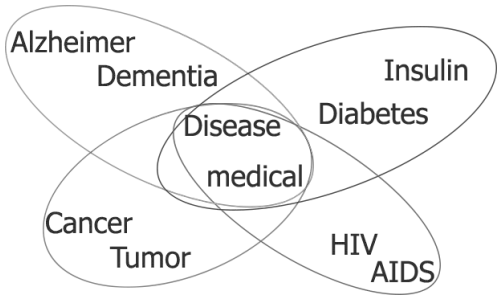

Figura 3.3: Conjunto de 10 termos agrupados em 4 grupos por método *fuzzy* com particionamento *soft*. Se fosse utilizado particionamento *hard*, os termos *"medical"* e *"Disease"* deveriam ser atribuídos a um só grupo, o que causaria desequilíbrio na identificação de assuntos, uma vez que esses dois termos são relacionados a todos os 4 grupos.

Com os termos agrupados, o objetivo agora é remover os termos não discriminativos de cada grupo, mantendo apenas os termos representativos, chamados de *termos tópicos chave*. Um algoritmo guloso é executado para realizar essa tarefa, seguindo os seguintes passos:

1. Calcula a pontuação de cada termo utilizando o método de seleção de atributos VAR-TFIDF, definido anteriormente na Equação 3.3.

2. Para cada grupo de termos, calcula a média das pontuações dos termos inclusos no grupo.

3. Em cada grupo de termos, os termos com pontuações abaixo da média do grupo são removidos. Os termos restantes são considerados *termos chave* e mantidos para a próxima etapa.

A saída dessa fase são K grupos de *termos chave*, encontrados pelo *fuzzy c-means* e selecionados pelo algoritmo guloso.

3.4.1.2 Encontrar documentos semente

Os grupos de *termos chave* gerados na fase anterior são utilizados nesta fase para encontrar os *documentos semente* de cada tópico. Os *documentos semente* de um grupo de termos são aqueles que são mais próximos uns aos outros, utilizando o sub-espaço gerado pelos *termos chave* do grupo. O método para extração dos *documentos semente* segue os seguintes passos:

1. Um termo-centróide (TC) é gerado utilizando a representação documentos-termos. Um vetor TC é a coluna média dos vetores correspondentes aos termos inclusos em um grupo de *termos chave*. Ele é um vetor com dimensionalidade igual ao número de documentos.

2. O algoritmo K-médias, com $K = 2$, é aplicado ao TC (no espaço unidimensional) para dividir os elementos em dois grupos. Um dos grupos contém os elementos com valores próximos de zero e o outro grupo inclui os elementos com valores elevados, diferentes de zero. Os elementos com valores próximos de zero representam os documentos que os termos do conjunto de *termos chave* possuem baixa frequência. Os outros documentos são os que melhor representam o grupo de termos e são chamados de *documentos semente*.

Um mesmo documento pode ser semente em mais de um tópico. O valor TF-IDF do documento no TC é utilizado para ponderar a importância do documento em cada grupo de termos. Esses pesos serão utilizados ao calcular documentos-centróides.

3.4.1.3 Agrupamento de documentos

Esta fase do algoritmo tem como entrada os *documentos semente* gerados no passo anterior. Dados os *documentos semente*, os documentos são agrupados seguindo os seguintes passos:

1. Para cada grupo de termos, é calculado um documento centróide sobre os seus *documentos semente*. Um documento-centróide (DC) é a linha média dos vetores de documentos correspondentes aos *documentos semente* (DS):

$$DC_p = \frac{\sum_{d_i \in DS_p} w_i \times d_i}{|DS_p|} \tag{3.8}$$

onde w_i é o valor médio TF-IDF do documento d_i no termo-centróide TC_p, e $|\cdot|$ indica a cardinalidade de um conjunto.

2. Cada documento do conjunto é atribuído ao centróide mais próximo.

Medidas como GTM, que considera apenas a ocorrência de termos nos cálculos de similaridade, podem ser prejudicadas pelo uso de centróides. Isso pode ocorrer pois, como o centróide é a média de vários documentos, todos os termos desses documentos passam a ter frequências não nulas. Como o GTM só considera a presença ou ausência de um termo para efetuar o cálculo de documento similaridade, a presença de muitos termos diminui a capacidade de discriminação dos objetos para os centróides. A fim de analisar tal hipótese, no Apêndice C é apresentado um estudo sobre o uso de centróides e medóides nessa etapa do algoritmo LDC.

3.4.2 *Ensemble Lexical-Semantic Document Clustering* (ELSDC)

Em [Nourashrafeddin et al., 2014], os autores mostram que a técnica de BOC, utilizando conceitos da Wikipédia, obtém resultados piores que a técnica BOW, quando utilizada no algoritmo de agrupamento LDC. O algoritmo *Ensemble Lexical-Semantic Document Clustering* (ELSDC) foi proposto por eles com o objetivo de unir as soluções obtidas por BOC e BOW. As soluções são unidas por um método de consenso. Os autores concluem que a junção das duas soluções melhora significantemente os resultados obtidos. Os passos do ELSDC são mostrados no Algoritmo 3.

Os passos são semelhantes aos já descritos no algoritmo LDC. A principal diferença esta na estruturação da coleção também para o formato BOC, onde ocorre extração de conceitos relevantes da Wikipédia, cálculo dos K centróides e formação dos K grupos para ambos os modelos (BOW e BOC). Por fim, as soluções de agrupamento são unidas por um método de consenso.

O método de consenso pode ser usado para unir agrupamentos obtidos por qualquer técnica ou medida de similaridade. Dados como entrada dois resultados de agrupamentos, o método de consenso executa 3 passos principais:

Algoritmo 3: *Ensemble Lexical-Semantic Double Clustering* (ELSDC)

Entrada: Conjunto de documentos, K
Saída: K grupos de documentos

1: **Para** cada documento **faça:**
2: Extrai conceitos relevantes da Wikipédia
3: **fim Para**
4: Usa *fuzzy c-means* para gerar K grupos de termos
5: Remove termos não discriminativos dos grupos de termos
6: **Para** cada grupo de termos selecionados **faça:**
7: Extrai conceitos relevantes da Wikipédia
8: **fim Para**
9: **Para** cada grupo de termos selecionado **faça:**
10: Encontra os *documentos semente* baseados no modelo BOW
11: Calcula os *documentos centróides* léxicos
12: Encontra os *documentos semente* baseados no modelo BOC
13: Calcula os *documentos centróides* semânticos
14: **fim Para**
15: Agrupa a coleção de documentos usando os *centróides léxicos*
16: Agrupa a coleção de documentos usando os *centróides semânticos*
17: Reúne os agrupamentos encontrando os documentos com mesmo rótulo em ambos agrupamentos.
18: Utiliza esses documentos como um conjunto de treinamento para induzir um classificador
19: Classifica os documentos restantes usando o classificador

1. Encontra os documentos com o mesmo rótulo[2] em ambos os agrupamentos.

2. Usa esses documentos como um conjunto de treinamento para treinar um classificador.

3. Classifica os documentos restantes.

3.5 Validação de Agrupamento

Índices de validação são utilizados para aferir a qualidade de resultados de agrupamento. Se os grupos obtidos refletem a estrutura dos dados, os índices de validação devem indicar um bom resultado [Milligan & Cooper, 1985; Vendramin et al., 2010].

[2]Os rótulos são atribuídos aos grupos de termos, logo, os agrupamentos e centróides devem ser gerados a partir da mesma fonte, ou seja, do mesmo grupo de termos.

Os índices de validação de agrupamento podem ser de três tipos: externos, internos e relativos. Os índices externos fazem uso de um agrupamento modelo do conjunto de dados. Ele compara o agrupamento obtido com um modelo conhecido, que é comumente chamado de "real" ou "*gold standard*". Já os índices internos não utilizam conhecimento externo. Os grupos são avaliados utilizando como parâmetro apenas os próprios dados. Os índices relativos também são internos, mas podem ser utilizados para comparar quantitativamente os índices obtidos [Halkidi et al., 2001; Milligan & Cooper, 1985].

Esta seção faz uma revisão de alguns dos principais índices de validação externos e internos da literatura. Como esses índices podem ser utilizados para avaliar os resultados de agrupamento e comparar as partições obtidas, eles são relativos.

3.5.1 Índices externos

Como já descrito anteriormente, os índices externos comparam com um agrupamento modelo para avaliar um resultado de agrupamento obtido. Como um resultado de agrupamento pode ser rotulado de forma diferente ao modelo, os índices de validação externos fazem uso do relacionamento entre pares de objetos ou intersecção entre grupos, para avaliar a semelhança entre os agrupamentos comparados. Quando a análise é feita por pares, os índices aferem em ambos os agrupamentos se cada par de objetos pertence ao mesmo grupo ou se foram atribuídos a grupos diferentes. As contagens de pares em que os objetos se relacionam de forma igual, em ambos agrupamentos, são utilizados pelos índices para determinar o grau de similaridade entre agrupamentos. Dois índices externos de validação, um baseado em pares e outro em intersecção, são apresentados a seguir.

3.5.1.1 *Ajusted Rand Index* (ARI)

O *Ajusted Rand Index* (ARI) foi proposto em [Hubert & Arabie, 1985] como uma melhoria ao índice Rand original [Rand, 1971]. Uma forte crítica ao índice Rand é que o índice não retorna valores próximos de zero para partições geradas aleatoriamente. Já a versão ajustada do Rand resulta em valores no intervalo $[-1, 1]$, onde partições aleatórias retornam valores próximos de zero. Para partições idênticas, o ARI retorna o valor 1.

Ao obter um agrupamento π_r de uma base, compara-se esse agrupamento com um agrupamento π_g da mesma base, já conhecido. A comparação entre partições leva em consideração cada par de objetos x_i e x_j $\forall\ i \neq j$, onde existem 4 possíveis situações:

- **a (11)**: x_i e x_j pertencem ao mesmo grupo em π_r e em π_g.
- **b (01)**: x_i e x_j pertencem a grupos diferentes em π_r e pertencem a grupos iguais em π_g.
- **c (10)**: x_i e x_j pertencem a grupos iguais em π_r e pertencem a grupos diferentes em π_g.
- **d (00)**: x_i e x_j pertencem a grupos diferentes em π_r e pertencem a grupos diferentes em π_g.

Considere que as variáveis a, b, c e d contenham a contagem de pares de termos que satisfaçam a sua própria condição, e $T = a+b+c+d$. O cálculo do ARI é dado pela seguinte equação:

$$ARI(\pi_r, \pi_g) = \frac{a - \frac{(a+b)(a+c)}{T}}{\frac{(a+b)+(a+c)}{2} - \frac{(a+b)(a+c)}{T}} \tag{3.9}$$

Para verificar o relacionamento entre todos os pares de objetos, são feitas $N \times (N-1)/2$ operações, logo, a complexidade do ARI é $O(N^2)$, onde N é a quantidade de objetos.

3.5.1.2 *Normalized Mutual Information* (NMI)

Assim como o ARI, o *Normalized Mutual Information* (NMI) [Bishop, 2006] também utiliza um modelo real do conjunto de dados para medir a qualidade do agrupamento obtido. Porém, o NMI calcula a qualidade da partição considerando a intersecção entre os grupos, ao contrário do ARI, que considera o relacionamento entre cada par de objetos. O valor máximo do índice NMI é 1, quando o agrupamento é perfeitamente igual ao modelo real, e tem 0 como valor mínimo. O índice NMI pode ser calculado como se segue:

$$NMI = \frac{I(\pi_r, \pi_g)}{[H(\pi_r) + H(\pi_g)]/2} \tag{3.10}$$

$$I(\pi_r, \pi_g) = \sum_{k=1}^{K} \sum_{j=1}^{K} \frac{|G_k \cap G'_j|}{N} \log \frac{N|G_k \cap G'_j|}{|G_k||G'_j|} \tag{3.11}$$

$$H(\pi_r) = -\sum_{k=1}^{K} \frac{|G_k|}{N} \log(\frac{|G_k|}{N}) \tag{3.12}$$

$$H(\pi_g) = -\sum_{k=1}^{K} \frac{|G'_k|}{N} \log(\frac{|G'_k|}{N}) \tag{3.13}$$

onde $\pi_r = \{G_1, G_2, \cdots, G_K\}$ e $\pi_g = \{G'_1, G'_2, \cdots, G'_K\}$ representam grupos e classes, respectivamente. $|G_K \cap G'_j|$ é o número de instâncias em comum entre G_K e G'_j, e N é o número de documentos.

Com uso de uma matriz de confusão, que é gerada com custo $O(K^2N)$, o NMI é calculado com $6K^2 + 6K$ operações. Desta forma, a complexidade computacional do NMI é $O(K^2N)$. Como K é sempre bem menor que N, a complexidade computacional do NMI é inferior à complexidade computacional do ARI, que é $O(N^2)$.

3.5.2 Índices internos relativos

Como já foi descrito anteriormente, índices internos relativos são utilizados para comparar a qualidade relativa de partições quantitativamente, sem utilizar conhecimento externo. Apenas os dados do conjunto são utilizados. Nesta seção são apresentados o índice de Silhueta e algumas de suas variações. O índice de Silhueta foi escolhido com base em estudos feitos em [Vendramin et al., 2009, 2010], onde a Silhueta foi pontuada entre os melhores índices.

3.5.2.1 Silhueta

O critério de Silhueta [Rousseeuw, 1987] mede o quão compactos e separados os grupos de determinado agrupamento estão [Tan et al., 2005].

A Silhueta de um objeto pode ser calculada conforme a Equação 3.14, onde $a(x_j)$ é a dissimilaridade média do documento x_j aos objetos do grupo $G_A \mid x_j \in G_A$. Já $b(x_j)$ é a menor dissimilaridade média do documento x_j para os documentos de um outro grupo $G_B \mid G_A \neq G_B$.

$$s(x_j) = \frac{b(x_j) - a(x_j)}{max(a(x_j), b(x_j))} \tag{3.14}$$

Se o documento é atribuído sempre ao centróide mais próximo, o valor de $s(x_j)$ será sempre entre 0 e 1. Se o valor da Silhueta é menor ou próximo a 0, indica que ele não deveria pertencer a esse grupo. Para calcular a Silhueta de um grupo G_i, basta calcular a média das Silhuetas de cada elemento x_j pertencente a G_i, conforme mostra a Equação 3.15. Já a Silhueta de um agrupamento completo é calculada pela média de todos os documentos do agrupamento, conforme mostra a Equação 3.16.

$$f(G_i) = \frac{1}{|G_i|} \sum_{x_j \in G_i} s(x_j) \tag{3.15}$$

$$f(\pi) = \frac{1}{N} \sum_{j=1}^{N} s(x_j) \tag{3.16}$$

O índice de Silhueta é mais apropriado para agrupamentos que seguem distribuições gaussianas multidimensionais hiperesféricas ou levemente alongadas [Vendramin et al., 2010].

3.5.2.2 Silhueta Simplificada

A Silhueta Simplificada (SS) foi proposta em [Hruschka et al., 2004] a fim de diminuir o custo computacional para a silhueta original. Na SS é feita uma alteração nos cálculos de $a(x_j)$ e $b(x_j)$. Nesta versão, $a(x_j)$ é a dissimilaridade do objeto x_j para o centróide de seu grupo, enquanto $b(x_j)$ é a dissimilaridade do objeto x_j para o centróide do grupo vizinho mais próximo. Com essa alteração, a complexidade do cálculo da Silhueta muda de $O(N^2)$ para $O(N)$. Segundo os autores e experimentos realizados em [Vendramin et al., 2013], as alterações mantém a qualidade próxima do índice original.

3.5.2.3 Silhueta *Fuzzy*

O índice de Silhueta *fuzzy* foi proposto em [Campello & Hruschka, 2006], projetado para considerar graus de associação entre objetos de partições *fuzzy*. O índice de Silhueta *fuzzy* pode ser calculado da seguinte forma:

$$\text{Silhueta } \textit{Fuzzy} = \frac{\sum_{j=1}^{N} (U_{p(j),j} - U_{q(j),j})^{\beta} s(x_j)}{\sum_{j=1}^{N} (U_{p(j),j} - U_{q(j),j})^{\beta}} \tag{3.17}$$

onde $s(x_j)$ é a Silhueta do objeto x_j de acordo com a Equação[3] 3.14, $U_{p(j),j}$ e $U_{q(j),j}$ são o primeiro e segundo maior elemento da j-ésima coluna da matriz de agrupamento *fuzzy*, respectivamente, e $\beta \geq 0$ é um fuzzificador. Esse fuzzificador é por padrão $\beta = 1$.

[3] Ao se calcular a Silhueta, pode-se utilizar a versão clássica ou simplificada, conforme mostrado na Seção 3.5.2.2.

3.6 Considerações finais

O objetivo deste capítulo foi apresentar os principais passos a serem seguidos em uma tarefa de agrupamento, desde o pré-processamento até a avaliação dos resultados. Nos próximos capítulos serão apresentadas as propostas e experimentos conduzidos nesse trabalho. Nesses capítulos, os algoritmos e métodos apresentados até aqui serão utilizados e comparados.

4 SIMILARIDADE SEMÂNTICA PARA AGRUPAMENTO

Os Capítulos 2 e 3 fizeram uma breve revisão dos principais conceitos necessários para compreender a proposta e desenvolvimento desta dissertação. Como abordado no Capítulo 3, medidas de similaridade estão intimamente ligadas à qualidade das partições obtidas por algoritmos de agrupamento. Neste capítulo são apresentadas as principais contribuições dessa dissertação. Aqui é proposta uma nova medida de similaridade semântica baseada em *tri-grams*, do Google [Brants & Franz, 2006], destinada a tarefas de agrupamento de textos. Também são propostas combinações de agrupamentos textuais a partir de diferentes medidas de similaridade.

Este capítulo está dividido da seguinte forma: na Seção 4.1 é feito um resumo dos trabalhos da literatura que propõem e comparam medidas de similaridade para textos, onde são apresentadas diferenças desses trabalhos para os métodos propostos nesta dissertação; a Seção 4.2 apresenta a medida de similaridade semântica que é proposta neste trabalho; a Seção 4.3 apresenta as propostas de combinações de medidas de similaridade em algoritmos de agrupamento; as análises de complexidade assintóticas das medidas de similaridade e combinações de agrupamentos são abordadas na Seção 4.4; já a Seção 4.5 trata das considerações finais deste capítulo.

4.1 Trabalhos Relacionados

Conforme descrito no Capítulo 3, o processo de agrupamento de textos é uma tarefa complexa, que depende da qualidade da medida de similaridade adotada para extrair bons grupos [Feldman & Sanger, 2006]. O Capítulo 2 descreve diferentes

tipos de medidas de similaridade, que podem ser utilizadas para diferentes tipos de dados. A literatura dispõe de diversos trabalhos que comparam e propõem medidas de similaridade para documentos textuais. Em [Huang, 2008], a autora compara 5 medidas de similaridade utilizando o algoritmo K-médias clássico para agrupamento de textos. São comparadas as medidas euclidiana [Aldenderfer & Roger, 1984], cosseno [Salton & McGill, 1986], Jaccard [Jaccard, 1912], Pearson [Pearson, 1895] e *Kullback-Leibler divergence* [Kullback & Leibler, 1951]. A autora concluiu que, com exceção da euclidiana (que obteve o pior resultado), as medidas obtiveram eficácia equivalente. Nenhuma das medidas comparadas pela autora baseia-se em semântica, que é a proposta deste trabalho.

Em [Islam & Inkpen, 2008], os autores propõem o método *Semantic Text Similarity* (STS), capaz de mensurar a similaridade entre dois textos utilizando informações de sintaxe e semântica. Além de uma função de análise semântica, o método implementa uma função de similaridade entre *strings*, caso alguma palavra tenha sido escrita errada, e uma função de similaridade para ordem em que as palavras aparecem na frase. A similaridade semântica entre palavras é mensurada pelo método *Second Order Co-occurrence PMI* (SOC-PMI) [Islam & Inkpen, 2006]. Os autores obtêm melhores coeficientes de correlação que a medida proposta em [Li et al., 2006], que utiliza corpus *WordNet*[1] [Miller, 1995] e *Brown Corpuse*[2]. Além disso, a complexidade de tempo do método STS é inferior à medida não-supervisionada PMI-IR [Mihalcea et al., 2006] e a medida supervisionada proposta em [Corley & Mihalcea, 2005].

No trabalho [Islam et al., 2012], que propõe a medida semântica *Google tri-grams*, os autores comparam a medida de similaridade com diversas outras medidas semânticas, como SPD-STS [Ho et al., 2010] e STS [Islam & Inkpen, 2008]. Os autores mostram que a medida utilizando *tri-grams* obtém melhores resultados que todas as outras medidas, perdendo apenas para o melhor participante humano comparado nos experimentos. Assim como em [Ferreira et al., 2014] e [Rakib et al., 2015], que também propuseram medidas de similaridade semântica, em [Islam et al., 2012; Islam & Inkpen, 2008; Ho et al., 2010; Li et al., 2006; Mihalcea et al., 2006], os autores utilizam as medidas apenas em tarefas de comparação de pares de sentenças curtas. As medidas não foram utilizadas ou testadas em algoritmos de agrupamento, que é o principal objetivo deste trabalho.

[1] O *WordNet* é uma ontologia léxica do conhecimento comum de palavra em inglês, expressa em termos de conceitos chamados conjuntos de sinônimo (*synsets*), mantidos por especialistas da Universidade de Princeton, nos Estados Unidos.

[2] http://www.helsinki.fi/varieng/CoRD/corpora/BROWN/

O uso de medida semântica para agrupamento de texto é abordado em [Wei et al., 2015]. Nele, os autores propõem um método para agrupamento de documentos e utilizam uma versão modificada da medida de similaridade Wu-Palmer [Wu & Palmer, 1994], baseada em *WordNet*. A medida Wu-Palmer é baseada em relação semântica explícita, ou seja, ela assume que os links entre conceitos representam distâncias. Porém, no *WordNet* não há conexão entre todos os pares de conceitos, o que limita o uso da medida. Os autores incorporam à medida Wu-Palmer o método de *glosses of senses* [Banerjee & Pedersen, 2003], que computa a similaridade entre conceitos baseado no número de palavras compartilhadas (*overlaps*) em suas definições (*glosses*). Embora essa modificação melhore a medida Wu-Palmer, os autores concluem que o uso de *WordNet* pode limitar a qualidade da medida de similaridade e dos agrupamentos. Algumas palavras importantes não estão inclusas ou não são completamente representadas no *WordNet*, o que pode causar deficiência no cálculo da similaridade.

Um método para treinar uma medida de similaridade entre documentos baseada em julgamentos humanos é proposta em [Huang et al., 2012]. O método proposto usa técnicas de aprendizado de máquina e um conjunto de documentos com similaridades avaliadas por humanos, chamado HE50 [Lee & Welsh, 2005], para treinar e construir a medida. A medida é baseada no modelo BOC e usa conceitos da Wikipédia [Hu et al., 2008] e *WordNet*. Os resultados experimentais mostram que o uso da Wikipédia é mais eficaz que *WordNet* para esta aplicação. Os autores aplicam a medida para classificar e agrupar documentos de 4 coleções. Para a tarefa de agrupamento, a medida supera a similaridade cosseno nos algoritmos hierárquico e K-médias. O modelo BOC usando conceitos da Wikipédia também é implementado neste trabalho.

O algoritmo *Ensemble Lexical-Semantic Document Clustering* (ELSDC) é proposto em [Nourashrafeddin et al., 2014] como uma nova versão do algoritmo LDC [Nourashrafeddin et al., 2013]. O ELSDC tem como objetivo unir um agrupamento obtido pelo modelo BOW com um agrupamento obtido pelo modelo BOC, que usa conceitos da Wikipédia [Hu et al., 2008]. Com o objetivo de alcançar consenso, documentos atribuídos aos mesmos grupos em ambos os agrupamentos são utilizados como conjunto de treinamento para o classificador Naive Bayes [Manning et al., 2008]. Finalmente, o classificador induzido é usado para classificar os documentos restantes. Os autores mostram que o uso de BOW com BOC supera os resultados obtidos pelo algoritmo LDC. Neste trabalho, o LDC e ELSDC são descritos nas Seções 3.4.1 e 3.4.2, e implementados com e sem nossa medida de similaridade proposta.

