Unsupervised Machine Learning Based on Heterogeneous Networks for Text Clustering

José Vitor Gualdi dos Santos[1], **Rafael Geraldeli Rossi**[1]

[1]Universidade Federal de Mato Grosso do Sul (UFMS)
Caixa Postal 210, CEP 79620-080 – Três Lagoas – MS – Brazil

jvgualdi1@gmail.com, rafael.g.rossi@ufms.br

Resumo. *As representações em redes permitem modelar diferentes tipos de relações entre textos, são capazes de capturar padrões dificilmente capturados pelo modelo espaço vetorial, e algoritmos de agrupamento em redes, como a propagação de rótulos, possuem complexidade linear. Porém, o agrupamento em redes não tem sido explorado (i) especificamente no agrupamento de textos; e (ii) com as diferentes possibilidades de representar textos em redes. Com isso, o objetivo deste artigo é a exploração e análise de técnicas de agrupamento aplicadas a diferentes tipos de representações em redes. Foram realizados experimentos utilizando 30 coleções de diferentes domínios, representadas em formato bag-of-words, redes de similaridade do tipo k-Nearest Neighbors e redes bipartidas. A abordagem de propagação de rótulos em redes de similaridade obteve os melhores resultados para a maioria das medidas de avaliação e para a maioria das coleções de textos. O algoritmo de propagação de rótulos demonstrou-se promissor, principalmente quando aplicado a redes de similaridade.*
Palavras-chaves*: agrupamento de textos; representações em redes; propagação de rótulos.*

Abstract. *Network-based representations can model different types of relationships between texts, they are capable of capturing patterns that are hardly captured by vector space model, and network-based clustering algorithms, such as label propagation, have linear complexity. However, network-based clustering has not been explored (i) specifically in clustering texts; and (ii) with different possibilities of text representations on networks. Thus, this article's objective is to explore and analyze clustering techniques applied to different types of network representations. Experiments were performed using 30 collections from different domains, represented in the bag-of-words format, similarity networks of the type k-Nearest Neighbors, and bipartite networks. The label propagation approach applied in similarity networks presented the best results for most evaluation measures and most text collections.*
Keywords*: text clustering; network-based representations; label propagation.*

1. Introdução

Grande parte dos dados digitais produzido está no formato textual, como *e-mails*, conteúdos de páginas *web*, e artigos. Porém, a análise e extração os conhecimentos embutidos nos textos é humanamente inviável devido o esforço e tempo necessários. Logo,

a automatização de tarefas e extração de conhecimento se torna necessário, e para isso, pode-se utilizar técnicas da área de Mineração de Textos [Aggarwal 2018].

Uma dessas técnicas é o agrupamento de textos [Aggarwal 2018]. O objetivo do agrupamento é separar os documentos em grupos de forma que todos documentos no mesmo grupo são mais similares entre si do que aos documentos de outros grupos. Sendo assim, propõe-se extrair grupos de mesmo tema ou assunto sem necessidade de interferência do usuário, uma vez que geralmente o agrupamento é feito de maneira não supervisionada. Vale ressaltar que além da aplicação direta do agrupamento de textos para organização e gerenciamento de coleções de textos em grupos temáticos, o agrupamento de textos é utilizado para diversas outras finalidades e aplicações como a organização de resultados de busca [Khennak et al. 2019], seleção de exemplos para o aprendizado ativo [Ienco et al. 2013], e aprendizado baseado em uma única classe [Golo and Rossi 2019].

Para execução dos algoritmos de agrupamento, é necessário que os textos sejam representados em um formato estruturado para que os algoritmos possam processá-los. Normalmente essa representação é feita no modelo espaço vetorial (MEV) no qual os documentos são representados por vetores, e suas dimensões correspondem geralmente aos termos da coleção, e o valor de cada dimensão corresponde ao peso do termo no documento por exemplo, a frequência do termo no documento [Aggarwal 2018].

Algoritmos de agrupamento que implementam suas estratégias baseadas no MEV normalmente obtêm grupos de formatos pré-definidos, o que pode fazer com que documentos que pertençam a um mesmo tema ou tópico sejam atribuídos a grupos distintos, ou ainda documentos de temas ou tópicos distintos sejam alocados no mesmo grupo [Tan et al. 2019]. Outros algoritmos baseados em MEV quando não apresentam tal caraterística, possuem complexidade de tempo e espaço cúbicas, fazendo com que sejam opções inviáveis para grandes conjuntos de dados [Aggarwal 2018].

As coleções textuais também podem ser representadas utilizando redes [Rossi et al. 2016]. As redes podem ser utilizadas para representar coleções de textos por meio da utilização de diferentes tipos de objetos e tipos de relações entre os objetos. As representações em rede permitem a extração de padrões e agrupamentos em formatos que dificilmente são captados por algoritmos baseados em MEV [Breve et al. 2011]. Por conta disso, algoritmos baseados em redes vêm sendo utilizados com sucesso na literatura [Mei et al. 2019, Rossi et al. 2016, Sun et al. 2009]. Porém, o seu impacto na performance de agrupamento de textos ainda não foi amplamente explorado.

Dada essa lacuna, o objetivo deste trabalho foi a realização de uma avaliação empírica extensa de forma a comparar algoritmos de aprendizado não supervisionado baseados no MEV e algoritmos baseados em redes, bem como em diferentes tipos de representações em redes. Com base nos resultados, demonstrou-se que a técnica de propagação de rótulos em redes, a qual possui complexidade de tempo linear, obteve os melhores resultados para diferentes medidas de qualidade de agrupamento.

O restante deste artigo está dividido da seguinte forma. Na Seção 2 são apresentados os conceitos necessários para o entendimento deste trabalho. Na Seção 3, são apresentados os detalhes do método de pesquisa utilizado. Na Seção 4 são apresentados os resultados e discussões. Por fim, na Seção 5 são apresentadas as considerações finais e trabalhos futuros.

2. Conceitos

Nesta seção são apresentados conceitos e trabalhos relacionados a este artigo, como pré-processamentos e estruturas utilizadas para representação de textos, além dos algoritmos de agrupamento de textos.

2.1. Representações Estruturadas

Para que as coleções textuais possam ser interpretadas pelos algoritmos de aprendizado de máquina, é preciso estruturá-las. Assume-se que a coleção é composta por n documentos, $\mathcal{D} = \{d_1, d_2, \ldots, d_n\}$, e por m termos, $\mathcal{T} = \{t_1, t_2, \ldots, t_m\}$.

Uma das formas mais utilizadas de representação de textos é o modelo espaço vetorial (MEV). No MEV, os textos são representados como vetores nos quais as dimensões correspondem às características dos textos. Em sua forma mais simples, as características correspondem às palavras simples, formando assim a representação denominada *bag-of-words*. A representação *bag-of-words* é caracterizada por ter alta dimensionalidade e esparsidade, uma vez que existem muitas palavras distintas em uma coleção de textos e em sua grande maioria estão presentes apenas em uma pequena parcela de documentos [Aggarwal 2018]. Alguns pré-processamentos dos textos são realizados para diminuir o número de termos e melhorar os resultados das técnicas de mineração de textos, como a padronização de caixas, radicalização das palavras e remoção de *stopwords* [Aggarwal 2018].

Os textos interpretados pelos algoritmos também podem ser representados no formato de redes [Rossi et al. 2015, Mihalcea and Radev 2011]. Formalmente uma rede pode ser definida como uma tripla $N = \langle \mathcal{O}, \mathcal{R}, \mathcal{W} \rangle$, na qual $\mathcal{O}$ representa o conjunto de objetos da rede, $\mathcal{R}$ representa o conjunto das relações entre os objetos e $\mathcal{W}$ representa o conjunto de pesos das relações entre os objetos. Os pesos das relações na rede podem ser diferenciados, fazendo assim com que seja uma rede ponderada. Isso é recorrente caso queira se considerar o número de vezes um objeto ocorreu ou se relacionou com outro em um conjunto de dados, ou o valor da similaridade entre os objetos. As redes também podem ser homogêneas, i.e., formadas por objetos de um único tipo além de um tipo de relação também, ou heterogêneas, as quais têm como característica, dois ou mais tipos de objetos na sua rede, e um ou mais tipos de relações dentre os objetos [Cao et al. 2020].