4.2 *Frequency Google Tri-grams Measure* (FGTM)

Nesta seção, propomos o *Frequency Google Tri-grams Measure* (FGTM). O FGTM é uma medida de similaridade semântica baseada no *corpus n-grams* do Google [Brants & Franz, 2006]. A medida está dividida em duas etapas, onde na primeira é calculada a similaridade entre os termos do conjunto de documentos e, posteriormente, calcula-se a similaridade entre documentos. O cálculo de termo-similaridade é o mesmo da medida GTM, definido anteriormente pela Equação 2.3.

A ideia principal do FGTM é integrar a frequência TF-IDF dos termos que aparecem nos documentos como um novo parâmetro no cálculo de similaridade entre documentos. Além disso, propomos uma novo critério de seleção de similaridades entre termos, a fim de selecionar uma quantidade maior de pares de palavras para o cálculo de similaridade entre documentos. O método de similaridade entre documentos consiste nos 5 passos a seguir:

- **Passo 1:** Tem-se como entrada dois documentos pré-processados, com seus vetores TF-IDF, $P = \{p_1, p_2, ..., p_m\}$ e $R = \{r_1, r_2, ..., r_n\}$, onde m e n são os números de termos e $n \geq m$. Caso contrário, troque P e R. Sabendo que $freq(p_i)$ é a frequência TF-IDF do i-ésimo termo no documento P, a soma das frequências dos termos em cada documento é calculado da seguinte forma:

$$pTotal = \sum_{i=1}^{m} freq(p_i),$$

$$rTotal = \sum_{j=1}^{n} freq(r_j).$$

- **Passo 2:** Conta o número de termos que ocorrem em ambos os documentos (chama-se δ) e incrementa a menor frequência TF-IDF entre o mesmo termo em ambos documentos (chama-se $\delta Total$). Isto é, $\delta = \delta + 1$ e $\delta Total = \delta Total + min(freq(p_i), freq(r_j))$ para cada $p_i = r_j, \forall p \in P, \quad \forall r \in R$. Ao final da contagem, removemos os δ termos de P e R, então, $P = \{p_1, p_2, ..., p_{m-\delta}\}$ e $R = \{r_1, r_2, ..., r_{n-\delta}\}$. Se $m - \delta = 0$ pule para o Passo 5.

- **Passo 3:** Constrói uma matriz de similaridade semântica $(\alpha_{ij})_{(m-\delta)\times(n-\delta)}$, com o seguinte processo: $\alpha_{ij} \leftarrow Sim(p_i, r_j) \times min(freq(p_i), freq(r_j))$ (usando a Equação 2.3). Nós chamaremos essa matriz de M. Nesse passo, a menor

frequência TF-IDF entre dois termos analizados é multiplicada à similaridade semântica entre eles.

- **Passo 4:** Usando as notações μ para média e σ para desvio padrão, nós consideramos que cada uma das $(m-\delta)$ linhas da matriz M é um conjunto de $(n-\delta)$ elementos, então, calculamos μ e σ de cada linha e encontramos os elementos maiores ou iguais a $(\mu+\sigma)$ de cada linha. Se existem y_i elementos na linha i que satisfaça essa condição, este conjunto é chamado de A_i. Formalmente, A_i pode ser definido da seguinte forma:

$$A_i = \{\alpha_{ij} : \alpha_{ij} \in M_i, \quad \alpha_{ij} \geq (\mu(M_i) + \sigma(M_i))\} \tag{4.1}$$

  A média dos y_i elementos do conjunto A_i é $\mu(A_i)$. A soma de todos as $(m-\delta)$ linhas da matriz M é $\sum_{i=1}^{m-\delta} \mu(A_i)$.

- **Passo 5:** A similaridade entre os documentos P e R é dado pela média harmônica adicionando $\delta Total$ à soma obtida no Passo 4. Logo, $S(P,R) \in [0,1]$.

$$S(P,R) = \frac{(\delta Total + \sum_{i=1}^{m-\delta} \mu(A_i)) \times (pTotal + rTotal)}{2 \times pTotal \times rTotal} \tag{4.2}$$

A frequência TF-IDF no FGTM é usada para ponderar a similaridade ou dissimilaridade entre dois documentos. Comparando a frequência de um termo em um dado par de documentos, pode-se estimar o quão importante o termo é para a identificação do assunto do texto. Neste caso, se um termo aparece várias vezes no mesmo documento, considera-se que o termo é mais relevante do que um termo que aparece poucas ou apenas uma vez. Logo, a discrepância entre frequências pode caracterizar a dissimilaridade entre documentos. O mesmo critério é usado quando calculamos a similaridade semântica entre termos diferentes. No numerador da Equação 4.2, as menores frequências dos termos similares em ambos documentos são armazenadas. No denominador, a frequência total de todos os termos nos documentos é usada. Desta forma, uma vez que usamos a menor frequência entre dois termos no cálculo da similaridade, quanto maior a diferença entre as frequências, menor é a similaridade entre os documentos.

Outra modificação aplicada no FGTM em relação ao GTM original [Islam et al., 2012] é o uso de $\geq$ no lugar de $>$ na Equação 4.1. Quando elementos com valores iguais a $\mu+\sigma$ são considerados, previne-se que uma considerável parte da matriz de similaridade semântica seja ignorada. Isso acontece frequentemente

quando compara-se sentenças curtas ou quando os documentos possuem uma pequena quantidade de termos. Por exemplo, se a matriz M possui somente duas colunas, nenhuma similaridade entre termos será considerada no cálculo de documento similaridade do GTM. Isso porque $\mu(M_i) + \sigma(M_i) \geq \alpha_{ij} \quad \forall i \in (m - \delta), \forall j \in (n - \delta)$, desde que $n - \delta = 2$. Por exemplo, quando se compara as duas seguintes sentenças:

- $P =$*{He is professor}*
- $R =$*{She is teacher}*

o termo *"is"* é removido e a variável δ é incrementada, pois o termo ocorre em ambas as frases. As similaridades entre os termos restantes, obtidos pela Equação 2.3, são usados para construir a matriz M conforme mostra a Tabela 4.1.

Tabela 4.1: Matriz M das medidas GTM e FGTM para o exemplo de comparação das frases *"He is professor"* × *"She is teacher"*.

$M =$

	she	***teacher***
he	0.455	0.261
professor	0.262	0.478

O próximo passo é o cálculo da μ e σ de cada linha da tabela.

- **Linha 1:** $\mu = 0.358$, $\sigma = 0.097$; $(\mu + \sigma) = 0.455$
- **Linha 2:** $\mu = 0.370$, $\sigma = 0.108$; $(\mu + \sigma) = 0.478$

Neste caso, os valores dos elementos nunca serão maiores que $\mu + \sigma$, logo, nenhum termo similaridade será usado no cálculo de documento similaridade do GTM. Comparando o GTM com o FGTM, que considera valores iguais a $\mu + \sigma$, as similaridades entre as sentenças são:

- **GTM:** $Sim_{(P,R)} = 0.333$
- **FGTM:** $Sim_{(P,R)} = 0.644$

Embora as sentenças sejam semanticamente similares, a similaridade estimada pelo GTM é equivalente a 33%, ou seja, somente o termo *"is"* é considerado. O FGTM usa a similaridade entre diferentes termos, logo, a medida de similaridade se torna mais sensível nesses cenários. Apesar do exemplo, esta situação pode ocorrer com três ou maiores quantidades de termos, o que é muito comum em documentos.

4.3 Combinações de medidas de similaridade

Em [Schonhofen, 2006] e [Nourashrafeddin et al., 2014], os autores mostram que o uso de conceitos ou categorias da Wikipédia não é melhor que o uso da representação documento-termo clássica. Em [Huang et al., 2009], é proposta uma combinação linear da medida de similaridade cosseno em conjuntos baseados no modelo BOW e BOC. Uma proposta parecida é feita em [Hu et al., 2009], onde os termos de um conjunto de documentos são mapeados em conceitos e categorias da Wikipédia. Com os termos mapeados, a similaridade cosseno é utilizada para agrupar e combinar os resultados de documento-conceitos e categorias. Porém, nenhuma melhoria significante é encontrada nos algoritmos particionais para essa técnica e os melhores resultados são obtidos com algoritmo hierárquico. Por fim, em [Nourashrafeddin et al., 2014] os autores propõem o algoritmo ELSDC, projetado para unir agrupamentos obtidos por BOW e BOC, onde o segundo utiliza conceitos da Wikipédia. Os resultados são unidos por um método de consenso, que foi apresentado no final da Seção 3.4.2. Os resultados apresentados em [Nourashrafeddin et al., 2014] indicam que o ELSDC supera o uso dos modelos isoladamente.

Com base nos resultados apresentados em [Nourashrafeddin et al., 2014], assumimos como hipótese que: unir resultados de agrupamentos gerados por medidas de similaridade distintas supera o uso das medidas isoladamente. Para unir os resultados de agrupamento, foi utilizado o processo ilustrado pelo fluxograma da Figura 4.1. Cada etapa do fluxograma e os métodos utilizados são descritos a seguir.

Figura 4.1: Fluxograma do processo de combinação de agrupamentos gerados a partir de duas medidas ou técnicas de similaridade.

1. **Pré-processamento:** O pré-processamento é a etapa de adequação do conjunto de documentos para o formato a ser processado pelo algoritmo de agrupamento. Quando são utilizadas as medidas euclidiana, cosseno, GTM ou FGTM, é necessário o pré-processamento pelo modelo BOW. Já no algoritmo ELSDC, que também calcula a similaridade entre documentos baseada em conceitos, o pré-processamento BOC também deve ser realizado. A saída dessa etapa é a representação do conjunto de documentos no modelo escolhido.

2. **Algoritmo de Agrupamento:** Nessa etapa, o algoritmo de agrupamento recebe o conjunto pré-processado e agrupa os documentos duas vezes, com uso de duas medidas de similaridade. Os algoritmos de agrupamento utilizados foram o LDC e ELSDC [Nourashrafeddin et al., 2013, 2014], apresentados nas Seções 3.4.1 e 3.4.2, respectivamente. Esses algoritmos foram escolhidos por serem o estado da arte em tarefas de agrupamento de textos. A medida FGTM, proposta neste trabalho, foi utilizada e combinada com outras 3 medidas de similaridade: euclidiana, cosseno e BOC, que utiliza conceitos da Wikipédia. As medidas clássicas, euclidiana e cosseno, foram escolhidas pelo custo computacional linear e pelos bons resultados apresentados nos experimentos propostos em [Nourashrafeddin et al., 2014]. A medida semântica baseada em conceitos da Wikipédia, que utiliza pré-processamento BOC, foi escolhida por ser o estado da arte no algoritmo ELSDC. Desta forma, os algoritmos de agrupamento foram executados com as seguintes configurações:

   - **LDC:**
     - Euclidiana + FGTM
     - Cosseno + FGTM
   - **ELSDC:**
     - Wikipédia + FGTM

   Uma vez que a combinação "Wikipédia + FGTM" faz uso de dois *corpus*, a complexidade computacional é mais elevada que as demais combinações. Porém, partimos da hipótese de que, utilizar duas fontes distintas de *corpus* pode enriquecer a percepção de semelhança dos métodos utilizados. Por isso, essa combinação foi considerada.

3. **Método de Consenso:** Tem a função unir os dois resultados de agrupamento obtidos na etapa anterior. O agrupamento resultante é o agrupamento final do processo. Nesta etapa, foi utilizado o método de consenso do algoritmo ELSDC, apresentado no final da Seção 3.4.2. Embora o algoritmo ELSDC tenha sido desenvolvido para trabalhar apenas com conceitos da Wikipédia, o método de consenso pode ser utilizado em outros algoritmos, como o LDC, para unir agrupamentos obtidos por qualquer medida de similaridade. Ele foi escolhido por ter complexidade linear e por ser o único a superar os resultados do modelo BOW, conforme descrito no início desta seção.

Na próxima seção, é feita a análise de complexidade de todos os passos dos algoritmos, medidas e combinações aqui apresentadas. No Capítulo 5, essas combinações serão comparadas com o uso das medidas euclidiana, cosseno, GTM e FGTM de forma individual. A configuração original do algoritmo ELSDC (Euclidiana + Wikipédia) [Nourashrafeddin et al., 2014] também é utilizada nos experimentos comparativos.

4.4 Análise de complexidade

Tarefas de agrupamento de documentos geralmente são aplicadas a conjuntos grandes, que são difíceis de ser analisados por um humano. Por lidar com entradas grandes, é importante que os algoritmos utilizados tenham a menor complexidade computacional possível. Nesta seção são feitas as análises de complexidade computacional dos algoritmos utilizados neste trabalho. Além dos algoritmos de agrupamento, os custos computacionais para pré-processamentos também são apresentados.

4.4.1 Análise de complexidade do pré-processamento

Neste trabalho, dois métodos de pré-processamento foram considerados: BOW e BOC. No método BOW, o pré-processamento executa $\sum_i^N T_i$ operações para separar e contar os termos, onde N é o número de documentos e T_i é o número de termos, incluindo repetições, contidos no i-ésimo documento. Para remover as *stopwords*, $S \cdot \log T$ operações são executadas, onde S é o tamanho da lista[3] de *stopwords* e T é o número total de termos, sem repetição, no conjunto de documentos. A complexidade computacional da normalização TF-IDF é $O(NT)$. A seleção de atributos VAR-TFIDF [Kogan et al., 2003] também possui complexidade $O(NT)$. Logo, a complexidade total do pré-processamento BOW é $O((T_1+T_2+\cdots+T_N)+(S\cdot\log T)+(NT)+(NT))$, que é equivalente a $O(N\bar{T}+NT)$, onde $\bar{T}$ é o número médio de termos no conjunto de documentos. Como a busca e remoção de *stopwords* são executadas em tempo logarítmico e o número N de documentos em uma coleção é normalmente maior que a lista com S *stopwords*, o custo $S \cdot \log T$ é omitido na notação O. Uma vez que os valores de $\bar{T}$ e T dependem das características do conjunto de documentos, não é possível afirmar que um é maior que o outro. Em um conjunto de documentos curtos, mas com alta diversidade de assuntos e vocabulário, o número de termos T é normalmente muito maior que

[3]Como forma de ilustração, a lista de *stopwords* usada neste trabalho contém 318 palavras.

o número médio de termos nos documentos, $\bar{T}$. Por outro lado, em um conjunto com documentos grandes, mas com pequena diversidade de assuntos e vocabulário, o número médio de termos nos documentos, $\bar{T}$, pode ser maior que o número de termos, T. O número de termos após a seleção VAR-TFIDF é chamado de T'.

No método BOC, o *Wikipedia Miner toolkit* [Milne & Witten, 2013] é usado para identificar conceitos nos documentos. Dado como entrada o i-ésimo documento com tamanho T_i, o *Wikipedia Miner toolkit* constrói uma lista com todos *n-grams* possíveis no documento. Um *n-gram* é uma sequência de palavras encontradas no texto que não podem estar separadas por pontuação ou quebra de linha. Por exemplo, na sentença *"I love cookies"* são obtidos os seguintes *n-grams*:

- *"I love cookies"*
- *"I love"*
- *"love cookies"*
- *"I"*
- *"love"*
- *"cookies"*

Como não existe um valor máximo para n, no pior caso, em que o documento não tenha pontuação ou quebras de linha, a lista de *n-grams* conterá, aproximadamente, a seguinte quantidade de entradas:

$$\sum_{j=1}^{T_i} j = \frac{T_i(T_i+1)}{2} \approx T_i^2$$

Após construir a lista de *n-grams*, o *Wikipedia Miner toolkit* pesquisa se existe conceitos relacionados a cada um dos *n-grams* encontrados. A ferramenta utiliza a estrutura de *hashmap* para consultar os conceitos. Podemos assumir que a complexidade para consultas em *hashmap* aproxima-se de $O(1)$ [Lafore, 2002]. Em seguida, a ferramenta calcula o relacionamento entre todos os conceitos encontrados. Se C_i é a quantidade de conceitos encontrados no documento i, a complexidade para calcular o relacionamento entre todos os conceitos no i-ésimo documento é $O(C_i^2)$. Uma vez calculado os relacionamentos, a ferramenta percorre e remove os conceitos com baixo peso. O limiar é definido pelo usuário. A remoção dos conceitos tem complexidade $O(C_i)$.

Após utilizar o *Wikipedia Miner toolkit* para todos os N documentos, a tabela de documentos-conceito é construída e normalizada pelo método TF-IDF. Esta operação tem complexidade $O(NC')$, onde C' é a quantidade de conceitos relevantes em todo o conjunto de documentos. Desta forma, a complexidade total do pré-processamento pelo método BOC é $O((T_1^2+T_2^2+\cdots+T_N^2)+(C_1^2+C_2^2+\cdots+C_N^2)+(C_1+C_2+\cdots+C_N)+(NC'))$. Como o tamanho de cada documento pode variar, não é possível simplificar a complexidade. Se assumirmos que todos os documentos tem tamanhos iguais e a mesma quantidade de conceitos, a complexidade pode ser representada como $O((N\bar{T}^2)+(N\bar{C}^2)+(N\bar{C})+(NC'))$, onde $\bar{T}$ e $\bar{C}$ são as médias das quantidades de palavras e conceitos, respectivamente, na coleção de documento. Como a quantidade de conceitos em um documento é menor que a quantidade de palavras no mesmo documento, então $N\bar{T}^2 \geq N\bar{C}^2 \geq N\bar{C}$. Dessa forma, $(N\bar{C}^2)$ e $(N\bar{C})$ são omitidos na notação O. Assim como acontece no modelo BOW, não é possível afirmar que $\bar{T}^2 \geq C'$, portanto, a complexidade do pré-processamento BOC é equivalente à $O(N\bar{T}^2 + NC')$.

4.4.2 Análise de complexidade das medidas de similaridade

O cálculo de similaridade entre dois documentos utilizando a distância euclidiana é feito com $2 \cdot T'$ operações, onde T' é o número total de termos após o pré-processamento. Logo, a complexidade é $O(T')$. Para o cálculo da similaridade utilizando o cosseno, são feitas $3 \cdot T'$ operações, portanto, a complexidade também é $O(T')$.

As medidas GTM e FGTM possuem ambas o mesmo custo computacional. Primeiro é montada uma tabela com as similaridades entre todos os termos. Para o cálculo da similaridade entre os termos, são feitas consultas dos *uni-grams* e *tri-grams* de cada par de palavras no corpus do Google. O corpus está estruturando em *hashmap*, logo, podemos assumir que a complexidade para consultas aproxima-se de $O(1)$ [Lafore, 2002]. A complexidade para montar a tabela termo-similaridade é $O(T'^2)$. Para o cálculo da similaridade entre documentos, no pior caso, em que todos os termos entre os dois documentos são diferentes, são realizadas $4 \cdot \left(\frac{T'}{2}\right)^2 + 3 \cdot \frac{T'}{2}$ operações, logo, a complexidade para cálculo entre dois documentos é $O(T'^2)$.

4.4.3 Análise de complexidade dos algoritmos de agrupamento

Na fase de agrupamento de documentos deste trabalho, foram utilizados os algoritmos LDC e ELSDC. Também foram feitos experimentos utilizado o método de consenso no algoritmo LDC, para unir agrupamentos obtidos por duas medidas de similaridade.

No algoritmo LDC, a fase de agrupamento de termos e seleção de *termos chave* possui complexidade $O(NT'K^2I)$ [Nourashrafeddin, 2014], onde K é a quantidade de grupos e I é o número de iterações[4] do algoritmo *fuzzy c-means*. A fase de encontrar *documentos semente* possui complexidade $O(NT''K + NKI)$ [Nourashrafeddin, 2014], onde T'' é a quantidade de *termos chave*. A fase de agrupamento de documentos realiza dois passos principais. Primeiro são calculados os documentos centróides, o que possui complexidade $O(NT'K)$. Segundo, é calculada a distância de todos os N documentos à todos os K centróides. Utilizando a distância euclidiana ou similaridade cosseno, esse passo possui complexidade $O(NT'K)$, logo, a complexidade dessa fase é $O(NT'K)$ [Nourashrafeddin, 2014]. Utilizando a medida GTM ou FGTM, a complexidade dessa fase é $O(NT'^2K)$.

No algoritmo ELSDC, a fase de agrupamento de termos e seleção de *termos chave* permanece a mesma e possui complexidade $O(NT'K^2I)$. Em seguida, para cada *termo chave* extraído, o algoritmo consulta e extrai os conceitos relevantes da Wikipédia. Esse passo possui complexidade $O(T'')$. Feito isso, os *documentos semente* dos modelos BOW e BOC são encontrados. Essa etapa, para o modelo BOW, possui complexidade $O(NT''K + NKI)$ e para o modelo BOC possui complexidade $O(NC''K + NKI)$, onde C'' é a quantidade de conceitos extraídos a partir dos *termos chave*. Na fase seguinte, os centróides de ambos os modelos, BOW e BOC, são calculados. As distâncias entre todos N documentos para os K centróides BOW e BOC são também calculados. Para o modelo BOW, a complexidade é $O(NT'K)$ com distância euclidiana ou similaridade cosseno e $O(NT'^2K)$ para as medidas GTM ou FGTM. Considerando o modelo BOC, a complexidade é $O(NC'K)$, onde C' é o número total de conceitos no conjunto de documentos.

O método de consenso é implementado como o último passo do ELSDC[5]. Primeiramente, o método de consenso encontra os documentos com o mesmo rótulo

[4] Assim como em [Nourashrafeddin, 2014], o número de interações do *fuzzy c-means* é definido como 50 neste trabalho.

[5] Embora o método de consenso não seja parte do algoritmo LDC original, ele pode ser implementado como um passo adicional. Conforme descrito na Seção 4.4.4, essa implementação é proposta neste trabalho.

em ambos agrupamentos. Esta tarefa tem complexidade $O(N)$. Em seguida, os documentos com mesmos rótulos são usados para treinar o classificador Naive Bayes, que então, classifica os documentos. O treinamento e classificação do Naive Bayes têm complexidade $O(NT')$ [Manning et al., 2008]. Logo, a complexidade do método de consenso é $O(NT' + N)$, que é equivalente à $O(NT')$.

As complexidades assintóticas de cada fase e algoritmo são apresentados na Tabela 4.2.

Tabela 4.2: Análise de complexidade de cada algoritmo e notações.

Algoritmo	**Tarefa**	**Complexidade de Tempo**
Pré-processamento	BOW	$O(N\bar{T} + NT)$
	BOC	$O(N\bar{T}^2 + NC')$
LDC	Agrupamento de termos	$O(NT'K^2I)$
	Documentos semente	$O(NT''K + NKI)$
	Agrupamento por distância	$O(NT'K)$
	Agrupamento por semântica	$O(NT'^2K)$
ELSDC	Agrupamento de termos	$O(NT'K^2I)$
	Conceitos relevantes	$O(T'')$
	Documentos semente BOW	$O(NT''K + NKI)$
	Documentos semente BOC	$O(NC''K + NKI)$
	Agrupamento por distância	$O(NT'K + NC'K)$
	Agrupamento por semântica	$O(NT'^2K + NC'K)$
	Método de consenso	$O(NT')$

Notações:

C' → Número total de conceitos
C'' → Número de conceitos selecionados
I → Número de iterações c-means
K → Número de grupos
N → Número de documentos
T → Número total de termos
T' → Número de termos selecionados
T'' → Número de termos chave
$\bar{T}$ → Número médio de termos no conjunto

4.4.4 Análise de complexidade da combinação de algoritmos

Neste trabalho, oito variantes dos algoritmos LDC e ELSDC são comparadas de duas maneiras: assintoticamente, com objetivo de avaliar os custos computacionais, e experimentalmente (na Seção 5.1), para avaliar a qualidade dos resultados. Considerando as complexidades mostradas na Tabela 4.2, os custos computacionais assintóticos de cada uma dessas variantes são calculados. Os nomes de cada uma das variantes são destacados em negrito e enumerados a seguir:

1. **Euclidiana:** esta variante usa pré-processamento BOW e agrupamento por distância euclidiana no algoritmo LDC. Complexidade: $O(N\bar{T} + NT +$

$NT'K^2I + NT''K + NKI + NT'K) = O(N\bar{T} + NT + NT'K^2I)$, porque $T' > T''$, logo, $NT'K^2I > NT'K$, $NT''K$, NKI e NT'.

2. **Cosseno:** esta variante usa pré-processamento BOW e agrupamento por distancia (similaridade cosseno) no algoritmo LDC. Possui a mesma complexidade da variante "Euclidiana", logo: $O(N\bar{T} + NT + NT'K^2I)$.

3. **GTM:** esta variante usa pré-processamento BOW e agrupamento por semântica (GTM) no algoritmo LDC. Complexidade: $O(N\bar{T} + NT + NT'K^2I + NT''K + NKI + NT'^2K) = O(N\bar{T} + NT + NT'^2K)$, porque $T' > T''$ e K, logo, $NT'^2K > NT'K^2I$, $NT''K$, NKI e NT'.

4. **FGTM:** esta variante usa pré-processamento BOW e agrupamento por semântica (FGTM) no algoritmo LDC. Possui a mesma complexidade da variante "GTM", logo: $O(N\bar{T} + NT + NT'^2K)$.

5. **Euc+Wik:** esta variante usa pré-processamento BOW e BOC. O agrupamento é feito pela distância euclidiana no algoritmo ELSDC. Complexidade: $O(N\bar{T} + NT + N\bar{T}^2 + NC' + NT'K^2I + T'' + NT''K + 2NKI + NC''K + NT'K + NC'K + NT') = O(N\bar{T}^2 + NT + NT'K^2I)$, porque normalmente $T' > T''$, C', C'' e I.

6. **Euc+FGTM:** esta variante usa pré-processamento BOW, agrupamento por distância euclidiana e semântica (FGTM) no algoritmo LDC. Os resultados são combinados pelo método de consenso. Complexidade: $O(N\bar{T} + NT + NT'K^2I + NT''K + NKI + NT'K + NT'^2K + NT') = O(N\bar{T} + NT + NT'^2K)$, pois $T' > K$.

7. **Cos+FGTM:** esta variante usa pré-processamento BOW, agrupamento por distância (similaridade cosseno) e semântica (FGTM) no algoritmo LDC. Os resultados são combinados pelo método de consenso. A complexidade é a mesma de "Euc+FGTM", logo: $O(N\bar{T} + NT + NT'^2K)$.

8. **Wik+FGTM:** esta variante usa pré-processamento BOW e BOC. O agrupamento é feito por semântica (FGTM) no algoritmo ELSDC. Complexidade: $O(N\bar{T} + NT + N\bar{T}^2 + NC' + NT'K^2I + T'' + NT''K + 2NKI + NC''K + NT'^2K + NC'K + NT') = O(N\bar{T}^2 + NT + NT'^2K)$.

As configurações "Euclidiana" e "Cosseno" possuem o menor custo computacional dentre as configurações. Já a "GTM" e "FGTM" possuem um maior custo

apenas na parte de agrupamento de documentos. Enquanto "Euclidiana" e "Cosseno" executam a fase de agrupamento de documentos com complexidade $NT'K$, "GTM" e "FGTM" possuem complexidade NT'^2K.

Quando comparamos "GTM" e "FGTM" com a "Euc+Wik", percebemos que "Euc+Wik" executa $N\bar{T}^2 + NC' + T'' + NKI + NC''K + NC'K + NT'$ operações que "GTM" e "FGTM" não executam. Porém, enquanto a fase de agrupamento de documentos de "GTM" e "FGTM" utilizam semântica, que é executada com complexidade NT'^2K, em "Euc+Wik" o agrupamento é feito por distância, com complexidade $NT'K$. Neste caso, a comparação de complexidade entre as configurações depende das características do conjunto a ser agrupado. Quando $T' >> \bar{T}$, "GTM" e "FGTM" têm um maior custo computacional que "Euc+Wik". Se $\bar{T} > T'$ ou T' não é muito maior que $\bar{T}$, o custo computacional de "Euc+Wik" é maior que "GTM" e "FGTM".

O mesmo acontece quando comparamos "Euc+Wik" com "Euc+FGTM" ou "Cos+FGTM". A variante "Euc+Wik" executa $N\bar{T}^2 + NC' + T'' + NKI + NC''K + NC'K$ operações que não são executadas em "Euc+FGTM" e "Cos+FGTM". Porém, "Euc+FGTM" e "Cos+FGTM" utilizam o agrupamento por semântica, com custo NT'^2K. Desta forma, "GTM", "FGTM" e "Euc+Wik" possuem menor custo computacional se $T' >> \bar{T}$. Já as configurações "Euc+FGTM" e "Cos+FGTM" executam $NT'K + NT'$ mais operações que "GTM" e "FGTM".

A "Wik+FGTM" é a que possui maior custo computacional dentre as variantes. Quando comparada com a outra variante baseada em ELSDC, "Euc+Wiki", "Wik+FGTM" possui um custo de NT'^2K para agrupar os documentos por semântica, enquanto "Euc+Wik" possui um custo de $NT'K$ usando distância euclidiana. As demais operações em ambas configurações são iguais. Já quando comparada com "Euc+FGTM" e "Cos+FGTM", a variante "Wik+FGTM" executa $N\bar{T}^2 + NC' + T'' + NKI + NC''K + NC'K$ mais operações.

4.5 Considerações finais

Neste capítulo foram apresentadas as principais propostas desta dissertação. Foi proposta uma nova medida de similaridade semântica, FGTM e diversas combinações de agrupamentos com uso de diferentes medidas de similaridade. Foi feita uma análise de complexidade assintótica de todos os algoritmos, medidas e combinações propostas neste capítulo. O próximo capítulo apresentará os experimentos conduzidos para testar empiricamente os métodos aqui propostos.

5 EXPERIMENTOS

Este capítulo trata dos experimentos conduzidos para avaliação empírica da medida de similaridade e combinações de medidas propostas neste trabalho. O capítulo está dividido em três partes: na Seção 5.1, são apresentados os experimentos e resultados para a tarefa de agrupamento de documentos. Nela são descritos os conjuntos de documentos e pré-processamentos utilizados, além dos resultados experimentais e análises estatísticas; na Seção 5.2, é abordado o problema de comparação de sentenças curtas, onde os resultados são apresentados, comparados e discutidos; por fim, a Seção 5.3 trata das considerações finais deste capítulo.

5.1 Agrupamento

O problema de agrupamento, conforme já descrito no Capítulo 3, é um dos mais complexos da área de Mineração de Dados. Isso porque ele deve encontrar grupos de objetos semelhantes de forma não supervisionada, ou seja, sem o uso de conhecimento externo [Rezende et al., 2011]. A medida FGTM, proposta no Capítulo 4, foi implementada e testada em algoritmos de agrupamento. Esta seção descreve os experimentos de agrupamento, onde são abordados os conjuntos de dados utilizados, passos de pré-processamento, resultados e análises experimentais.

5.1.1 Conjunto de dados

Em nossos experimentos, utilizamos 18 conjuntos de documentos, cujas características são resumidas na Tabela 5.1. Foram utilizados todos os conjuntos de documentos disponibilizados e citados na literatura dos trabalhos relacionados abordados na Seção 4.1. Também foram criados 4 novos conjuntos de documentos a

partir de artigos disponibilizados na web. A Tabela 5.1 contém o nome, quantidade de grupos (K), número de documentos (N), número total de termos (T) e número de termos restantes após a seleção de atributos VAR-TFIDF (T') para cada conjunto de dados.

Tabela 5.1: Descrição dos conjuntos de documentos utilizados neste trabalho

Conjunto	***K***	***N***	***T***	***T'***
20ng-subset	20	7.532	91.120	18.157
20ng-whole	20	18.845	172.035	29.116
Articles-1442-5	5	253	31.354	4.564
cbr-ilp-ir-son	4	675	26.975	5.716
cbr-ilp-ir-son-int	5	681	27.226	5.740
Classic4	4	7.095	27.502	5.548
News-10	5	1.000	28.929	6.264
News-multi7	7	6.633	62.728	12.931
News-multi10	10	9.587	81.111	16.376
News-rel3	3	2.625	38.949	8.361
News-sim3	3	2.946	79.383	13.538
Pubmed2000sel	4	2.000	14.459	3.727
Pubmed2000non	4	2.000	15.836	4.191
Pubmed4000	4	4.000	21.257	5.392
Reauters8-whole	8	6.688	24.284	5.596
Scopus2800	7	2.800	34.350	8.292
SMS	2	5.574	7.801	1.799
WebKb	4	4.199	77.108	9.323

1. **20Newsgroups**[1]**:** Esta coleção contém aproximadamente 20.000 documentos divididos em 20 diferentes classes. Em [Nourashrafeddin, 2014], o autor cria diversos subconjuntos a partir dessa coleção. Neste trabalho foram usados os seguintes subconjuntos:

   - **20ng-subset** inclui todas as classes da coleção, porém apenas 40% do número total de documentos.
   - **20ng-whole** inclui todos documentos de todas as classes.
   - **News-multi7** inclui todos documentos de 7 diferentes classes, que são: *alt.atheism, comp.sys.mac.hardware, misc.forsale, rec.sport.hockey, sci.crypt, alt.politics.guns,* e *soc.religion.Christian.*

[1]Disponível em http://qwone.com/~jason/20Newsgroups/ acessado em 23 de agosto de 2016.

- **News-multi10** inclui todos documentos de 10 diferentes classes, que são: *alt.atheism, comp.sys.mac.hardware, misc.forsale, rec.autos, rec.sport.hockey, sci.crypt, sci.med, sci.eletronics, sci.space,* e *talk.politics.guns.*
- **News-rel3** inclui todos os documentos de 3 classes relacionadas, que são: *talks.politics.misc, talks.politics.guns* e *talk.politics.mideast.*
- **News-sim3** inclui todos documentos de 3 classes similares, que são: *comp.graphics, comp.os.ms-windows.misc* e *comp.windows.x*

2. **Articles-1442-5:** Esta coleção contém 253 artigos obtidos a partir de 5 periódicos internacionais diferentes, eles são: *American Political Science Review, DNA Research, Monthly Weather Review, British Food Journal* e *Transactions on Mobile Computing* [Naldi et al., 2011].

3. **cbr-ilp-ir-son**[2]**:** Esta coleção contém 675 documentos com 4 diferentes classes, que são: *case based reasoning, inductive logic programming, information retrieval* e *sonification papers* [Paulovich et al., 2008].

4. **cbr-ilp-ir-son-int**[2]**:** Esta coleção contém 681 documentos de 5 diferentes classes. Além das classes descritas no conjunto anterior, esta coleção inclui alguns documentos sobre *data visualization* produzidos na Universidade de São Paulo (USP) [Paulovich et al., 2008].

5. **Classic4:** Esta coleção foi criada para o repositório de dados *SMART*[3] e é composta por 7.095 artigos separados em 4 classes: *medical, information retrieval, aerodynamics* e *computing algorithms* [Nourashrafeddin et al., 2014].

6. **News-10**[2]**:** Esta coleção contém 1.000 documentos de notícias separados em 5 classes [Paulovich et al., 2008].

7. **PUBMED:** A coleção PUBMED[4] foi criada a partir de vários artigos médicos. Neste trabalho, nós coletamos artigos de 4 diferentes classes, que são: *Alzheimer, cancer, diabetes* e *HIV*. Esses artigos foram usados para criar os seguintes subconjuntos de documentos:

[2]Disponível em http://infoserver.lcad.icmc.usp.br/infovis2/DataSets acessado em 23 de agosto de 2016.
[3]Disponível em ftp://ftp.cs.cornell.edu/pub/smart/ acessado em 23 de agosto de 2016.
[4]http://www.ncbi.nlm.nih.gov/pubmed

- **PUBMED4000** contém 4.000 documentos, onde os primeiros 1.000 artigos de cada assunto (classe) foram coletados. Não existe nenhuma restrição quanto a existência de palavras específicas nos artigos.
- **PUBMED2000non** contém 2.000 documentos, onde os primeiros 500 artigos de cada assunto foram coletados. Também não foram impostas restrições para os artigos selecionados para esse subconjunto.
- **PUBMED2000sel** contém 2.000 documentos, onde são 500 de cada classe. Nesse conjunto, todos os artigos da classe "*Alzheimer*" possuem as palavras "*alzheimer*" e "*dementia*". Todos os artigos da classe "*Cancer*" possuem as palavras "*cancer*" e "*tumor*". Os artigos da classe "*Diabetes*" possuem as palavras "*diabetes*" e "*insulin*". Por fim, todos os artigos da classe "*HIV*" possuem as palavras "*HIV*" e "*AIDS*".

8. **Reuters8-whole:** É um subconjunto da coleção Reuters-21578[5]. Este subconjunto contém todos documentos de 8 diferentes classes da coleção original, que são: *acq, crude, earn, grain, interest, money-fx, ship* e *trade* [Nourashrafeddin, 2014].

9. **Scopus2800[6]:** Esta coleção foi criada usando resumos de artigos extraídos da base de dados Scopus[7]. Esta coleção contém 2.800 documentos, onde cada documento é composto pelo título e resumo de um artigo. Os artigos são separados em 7 classes (400 artigos em cada), que são: *tectonic plates, neural network, photosyntesis, concrete, proton, hyperactivity* e *investiment.*

10. **SMS[8]:** Esta coleção contém 5.574 mensagens de texto rotuladas como *spam* ou *non-spam* [Cormack et al., 2007].

11. **WebKb[9]:** Esta coleção contém 7 diferentes classes de paginas da *Web* coletadas a partir de 4 departamentos de ciência da computação em Cornell, Texas, Washington e Winsconsin. Nesta versão, somente 4 classes são usadas, que são: *student, faculty, course* e *project* [Nourashrafeddin, 2014].

[5]Disponível em http://www.daviddlewis.com/resources/testcollections/reuters21578/ acessado em 23 de agosto de 2016.

[6]Conjunto cedido gentilmente pelo colega Paulo Gustavo Lopes Cândido

[7]http://www.scopus.com/

[8]Disponível em http://www.dt.fee.unicamp.br/~tiago/smsspamcollection/ acessado em 23 de agosto de 2016.

[9]Disponível em http://www.cs.cmu.edu/afs/cs.cmu.edu/project/theo-20/www/data/ acessado em 23 de agosto de 2016.

5.1.2 Pré-processamento

Para o modelo BOW, os seguintes passos de pré-processamentos foram aplicados sobre todos os conjuntos de documentos:

- Remove todas as *stopwords*.
- Cria a matriz de documento-termo. A significância dos termos nos documentos são mensurados usando o método TF-IDF [Salton & Buckley, 1988].
- Aplica a norma L2 [Horn & Johnson, 2012] sobre todos documentos, com objetivo de normalizar o comprimento dos vetores de documentos para 1.
- Aplica a seleção de atributos VAR-TFIDF com corte descrito na Seção 5.1.2.1.
- Aplica a norma L2 sobre todos documentos uma segunda vez.

Para o modelo BOC, os seguintes passos de pré-processamento foram aplicados sobre todos os conjuntos de documentos:

- Usa o *Wikipedia Miner Toolkit* [Milne & Witten, 2013] para extrair conceitos de todos os documentos.
- Cria uma matriz de documento-conceitos. A significância de um conceito nos documentos são medidos usando o método TF-IDF.
- Aplica a norma L2 sobre todos os documentos.

5.1.2.1 Corte VAR-TFIDF

Em [Nourashrafeddin et al., 2013], os autores mostram que o uso do método de ranqueamento VAR-TFIDF, apresentado na Seção 3.2.2, para agrupamento de textos supera outros métodos, tais como *Mean-TFIDF*, *Entropy Rank* e *Term Contribution*. Por essa razão, o método VAR-TFIDF é usado neste trabalho.

Os termos discriminantes de um conjunto de documentos são os que possuem maiores pontuações de VAR-TFIDF. Para definir o *threshold* de seleção, foram conduzidos experimentos que são apresentados no Apêndice A. O processo de seleção de atributos utilizado neste trabalho segue 3 passos principais, conforme definidos em [Nourashrafeddin, 2014]:

- Atribuir a pontuação Var-TFIDF para cada um dos T termos do conjunto usando a Equação 3.3, definida anteriormente.

- A média das pontuações é calculada utilizando a seguinte fórmula:

$$avgVar = \frac{1}{T}\sum_{j=1}^{T} \text{VAR-TFIDF}_j$$

- Todos os termos com pontuação maior que *avgVar* são mantidos e os termos restantes são descartados.

5.1.3 Resultados

Cada variante do LDC e ELSDC, apresentadas na Seção 4.4.4, foram executadas 50 vezes para cada conjunto de documentos da Seção 5.1.1. A Tabela 5.2 mostra os valores médios e desvio padrão do *Adjusted Rand Index* (ARI) [Hubert & Arabie, 1985] obtido sobre essas execuções. Além do ARI, foi calculado o critério *Normalized Mutual Information* (NMI) [Bishop, 2006], que também é amplamente usado na prática e em trabalhos da literatura. No entanto, uma vez que os resultados são muito similares e as conclusões principais não mudam, são reportados apenas os resultados ARI, a fim de compactação.

Com o objetivo de avaliar a significância dos resultados experimentais, testes de hipótese foram adotados sobre os valores ARI. *Analysis of variance* (ANOVA) [Walpole et al., 2007] e *Friedman test* [Hollander & Wolfe, 1999] foram aplicados com 95% de confiança. Assume-se que, quando a hipótese nula é rejeitada em ambos os testes, há evidência estatística de que os resultados comparados são diferentes. O *Bonferroni procedure* [Hochberg & Tamhane, 1987; Hochberg, 1988] foi aplicado (usando Matlab®) para os valores críticos, para compensar as multi-comparações, mantendo o atual nível de confiança estatística em 95%

O melhor resultado e os resultados sem diferença (estatisticamente) significante para o melhor índice estão destacados em negrito na Tabela 5.2. A linha "Melhor resultado" na Tabela 5.2 contém o número de conjuntos de documentos em que a configuração de experimento obteve o melhor resultado geral, enquanto a linha "Equivalente ao melhor" contém o número de conjuntos de documentos em que a configuração resultou em valores sem diferença estatisticamente significante para o melhor resultado. A linha "Melhor absoluto" na Tabela 5.2 contém o número de conjuntos de documentos em que a configuração apresentou um valor médio absolutamente melhor, ou seja, com diferença significante para todas as outras configurações.

Tabela 5.2: Valores da média e desvio padrão do ARI sobre 50 execuções do algoritmo LDC com as medidas euclidiana, cosseno, GTM e FGTM; e combinações de (Euc+FGTM) e (Cos+FGTM) pelo método de consenso. Os resultados das combinações (Euc+Wik) e (Wik+FGTM) foram obtidos pelo algoritmo ELSDC. Os valores destacados em negrito são os melhores índices obtidos por cada conjunto de documento. As linhas que possuem mais de um valor destacados em negrito mostram que os resultados em negrito não possuem diferença estatisticamente significante.

Conjunto de Documentos	Euclidiana	Cosseno	GTM	FGTM	Euc+Wik	Euc+FGTM	Cos+FGTM	Wik+FGTM
20ng-subset	**0.409±0.022**	0.355±0.029	0.349±0.019	0.267±0.099	0.293±0.030	**0.432±0.027**	**0.429±0.023**	0.176±0.054
20ng-whole	0.397±0.057	0.381±0.013	0.343±0.013	0.349±0.087	0.197±0.032	**0.452±0.064**	**0.450±0.011**	0.144±0.046
Articles-1442-5	**0.940±0.107**	**0.937±0.115**	**0.936±0.117**	**0.943±0.101**	**0.900±0.133**	**0.943±0.100**	**0.966±0.078**	**0.972±0.070**
Cbr-ilp-ir-son	0.787±0.115	0.805±0.043	0.746±0.050	**0.820±0.041**	**0.842±0.033**	**0.812±0.150**	**0.807±0.053**	**0.843±0.031**
Cbr-ilp-ir-son-int	0.742±0.089	0.759±0.094	0.698±0.074	**0.779±0.059**	**0.811±0.062**	**0.793±0.067**	**0.783±0.089**	**0.824±0.049**
Classic4	**0.805±0.001**	0.756±0.104	0.359±0.004	0.704±0.001	0.619±0.001	**0.814±0.000**	**0.800±0.001**	0.585±0.001
News-10	0.456±0.076	0.494±0.058	0.430±0.047	**0.587±0.045**	**0.581±0.037**	**0.617±0.062**	**0.609±0.041**	**0.603±0.038**
News-multi7	0.629±0.029	0.649±0.041	0.635±0.029	0.701±0.034	0.693±0.031	**0.754±0.041**	**0.736±0.025**	0.679±0.033
News-multi10	0.584±0.030	0.560±0.012	0.523±0.026	0.567±0.022	0.587±0.044	**0.678±0.030**	**0.654±0.035**	0.578±0.045
News-rel3	0.412±0.088	**0.464±0.120**	0.438±0.109	**0.462±0.150**	**0.480±0.094**	**0.511±0.154**	**0.523±0.168**	**0.474±0.095**
News-sim3	0.384±0.098	**0.430±0.085**	0.350±0.078	**0.432±0.073**	0.286±0.018	**0.483±0.084**	**0.472±0.116**	0.269±0.023
Pubmed2000sel	0.907±0.123	**0.940±0.022**	0.640±0.022	**0.957±0.079**	**0.973±0.003**	**0.953±0.083**	**0.962±0.016**	**0.974±0.004**
Pubmed2000non	0.415±0.058	0.446±0.037	0.332±0.051	**0.512±0.029**	0.448±0.051	0.479±0.025	0.462±0.060	0.472±0.061
Pubmed4000	0.461±0.015	0.475±0.020	0.364±0.028	**0.559±0.018**	0.502±0.010	0.496±0.031	0.505±0.005	**0.560±0.016**
Reuters8-whole	0.341±0.041	0.351±0.055	0.236±0.023	**0.394±0.162**	**0.425±0.022**	**0.429±0.028**	**0.408±0.048**	0.374±0.104
Scopus2800	0.828±0.068	**0.882±0.056**	0.813±0.076	**0.884±0.069**	**0.884±0.065**	**0.902±0.067**	**0.911±0.034**	**0.890±0.065**
SMS	**0.496±0.154**	0.262±0.132	0.077±0.075	0.056±0.112	0.192±0.106	**0.347±0.121**	0.212±0.123	0.200±0.129
webkb	0.122±0.046	0.123±0.032	0.058±0.021	0.137±0.046	0.113±0.053	**0.237±0.084**	**0.238±0.058**	0.067±0.033
Melhor resultado:	1	0	0	1	0	**8**	3	5
Equivalente ao melhor:	4	5	1	11	8	**16**	15	8
Melhor absoluto:	0	0	0	**1**	0	0	0	0

5.1.4 Análise dos resultados

Olhando para os valores destacados na Tabela 5.2, é notável que os melhores resultados estão concentrados nas configurações que utilizam método de consenso, especialmente os que implementam FGTM, a medida de similaridade proposta neste trabalho. A variante com mais resultados *Equivalente ao melhor*, ou seja, não significantemente diferente do melhor resultado, são as "Euc+FGTM" e "Cos+FGTM", que obtiveram 16 e 15, respectivamente, dos melhores resultados em 18 conjuntos de dados. A "Euc+FGTM" e "Cos+FGTM" também obtiveram *Melhor Resultado*, i.e., a média absoluta mais alta, em 8 e 3 conjuntos, respectivamente.

Ambas FGTM e "Euc+Wik" são medidas de similaridade semântica. A medida FGTM, mesmo quando usada sozinha, consegue resultados competitivos. A FGTM sozinha obteve resultados *equivalente ao melhor* em 11 de 18 conjuntos de documentos, *melhor resultado* e *melhor absoluto*, ou seja, significantemente melhor que todos os outros experimentos, em 1 conjunto de documentos. A variante "Euc+Wik" obteve resultados *equivalente ao melhor* em 8 conjuntos de documentos, mas ela não obteve *melhor resultado* em nenhum conjunto de documentos. A FGTM utiliza o *corpus* Google n-grams, e a "Euc+Wik", que é executada no algoritmo ELSDC, usa conceitos da Wikipédia. Como mostrado na Seção 4.4, embora o ELSDC seja mais complexo que o LDC, o cálculo FGTM possui complexidade $O(T'^2)$, enquanto a distância euclidiana, que é usada em "Euc+Wik", possui complexidade $O(T')$. Desta forma, a comparação de custo computacional depende das características do conjunto de documentos utilizado. Quando $T' >> \bar{T}$, o custo computacional do FGTM é maior que "Euc+Wik". Quando $\bar{T} > T'$, ou T' não é muito maior que $\bar{T}$, o custo computacional de "Euc+Wik" é maior que FGTM.