Existem diversos formatos de redes para representação de textos [Rossi et al. 2015]. Neste trabalho foram utilizadas dois tipos tradicionais de redes para representar coleções de documentos: redes homogêneas de documentos e redes heterogêneas bipartidas. Em uma rede de documentos, $\mathcal{O} = \mathcal{D}$, isto é, os objetos correspondem aos documentos da coleção. As relações entre os documentos podem ser dados de maneira explícita, como *hiperlinks* ou citações, ou de forma implícita, que são informações geradas por meio de cálculos de similaridade entre os objetos que compõem a rede considerando os valores dos atributos dos respectivos objetos [Angelova and Weikum 2006]. Na Figura 2 é apresentada um exemplo de uma rede de documentos baseada em similaridade.

O uso de redes de documentos geradas por meio de similaridades geralmente provêm os melhores resultados [Angelova and Weikum 2006]. Neste artigo foi utilizada a *k-Nearest Neighbors* (*k-NN*) para gerar uma rede de similaridade entre os documentos [de Sousa et al. 2013]. Nessa rede, cada documento se conecta aos seus *k* vizinhos

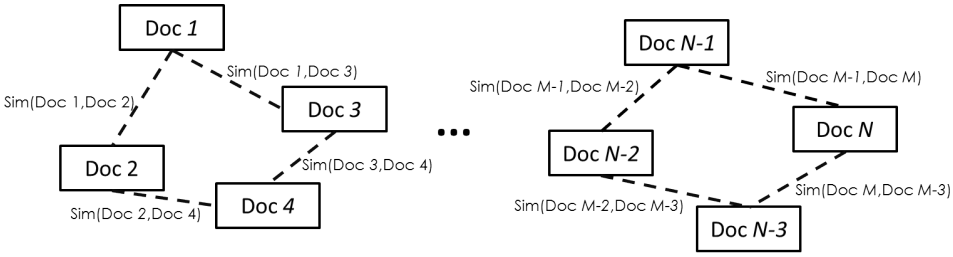

Figure 1. Ilustração de uma rede de documentos baseada em similaridade.

mais próximos. O valor de k irá controlar a densidade da rede. Porém, valores de *k* muito baixos podem gerar uma muito rede esparsa e desconexa, enquanto que valores de k muito altos pode gerar muitas relações e consequentemente relacionar documentos de tópicos diferentes e aumentar o custo computacional. Ambas as situações podem degradar a qualidade do agrupamento. Já em uma rede bipartida, $\mathcal{O} = \mathcal{D} \cup \mathcal{T}$, isto é, os objetos da rede correspondem a documentos e termos. Os termos se conectam aos documentos em que eles ocorrem, e o peso da relação pode ser dado pela frequência de ocorrência [Rossi et al. 2016]. Portanto, as relações aqui são sempre dadas de maneira explícita, i.e., são obtidas diretamente ao analisar os textos. Na Figura 2 é apresentada um exemplo de uma rede bipartida representado uma coleção de textos.

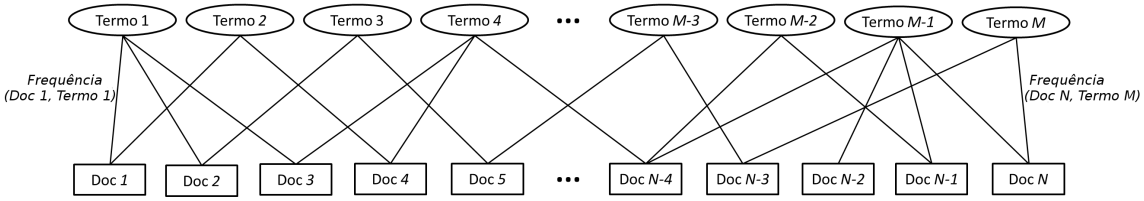

Figure 2. Ilustração de uma rede bipartida.

2.2. Agrupamento de Dados

Uma vez que as coleções textuais são estruturadas, podem ser interpretadas pelos algoritmos. São apresentados diferentes algoritmos suas diferentes configurações para realizar o agrupamento de textos, considerando as diferentes representações dos conjuntos de dados apresentados.