Embora "Euclidiana" e "Cosseno" tenham o menor custo computacional, elas obtiveram resultados *equivalente ao melhor* somente em 4 e 5 conjuntos de documentos, respectivamente. A medida GTM, que possui mesma complexidade computacional da FGTM, só obteve bom resultado, comparado com as outras medidas, no conjunto de documentos "Articles-1442-5", em que todas as medidas de similaridade não são significantemente diferentes, segundo o teste estatístico aplicado.

Dentre as combinações usando FGTM, a união com conceitos da Wikipédia tem menor quantidade de resultados *equivalente ao melhor*, obtendo em 8 dos 18 conjuntos. No entanto, esta combinação obteve *melhor resultado* em 5 deles. É importante ressaltar que, como mostrado na Seção 4.4, a "Wik+FGTM" é a configuração de experimento com maior custo computacional assintótico dentre todos experimentos.

Quando comparamos as combinações "Euc+Wik" e "Euc+FGTM", é notável que FGTM supera significantemente o uso de conceitos da Wikipédia em 8 conjuntos de documentos e não é superado em nenhum conjunto. Nos conjuntos restantes, não existe diferença estatisticamente significante entre elas. Quando consideramos as médias absolutas, a FGTM obteve melhores resultados que os conceitos da Wikipédia em 13 dos 18 conjuntos de documentos. A diferença de custo computacional entre essas duas combinações (e usando somente o FGTM) depende dos valores de T' e $\bar{T}$, ou seja, o número de termos selecionados em um conjunto de documentos e o número médio de termos em todos os documentos do conjunto. Desta forma, se $T' >> \bar{T}$, a "Euc+FGTM" possui custo computacional maior que "Euc+Wik". Por outro lado, se $\bar{T} > T'$ ou T' não é muito maior que $\bar{T}$, o custo computacional de "Euc+Wik" é maior que "Euc+FGTM".

Considerando o custo computacional e a qualidade dos resultados, o uso da combinação "Euc+FGTM" é a mais eficiente dentre as configurações comparadas. Ela possui melhores resultados que todas as outras configurações, possui menor custo computacional que "Wik+FGTM" e o mesmo custo que "Cos+FGTM", ambos considerados as melhores configurações. A "Euc+FGTM" executa $NT'K + NT'$ mais operações que o FGTM sozinho, ou seja, a diferença de custo é linear para as variáveis N, T' e K. Embora a "Euc+FGTM" não tenha sempre o custo computacional inferior ao da "Euc+Wik", a qualidade dos resultados da "Euc+FGTM" é significantemente melhor. Embora as medidas Euclidiana e Cosseno tenham um menor custo computacional que "Euc+FGTM", como mostrado na Seção 4.4, essas medidas não obtiveram bons resultados na maioria dos conjuntos de documentos.

5.2 Comparação de sentenças curtas

Nesta seção é abordada a tarefa de comparação de sentenças curtas disponibilizadas pela conferência *Semantic Evaluation* (SemEval[10]). SemEval é uma série contínua de avaliações de sistemas de análise semântica computacional, que é organizada e amparada pelo *Special Interest Group on the Lexicon of the Association for Computational Linguistics* (SIGLEX). O objetivo da tarefa é mensurar a similaridade semântica entre duas dadas sentenças em uma escala de 0 a 5, tentando emular a ideia de compreensão da linguagem humana [Rychalska et al., 2016].

A tarefa de comparação de sentenças curtas do SemEval contém, geralmente, 5 conjuntos de frases pareadas, coletadas de diferentes fontes. Os pares de frases

[10] `http://en.wikipedia.org/wiki/SemEval`

são então julgados e anotados por humanos, onde cada pessoa recebe 5 pares de frases de uma só vez. Ao todo, são coletadas 5 anotações por par de frases [Agirre et al., 2015]. As frases são então disponibilizadas na página da conferência, onde os candidatos devem utilizar métodos automáticos para julgar a semelhança entre cada par de frases. Após os candidatos submeterem os resultados de avaliação obtidos por seus métodos, o SemEval divulga os índices modelo (ou *gold standard*), que são os índices avaliados pelos humanos. Os métodos são então avaliados pelo coeficiente de *Pearson* [Pearson, 1895] obtido entre os resultados do método e os rótulos de *gold standard* divulgados pelo SemEval.

5.2.1 Conjuntos de Sentenças

Nos experimentos deste trabalho foram utilizados os conjuntos de sentenças curtas das tarefas do SemEval de 2015 [Agirre et al., 2015] e 2016 [Agirre et al., 2016]. A Tabela 5.3 descreve os nomes, quantidades e fontes dos conjuntos.

Tabela 5.3: Descrição dos conjuntos de sentenças da SemEval, utilizados neste trabalho.

Ano	Conjunto	Pares	Fonte
2015	*HDL*	750	Manchetes do *Newswire*
	Images	750	Descrições de imagens
	Ans.-student	750	Respostas de estudantes
	Ans.-forum	375	Respostas do fórum Q&A
	Belief	375	*Committed belief*
2016	*HDL*	249	Manchetes do *Newswire*
	Plagiarism	230	respostas curtas plagiadas
	Postediting	244	MT *postedits*
	Ans.-Ans.	254	Respostas do fórum Q&A
	Quest.-Quest.	209	Perguntas do fórum Q&A

5.2.2 Resultados

Como descrito anteriormente, para avaliar uma medida de similaridade para tarefas de sentenças curtas, é necessário julgar todos os pares de sentenças com uma pontuação de 0 a 5. Após ter feito isso, os rótulos de *gold standard* são utilizados para comparar com os resultados obtidos pela medida de similaridade. A comparação é feita com uso do coeficiente de *Pearson*.

As frases foram julgadas utilizando 4 medidas de similaridade abordadas nesse trabalho: cosseno (apresentada na Seção 2.1.2), M3 (apresentada na Seção 2.2.3), GTM (apresentada na Seção 2.2.1) e FGTM (proposta na Seção 4.2). Os relacionamentos entre termos na medida M3 foram calculados pela Equação 2.3 da medida GTM. Também são apresentados os resultados dos métodos com melhor coeficiente nos respectivos anos em que a tarefa foi proposta. O método *DLS@CU-S1* [Sultan et al., 2015] foi o vencedor em 2015, enquanto o método *Samsung Poland NLP Team* [Rychalska et al., 2016] foi o vencedor em 2016.

Os resultados são apresentados nas Tabelas 5.4 e 5.5, que são as tarefas SemEval-2015 e 2016, respectivamente.

Tabela 5.4: SemEval-2015. Comparação dos coeficientes obtidos pelas medidas cosseno, M3, GTM e FGTM com o melhor método publicado na SemEval. O rank considera todos os métodos submetidos no evento (com 78 no total).

Run	**HDL**	**Images**	**Ans.-student**	**Ans.-forum**	**Belief**	**Média**	**Rank**
DLS@CU-S1	0,7390	0,7725	0,7491	0,8250	0,8644	0,8015	1
FGTM	0,6378	0,6292	0,6614	0,7131	0,7575	0,6798	43
COSSENO	0,6339	0,5992	0,5735	0,7111	0,7686	0,6573	46
GTM	0,6343	0,5790	0,5737	0,7003	0,6735	0,6322	54
M3	0,3582	0,3527	0,1820	0,4944	0,5239	0,3823	75

Tabela 5.5: SemEval-2016. Comparação dos coeficientes obtidos pelas medidas cosseno, M3, GTM e FGTM com o melhor método publicado na SemEval. O rank considera todos os métodos submetidos no evento (com 127 no total).

Run	**HDL**	**Plagiar.**	**Posted.**	**Ans.-Ans.**	**Quest.-Quest.**	**Média**	**Rank**
Samsung P. NLP	0,7390	0,7725	0,7491	0,8250	0,8644	0,8015	1
FGTM	0,6858	0,7022	0,7775	0,7779	0,6618	0,7210	50
COSSENO	0,4587	0,7014	0,7679	0,7033	0,6510	0,6565	89
GTM	0,5417	0,6315	0,7469	0,6955	0,5213	0,6274	100
M3	0,3011	0,5293	0,5098	0,3458	0,4938	0,4359	118

5.2.3 Análise dos resultados

A medida GTM foi projetada para tarefas de comparação de sentenças curtas [Islam et al., 2012]. Embora a medida FGTM, proposta neste trabalho, tenha sido projetada para agrupamento de documentos, ela supera os resultados obtidos pela GTM em ambas tarefas SemEval, conforme mostrado nas Tabelas 5.4 e 5.5. A medida cosseno, mesmo não utilizando conceitos semânticos, supera a medida M3, que é a versão semântica da similaridade cosseno.

Apesar de a medida FGTM superar as medidas apresentadas nesse trabalho, existem diversos métodos no SemEval que superam o FGTM. Isso ocorre pois esses métodos são projetados especificamente para essa tarefa de comparação de sentenças curtas. O método *Samsung Poland NLP Team*, por exemplo, utiliza os resultados de todos os desafios SemEval, desde de 2012, como conjunto de treinamento [Rychalska et al., 2016]. Além disso, são executadas diversas técnicas de complexidade elevada para mensurar as similaridades, o que seria inviável para tarefas complexas, que utilizam grandes quantidades de textos, como é o caso do agrupamento de documentos.

5.3 Considerações finais

Neste capítulo, experimentos e resultados foram apresentados e discutidos. A medida de similaridade FGTM, proposta neste trabalho, foi testada empiricamente em tarefas de agrupamento de documentos e comparação de sentenças curtas. Na tarefa de agrupamentos, a medida FGTM, quando utilizada no algoritmo LDC, supera todas as demais medidas usadas sozinhas. O FGTM também consegue resultados competitivos quando comparado a métodos mais complexos, como é o caso do algoritmo ELSDC, que mescla agrupamentos obtidos por BOW e BOC. Quando o FGTM é combinado com métricas de distância, como a euclidiana, ele supera todas as outras medidas e combinações comparadas.

Na tarefa de comparação de sentenças curtas, o FGTM é o 43^o (de 78) no SemEval-2015 [Agirre et al., 2015] e 50^o (de 127) no SemEval-2016 [Agirre et al., 2016]. Isso ocorre pois, o FGTM é uma medida desenvolvida para agrupamento de textos, logo, precisa lidar com grandes quantidades de textos com o menor custo computacional possível. Como o SemEval utiliza apenas frases curtas e não julga o tempo para realizar o cálculo, os melhores métodos propostos tendem a ser mais dispendiosos. Além disso, eles são desenvolvidos especificamente para essa tarefa, como é o caso do método vencedor do SemEval-2016, que utiliza todos os resultados SemEval, desde 2012, como conjunto de treinamento. Porém, a medida FGTM supera sua antecessora, GTM, e medidas como cosseno e M3, que é uma versão semântica do cosseno clássico.

6 AGRUPAMENTO COM NÚMERO VARIÁVEL DE GRUPOS

O problema de agrupamento de dados foi abordado inicialmente no Capítulo 3. Neste capítulo é investigado um outro problema envolvendo algoritmos de agrupamento: a estimativa do número K de grupos de forma automática. Este capítulo está dividido da seguinte forma: na Seção 6.1, o problema de estimar o número K de grupos é definido e os principais aspectos são abordados; a Seção 6.2 faz uma revisão de uma heurística e uma meta-heurística da literatura, desenvolvidas para estimar o número de grupos em agrupamentos de dados. Além disso, uma nova versão da heurística estatística *G-means* é proposta; já na Seção 6.3, é feito um estudo sobre os grupos dos conjuntos textuais quando são agrupados por algoritmos baseados em K-médias e guiados por índice de validação interno; na Seção 6.4, são apresentados os experimentos e resultados de agrupamento utilizando as heurísticas descritas neste capítulo; por fim, a Seção 6.5 trata das considerações finais deste capítulo.

6.1 Definição e aspectos principais

Como já abordado no Capítulo 3, algoritmos de agrupamento, como K-médias, buscam encontrar grupos onde os dados pertencentes a um mesmo grupo sejam altamente similares entre si e dissimilares aos dados dos outros grupos [Pinheiro, 2008].

Esse processo é realizado por meio das medidas de similaridade entre objetos. Em 2009, o algoritmo particional K-médias foi eleito entre os dez mais influentes algoritmos de Mineração de Dados [Wu & Kumar, 2009]. Contudo, ele possui limitações, como a sensibilidade na seleção dos centróides iniciais [Jain et al., 1999] e a necessidade de especificação do número K de agrupamentos a priori. Como os algoritmos de agrupamento processam dados sem nenhum conhecimento prévio, a necessidade de estimar o número K de grupos torna-se um fator bastante restritivo. Aqui é tratado o problema de agrupamento sem a necessidade de pré-estabelecer um número de K. Nesse cenário, são empregadas heurísticas para encontrar um número de grupos coesos, sem exigir parâmetros do usuário.

A sensibilidade na seleção de protótipos iniciais também é uma limitação a ser considerada. Mesmo que a quantidade de grupos seja conhecida, algoritmos como o K-médias podem não encontrar os melhores grupos. A Figura 6.1 ilustra um caso de uma base de dados com três grupos bem definidos, mas que o K-médias não consegue chegar ao resultado esperado devido à má inicialização dos protótipos.

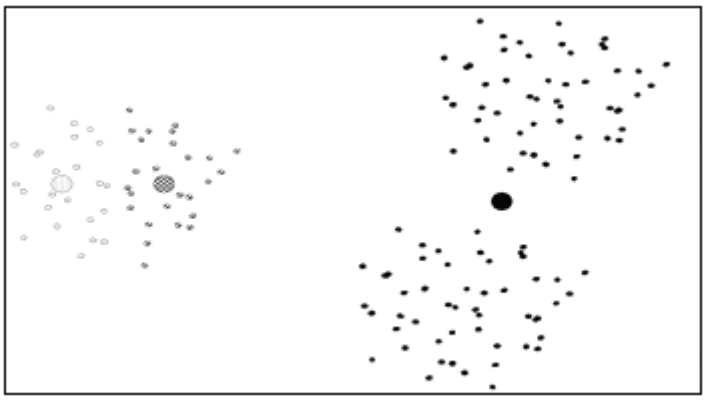

Figura 6.1: Exemplo de mau resultado do K-médias que pode ser causado pela má inicialização dos protótipos iniciais.
Fonte: [Linden, 2009]

Na próxima seção, são apresentadas heurísticas e meta-heurísticas que buscam identificar a quantidade de grupos em um conjunto de dados de forma não supervisionada. O problema de inicialização dos protótipos também é abordado.

6.2 Heurísticas e meta-heurísticas para identificação de K

Heurísticas e meta-heurísticas são técnicas empregadas a problemas de otimização em que não se conhece algoritmos eficientes para obter uma solução ótima.

Em tarefas de agrupamento de dados, heurísticas estatísticas são amplamente empregadas. Essas heurísticas utilizam de testes estatísticos para encontrar padrões de distribuição nos dados. Se um teste de hipótese estatístico é aceito para um determinado grupo de dados, esse grupo é considerado coeso e é mantido para o agrupamento final. No caso de meta-heurísticas, os Algoritmos Evolutivos (AEs) são bem conhecidos na literatura. Os AEs consideram soluções como indivíduos de uma população. As características desses indivíduos são combinadas a fim de obter indivíduos melhores a cada iteração (geração). A qualidade desses indivíduos é mensurada por um índice de aptidão (ou *fitness*). Esses algoritmos seguem a teoria darwiniana de sobrevivência do mais apto, e seleção natural das espécies [Beasley et al., 1993].

A Seção 6.2.1 apresenta uma meta-heurística evolutiva para o problema de identificação do número K de grupos. Na Seção 6.2.2 é apresentado o algoritmo *G-means*, uma heurística para agrupamento de dados baseado em distribuições estatísticas. Por fim, uma versão monitorada do algoritmo *G-means* é proposta na Seção 6.2.3.

6.2.1 *Fast Evolutionary Algorithm for Clustering* (F-EAC)

O algoritmo evolutivo *Fast Evolutionary Algorithm for Clustering* (F-EAC) foi inicialmente proposto em [Alves et al., 2006] como um aprimoramento ao *Evolutionary Algorithm for Clustering* (EAC) [Hruschka et al., 2004]. O F-EAC é uma meta-heurística criada para evoluir partições de dados obtidas pela execução de K-médias de forma eficiente. Para isso, o F-EAC utiliza medidas probabilísticas como critério de seleção ou mutação dos indivíduos. Para avaliação dos grupos gerados, é utilizado o índice de Silhueta Simplificada [Hruschka et al., 2004], apresentado na Seção 3.5.2.2.

O algoritmo F-EAC utiliza os seguintes parâmetros para execução:

- $|\boldsymbol{P}|$: É o tamanho da população que será gerada e mantida durante a execução do F-EAC. Em [Naldi et al., 2011] os autores mostram que altos valores de $|P|$ conseguem bons resultados com um menor número de gerações, mas exigem um maior tempo computacional. Para valores $|P|$ menores, o tempo computacional para execução é reduzido, porém, são necessárias mais gerações para alcançar melhores valores de validação. Os autores definem que $|P| = 10$ é uma quantidade aceitável para esse problema. Esse é o valor utilizado nesse trabalho.

- $\boldsymbol{K_{min}}$: É utilizado como limite inferior do valor de K no sorteio dos indivíduos iniciais. Este parâmetro é considerado apenas na inicialização dos indivíduos, não limitando o valor de K durante a execução do F-EAC. Foi definido $K_{min} = 2$, uma vez que é esperado que o conjunto tenha no mínimo mais de um grupo.

- $\boldsymbol{K_{max}}$: É utilizado como limite superior do valor de K na definição dos indivíduos iniciais. Isso não significa que durante a execução do F-EAC o valor de K não se tornará maior que este parâmetro. Caso o algoritmo consiga melhores índices de aptidão com valores de K maiores que K_{max}, o parâmetro não servirá como interrupção da evolução. O K_{max} definido neste trabalho foi 20.

- $\boldsymbol{t}$: É a quantidade máxima de iterações do algoritmo K-médias. Um estudo realizado em [Anderberg, 1973] sugere que $t = 5$, ou menos interações, normalmente serão suficientes. Esse valor é utilizado neste trabalho.

- $\boldsymbol{g_{max}}$: É a quantidade máxima de gerações que o F-EAC será executado. Em alguns casos, o algoritmo pode ser ajustado para que o critério de parada seja quando um valor específico para um índice de validação seja alcançado. Neste trabalho o algoritmo para com 10 gerações.

A inicialização da população é feita com valores de K e protótipos aleatórios. Cada indivíduo possui um valor inicial de K, onde $K_{min} \leq K \leq K_{max}$, e K protótipos iniciais. A cada geração, é calculado o índice de silhueta simplificada de cada grupo e do agrupamento total de cada indivíduo. Ao utilizar a estratégia elitista, o genótipo com maior aptidão (maior índice de silhueta) é copiado para permanecer na próxima geração. Já os genótipos com menor índice de silhueta possuem maior probabilidade de sofrerem mutação. Para isso, utiliza-se a estratégia da roleta [Mitchell, 1998].

6.2.1.1 Operadores de mutação:

Para escolher os indivíduos que sofrerão mutação, a roleta utiliza a probabilidade de cada indivíduo ser selecionado, que é dada pelo inverso do índice da silhueta simplificada. Ou seja, quanto menor for o índice de silhueta de um agrupamento, maior a chance do indivíduo sofrer mutação.

O F-EAC emprega dois operadores de mutação. O primeiro operador (MO_1) só pode ser aplicado a indivíduos que codificam mais do que dois grupos. O MO_1 elimina um ou mais grupos do genótipo, onde os objetos que ficarem sem grupos

são atribuídos aos grupos mais próximos. A chance de um grupo ser removido é proporcional ao seu valor de validação.

O segundo operador (MO_2) só pode ser aplicado a grupos que possuam mais de dois objetos. O MO_2 seleciona um ou mais grupos do genótipo e divide o grupo em dois. A grosso modo, dado que um grupo G_i tenha sido escolhido para sofrer a mutação MO_2, a divisão do grupo G_i é feita escolhendo um elemento do grupo aleatoriamente. Chamaremos esse elemento de x_j, que será a nova semente (ou medóide) do primeiro grupo da divisão. Feito isso, é buscado o vizinho mais distante a x_j e que pertença ao grupo G_i. Formalmente, procura-se um $x_k \in G_i,\ dist(x_j, x_k) \geq x_l\ \forall l = 1, ..., N_i$, onde N_i é a quantidade de elementos do grupo G_i. Esse vizinho mais distante será a semente do segundo grupo da divisão.

6.2.1.2 Considerações finais sobre o F-EAC:

Cada indivíduo possui uma variável que armazena qual a mutação aplicada na última geração. Caso a mutação tenha causado uma piora no índice de silhueta, a mutação não será repetida naquele indivíduo na geração seguinte. Caso a mutação tenha causado uma melhora no índice de silhueta, a mesma mutação será aplicada mais uma vez. Ou seja, a taxa de aplicação de cada operador de mutação é dinamicamente auto-ajustada pelo F-EAC, conforme o agrupamento melhora ou piora.

Os operadores de mutação são responsáveis por reduzir ou aumentar a quantidade de grupos dos indivíduos, bem como selecionar novos protótipos. Essas características solucionam as limitações do K-médias clássico, o que não torna necessário fixar um valor de K e possibilita a criação de grupos com diferentes protótipos iniciais.

6.2.2 *The Gaussian-means* (G-*means*)

O *G-means* [Hamerly & Elkan, 2003] é uma heurística de agrupamento de dados baseada em conceitos estatísticos de distribuição. A execução do *G-means* começa com um número pequeno de grupos, onde, a cada iteração, os conjuntos que não seguem uma distribuição gaussiana são divididos em dois grupos. O *G-means* é normalmente iniciado com $K = 1$, mas pode ser iniciado com diferentes valores de K, principalmente quando há um conhecimento prévio do conjunto a ser agrupado [Debatty et al., 2014]. Dado como entrada um conjunto X de N objetos, um nível α de confiança e um valor inicial para K, o *G-means* é executado conforme descrito no Algoritmo 4.

Algoritmo 4: Passos do algoritmo de agrupamento *G-means*

Entrada: X, K, α

Saída: G grupos Gaussianos

1: Seja C um conjunto com K centróides (geralmente $C \leftarrow \{\bar{x}\}$).
2: $C \leftarrow K$-médias(C, X)
3: Seja $\{x_i | classe(x_i) = j\}$ um conjunto de dados atribuídos ao centróide c_j
4: Usa teste estatístico para detectar se cada $\{x_i | \text{classe}(x_i) = j\}$ segue uma distribuição Gaussiana (com nível α de confiança).
5: Se os dados são Gaussianos, mantém c_j. Caso contrário, divide c_j em dois centróides.
6: Volta ao passo 2 até que mais nenhum centro seja adicionado.

Os protótipos iniciais e os escolhidos para se tornarem os novos protótipos dos grupos que serão divididos são escolhidos aleatoriamente. Após escolher os centróides, o algoritmo K-médias é executado para refinar os grupos. É verificado se cada grupo segue uma distribuição gaussiana, caso contrário, divide o grupo em dois. O *G-means* encerra quando todos os grupos seguirem uma distribuição gaussiana.

O teste estatístico utilizado é baseado no teste de Anderson-Darling. Dada uma lista de valores normalizados, para que tenha média 0 e variância 1, onde s_i é o i-ésimo valor ordenado desta lista; e dado $z_i = F(s_i)$, onde F é o valor da função de distribuição acumulada, o teste de Anderson-Darling é calculado pela seguinte equação:

$$A^2(Z) = -\frac{1}{n}\sum_{i=1}^{n}(2i-1)[\log(z_i) + \log(1 - z_{n+1-i})] - n \tag{6.1}$$

Como o teste é calculado para grupos de dados e não se conhece a média e desvio padrão de toda base, deve-se aplicar a seguinte correção estatística:

$$A^2_*(Z) = A^2(Z)(1 + 4/n - 25/(n^2)) \tag{6.2}$$

Como os conjuntos utilizados em tarefas de agrupamento são multi-dimensionais e o teste Gaussiano é realizado para dados uni-dimensionais, alguns passos são seguidos a fim de possibilitar a análise. Dado um subconjunto G_i de N_i dimensões que pertence ao centro c_i, e as Equações 6.1 e 6.2 do teste de Anderson-Darling, o teste de hipótese é feito, especificadamente, da seguinte forma:

1. Escolhe-se um nível de significância α para o teste.

2. Escolhe-se dois centróides, chamados "filho" de c_i.

3. Executa o K-médias apenas no subconjunto G_i utilizando esses dois centróides. Os centróides obtidos pela execução do K-médias são chamados c_{i1} e c_{i2}.