2.2.1. Modelo Espaço Vetorial

Este trabalho apresenta 5 diferentes algoritmos baseados em MEV comumente utilizados na literatura para o agrupamento de textos [Aggarwal 2018]. O emprego desses algoritmos nesse trabalho será para o agrupamento do tipo *hard*, ou seja, os algoritmos irão dividir a coleção de documentos em k grupos disjuntos, i.e., $\mathcal{D} = G_1 \cup G_2 \cup \ldots \cup G_k$. A seguir são apresentados os detalhes dos algoritmos implementados e utilizados neste trabalho. Os algoritmos estão divididos em duas categorias: baseados em representantes de grupos e baseados em projeção no espaço semântico.

Os algoritmos baseados em representantes de grupos visam induzir um objeto ou uma distribuição para representar os grupos do conjunto de dados são [Aggarwal 2018]:

***k-Means*:** esta técnica é baseada no posicionamento de centroides para gerar os grupos do conjunto de dados. Se o objetivo é gerar k grupos, então k centroides serão gerados. Os centroides são objetivos artificiais que correspondem à média dos exemplos que estão mais próximos a este em relação a outros centroides. O algoritmo inicializa centroides com valores aleatórios, e assim iterativamente atribui os documentos a um grupo conforme o seu centroide mais próximo, e em seguida, após atribuir todos os documentos a um grupo, recalcula a posição dos centroides. As iterações do k-*Means* visam minimizar a seguinte função denominada coesão, que é dada por:

$$\textit{coesão} = \sum_{j=1}^{k} \sum_{d_i \in G_j} cosine(\mathbf{w}_{d_i}, \mathbf{c}_j), \tag{1}$$

na qual $\mathbf{c}_j$ é o centroide do grupo G_j, $\mathbf{w}_{d_i}$ é o vetor de atributos do documento d_i e *cosine* é a medida de similaridade cosseno. A atribuição dos documentos aos grupos e o reposicionamento dos centroides é feito até que não se altere os documentos nos grupos, até um determinado número de iterações, ou até que se atinja uma coesão mínima ou uma variação de coesão mínima.

***Bisecting k-Means* (Bk-Means):** este algoritmo é baseado no algoritmo *k-Means*. Inicialmente todos os documentos pertencem a um único grupo. Iterativamente, o algoritmo *k-Means* com $k = 2$ é aplicado a cada grupo, até atingir o número de grupos desejado.

***Gaussian Mixture Models (GMMs)*:** este algoritmo visa encontrar modelos estatísticos que geram os dados de uma coleção. No GMM, considera-se que cada grupo de pontos é gerado por um modelo, e o objetivo algoritmo é encontrar os parâmetros do modelo. Normalmente assume-se que o modelo de geração dos dados é uma função gaussiana, i.e, a probabilidade de um documento d_i pertencer à um grupo g_j é dada por:

$$p_j(d_i|\Theta_j) = \frac{1}{\sqrt{2\pi}\sigma_j} e^{\frac{-(\mathbf{w}_{d_i} - \mu_j)}{2\sigma_j^2}}, \tag{2}$$

na qual Θ_j corresponde aos parâmetros do modelo do j-ésimo grupo, que neste caso correspondem à média (μ_j) e ao desvio padrão (σ_j). O objetivo portanto é encontrar valores de μ e σ para as distribuições de forma a maximizar o produtório $\prod(d_i|\Theta)$ para todo $d_i \in \mathcal{D}$. A obtenção dos parâmetros dos modelos é feita de maneira iterativa utilizando o algoritmo *Expectation-Maximization* (*EM*).

Já os algoritmos baseados em projeção no espaço semântico visam representar os documentos em um novo espaço, de forma que o novo espaço terá k dimensões, usualmente $k \ll |\mathcal{T}|$ as quais representação os k grupos. A pertinência de um documento ao grupo é proporcional ao valor atribuído a cada uma das dimensões, sendo que quanto maior o valor, maior a pertinência. O termo "espaço semântico" é utilizado pelo fato de que para reduzir o número de dimensões, termos semanticamente relacionados são agrupados no mesmo atributo. O resultado da aplicação desses algoritmos é a geração de uma matriz chamada documento-tópico, que irá representar a pertinência dos documentos a cada grupo/tópico e que será utilizada para definir o grupo aos quais os documentos pertence, e termo-tópico, a qual irá representar a pertinência de um termo a cada um dos grupos.

Os algoritmos baseados em projeção no espaço semântico utilizados neste trabalho são:

***Non Negative Matrix Factorization (NMF)*:** esta técnica de agrupamento é baseada na fatoração de matrizes, isto é, a decomposição de uma matriz e duas outras matrizes, de forma que matriz resultante da multiplicação das matrizes fatoradas seja próxima a matriz original. No caso do NMF, todos os valores das matrizes geradas são não negativos, i.e., maiores ou iguais a zero. O NMF baseia-se na minimização da seguinte função objetivo:

$$NFM(\mathbf{W}) = \frac{1}{2}||\mathbf{W} - \mathbf{V}\mathbf{U}^T||, \tag{3}$$

na qual $\mathbf{W}$ representa a matriz documento-termo, ($\mathbf{V}$) é a matriz documento-tópico e ($\mathbf{U}$) é a matriz termo-tópico. Neste trabalho foi utilizada uma solução iterativa [Aggarwal 2018], cujo critério de parada é um número máximo de iterações ou até que a diferença absoluta somada de todas células da matriz documento termo em iterações consecutivas seja abaixo de um limiar.

***Latent Dirichlet Allocation (LDA)*:** é um modelo de projeção no espaço semântico probabilístico. Este algoritmo possui a mesma base do algoritmo GMM, apresentado anteriormente, mas a diferença se dá pelo fato do modelo generativo ser baseado na distribuição de Dirichlet, a qual é mais adequada para dados textuais.

2.2.2. Redes

Há vários algoritmos para gerar partições nas redes, como os algoritmos baseados em corte, remoção de arestas, propagação de rótulos, e maximização de modularidade [Newman 2018]. Neste trabalho foram considerados 3 algoritmos de custo linear que podem ser aplicados redes homogêneas e heterogêneas:

***Label Propagation* (LP) [Šubelj 2019]:** dada uma rede, no algoritmo *Label Propagation* inicialmente é atribuído um rótulo único de grupo a cada nó da rede (ex: $label(o_i) = i$), na qual $label(o_i)$ é uma função que retorna um rótulo do objeto o_i. Após a inicialização, iterativamente cada objeto da rede irá receber o rótulo majoritário dos seus vizinhos. Caso a rede seja pesada, o voto levará em conta o peso da conexão. Com isso, seja $\mathcal{L} = \{l_1, l_2, \ldots, l_{|\mathcal{O}|}\}$ um conjunto de rótulos, a regra de propagação de rótulos para uma rede não pesada é dada por:

$$label(o_i) = \arg\max_{l_k \in \mathcal{L}} \sum_{o_i, o_j \in \mathcal{O}} w_{o_i,o_j} \delta(o_j, l_k), \tag{4}$$

na qual $\delta(o_j, l_k) = 1$ se o objeto o_j possui o rótulo l_k, e 0 caso contrário. A propagação de rótulos pode ser realizada de maneira síncrona, i.e., os rótulos dos nós são atualizados simultaneamente com base nas informações dos rótulos de uma iteração anterior, ou de maneira assíncrona, i.e., na qual o rótulo de um nó é atualizado com base nós rótulos correntes dos vizinhos. A técnica de propagação de rótulos é repetida até que os objetos não mudem mais seus rótulos ou até um determinado número de iterações. Ao final, objetos com o mesmo rótulo pertencerão ao mesmo grupo. Este algoritmo pode ser tanto aplicado a rede de documentos quanto rede bipartida.

***Simple Ranking* (Sim. Rank.) [Sun et al. 2009]:** esta técnica funciona de forma similar ao *Label Propagation* em rede bipartida. Porém, em vez de propagar um único valor, são propagados *k* valores, sendo que os valores correspondem aos *rankings* dos objetos para os grupos. A ideia é obter o *ranking* do termo para cada grupo g_j e definir a pertinência dos documentos aos grupos com base nos *rankings* de seus termos, onde o *ranking* de um documento d_i para um grupo é dado por:

$$\textit{rank}(d_i, g_j) = \left(\frac{\sum_{t_o \in \mathcal{T}} w_{(d_i,t_o)} r_{t_o,g_j}}{\sum_{j=1}^{k} \sum_{t_o \in \mathcal{T}} w_{(d_i,t_o)} r_{t_o,g_j}} \right), \tag{5}$$