4. Encontra-se um vetor $\vec{v} = \vec{c_{i1}} - \vec{c_{i2}}$, ou seja, o vetor que conecta c_1 a c_2. Então, calcula-se o escalar da projeção de todos os objetos de G_i em $\vec{v}$: $(s_i, \vec{v})/||\vec{v}||^2$. O resultado é um conjunto U_i', que é uma representação uni-dimensional do conjunto X projetado em $\vec{v}$.

5. Transforma U_i' para que ele tenha média 0 e variância 1.

6. Dado $z_i = F(s_i')$, onde s_i' é o i-ésimo elemento do conjunto U_i'. Se $A_*^2(Z)$ está no intervalo não-crítico no nível de confiança α, então mantém o centróide original e descarta $\{c_{i1}, c_{i2}\}$. Caso contrário, mantém $\{c_{i1}, c_{i2}\}$ no lugar do centróide original.

Segundo descrito em [Debatty et al., 2014], o nível de significância de 95% é amplamente usado na literatura em testes como este. Desta forma, o valor definido de α neste trabalho é 0.95.

6.2.3 *Monitored Gaussian-means* (MG-*means*)

Em [Debatty et al., 2014], os autores afirmam que o *G-means*, muitas vezes, superestima a quantidade de grupos. Isso ocorre quando os grupos que se deseja encontrar não seguem uma distribuição Gaussiana, logo, a distribuição só é encontrada em pequenos subgrupos. Diferente do F-EAC, o *G-means* não utiliza um índice de aptidão para guiar a busca. A decisão de continuar ou não as divisões depende somente da distribuição estatística. Pensando nisso, propomos aqui uma versão monitorada do *G-means*, chamada de *Monitored Gaussian-means* (*MG-means*).

O *MG-means* segue o mesmo critério de execução do *G-means* original. A diferença está em calcular a qualidade da solução a cada iteração, e ao final da execução retornar a solução com melhor índice. Embora o *MG-means* possa finalizar a execução com uma quantidade superestimada de grupos, a solução utilizada será a que obtiver o melhor índice de validação durante a execução. Essa forma de execução também difere de soluções incrementais, conhecidas na literatura. Algoritmos incrementais, como o *Ordered Multiple runs of K-means* (OMRK) [Jain et al., 1999], apenas executam o K-médias para crescentes valores de K. De diferente forma, o *MG-means* divide grupos baseado em probabilidade estatística. O Algoritmo 5 apresenta os passos para execução do *MG-means*.

Algoritmo 5: Passos do algoritmo de agrupamento *MG-means*

Entrada: X, K, α

Saída: Agrupamento com melhor índice de aptidão

1: Seja C um conjunto de K centróides (geralmente $C \leftarrow \{\bar{x}\}$).
2: $C \leftarrow$k-médias(C, X)
3: Seja $\{x_i | classe(x_i) = j\}$ um conjunto de dados atribuídos ao centróide c_j
4: Usa teste estatístico para detectar se cada $\{x_i | classe(x_i) = j\}$ segue uma distribuição Gaussiana (com nível α de confiança).
5: Se os dados são Gaussianos, mantém c_j. Caso contrário, divide c_j em dois centróides.
6: Avalia a solução obtida. Se é a melhor conhecida, armazena a solução.
7: Repete do passo 2 até que mais nenhum centro seja adicionado.
8: Retorna a melhor solução encontrada.

Assim como no F-EAC, a Silhueta Simplificada é utilizada no *MG-means* para avaliar as soluções a cada iteração. O valor de $\alpha = 0.95$ também é utilizado para o *MG-means*.

6.3 Análise de grupos dos conjuntos de textos

As heurísticas abordadas nesse capítulo são todas baseadas em K-médias. O algoritmo K-médias busca por grupos bem comportados (volumétricos com formas aproximadamente hiperesféricas). O índice de Silhueta também é mais apropriado para agrupamentos que seguem distribuições gaussianas multidimensionais hiperesféricas ou levemente alongadas [Vendramin et al., 2010]. As classes conhecidas dos conjuntos de documentos nem sempre formam grupos. Conforme ilustrado na Figura 3.1, há subjetividade na definição do que realmente é um grupo. Em razão disso, se as classes conhecidas do conjunto de documentos não formam grupos bem definidos, as heurísticas não irão procurar por essas classes, mas sim, uma partição que tenha grupos coesos e bem separados.

Nesta seção, experimentos são conduzidos com o objetivo de analisar a estrutura dos grupos conhecidos nos conjuntos de documentos.

6.3.1 Metodologia de análise

O objetivo dessa análise é saber se a estrutura dos conjuntos, considerando as classes conhecidas, se aproximam de um formato que possa ser identificado por algoritmos baseado em K-médias, que busca por grupos volumétricos e com formas aproximadamente hiperesféricas. Para isso, foram adotados métodos supervisiona-

dos para selecionar apenas os termos adequados ao teste. Isso impede que termos não significativos possam comprometer a estrutura dos conjuntos e resultar em partições distorcidas. O método de seleção utilizado foi o *CfsSubsetEval* [Witten et al., 2011], que retorna um subconjunto de termos representativos, considerando o valor preditivo de cada atributo, juntamente com o grau de redundância entre eles.

Após ser feita a seleção supervisionada de termos, os conjuntos foram agrupados utilizando o algoritmo F-EAC (descrito na Seção 6.2.1), que busca maximizar a silhueta dos grupos. Ao final da execução, as maiores silhuetas obtidas foram armazenadas. Também foram computadas as silhuetas do agrupamento modelo de cada conjunto, ou seja, a silhueta que seria obtida caso o agrupamento resultante fosse exatamente igual ao modelo conhecido. Se a silhueta obtida pelo algoritmo F-EAC é próxima ou inferior à silhueta do agrupamento modelo, consideramos que o agrupamento modelo é coeso. Como o objetivo do F-EAC é maximizar a silhueta, é esperado que o índice de silhueta do agrupamento modelo seja elevado. Caso a silhueta do agrupamento modelo seja muito inferior à silhueta obtida no algoritmo F-EAC, os grupos do modelo não são coesos e, logo, algoritmos baseados em K-médias terão dificuldade em encontra-los. Para essa verificação, foram utilizadas as medidas cosseno, GTM e FGTM. Como o F-EAC é um algoritmo probabilístico, cada configuração de agrupamento foi replicada 10 vezes. Os resultados dessa análise são apresentados na próxima seção.

6.3.2 Resultados da análise

A Tabela 6.1 apresenta os resultados de silhueta simplificada para execuções do algoritmo F-EAC com uso de 14 conjuntos de documentos descritos na Seção 5.1.1. Os índices de silhueta do agrupamento modelo também são apresentados, que são baseados nas classes conhecidas. Como o F-EAC é probabilístico, cada configuração de experimento foi replicada 15 vezes. O resultado é apresentado pela média das execuções. Os índices cuja diferença do F-EAC para o modelo sejam menor que 0.20 são destacados em negrito.

Como é mostrado na Tabela 6.1, a maioria dos índices de silhueta dos modelos são bem inferiores aos índices obtidos pelo algoritmo F-EAC. Isso ocorre quando os grupos conhecidos não são coesos. Desta forma, o F-EAC não irá procurar por esses grupos, mas sim, por partições que tenham uma estrutura de grupos coesos e bem separados.

Consideramos que, se a diferença entre o índice obtido pelo F-EAC e o índice do modelo é menor que 0.20, é possível alcançar um resultado de agrupamento

Tabela 6.1: Comparação dos índices médios de silhueta simplificada obtidos pelo algoritmo F-EAC, sobre bases com seleção supervisionada, com o índice do agrupamento modelo, que é obtido baseado nas classes conhecidas.

Conjuntos	**Cosseno**		**GTM**		**FGTM**	
	F-EAC	**Modelo**	**F-EAC**	**Modelo**	**F-EAC**	**Modelo**
20ng-subset	0,7688	0,1190	0,3912	0,0069	0,3095	0,0046
Articles-1442-5	**0,8443**	**0,8431**	**0,5152**	**0,5151**	**0,6001**	**0,6001**
Cbr-ilp-ir-son	**0,6183**	**0,5949**	**0,3868**	**0,3557**	**0,4182**	**0,4131**
Cbr-ilp-ir-son-int	**0,6077**	**0,5334**	**0,3837**	**0,3386**	**0,4111**	**0,2206**
News-10	**0,8442**	**0,7802**	**0,7997**	**0,7351**	**0,6030**	**0,6546**
News-multi7	0,8137	0,2982	0,3866	0,1781	0,4302	0,1097
News-multi10	0,7833	0,1573	0,4587	0,1023	0,3898	0,0814
News-rel3	0,8179	0,2663	0,5493	0,2545	0,4273	0,2076
News-sim3	0,8175	0,2447	0,5186	0,1239	0,5063	0,1744
Pubmed2000sel	**0,8587**	**0,8068**	**0,8384**	**0,8359**	**0,5718**	**0,6698**
Pubmed2000non	0,8282	0,5526	0,6656	0,4574	0,6511	0,4382
Pubmed4000	0,8051	0,5077	0,5507	0,3817	**0,3768**	**0,3993**
SMS	0,8141	0,1613	0,6256	0,1209	0,6126	0,1247
webkb	0,8157	0,1618	0,3619	0,1566	0,3095	0,0817

próximo ao modelo. Se é a diferença entre os índices é maior que 0.20, entende-se que a silhueta das classes é baixa e, portanto, não formam grupos coesos. Dessa forma, os conjuntos de documentos que se encaixam nesse critério são:

- Articles-1442-5
- Cbr-ilp-ir-son
- Cbr-ilp-ir-son-int
- News-10
- Pubmed2000sel
- Pubmed4000

Para o leitor interessado no comportamento dos conjuntos de documentos para outros algoritmos de agrupamento, recomendamos a leitura do Apêndice D, que apresenta um estudo sobre o agrupamento não-supervisionado de termos.

6.4 Experimentos

Nesta seção, são conduzidos experimentos comparativos entre os algoritmos F-EAC, *G-means* e *MG-means*, apresentados nesse capítulo. Para essa tarefa, são utilizados os conjuntos de documentos cujo modelo conhecido possui valor satisfatório de silhueta, identificados na Seção 6.3. Foram utilizadas as medidas de similaridade cosseno, GTM e FGTM em cada um dos algoritmos. A avaliação foi feita utilizando o índice ARI [Hubert & Arabie, 1985]. A fim de obter confiança estatística, cada experimento foi replicado 15 vezes. Os valores ARI são apresentados pela média (μ) e desvio padrão (σ).

Assim como feito na Seção 5.1, testes de hipótese foram adotados com o objetivo de avaliar a significância dos resultados experimentais. Assumimos que, quando a hipótese nula é rejeitada nos testes ANOVA [Walpole et al., 2007] e *Friedman test* [Hollander & Wolfe, 1999], há evidência estatística de que os resultados comparados são diferentes. Os teste foram aplicados com 95% de confiança. O *Bonferroni procedure* [Hochberg & Tamhane, 1987; Hochberg, 1988] foi aplicado para os valores críticos, para compensar as multi-comparações, mantendo o atual nível de confiança estatística em 95%

Os resultados e análises são abordados de duas maneiras. Primeiro, na Seção 6.4.1, é comparada a qualidade dos agrupamentos obtidos por cada heurística apresentada neste capítulo. Segundo, na Seção 6.4.2, é comparada a qualidade dos agrupamentos obtidos por cada medida de similaridade nas heurísticas.

6.4.1 Comparação das heurísticas para agrupamento

Nesta seção, comparamos a meta-heurística F-EAC com as heurística *G-means* e *MG-means*, onde a segunda foi proposta neste trabalho. A Tabela 6.2 apresenta as médias (μ) e desvios padrão (σ) do ARI para a execução de cada heurística sobre cada um dos conjuntos de documentos, com uso das medidas Cosseno, GTM e FGTM. Desta forma, cada μ e σ apresentados na Tabela 6.2 são computados a partir das 45 execuções (15 para cada medida de similaridade) da heurística sobre o conjunto de documentos. Os valores destacados em negrito na Tabela 6.2 representam os índices que não são estatisticamente diferentes do melhor índice obtido em um conjunto de documentos.

Os resultados apresentados na Tabela 6.2 mostram que a meta-heurística F-EAC obteve os melhores índices de validação em todos os conjuntos de documentos.

Tabela 6.2: Média e desvio padrão (μ e σ) do ARI para 45 execuções da meta-heurística F-EAC e heurísticas *G-means* e *MG-means* para cada conjunto de documentos. Foram utilizadas as medidas cosseno, GTM e FGTM, onde os experimentos foram replicados 15 vezes para cada medida de similaridade.

Conjunto	**F-EAC**		***G-means***		***MG-means***	
	μ	σ	μ	σ	μ	σ
Articles-1442-5	**0,9935**	0,0046	0,5262	0,2911	0,6994	0,1809
cbr-ilp-ir-son	**0,8064**	0,0305	0,4835	0,1644	0,6549	0,2521
cbr-ilp-ir-son-int	**0,6723**	0,2353	0,4444	0,1825	**0,6040**	0,2357
news-10	**0,7608**	0,1828	0,2971	0,1491	**0,6976**	0,1265
PubMed2000sel	**0,9051**	0,0343	0,3129	0,1966	0,6076	0,2017
PubMed4000	**0,3181**	0,2007	0,1221	0,1028	**0,2897**	0,1637

Já a heurística *MG-means*, proposta neste capítulo, obteve resultados não estatisticamente diferentes do F-EAC em 3 dos 6 conjuntos de documentos. Além disso, o *MG-means* supera o *G-means* original em todos os conjuntos de documentos. Isso ocorre pois, conforme afirmado por Debatty et al. [2014], o *G-means*, muitas das vezes, superestima a quantidade de grupos. Como o *MG-means* acompanha o desenvolvimento das partições medindo a qualidade de cada agrupamento, logo, o melhor agrupamento será retornado, mesmo que o resultado final das divisões gaussianas seja superestimado. Por outro lado, isso torna o algoritmo dependente da qualidade do índice de validação, nesse caso, a silhueta simplificada. Como o bom funcionamento da silhueta limita-se a agrupamentos concisos e bem separados, o agrupamento modelo pode não ser bem pontuado pela silhueta se o conjunto de dados não possuir uma estrutura concisa.

Uma vantagem do F-EAC está nos métodos de mutação, que são projetados para criar novos ou excluir grupos existentes. Cada mutação é aplicada conforme a aptidão do indivíduo melhora ou piora. Já as heurísticas *G-means* e *MG-means* não são guiadas por um índice de validação. Elas apenas duplicam grupos existentes. Embora o *MG-means* calcule a qualidade dos agrupamentos a cada iteração, o índice obtido não interfere na divisão dos grupos. O índice é utilizado apenas para definir qual dos agrupamentos obtidos será retornado ao final da execução. Outra vantagem do F-EAC é o uso de população, onde indivíduos com diferentes inicializações são processados ao mesmo tempo. Isso contorna o problema de má inicialização do algoritmo K-médias, descrito na Seção 6.1 e ilustrado pela Figura 6.1. O *G-means* e *MG-means* partem de uma única inicialização aleatória por execução, o que é um fator restritivo.

6.4.2 Comparação das medidas de similaridade nas heurísticas de agrupamento

Nesta seção, comparamos a qualidade das partições obtidas por cada medida de similaridade nas heurísticas apresentadas neste capítulo. A Tabela 6.3 apresenta as médias (μ) e desvios padrão (σ) do ARI obtido pelas medidas de similaridade sobre cada um dos conjuntos de documentos na meta-heurística F-EAC e heurísticas *G-means* e *MG-means*. A configuração dos experimentos é a mesma descrita na seção anterior. Desta forma, cada μ e σ apresentados na Tabela 6.3 são computados a partir das 45 execuções (15 para cada heurística) com uso da medida de similaridade sobre o conjunto de documentos. Os valores destacados em negrito na Tabela 6.3 representam os índices que não são estatisticamente diferentes do melhor índice obtido em um conjunto de documentos.

Tabela 6.3: Média e desvio padrão (μ e σ) do ARI de 45 resultados de agrupamento obtidos com uso das medidas de similaridade cosseno, GTM e FGTM para cada conjunto de documentos. Foi utilizada a meta-heurística F-EAC e heurísticas *G-means* e *MG-means*, onde os experimentos foram replicados 15 vezes para cada algoritmo.

Conjunto	**Cosseno**		**GTM**		**FGTM**	
	μ	σ	μ	σ	μ	σ
Articles-1442-5	0,6689	0,2338	0,7067	0,3511	**0,9121**	0,1134
cbr-ilp-ir-son	**0,7400**	0,2066	**0,7777**	0,2396	**0,7688**	0,2108
cbr-ilp-ir-son-int	**0,7641**	0,1664	**0,7024**	0,2520	**0,6913**	0,2410
news-10	0,5590	0,0969	0,6149	0,2201	**0,7825**	0,1824
PubMed2000sel	**0,6449**	0,2315	0,5125	0,2779	**0,6690**	0,3521
PubMed4000	0,2311	0,1775	0,1268	0,1475	**0,3266**	0,1242

A Tabela 6.3 mostra que a medida FGTM supera significantemente a sua antecessora, GTM, em 4 dos 6 conjuntos de documentos e não é superada em nenhum conjunto. Quando comparada com a medida cosseno, o FGTM é significantemente melhor em 3 dos 6 conjuntos e não significantemente diferente nos outros 3 conjuntos.

Neste cenário de experimentos, a seleção de atributos é supervisionada, onde o método de seleção mantém todos os atributos discriminativos, forçando que documentos da mesma classe compartilhem termos. Isso evita os problemas de uso da linguagem, descritos na Seção 2.2, onde documentos semelhantes podem ser representados por termos com sintaxes diferentes, compartilhando poucos ou nenhum termo. Como o objetivo dos experimentos deste capítulo é a avaliação das heurísticas, as medidas de similaridade deveriam influenciar o mínimo possível. Por esta razão, foi utilizada a seleção supervisionada de termos. Embora esse cenário de

experimentos seja favorável à medida cosseno, os resultados apresentados na Tabela 6.3 mostram que a medida FGTM pode obter melhores índices mesmo quando a mensuração semântica não é necessária.

6.5 Considerações finais

Neste capítulo foi abordado o problema de identificação do número de grupos em tarefas de agrupamento de textos. Uma heurística e uma meta-heurística da literatura foram apresentadas. Além disso, uma variação da heurística *G-means* foi proposta, a *MG-means*.

Foi mostrado que a maioria dos modelos de agrupamento conhecidos dos conjuntos de documentos não podem ser encontrados por algoritmos baseados em K-médias. Dos 14 conjuntos de documentos testados, apenas 6 modelos foram julgados como alcançáveis e utilizados nos experimentos comparativos. Nesses experimentos, a meta-heurística F-EAC superou as demais heurísticas. Por outro lado, a heurística *MG-means* superou significativamente a sua antecessora na tarefa de agrupamento de documentos textuais.

Considerando a qualidade das medidas de similaridade, a medida FGTM superou significantemente as medidas cosseno e GTM em 3 e 4 dos 6 conjuntos de documentos, respectivamente. Além disso, a FGTM não foi superada de forma estatisticamente significante em nenhum conjunto. Isso mostra que a medida FGTM é competitiva mesmo em cenários que o número K de grupos não é conhecido.

7 CONCLUSÕES GERAIS E TRABALHOS FUTUROS

O trabalho apresentado nessa dissertação contribui para a área de mineração de textos, onde o objetivo principal é a obtenção de melhores resultados em tarefas de agrupamento de documentos textuais. Com essa finalidade, o estudo abordado no Capítulo 2 permitiu entender o funcionamento, pontos fortes e fracos de diversas medidas de similaridade da literatura. A partir desse estudo, foi possível concluir que, ao trabalhar com textos, é necessário empregar técnicas para contornar problemas de uso da linguagem. Uma das soluções é o uso de medidas de similaridade semântica.

Assim como existem medidas de similaridade específicas para documentos textuais, também existem processos e algoritmos de agrupamento projetados para esse tipo de tarefa. No Capítulo 3, é feito um estudo do processo completo para agrupamento de documentos textuais. A partir desse estudo, foi possível definir os passos a serem seguidos para o desenvolvimento da proposta deste trabalho. Em primeiro lugar, é abordada a importância de se aplicar uma boa seleção de atributos, a fim de manter uma baixa dimensionalidade dos conjuntos de documentos. Manter uma baixa dimensionalidade mantém a capacidade de discriminação dos atributos do conjunto. Em segundo lugar, são apresentados alguns algoritmos de agrupamento clássicos e específicos para documentos textuais. Essa apresentação permite entender os conceitos de centróides e medóides, agrupamento *fuzzy* e a importância desses conceitos nos algoritmos de agrupamento de textos, como o *Lexical Document Clustering* (LDC) [Nourashrafeddin et al., 2013]. Por fim, ainda no Capítulo 3, é feito um estudo sobre índices de validação, que são utilizados para medir a qualidade das partições obtidas pelos algoritmos de agrupamento.

As principais propostas desse trabalho foram apresentadas no Capítulo 4 e experimentadas no Capítulo 5. Dentre elas, uma nova medida de similaridade semântica, baseada no *corpus* tri-gramas do Google. Essa nova medida de similaridade, *Frequency Google Tri-grams* (FGTM), foi desenvolvida com o objetivo de obter melhores resultados em tarefas de agrupamento de textos. A medida combina métodos de similaridade semântica com as frequências TF-IDF dos termos nos documentos. Com isso, mostramos que a medida proposta pode superar outras medidas em tarefas de agrupamento e de comparação de sentenças curtas.

Além da nova medida de similaridade, propomos, também, combinações de medidas para agrupamento de textos. Essas combinações são feitas utilizando um método de consenso. Nós concluímos, através de testes empíricos e de complexidade teórica, que o uso de medidas de similaridade combinadas supera qualitativamente o uso de uma única medida similaridade, onde a complexidade depende das características do conjunto de documentos a ser agrupado. Os métodos propostos foram testados em 18 conjuntos reais de textos, obtendo melhores índices de validação na maioria deles. As propostas de combinações de medidas superam o estado da arte dos algoritmos de agrupamento LDC [Nourashrafeddin et al., 2013] e ELSDC [Nourashrafeddin et al., 2014].

No Capítulo 6, é introduzido um outro problema que envolve agrupamento de dados: a identificação do número K de grupos. Uma variação da heurística *G-means* é proposta. Em [Debatty et al., 2014], os autores afirmam que o *G-means*, muitas vezes, superestima a quantidade de grupos. Em razão disso, propomos uma versão monitorada do *G-means*, que calcula e armazena o índice de validação de cada agrupamento a cada iteração. Ao final da execução, a partição com melhor índice é retornado. Os resultados apresentados permitem concluir que o uso de monitoramento por índice de validação melhora significantemente os resultados da heurística *G-means* original.

Um teste utilizando os grupos conhecidos dos conjuntos de documentos também é conduzido no Capítulo 6. Esse teste permite identificar se o modelo (classes) do conjunto de documento é alcançável quando agrupado por algoritmos baseados em K-médias. O estudo permite concluir que a maioria dos conjuntos de documentos utilizados neste trabalho não possuem estruturas conhecidas detectáveis pelo K-médias. Isso porque os algoritmos aplicados tendem a encontrar grupos mais coesos e separados do que os grupos conhecidos.