na qual r_{t_o,g_j} corresponde ao *ranking* do termo o para aquele grupo. O *ranking* dos termos é definido de maneira análoga à dos documentos, porém, os *rankings* atribuídos aos documentos são usados para definir os *rankings* dos termos. A atualização dos *rankings* dos diferentes tipos de objetos é dada de maneira alternada, i.e., primeiro atualiza-se os *rankings* de um tipo de objeto e depois do outro. Por questões de convergência, os rankings dos documentos são atualizados de acordo com uma taxa α, i.e., o ranking de um documento d_i na iteração $i+1$ é dada por:

$$\textit{rank}(d_i, g_j)^{(i+1)} = \alpha * \textit{rank}(d_i, g_j)^{(i+1)} + (1 - \alpha) * \textit{rank}(d_i, g_j)^{(i)}. \tag{6}$$

O processo de atualização dos *rankings* dos objetos da rede se repete até que a diferença dos *rankings* em iterações consecutivas esteja abaixo de um limiar ou até um número máximo de iterações. Ao final, um documento será atribuído ao grupo correspondente ao maior valor de *ranking*.

***Ranking Clustering* (Rank. Clus.) [Sun et al. 2009]:** o algoritmo *Ranking Clustering* pode ser considerado como uma extensão do algoritmo *Simple Ranking*, porém, *rankings* e agrupamentos são combinados para se obter a solução. Neste caso, são gerados grupos com base no vetor de *rankings* e *rankings* são gerados considerando os exemplos em cada grupo, i.e., os rankings são condicionais aos grupos.

3. Método de Pesquisa

Nesta seção são apresentadas as coleções textuais utilizadas na pesquisa, as métricas de parâmetros para os algoritmos implementados durante a pesquisa e as medidas de avaliação utilizadas como critério para avaliação dos testes realizados.

3.1. Coleções de Textos

Para a avaliação experimental, foram utilizadas 30 coleções textuais de *benchmarking* de diferentes domínios e características [Rossi et al. 2013]. Na Tabela 1 são apresentadas características das coleções textuais utilizadas neste trabalho: número de documentos contidos na coleção textual $|\mathcal{D}|$, número de termos contidos nos documentos da coleção textual $|\mathcal{T}|$, número médio de termos por documento $\overline{|\mathcal{T}|}$, número de classes $|\mathcal{C}|$, porcentagem de documentos pertencentes à classe majoritária $max(\mathcal{C})$, e seus domínios (documentos médicos - DM, páginas *web* - PW, documentos científico - DC, análise de sentimento AS, documentos de recuperação de informação DT, e artigos de notícias - AN).

Table 1. Características das coleções textuais utilizadas na avaliação experimental realizada neste projeto.

Collection	$\|\mathcal{D}\|$	$\|\mathcal{T}\|$	$\overline{\|\mathcal{T}\|}$	$\|\mathcal{C}\|$	$max(\mathcal{C})$	Dom.
Classic4	7095	7749	35.28	4	45.16%	DC
CSTR	299	1726	54.27	4	42.81%	DC
Dmoz-Bus.	18500	8303	11.92	37	2.70%	PW
Dmoz-Comp.	9500	5011	10.83	19	5.26%	PW
Dmoz-Health	6500	4217	12.39	13	7.69%	PW
Dmoz-Science	6000	4821	11.52	12	8.33%	PW
Dmoz-Sports	13500	5682	11.87	27	3.70%	PW
FBIS	2463	2001	159.24	17	20.54%	AN
Hitech	2301	12942	141.93	6	26.21%	AN
Irish-Sentiment	1660	8659	112.64	3	39.46%	AS
La1s	3204	13196	144.63	6	29.43%	AN
La2s	3075	12433	144.82	6	29.43%	AN
LATimes	6279	10020	42.19	6	29.43%	AN
NSF	10524	3888	6.55	16	13.39%	DC
Oh0	1003	3183	52.50	10	19.34%	DM
Oh5	918	3013	54.43	10	16.23%	DM
Oh10	1050	3239	55.63	10	15.71%	DM
Oh15	3101	54142	17.46	10	5.06%	DM
Opinosis	6457	2693	7.55	51	8.18%	AS
Re0	1504	2887	51.72	13	40.43%	AN
Re1	1657	3759	52.69	25	22.39%	AN
Re8	7674	8901	35.30	8	51.12%	AN
SyskillWebert	334	4340	93.15	4	41.02%	PW
Tr11	414	6430	281.66	9	31.88%	DT
Tr12	313	5805	273.59	8	29.71%	DT
Tr21	336	7903	469.86	6	68.75%	DT
Tr23	204	5833	385.29	6	44.61%	DT
Tr31	927	10129	268.49	7	37.97%	DT
Tr41	8778	7455	19.53	10	2.77%	DT
Tr45	690	8262	280.58	10	23.19%	DT