Em resumo, a investigação proposta permite concluir que o uso de medida de similaridade semântica com as frequências TF-IDF dos termos nos documentos são capazes de superar medidas da literatura. Também é possível melhorar a qualidade

de partições unindo resultados de agrupamento obtidos por duas medidas de similaridade. Adicionalmente, é possível melhorar a qualidade de agrupamentos obtidos por heurísticas estatísticas integrando índices de validação para monitorar as partições obtidas a cada iteração. Esses resultados abrem caminho para novos estudos, como é abordado na Seção 7.1.

7.1 Trabalhos Futuros

Baseado nos estudos realizados nessa dissertação, diversos novos projetos de pesquisas podem ser desenvolvidos. A medida FGTM, proposta no Capítulo 4, obteve bons resultados quando utilizada no algoritmo particional LDC. No entanto, há na literatura diversos outros tipos de algoritmos de agrupamento, em que a medida FGTM pode ser implementada, como é o caso dos algoritmos hierárquicos e de densidade. Além disso, a medida FGTM possui complexidade quadrática em função do número de atributos. Isso pode tornar a medida inviável quando utilizada em conjuntos de documentos muito grandes, como é o caso de aplicações em *Big Data*. Algumas alternativas seriam desenvolver métodos de aproximação ou estruturas de dados para minimizar a quantidade de consultas ao *corpus*.

Conforme apresentado no Capítulo 5, a medida FGTM obtém bons resultados para tarefas de agrupamentos e supera a GTM, sua antecessora, em tarefas de comparação de sentenças curtas (SemEval). No entanto, diversos métodos do SemEval superam a FGTM para essa tarefa específica. Embora esse não seja o propósito principal da medida, aperfeiçoá-la em tarefas do SemEval, como a comparação de sentenças curtas, pode melhorar os resultados em tarefas de agrupamento. Uma ideia seria integrar métodos de aprendizado de máquina para treinar a medida, assim como é feito no método proposto em [Rychalska et al., 2016], que utiliza todos os pares de sentenças do SemEval, desde 2012, como conjunto de treinamento.

O estudo apresentado no Capítulo 6 abre caminho para diversas outras pesquisas. Nesse capítulo, foi mostrado que a maioria dos conjuntos de documentos possuem classes conhecidas que formam grupos não coesos. Esses grupos não são identificados por K-médias, ou por algoritmos guiados por índice de silhueta. Isso sugere que outros métodos de agrupamento e índices de validação sejam investigados, a fim de identificar a estrutura dos grupos de documentos de forma não supervisionada.

No Capítulo 6, também é proposta uma nova versão do algoritmo *G-means*, chamada *MG-means*. Nos experimentos utilizando documentos textuais, o *MG-*

means supera seu antecessor, mas é superado pela meta-heurística F-EAC. No entanto, o *MG-means* pode ser uma heurística promissora para outros tipos de dados. Um objetivo seria testá-la em diversos conjuntos de dados, reais e artificiais, com diferentes padrões e estruturas. Desta forma, seria possível conduzir uma comparação mais precisa com outras heurísticas da literatura.

Por fim, uma proposta para trabalhos futuros é a implementação de uma ferramenta para agrupamento de textos. Como o problema de agrupamento, quando considerado desde o pré-processamento, exige a implementação de diversos métodos e algoritmos, a integração dos melhores métodos em uma ferramenta automatizada pouparia grande tempo para se obter um resultado de agrupamento. Além disso, uma versão estruturada do *corpus* do Google poderia ser integrado à ferramenta, para que a medida FGTM, proposta neste trabalho, possa ser facilmente utilizada. A ferramenta seria capaz de estruturar os textos, selecionar atributos, identificar a quantidade de grupos, realizar o agrupamento e retornar o resultado para o usuário. O resultado pode ser apresentado utilizando técnicas de visualização de dados ou simplesmente, separando os textos em arquivos ou pastas.

Referências Bibliográficas

Agirre, E.; Banea, C.; Cardie, C.; Cer, D. M.; Diab, M. T.; Gonzalez-Agirre, A.; Guo, W.; Lopez-Gazpio, I.; Maritxalar, M.; Mihalcea, R.; Rigau, G.; Uria, L. & Wiebe, J. (2015). Semeval-2015 task 2: Semantic textual similarity, english, spanish and pilot on interpretability. In Cer, D. M.; Jurgens, D.; Nakov, P. & Zesch, T., editores, *SemEval@NAACL-HLT*, pp. 252–263. The Association for Computer Linguistics.

Agirre, E.; Banea, C.; Cer, D. M.; Diab, M. T.; Gonzalez-Agirre, A.; Mihalcea, R.; Rigau, G. & Wiebe, J. (2016). Semeval-2016 task 1: Semantic textual similarity, monolingual and cross-lingual evaluation. In Bethard, S.; Cer, D. M.; Carpuat, M.; Jurgens, D.; Nakov, P. & Zesch, T., editores, *SemEval@NAACL-HLT*, pp. 497–511. The Association for Computer Linguistics.

Aldenderfer, M. S. & Roger, K. B. (1984). Cluster analysis. *Beverly Hills: Sage Publications*.

Alves, V.; Campello, R. & Hruschka, E. (2006). Towards a fast evolutionary algorithm for clustering. In *IEEE Congress on Evolutionary Computation*, pp. 1776--1783, Vancouver, Canada. IEEE.

Anderberg, M. R. (1973). *Cluster Analysis for Applications*. Academic Press.

Aranha, C. N. (2007). *Uma Abordagem de Pré-Processamento Automático para Mineração de Textos em Português: Sobe o Enfoque da Inteligência Computacional*. PhD thesis, PUC - Rio de Janeiro, Brasil.

Banerjee, S. & Pedersen, T. (2003). Extended gloss overlaps as a measure of semantic relatedness. In *Proceedings of the 18th International Joint Conference on Artificial Intelligence*, IJCAI'03, pp. 805--810, San Francisco, CA, USA. Morgan Kaufmann Publishers Inc.

Beasley, D.; Bull, D. R. & Martin, R. R. (1993). An Overview of Genetic Algorithms: Part 2, research topics. *University Computing*, 15:170--181.

Bellman, R. E. (1961). *Adaptive control processes - A guided tour*. Princeton University Press, Princeton, New Jersey, U.S.A.

Beyer, K. S.; Goldstein, J.; Ramakrishnan, R. & Shaft, U. (1999). When is "nearest neighbor" meaningful? In *Proceedings of the 7th International Conference on Database Theory*, ICDT '99, pp. 217--235, London, UK, UK. Springer-Verlag.

Bishop, C. M. (2006). *Pattern Recognition and Machine Learning*. Number 4 in Information science and statistics. Springer.

Brants, T. & Franz, A. (2006). Web 1t 5-gram corpus version 1. Technical report, Google Research.

Campello, R. J. G. B. & Hruschka, E. R. (2006). A fuzzy extension of the silhouette width criterion for cluster analysis. *Fuzzy Sets and Systems*, 157(21):2858–2875.

Corley, C. & Mihalcea, R. (2005). Measuring the semantic similarity of texts. In *Proceedings of the ACL Workshop on Empirical Modeling of Semantic Equivalence and Entailment*, EMSEE '05, pp. 13--18, Stroudsburg, PA, USA. Association for Computational Linguistics.

Cormack, G. V.; Hidalgo, J. M. G. & Sánz, E. P. (2007). Feature engineering for mobile (sms) spam filtering. In *Proceedings of the 30th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '07, pp. 871--872, New York, NY, USA. ACM.

Debatty, T.; Michiardi, P.; Thonnard, O. & Mees, W. (2014). Determining the k in k-means with mapreduce. In *17th International Conference on Database Theory, in conjunction with EDBT/ICDT*, pp. 24--28, Athens, Greece.

Dunn, J. C. (1973). A fuzzy relative of the isodata process and its use in detecting compact well-separated clusters. *Journal of Cybernetics*, 3(3):32–57.

Faceli, K.; Gama, J.; Carvalho, A. C. P. L. d. & Lorena, A. C. (2011). *Inteligência Artificial, Uma Abordagem de Aprendizado de Máquina*, volume 1. LTC.

Feldman, R. & Sanger, J. (2006). *The Text Mining Handbook: Advanced Approaches in Analyzing Unstructured Data*. Cambridge University Press.

Ferreira, R.; Lins, R. D.; Freitas, F.; Simske, S. J. & Riss, M. (2014). A new sentence similarity assessment measure based on a three-layer sentence representation. In *Proceedings of the 2014 ACM Symposium on Document Engineering*, DocEng '14, pp. 25--34, New York, NY, USA. ACM.

Halkidi, M.; Batistakis, Y. & Vazirgiannis, M. (2001). On clustering validation techniques. *J. Intell. Inf. Syst.*, 17(2-3):107--145.

Hamerly, G. & Elkan, C. (2003). Learning the k in k-means. In *Advances in Neural Information Processing Systems*, volume 17. MIT Press.

Hastie, T.; Tibshirani, R. & Friedman, J. (2009). *The elements of statistical learning: data mining, inference and prediction*. Springer, 2 edição.

Ho, C.; Murad, M. A. A.; Kadir, R. A. & Doraisamy, S. C. (2010). Word sense disambiguation-based sentence similarity. In *Proceedings of the 23rd International Conference on Computational Linguistics: Posters*, COLING '10, pp. 418--426, Stroudsburg, PA, USA. Association for Computational Linguistics.

Hochberg, Y. (1988). A sharper bonferroni procedure for multiple tests of significance. *Biometrika*, 75:800–802.

Hochberg, Y. & Tamhane, A. C. (1987). *Multiple Comparison Procedures*. John Wiley & Sons, Inc., New York, NY, USA.

Hollander, M. & Wolfe, D. A. (1999). *Nonparametric statistical methods*. Wiley series in probability and statistics. Wiley, New York. A Wiley-Interscience publication.

Horn, R. A. & Johnson, C. R. C. R. (2012). *Matrix Analysis*. Cambridge University Press, England, segunda edição.

Horta, D. (2013). *Algoritmos e técnicas de validação em agrupamento de dados multi-representados, agrupamento possibilístico e bi-agrupamento*. PhD thesis, USP - São Carlos - Brasil.

Hotho, A.; Nürnberger, A. & Paaß, G. (2005). A brief survey of text mining. *LDV Forum - GLDV Journal for Computational Linguistics and Language Technology*, 20(1):19–62.

Hruschka, E.; Castro, L. d. & Campello, R. (2004). Evolutionary algorithms for clustering gene-expression data. In *IEEE Int. Conf. on Data Mining*, pp. 403--406, Brighton/England. IEEE.

Hu, J.; Fang, L.; Cao, Y.; Zeng, H.-J.; Li, H.; Yang, Q. & Chen, Z. (2008). Enhancing text clustering by leveraging wikipedia semantics. In *Proceedings of the 31st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '08, pp. 179--186, New York, NY, USA. ACM.

Hu, X.; Zhang, X.; Lu, C.; Park, E. K. & Zhou, X. (2009). Exploiting wikipedia as external knowledge for document clustering. In *Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '09, pp. 389--396, New York, NY, USA. ACM.

Huang, A. (2008). Similarity Measures for Text Document Clustering. In Holland, J.; Nicholas, A. & Brignoli, D., editores, *New Zealand Computer Science Research Student Conference*, pp. 49--56, Christchurch, New Zealand.

Huang, A.; Milne, D.; Frank, E. & Witten, I. H. (2008). Clustering documents with active learning using wikipedia. In *Proceedings of the 2008 Eighth IEEE International Conference on Data Mining*, ICDM '08, pp. 839--844, Washington, DC, USA. IEEE Computer Society.

Huang, A.; Milne, D.; Frank, E. & Witten, I. H. (2009). Clustering documents using a wikipedia-based concept representation. In *Proceedings of the 13th Pacific-Asia Conference on Advances in Knowledge Discovery and Data Mining*, PAKDD '09, pp. 628--636, Berlin, Heidelberg. Springer-Verlag.

Huang, L.; Milne, D.; Frank, E. & Witten, I. H. (2012). Learning a concept-based document similarity measure. *Journal of the American Society for Information Science and Technology*, 63(8):1593--1608.

Hubert, L. & Arabie, P. (1985). Comparing partitions. *Journal of Classification*, 2(1):193--218.

Islam, A. & Inkpen, D. (2008). Semantic text similarity using corpus-based word similarity and string similarity. *ACM Trans. Knowl. Discov. Data*, 2(2):10:1--10:25.

Islam, A.; Milios, E. & Keselj, V. (2012). Text similarity using google tri-grams. In *Proceedings of the 25th Canadian Conference on Advances in Artificial Intelligence*, Canadian AI'12, pp. 312--317, Berlin, Heidelberg. Springer-Verlag.

Islam, M. A. & Inkpen, D. (2006). Second order co-occurrence pmi for determining the semantic similarity of words. In *In Proceedings of the International Conference on Language Resources and Evaluation*, pp. 1033--1038, Genoa, Italy.

Jaccard, P. (1912). The Distribution of the Flora in the Alpine Zone. *New Phytologist*, 11(2):37--50.

Jain, A. K. & Dubes, R. C. (1988). *Algorithms for Clustering Data*. Prentice-Hall, Inc., Upper Saddle River, NJ, USA.

Jain, A. K.; Murty, M. N. & Flynn, P. J. (1999). Data clustering: A review. *ACM Comput. Surv.*, 31(3):264--323.

Kaplan, A. (1955). An experimental study of ambiguity and context. *Mechanical Translation*, 2:39--46.

Kogan, J.; Nicholas, C. & Volkovich, V. (2003). Text mining with information-theoretic clustering. *Computing in Science and Engg.*, 5(6):52--59.

Krishnapuram, R.; Joshi, A.; Nasraoui, O. & Yi, L. (2001). Low-complexity fuzzy relational clustering algorithms for web mining. *Transactions on Fuzzy Systems*, 9(4):595--607.

Kullback, S. & Leibler, R. A. (1951). On Information and Sufficiency. *The Annals of Mathematical Statistics*, 22(1):79--86.

Lafore, R. (2002). *Data Structures and Algorithms in Java*. Sams, Indianapolis, IN, USA, segunda edição.

Larose, D. T. (2005). *Discovering Knowledge in Data: an introduction to data mining*. Wiley-Interscience, primeira edição.

Lee, M. D. & Welsh, M. (2005). An empirical evaluation of models of text document similarity. In *In CogSci2005*, pp. 1254--1259. Erlbaum.

Li, Y.; McLean, D.; Bandar, Z. A.; O'Shea, J. D. & Crockett, K. (2006). Sentence similarity based on semantic nets and corpus statistics. *IEEE Trans. on Knowl. and Data Eng.*, 18(8):1138--1150.

Linden, R. (2009). Técnicas de agrupamento. *Revista de Sistemas de Informação da FSMA*, 4:18--36.

Manning, C. D.; Raghavan, P. & Schütze, H. (2008). *Introduction to Information Retrieval*. Cambridge University Press, New York, NY, USA.

Mihalcea, R.; Corley, C. & Strapparava, C. (2006). Corpus-based and knowledge-based measures of text semantic similarity. In *Proceedings of the 21st National Conference on Artificial Intelligence - Volume 1*, AAAI'06, pp. 775--780. AAAI Press.

Miller, G. A. (1995). Wordnet: A lexical database for english. *Commun. ACM*, 38(11):39--41.

Milligan, G. W. & Cooper, M. C. (1985). An examination of procedures for determining the number of clusters in a data set. *Psychometrika*, 50(2):159--179.

Milne, D. & Witten, I. H. (2013). An open-source toolkit for mining wikipedia. *Artif. Intell.*, 194:222--239.

Mitchell, M. (1998). *An Introduction to Genetic Algorithms*. MIT Press, Cambridge, MA, USA.

Naldi, M. C.; Campello, R. J. G. B.; Hruschka, E. R. & Carvalho, A. C. P. L. F. (2011). Efficiency issues of evolutionary k-means. *Appl. Soft Comput.*, 11(2):1938--1952.

Nourashrafeddin, S. (2014). *Interactive User-Supervised Text Document Clustering*. PhD thesis, Dalhousie University - Halifax - Canada.

Nourashrafeddin, S.; Milios, E. & Arnold, D. (2013). Interactive text document clustering using feature labeling. In *Proceedings of the 2013 ACM Symposium on Document Engineering*, DocEng '13, pp. 61--70, New York, NY, USA. ACM.

Nourashrafeddin, S.; Milios, E. & Arnold, D. V. (2014). An ensemble approach for text document clustering using wikipedia concepts. In *Proceedings of the 2014 ACM Symposium on Document Engineering*, DocEng '14, pp. 107--116, New York, NY, USA. ACM.

Parikh, D. & Tirkha, P. (2013). Data Mining & Data Stream Mining - Open Source Tools . *Nature*, 2:5234--5239.

Paulovich, F. V.; Nonato, L. G.; Minghim, R. & Levkowitz, H. (2008). Least square projection: A fast high-precision multidimensional projection technique and its application to document mapping. *IEEE Transactions on Visualization and Computer Graphics*, 14(3):564--575.

Pearson, K. (1895). Note on regression and inheritance in the case of two parents. *Proceedings of the Royal Society of London*, 58(347-352):240--242.

Pinheiro, C. A. R. (2008). *Inteligência Analítica*. Ciência Moderna, 1 edição.

Rajaraman, A. & Ullman, J. D. (2011). *Mining of Massive Datasets*. Cambridge University Press, New York, NY, USA.

Rakib, M.; Islam, A. & Milios, E. (2015). Trwp: Text relatedness using word and phrase relatedness. In *Proceedings of the SemEval-2015*, pp. 90--95, New York, NY, USA. ACM.

Rand, W. (1971). Objective criteria for the evaluation of clustering methods. *Journal of the American Statistical Association*, 66(336):846–850.

Rezende, S. O.; Marcacini, R. M. & Moura, M. F. (2011). O uso da mineração de textos para extração e organização não supervisionada de conhecimento. *Revista de Sistemas de Informação da FSMA*, 7:7–21.

Rocha, C. A. J. (1999). Redes bayesianas para extração de conhecimento de bases de dados, considerando a incorporação de conhecimento de fundo e o tratamento de dados incompletos. Master's thesis, ICMC-USP, São Carlos - SP.

Rousseeuw, P. (1987). Silhouettes: A graphical aid to the interpretation and validation of cluster analysis. *Journal of Computational and Applied Mathematics*, 20(1):53--65.

Rychalska, B.; Pakulska, K.; Chodorowska, K.; Walczak, W. & Andruszkiewicz, P. (2016). Samsung poland nlp team at semeval-2016 task 1: Necessity for diversity; combining recursive autoencoders, wordnet and ensemble methods to measure semantic similarity. In *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016)*, pp. 614--620, San Diego, California. Association for Computational Linguistics.

Salton, G. & Buckley, C. (1988). Term-weighting approaches in automatic text retrieval. *Inf. Process. Manage.*, 24(5):513--523.

Salton, G. & McGill, M. J. (1986). *Introduction to Modern Information Retrieval*. McGraw-Hill, Inc., New York, NY, USA.

Schonhofen, P. (2006). Identifying document topics using the wikipedia category network. In *Proceedings of the 2006 IEEE/WIC/ACM International Conference*

on Web Intelligence, WI '06, pp. 456--462, Washington, DC, USA. IEEE Computer Society.

Sebastiani, F. (2002). Machine learning in automated text categorization. *ACM Comput. Surv.*, 34(1):1--47.

Sultan, M. A.; Bethard, S. & Sumner, T. (2015). Dls@cu: Sentence similarity from word alignment and semantic vector composition. In *Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval 2015)*, pp. 148--153, Denver, Colorado. Association for Computational Linguistics.

Tan, P.-N.; Steinbach, M. & Kumar, V. (2005). *Introduction to Data Mining, (First Edition)*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA.

Tang, B.; Shepherd, M.; Milios, E. & Heywood, M. (2005). Comparing and combining dimension reduction techniques for efficient text clustering. In *SIAM International Workshop on Feature Selection for Data Mining - Interfacing Machine Learning and Statistics*, in conjunction with 2005 SIAM International Conference on Data Mining, pp. 1--10, Newport Beach, California.

Vendramin, L.; Campello, R. J. G. B. & Hruschka, E. R. (2009). On the comparison of relative clustering validity criteria. *Proceedings of the 2009 SIAM International Conference on Data Mining*, pp. 733–744.

Vendramin, L.; Campello, R. J. G. B. & Hruschka, E. R. (2010). Relative clustering validity criteria: A comparative overview. *Stat. Anal. Data Min.*, 3(4):209--235.

Vendramin, L.; Jaskowiak, P. A. & Campello, R. J. G. B. (2013). On the combination of relative clustering validity criteria. In *Proceedings of the 25th International Conference on Scientific and Statistical Database Management*, SSDBM, pp. 4:1--4:12, New York, NY, USA. ACM.

Walpole, R. E.; Myers, R. H.; Myers, S. L. & Ye, K. (2007). *Probability & statistics for engineers and scientists*. Pearson Education, Upper Saddle River, 8th edição.

Wang, X. (2015). Text document similarities based on wikipedia concept relatedness. Master's thesis, Dalhousie University, Halifax - Canada.

Wei, T.; Lu, Y.; Chang, H.; Zhou, Q. & Bao, X. (2015). A semantic approach for text clustering using wordnet and lexical chains. *Expert Systems with Applications*, 42(4):2264 – 2275.

Witten, I. H.; Frank, E. & Hall, M. A. (2011). *Data Mining: Practical Machine Learning Tools and Techniques*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 3rd edição.

Wu, X. & Kumar, V. (2009). *The Top Ten Algorithms in Data Mining*. Chapman & Hall/CRC, primeira edição.

Wu, X.; Kumar, V.; Ross Quinlan, J.; Ghosh, J.; Yang, Q.; Motoda, H.; McLachlan, G. J.; Ng, A.; Liu, B.; Yu, P. S.; Zhou, Z.-H.; Steinbach, M.; Hand, D. J. & Steinberg, D. (2007). *Top 10 Algorithms in Data Mining*, volume 14. Springer-Verlag New York, Inc., New York, NY, USA.

Wu, Z. & Palmer, M. (1994). Verbs semantics and lexical selection. In *Proceedings of the 32Nd Annual Meeting on Association for Computational Linguistics*, ACL '94, pp. 133--138, Stroudsburg, PA, USA. Association for Computational Linguistics.

Apêndice A

Thresholds para corte VAR-TFIDF

Com o objetivo de definir um *threshold* para corte VAR-TFIDF [Kogan et al., 2003], foram conduzidos experimentos de agrupamento com diferentes valores de corte. Foram utilizados quinze conjuntos de documentos (apresentados na Seção 5.1.1) sem aplicação de *stemming* e três *thresholds*. Os *thresholds* definidos selecionam termos com pontuação VAR-TFIDF acima de:

- Média das pontuações
- Média + desvio padrão das pontuações
- Média + 3× desvio padrão das pontuações

A Tabela A.1 apresenta a porcentagem de corte de termos após ser aplicada a seleção VAR-TFIDF pra cada um dos três *thresholds*.

Quando é feita uma seleção mantendo poucos atributos, corre-se o risco de que documentos não sejam representados por nenhum termo, sendo então, considerados *outliers*. Em nossos experimentos, os *outliers* foram removidos do conjuntos de documentos. A Tabela A.2 apresenta a quantidade de *outliers* removidos de cada conjunto de documentos para cada nível de *threshold*.

Os experimentos foram conduzidos utilizando a medida de similaridade cosseno nos algoritmos de agrupamento k-medóides [Krishnapuram et al., 2001], k-médias [Larose, 2005] e LDC [Nourashrafeddin et al., 2013]. Os resultados são apresentados nas Tabelas A.3, A.4 e A.5. Os resultados são dados pela média do índice ARI [Hubert & Arabie, 1985] de 15 execuções de cada configuração. Os valores destacados em negrito representam o maior valor de ARI dentre os *thresholds*.

Tabela A.1: Porcentagem de corte nos conjuntos de dados para cada *threshold* variável

Dataset	**Qtd. Termos >AVG**	**Qtd. Termos >AVG + STD**	**Qtd. Termos >AVG + 3xSTD**
20ng-subset	80,07%	93,90%	98,28%
Articles-1442-5	85,44%	97,20%	99,05%
Cbr-ilp-ir-son	78,81%	94,47%	98,62%
Cbr-ilp-ir-son-int	78,92%	94,59%	98,65%
News-10	78,35%	93,47%	98,20%
News-multi7	79,39%	93,88%	98,27%
News-multi10	79,81%	94,03%	98,26%
News-rel3	78,53%	93,95%	98,40%
News-sim3	82,95%	94,33%	98,38%
Pubmed2000sel	74,22%	91,62%	98,25%
Pubmed2000non	73,53%	91,25%	98,12%
Pubmed4000	74,63%	92,63%	98,34%
Scopus2800	75,86%	91,87%	98,15%
SMS	76,94%	93,91%	98,36%
webkb	87,91%	96,98%	99,55%

Tabela A.2: Quantidade de *outliers*

Dataset	**>AVG+3xSTD**	**>AVG+STD**	**>AVG**
20ng-subset	2	0	0
Articles-1442-5	0	0	0
Cbr-ilp-ir-son	0	0	0
Cbr-ilp-ir-son-int	0	0	0
News-10	0	0	0
News-multi7	1	0	0
News-multi10	1	0	0
News-rel3	0	0	0
News-sim3	0	0	0
Pubmed2000sel	0	0	0
Pubmed2000non	0	0	0
Pubmed4000	2	0	0
Scopus2800	2	0	0
SMS	1178	483	141
webkb	5	0	0

Ao se analisar os resultados das Tabelas A.3 e A.4 é possível perceber que, tanto para o k-médias como para o k-medóides, os melhores resultados concentram-se na coluna do *threshold* >AVG+3xSTD, ou seja, com menor quantidade de termos. Isso sustenta a teoria de que quanto maior a dimensionalidade, menor a capacidade discriminativa dos atributos.

Tabela A.3: Média ARI dos experimentos com uso da medida Cosseno no algoritmo k-médias para diferentes *thresholds*. Foram utilizados os conjuntos sem *stemming* e para cada configuração de experimento foram feitas 15 execuções.

Dataset	**>AVG+3xSTD**	**>AVG+STD**	**>AVG**
20ng-subset	0,2307	0,2149	**0,2347**
Articles-1442-5	**0,8159**	0,7918	0,7555
Cbr-ilp-ir-son	**0,7320**	0,7222	0,7250
Cbr-ilp-ir-son-int	0,7219	**0,7521**	0,7239
News-10	**0,3787**	0,3303	0,3120
News-multi7	**0,6091**	0,5545	0,6002
News-multi10	**0,4829**	0,3981	0,4674
News-rel3	**0,4690**	0,3445	0,3400
News-sim3	**0,2123**	0,1722	0,1790
Pubmed2000sel	**0,9373**	0,8980	0,8223
Pubmed2000non	0,5690	**0,5861**	0,5285
Pubmed4000	**0,5859**	0,5500	0,5443
Scopus2800	**0,7504**	0,7304	0,7490
SMS	**0,0705**	0,0591	0,0276
webkb	0,0100	0,0144	**0,0177**

Tabela A.4: Média ARI dos experimentos com uso da medida Cosseno no algoritmo k-medóides para diferentes *thresholds*. Foram utilizados os conjuntos sem *stemming* e para cada configuração de experimento foram feitas 15 execuções.

Dataset	**>AVG+3xSTD**	**>AVG+STD**	**>AVG**
20ng-subset	0,0958	0,0991	**0,0995**
Articles-1442-5	0,7545	0,7112	**0,8372**
Cbr-ilp-ir-son	**0,6771**	0,6551	0,6043
Cbr-ilp-ir-son-int	**0,7120**	0,6727	0,6768
News-10	0,2275	**0,3051**	0,1566
News-multi7	0,3022	0,3013	**0,3064**
News-multi10	0,1338	0,1701	**0,1911**
News-rel3	**0,1952**	0,1905	0,1526
News-sim3	**0,1395**	0,0619	0,0966
Pubmed2000sel	**0,6046**	0,5873	0,4720
Pubmed2000non	**0,4209**	0,2762	0,3268
Pubmed4000	**0,3600**	0,3277	0,3431
Scopus2800	**0,4148**	0,3753	0,4129
SMS	**0,0955**	0,0615	0,0014
webkb	0,0045	0,0091	**0,0109**

Por outro lado, na Tabela A.5, que apresenta os resultados obtidos pelo algoritmo LDC, os melhores índices estão na coluna de *threshold* > AVG, ou seja, maior quantidade de termos. Isso pode ser justificado pelo fato de que o algoritmo LDC já possui uma etapa de seleção de termos. Após realizar o agrupamento de termos, o LDC remove os termos que possuem pontuação VAR-TFIDF abaixo da média de cada grupo. Por essa razão, fornecer uma base com dimensionalidade muito baixa

Tabela A.5: Média ARI dos experimentos com o algoritmo LDC para diferentes *thresholds*. Foram utilizados os conjuntos sem *stemming* e para cada configuração de experimento foram feitas 15 execuções.

Dataset	**>AVG+3xSTD**	**>AVG+STD**	**>AVG**
20ng-subset	0,3020	0,3503	**0,4065**
Articles-1442-5	0,8232	0,8489	**0,9203**
Cbr-ilp-ir-son	0,7212	0,7402	**0,7863**
Cbr-ilp-ir-son-int	0,6893	0,7055	**0,7608**
News-10	0,3743	0,3855	**0,4743**
News-multi7	0,5012	0,5214	**0,6051**
News-multi10	0,4256	0,4454	**0,554**
News-rel3	0,3221	0,3692	**0,4001**
News-sim3	0,3513	0,3719	**0,4261**
Pubmed2000sel	0,8750	0,8841	**0,9182**
Pubmed2000non	0,4198	0,4251	**0,4296**
Pubmed4000	0,4313	**0,4641**	0,4484
Scopus2800	0,7454	0,7528	**0,8107**
SMS	0,2013	0,2123	**0,3696**
webkb	0,0199	0,0136	**0,1107**

força o LDC a cortar ainda mais termos, causando a remoção de termos que possam ser importantes. Por essa razão, nos experimentos utilizando LDC nesse trabalho, foi adotado o *threshold* >AVG.

Apêndice B

Agrupamento léxico com e sem *stemming*

Ao utilizar medidas de similaridade que não acessam conhecimento externo, é necessário aplicar técnicas para contornar problemas na interpretação da linguagem, como variações morfológicas e termos sinônimos. Nesse cenário, técnicas de *stemming* são amplamente empregadas [Rezende et al., 2011]. *Stemming* é o processo de redução de uma palavra à sua raiz. Com esse processo, palavras como "cachorro", "cachorra", "cachorrinho" e "cachorrão" seriam todas representadas pelo mesmo radical, "cachor".

Experimentos utilizando bases com e sem *stemming* foram conduzidos no algoritmo LDC [Nourashrafeddin et al., 2013] original. O algoritmo LDC utiliza a medida cosseno na tarefa de agrupamento de termos e a distância euclidiana ao agrupar documentos. A Tabela B.1 apresenta as médias ARI [Hubert & Arabie, 1985] de 15 execuções do algoritmo LDC para cada configuração em 15 conjuntos de documentos com e sem *stemming*. Os melhores índices são destacados em negrito.

Como é possível ver na Tabela B.1, o pré-processamento sem *stemming* obteve melhores médias em 12 dos 15 conjuntos de documentos. Para muitos conjuntos as médias ficaram próximas. Como o objetivo é utilizar o algoritmo com a melhor configuração possível para comparar com as técnicas propostas nessa dissertação, os conjuntos de documentos foram pré-processados sem aplicação de *stemming*, neste trabalho.

Tabela B.1: Média ARI de 15 execuções do algoritmo LDC em 15 conjuntos de documentos com e sem aplicação de *stemming*.

Dataset	**com *stemming***	**sem *stemming***
20ng-subset	0,4033	**0,4065**
Articles-1442-5	**0,9515**	0,9203
Cbr-ilp-ir-son	0,7502	**0,7863**
Cbr-ilp-ir-son-int	0,7115	**0,7608**
News-10	0,4699	**0,4743**
News-multi7	0,5942	**0,6051**
News-multi10	**0,5775**	0,5540
News-rel3	0,4385	**0,4490**
News-sim3	0,3734	**0,4261**
Pubmed2000sel	0,9182	**0,9324**
Pubmed2000non	0,4138	**0,4296**
Pubmed4000	**0,4522**	0,4484
Scopus2800	0,8046	**0,8107**
SMS	0,3696	**0,4262**
webkb	0,0987	**0,1107**

Apêndice C

Agrupamento semântico com centróides ou medóides

A medida GTM [Islam et al., 2012] é uma medida de similaridade semântica que calcula a similaridade entre documentos baseado na similaridade entre termos. Nessa medida, apenas a ocorrência ou ausência de termos no documento são considerados. No algoritmo de agrupamento LDC [Nourashrafeddin et al., 2013], após encontrar os grupos de documentos semente dos grupos de termos, o LDC calcula os centróides, ponderando cada documento semente pela importância dele naquele grupo. Por mais que um centróide não seja um documento do conjunto de dados, é possível calcular a distância GTM ou FGTM até ele, pois ele pode ser considerado um pseudo-documento. Porém, quando se calcula a média de um conjunto de documentos, o centróide resultante contém valores não nulos para todos os termos ocorrentes em todos os documentos semente. Todos esses termos terão o mesmo peso no cálculo de similaridade GTM, o que pode ser prejudicial para a medida. Como alternativa a isso, pode-se usar medóides como representantes dos grupos. Medóide é o elemento mais central contido em um grupo.

A fim de verificar o comportamento da medida GTM com o uso de centróides ou medóides no algoritmo LDC, diversos experimentos foram conduzidos. Nos experimentos baseados em medóides, os documentos sementes são considerados medóides, onde eles são selecionados em ordem de centralidade, ou seja, quanto mais central no grupo de documentos semente, mais apto para ser um medóide. Foram utilizadas diferentes quantidades de medóides, sendo elas: 1, 10, 30, 50 e todos os documentos do grupo de documentos semente. Por exemplo, quando apenas 1 medóide é utilizado, dado um documento d_i e um grupo de documentos semente DS_j,

onde $d_i \in DS_j$, se d_i é o documento mais central[1] do grupo DS_j, a distância de qualquer documento d_p para o grupo DS_j é dado pela distância de d_i até d_p. Já quando se utiliza mais medóides (chamaremos a quantidade de m), encontra-se os m documentos mais centrais do grupo DS_j (chamaremos esse grupo de MD_j), onde a distância de um objeto d_p para o grupo DS_j é dado por:

$$dist(d_p, DS_j) = \frac{\sum_{i=1}^{m} dist(d_p, md_i)}{m} \quad \forall md \in MD_j \tag{C.1}$$

onde foi considerado que se $m > |DS_j|$, então $m \leftarrow |DS_j|$.

As Tabelas C.1 e C.2 apresentam as médias ARI [Hubert & Arabie, 1985] de 15 execuções do algoritmo LDC para cada configuração de experimento com as medidas GTM e FGTM, respectivamente. Os melhores índices estão destacados em negrito.

Tabela C.1: Média ARI de 15 execuções do algoritmo LDC com a medida GTM utilizando centróides e 1, 10, 30, 50 ou todos os documentos semente de um grupo como medóides.

Dataset	**1**	**10**	**30**	**50**	**Todos**	**Centróides**
20ng-subset	0,0163	0,0339	0,0465	0,0517	0,1676	**0,3514**
Articles-1442-5	**0,9493**	0,9402	0,9355	0,9277	0,9198	0,9258
Cbr-ilp-ir-son	0,1361	0,3611	0,4972	0,5758	0,5295	**0,7046**
Cbr-ilp-ir-son-int	0,1942	0,3938	0,5335	0,5792	0,6349	**0,6949**
News-10	0,2262	0,4779	0,5854	0,6003	**0,6161**	0,4336
News-multi7	0,0738	0,1983	0,2239	0,2404	0,4432	**0,6263**
News-multi10	0,0291	0,0941	0,1249	0,1292	0,3423	**0,5345**
News-rel3	0,0111	0,0645	0,1106	0,1418	0,3284	**0,4519**
News-sim3	0,0071	0,0364	0,0525	0,0891	0,2540	**0,3633**
Pubmed2000sel	0,7057	0,8601	0,9208	**0,9391**	0,9005	0,6473
Pubmed2000non	0,2458	0,3784	0,4276	0,4217	**0,4376**	0,3372
Pubmed4000	0,2901	0,4295	0,4707	0,4777	**0,5124**	0,3526
Scopus2800	0,5488	0,6732	0,6564	0,6958	0,7809	**0,8382**
SMS	0,0534	0,2931	0,2657	**0,3551**	0,1358	0,0712
webkb	-0,0022	0,0232	0,0404	0,0528	0,0399	**0,0597**

Observando os resultados da Tabela C.1 percebe-se que, embora o uso de medóides com a medida GTM supere o uso de centróides em alguns casos, na maioria dos conjuntos de dados os centróides se comportam melhor. Também é possível notar que quantidades baixas de medóides tendem a ser piores do que o uso de

[1]Documento mais central é o que minimiza o somatório das distâncias para todos os outros documentos do grupo

Tabela C.2: Média ARI de 15 execuções do algoritmo LDC com a medida FGTM utilizando centróides e 1, 10, 30, 50 ou todos os documentos semente de um grupo como medóides.

Dataset	**1**	**10**	**30**	**50**	**Todos**	**Centróides**
20ng-subset	0,0192	0,0425	0,0580	0,0622	0,2094	**0,2455**
Articles-1442-5	0,5356	0,8480	**0,9613**	0,9303	0,9424	0,9433
Cbr-ilp-ir-son	0,2167	0,3632	0,4778	0,5234	0,5911	**0,8254**
Cbr-ilp-ir-son-int	0,1897	0,4487	0,5640	0,5600	0,6545	**0,7749**
News-10	0,0798	0,2834	0,4277	0,4948	0,5003	**0,5866**
News-multi7	0,0549	0,1508	0,1663	0,1801	0,3977	**0,6685**
News-multi10	0,0380	0,0994	0,1072	0,1129	0,3276	**0,5581**
News-rel3	0,0327	0,0810	0,0881	0,1150	0,2775	**0,4315**
News-sim3	0,0101	0,0115	0,0254	0,0399	0,1427	**0,4405**
Pubmed2000sel	0,3533	0,6629	0,7364	0,7233	0,7766	**0,9289**
Pubmed2000non	0,1335	0,3627	0,3338	0,3207	0,3836	**0,5081**
Pubmed4000	0,1647	0,3272	0,3218	0,3312	0,3973	**0,5519**
Scopus2800	0,3691	0,5965	0,6308	0,6112	0,7660	**0,8483**
SMS	0,1943	0,3873	0,3861	**0,4142**	0,3474	0,0901
webkb	0,0021	0,0012	0,0079	0,0236	0,0605	**0,1321**

grandes quantidades, isso porque um ou poucos medóides dificilmente contém termos suficientes para representar os objetos de todo o grupo. Pelos resultados não é possível concluir que o uso de centróides ou medóides sejam sempre indicados para a medida GTM. No entanto, como o uso de centróides supera o uso de medóides para a maioria dos casos estudados, foram utilizados centróides com a medida GTM neste trabalho.

Quando a análise é feita na Tabela C.2, que utiliza a medida FGTM, fica nítido que o uso de centróides supera o uso de medóides. Isso porque a medida FGTM é sensível à frequência em que os termos aparecem nos documentos, logo, utilizar uma grande quantidade de termos com baixa frequência não é prejudicial à medida.

Embora cada conjunto de documentos possua características diferentes, se plotarmos um gráfico com a média dos resultados das medidas de similaridade para cada configuração do algoritmo sobre todos os conjuntos de documentos, podemos perceber o comportamento do uso de diferentes quantidades de medóides e o uso de centróides. A Figura C.1 ilustra isso. Cada ponto no gráfico representa a média dos resultados da medida de similaridade sobre todos os conjuntos de documentos.

Olhando o gráfico ilustrado na Figura C.1, fica nítido que, quanto maior a quantidade de medóides, melhor o índice obtido. Porém, mesmo utilizando todos os documentos como medóides, os resultados não superam o uso de centróide. Além

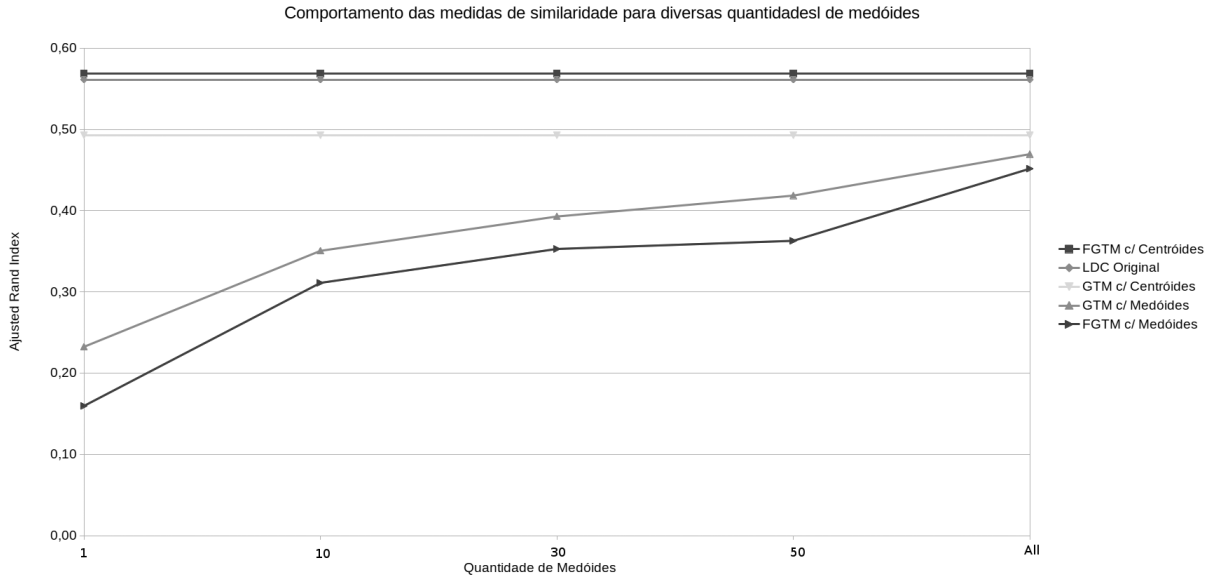

Figura C.1: Comparação do comportamento das medidas de similaridade para diversas quantidades de medóides. Os resultados são a média obtida considerando todos os conjuntos de documentos. As três linhas mais de cima são as execuções utilizando centróides, logo, não variam.

dos resultados aqui mostrados, a figura também ilustra o resultado obtido pelo algoritmo LDC original [Nourashrafeddin et al., 2013], que utiliza centróides e distância euclidiana. A média da medida FGTM supera a distância euclidiana nessa comparação.

Apêndice D

Teste de agrupamento por colunas

Conforme mostrado no Capítulo 6, para a maioria dos conjuntos de documentos, quando agrupados em algoritmos baseados em K-médias, não é possível chegar ao modelo (classes) conhecido. Como a maioria das heurísticas de identificação do número K são baseadas no algoritmo K-médias, ou o utilizam para refinar os grupos, é importante que o modelo dos conjuntos possuam uma estrutura de grupos coesa. Isso porque o K-médias é aplicável a grupos bem comportados, que tenham dados hiperesféricos ou levemente alongados. Também é necessário que os grupos sejam separados e bem definidos, caso contrário, o K-médias pode não ter a percepção dos grupos desejados. Alguns algoritmos, como o LDC [Nourashrafeddin et al., 2013], possuem duas fases de agrupamento. Primeiro são agrupados os termos (colunas da matriz de documentos-termos) e depois os documentos (linhas da matriz de documentos-termos). Os experimentos conduzidos na Seção 6.3 analisaram a estrutura dos conjuntos para a etapa de agrupamento dos documentos. Aqui são conduzidos experimentos para verificar se a estrutura dos grupos de termos são coesos para valores de K próximos ao número de grupos conhecidos do conjunto de documentos.

O algoritmo LDC recebe o parâmetro K do usuário e agrupa os termos utilizando o algoritmo *fuzzy c-means*. Os grupos de termos obtidos são utilizados para identificar os *documentos semente* e, posteriormente, agrupar os documentos. A verificação da estrutura do agrupamento de termos é conduzida a fim de verificar a possibilidade de utilização de heurísticas para identificação do número K de grupos no algoritmo LDC. O uso de heurísticas para agrupamento de termos no LDC tornaria o algoritmo totalmente não-supervisionado, ou seja, sem que haja a necessidade do usuário fornecer o número K.

Como não existe um agrupamento modelo para termos, como é o caso das clas-

ses nos documentos, foram conduzidos agrupamentos incrementais para o número de K, onde, para cada iteração, a silhueta simplificada é calculada. Esse método incremental é conhecido como *Ordered Multiple runs of K-means* (OMRK) [Jain et al., 1999]. Para cada conjunto de documentos, foi executado o algoritmo K-médias e *fuzzy c-means*, com quantidade de grupos partindo de $K = 2$ até $K = 25$. Como o K-médias e *fuzzy c-means* são sensíveis à escolha dos protótipos iniciais, os algoritmos foram executados 3 vezes para cada valor de K. Após cada execução, o índice de silhueta é calculado, onde, ao final das 3 execuções, apenas o melhor índice é armazenado. Para o algoritmo K-médias, o índice de Silhueta Simplificada foi adotado (apresentado na Seção 3.5.2.2). No algoritmo *fuzzy c-means*, foi adotado o índice de silhueta simplificada *fuzzy* (apresentado na Seção 3.5.2.3). A similaridade cosseno foi utilizada em ambos os algoritmos, por ser a mesma utilizada na fase de agrupamento de termos do algoritmo LDC [Nourashrafeddin et al., 2013]. Os resultados para cada um dos 18 conjuntos de documentos (descritos na Seção 5.1.1) são apresentados pelos gráficos nas Figuras D.1 até D.18.

Os resultados são similares para todos os conjuntos de documentos. A silhueta tem um comportamento crescente, à medida que o valor de K também cresce. Somente em 7 dos 36 gráficos, os melhores valores de silhueta não foram para $K = 25$. Ainda assim, o melhor índice de silhueta nesses gráficos ficaram distantes do índice obtido pelos agrupamentos com valor K igual à quantidade de classes dos conjuntos. A silhueta superestima a quantidade de grupos quando diversos objetos estão espalhados no espaço, não havendo formação de grupos coesos. Isso mostra que as estruturas dos grupos de termos desses conjuntos de documentos não são coesas para valores de K próximos ao número de classes, quando agrupados pelos algoritmos K-médias e *fuzzy c-means*. Desta forma, as heurísticas apresentadas no Capítulo 6 não são apropriadas para a identificação do número K de grupos no algoritmo LDC. Os resultados abrem caminho para que outras técnicas sejam investigadas a fim de estimar o número de grupos em conjuntos de documentos textuais. Essa é uma das propostas de trabalhos futuros.

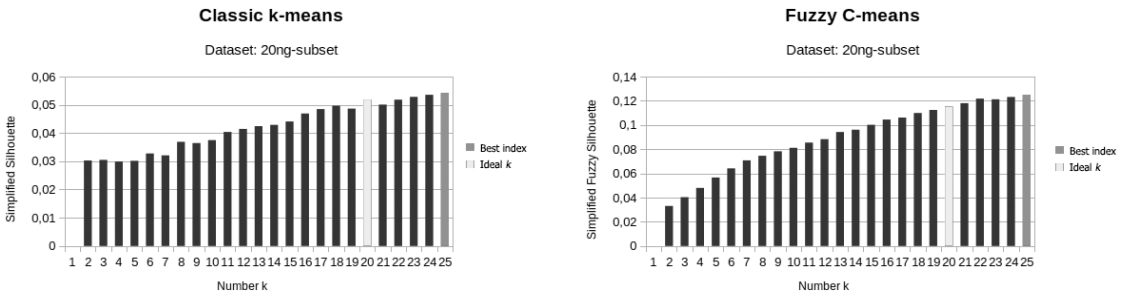

Figura D.1: Conjunto 20ng-subset. Gráficos de execução do $OMRK$ para $K = 2$ até $K = 25$. O gráfico da esquerda ilustra os índices de silhueta simplificada no algoritmo k-médias. O gráfico da direita ilustra os índices de silhueta simplificada *fuzzy* no algoritmo *fuzzy c-means*.

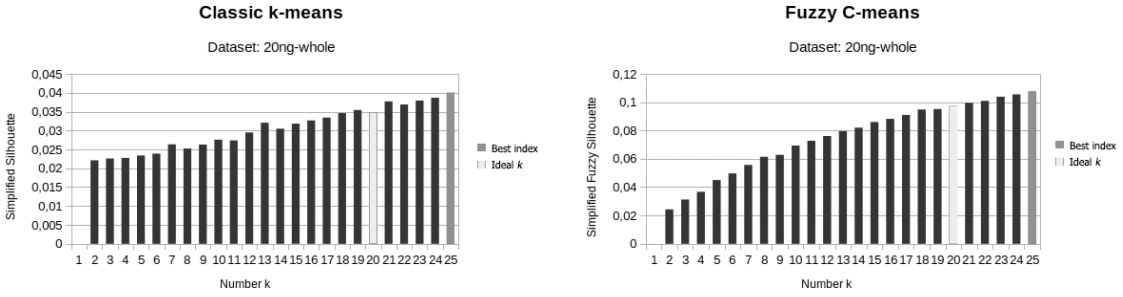

Figura D.2: Conjunto 20ng-whole. Gráficos de execução do $OMRK$ para $K = 2$ até $K = 25$. O gráfico da esquerda ilustra os índices de silhueta simplificada no algoritmo k-médias. O gráfico da direita ilustra os índices de silhueta simplificada *fuzzy* no algoritmo *fuzzy c-means*.

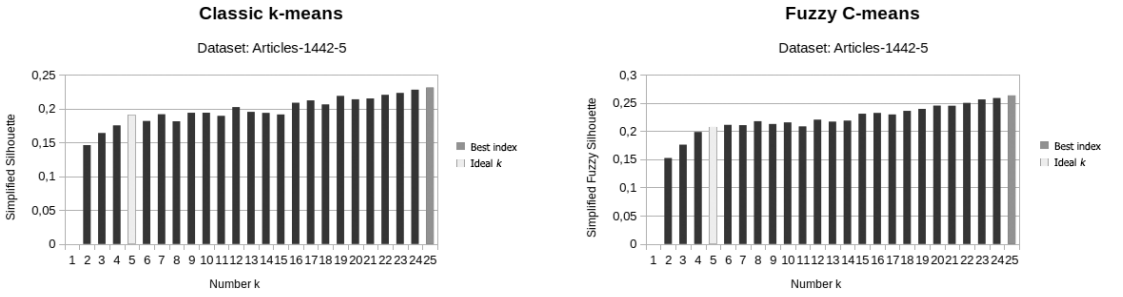

Figura D.3: Conjunto Articles-1442-5. Gráficos de execução do $OMRK$ para $K = 2$ até $K = 25$. O gráfico da esquerda ilustra os índices de silhueta simplificada no algoritmo k-médias. O gráfico da direita ilustra os índices de silhueta simplificada *fuzzy* no algoritmo *fuzzy c-means*.

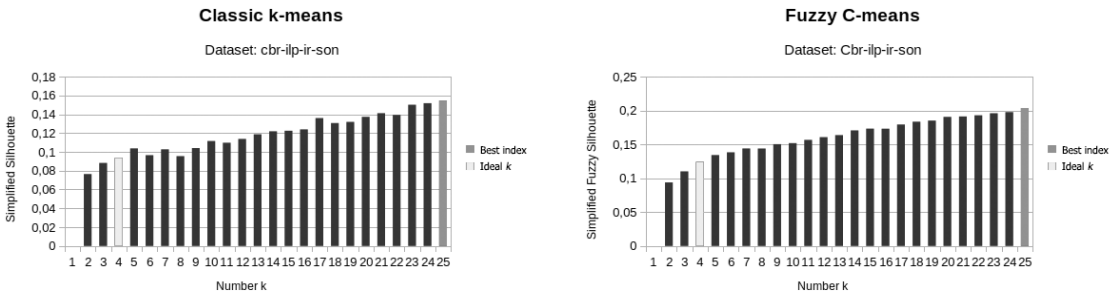

Figura D.4: Conjunto cbr-ilp-ir-son. Gráficos de execução do $OMRK$ para $K = 2$ até $K = 25$. O gráfico da esquerda ilustra os índices de silhueta simplificada no algoritmo k-médias. O gráfico da direita ilustra os índices de silhueta simplificada *fuzzy* no algoritmo *fuzzy c-means*.

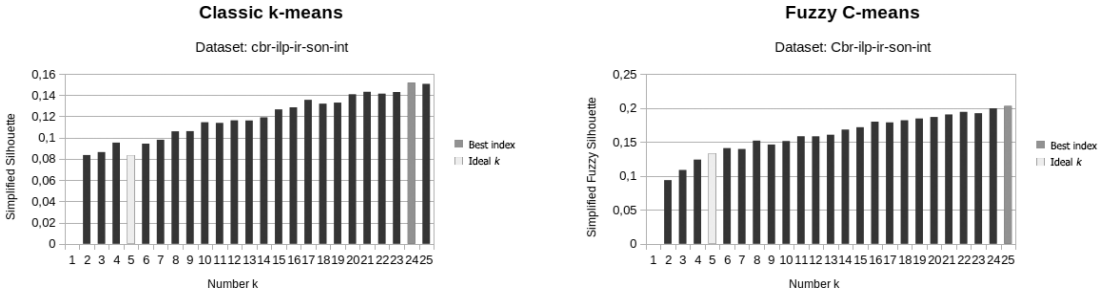

Figura D.5: Conjunto cbr-ilp-ir-son-int. Gráficos de execução do $OMRK$ para $K = 2$ até $K = 25$. O gráfico da esquerda ilustra os índices de silhueta simplificada no algoritmo k-médias. O gráfico da direita ilustra os índices de silhueta simplificada *fuzzy* no algoritmo *fuzzy c-means*.

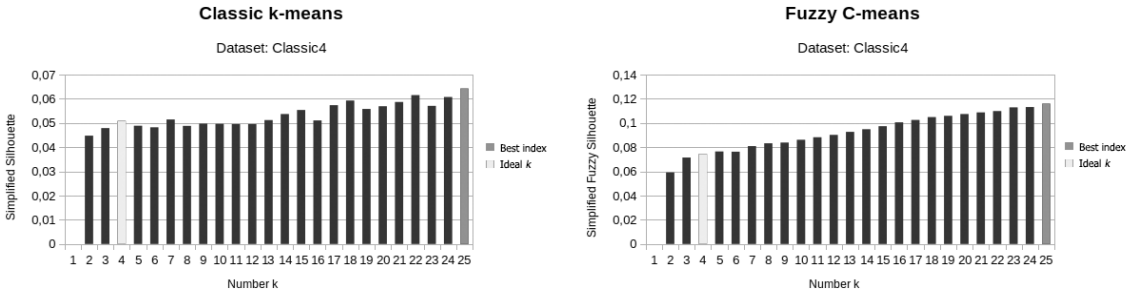

Figura D.6: Conjunto Classic4. Gráficos de execução do $OMRK$ para $K = 2$ até $K = 25$. O gráfico da esquerda ilustra os índices de silhueta simplificada no algoritmo k-médias. O gráfico da direita ilustra os índices de silhueta simplificada *fuzzy* no algoritmo *fuzzy c-means*.

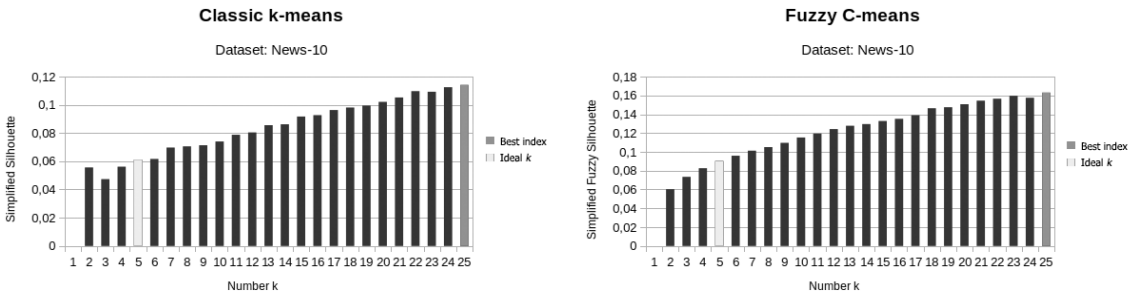

Figura D.7: Conjunto News-10. Gráficos de execução do $OMRK$ para $K = 2$ até $K = 25$. O gráfico da esquerda ilustra os índices de silhueta simplificada no algoritmo k-médias. O gráfico da direita ilustra os índices de silhueta simplificada *fuzzy* no algoritmo *fuzzy c-means*.

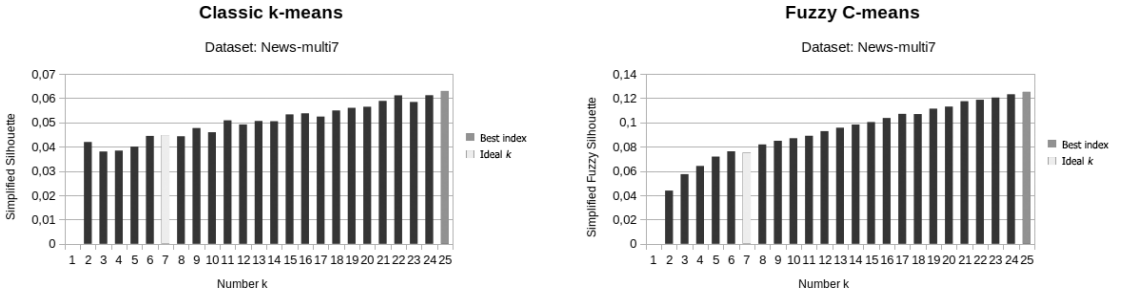

Figura D.8: Conjunto News-multi7. Gráficos de execução do $OMRK$ para $K = 2$ até $K = 25$. O gráfico da esquerda ilustra os índices de silhueta simplificada no algoritmo k-médias. O gráfico da direita ilustra os índices de silhueta simplificada *fuzzy* no algoritmo *fuzzy c-means*.

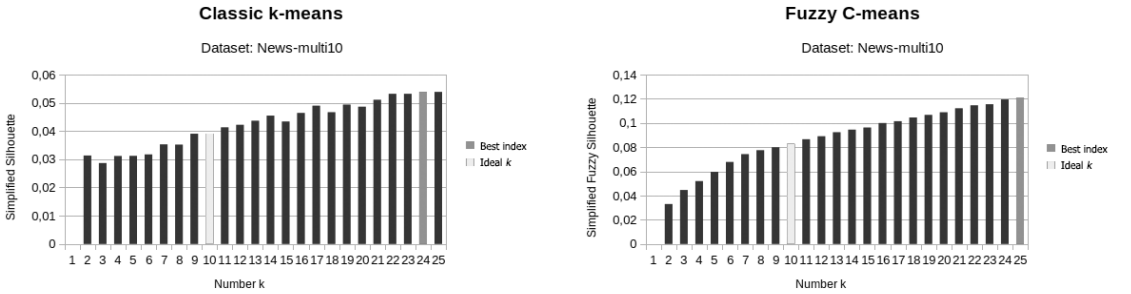

Figura D.9: Conjunto News-multi10. Gráficos de execução do $OMRK$ para $K = 2$ até $K = 25$. O gráfico da esquerda ilustra os índices de silhueta simplificada no algoritmo k-médias. O gráfico da direita ilustra os índices de silhueta simplificada *fuzzy* no algoritmo *fuzzy c-means*.

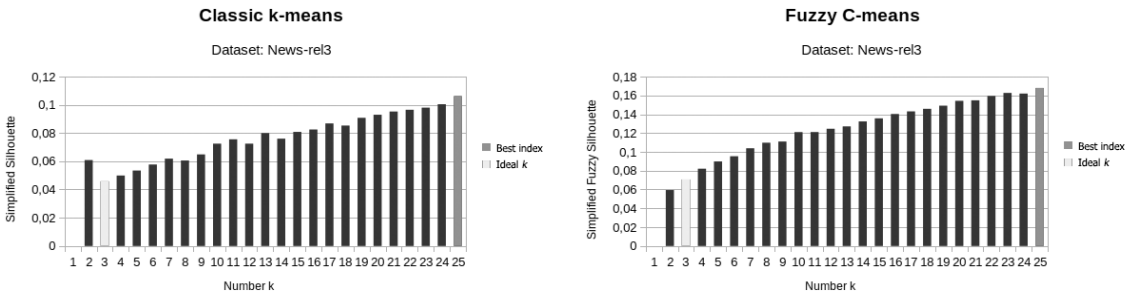

Figura D.10: Conjunto News-rel3. Gráficos de execução do $OMRK$ para $K = 2$ até $K = 25$. O gráfico da esquerda ilustra os índices de silhueta simplificada no algoritmo k-médias. O gráfico da direita ilustra os índices de silhueta simplificada *fuzzy* no algoritmo *fuzzy c-means*.

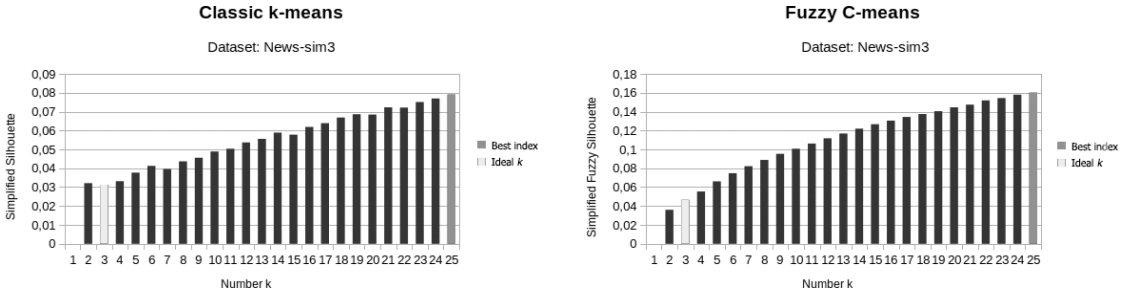

Figura D.11: Conjunto News-sim3. Gráficos de execução do $OMRK$ para $K = 2$ até $K = 25$. O gráfico da esquerda ilustra os índices de silhueta simplificada no algoritmo k-médias. O gráfico da direita ilustra os índices de silhueta simplificada *fuzzy* no algoritmo *fuzzy c-means*.

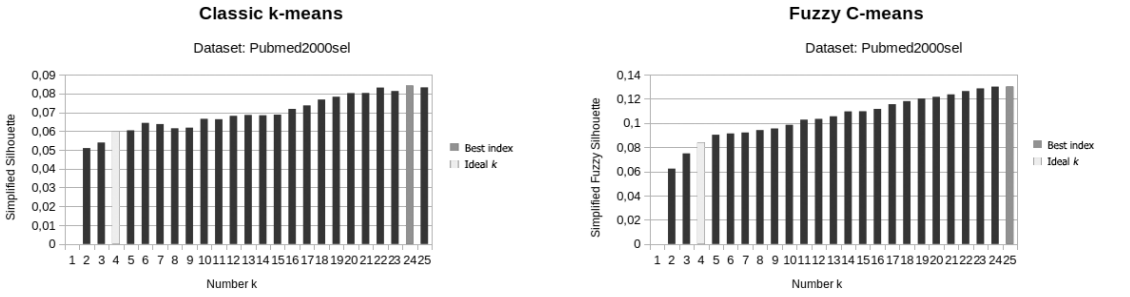

Figura D.12: Conjunto Pubmed2000sel. Gráficos de execução do $OMRK$ para $K = 2$ até $K = 25$. O gráfico da esquerda ilustra os índices de silhueta simplificada no algoritmo k-médias. O gráfico da direita ilustra os índices de silhueta simplificada *fuzzy* no algoritmo *fuzzy c-means*.

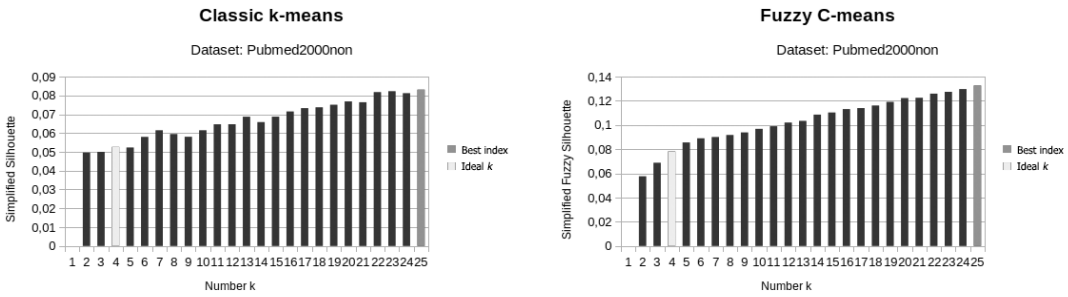

Figura D.13: Conjunto Pubmed2000non. Gráficos de execução do $OMRK$ para $K = 2$ até $K = 25$. O gráfico da esquerda ilustra os índices de silhueta simplificada no algoritmo k-médias. O gráfico da direita ilustra os índices de silhueta simplificada *fuzzy* no algoritmo *fuzzy c-means*.

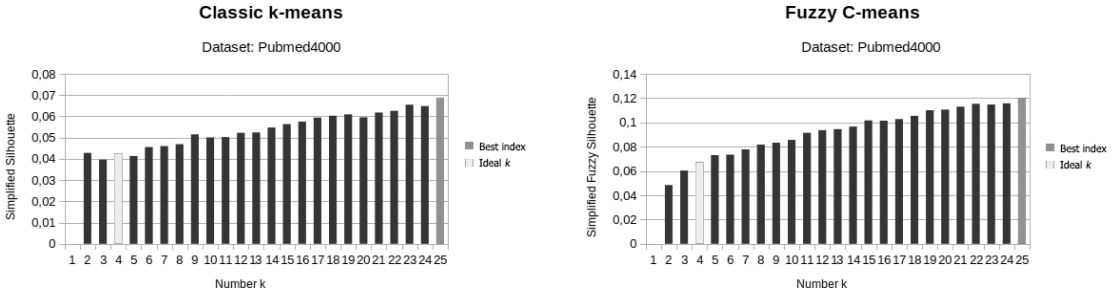

Figura D.14: Conjunto Pubmed4000. Gráficos de execução do $OMRK$ para $K = 2$ até $K = 25$. O gráfico da esquerda ilustra os índices de silhueta simplificada no algoritmo k-médias. O gráfico da direita ilustra os índices de silhueta simplificada *fuzzy* no algoritmo *fuzzy c-means*.

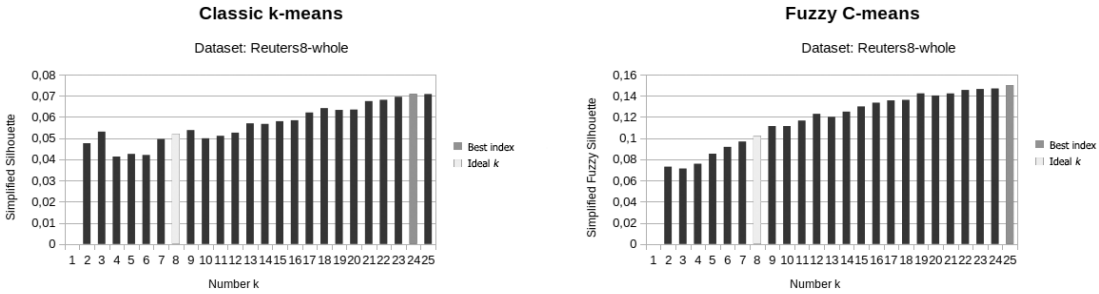

Figura D.15: Conjunto Reauters8-whole. Gráficos de execução do $OMRK$ para $K = 2$ até $K = 25$. O gráfico da esquerda ilustra os índices de silhueta simplificada no algoritmo k-médias. O gráfico da direita ilustra os índices de silhueta simplificada *fuzzy* no algoritmo *fuzzy c-means*.

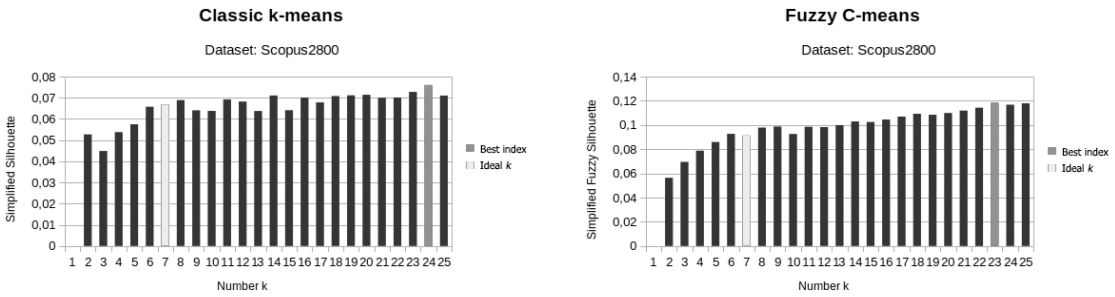

Figura D.16: Conjunto Scopus2800. Gráficos de execução do $OMRK$ para $K = 2$ até $K = 25$. O gráfico da esquerda ilustra os índices de silhueta simplificada no algoritmo k-médias. O gráfico da direita ilustra os índices de silhueta simplificada *fuzzy* no algoritmo *fuzzy c-means*.

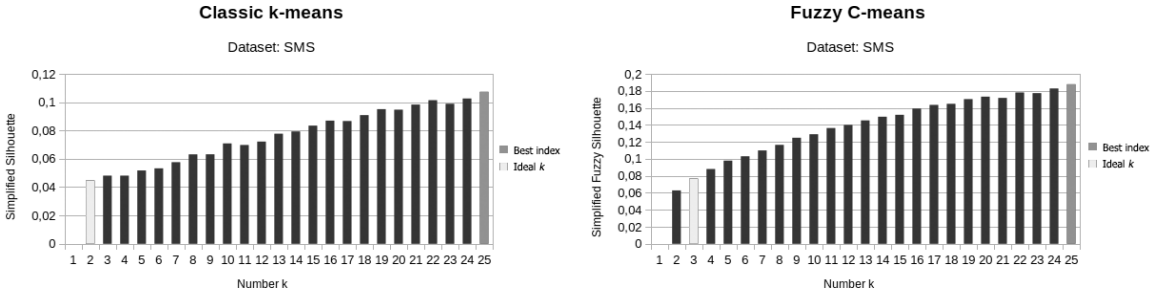

Figura D.17: Conjunto SMS. Gráficos de execução do $OMRK$ para $K = 2$ até $K = 25$. O gráfico da esquerda ilustra os índices de silhueta simplificada no algoritmo k-médias. O gráfico da direita ilustra os índices de silhueta simplificada *fuzzy* no algoritmo *fuzzy c-means*.

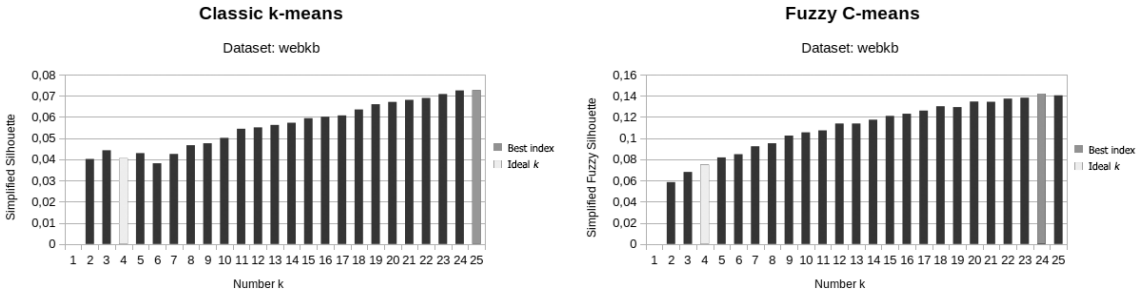

Figura D.18: Conjunto webkb. Gráficos de execução do $OMRK$ para $K = 2$ até $K = 25$. O gráfico da esquerda ilustra os índices de silhueta simplificada no algoritmo k-médias. O gráfico da direita ilustra os índices de silhueta simplificada *fuzzy* no algoritmo *fuzzy c-means*.