3.2. Algoritmos de Agrupamento e Parâmetros

Os algoritmos de agrupamento e seus respectivos parâmetros são:

- ***K-Means* e *Bisecting K-Means*:** taxa de coesão 0.01, número de iterações máximas igual a 100, e cosseno como medida de similaridade.
- ***Gaussian Mixture Model*:** número de iterações máximas 100 e nível de tolerância igual a 1×10^{-6}.
- ***Non-Negative Matrix Factorization*:** taxa de erro mínimo = 0.001, diferença limiar mínima 0.01 e número de iterações máximas 100.
- ***Label Propagation*:** número de iterações máximas igual a 100, considerando a rede pesada e sem peso, e versões síncrona e assíncrona. As redes de similaridade *k-Nearest Neighbors* foram geradas considerando $\forall$ k = 3 * z $\mid$ z $\in$ [1 – 20].
- ***Simple Ranking*:** valores de α = 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.8, 0.9, 1.0, número de iterações máximas 100 e coesão 0.01.
- ***Ranking Clustering*:** valores de α = 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.8, 0.9, 1.0, número de iterações máximas 100 e coesão 0.01.

Para todos os algoritmos de agrupamento, o número de grupos foi variado entre o número de classes e a raiz do número de documentos da coleção [Marcacini et al. 2012]. Vale ressaltar também que o esquema de peso dos termos utilizados nas coleções foi *term frequency* para todos os algoritmos de agrupamento e *term frequency-inverse document frequency* para todos exceto o LDA.

3.3. Medidas e Critérios de Avaliação

Neste trabalho foram utilizadas medidas de avaliação externa para avaliar a qualidade dos resultados dos algoritmos de agrupamento [Tan et al. 2019]. Essas medidas comparam os grupos com informações externas, que neste caso, correspondem às categorias dos documentos. Portanto, essas medidas visam avaliar o quão o processo de agrupamento é capaz de manter no mesmo grupo, documentos da mesma categoria. Para a explicação das medidas, considere que:

- c_i representa a i-ésima classe da coleção de documentos.
- g_j representa o j-ésimo grupo obtido através da técnica de agrupamento.

- n_{c_i} representa o número de documentos da classe c_i.
- n_{g_j} representa o número de documentos no grupo g_j.
- n_{g_j,c_i} representa o número de documentos do grupo g_j pertencentes à classe c_i.
- $\max_{g_j,c_i}$ retorna a classe majoritária dos documentos agrupados em um grupo g_j.
- k representa o número de grupos.
- l é o número total de classes.
- n é o número total de documentos da coleção textual.

Neste artigo foram utilizadas as seguintes medidas de agrupamento [Tan et al. 2019]:

***Entropy*:** nos permite medir o grau de desordem no resultado de agrupamento, logo, quanto menor o resultado, maior a qualidade do agrupamento:

$$\textit{Entropy} = -\sum_{j=1}^{k} \frac{n_{g_j}}{n} \left(\sum_{i=1}^{l} \frac{n_{g_j,c_i}}{n_{g_j}} \log_2 \frac{n_{g_j,c_i}}{n_{g_j}} \right), \tag{7}$$

***Purity*:** avalia se um grupo contém exemplos de apenas uma classe:

$$\textit{Purity} = \sum_{j=1}^{k} \frac{n_{g_j}}{n} \left(\max_{g_j,c_i} \frac{n_{g_j,c_i}}{n_{g_j}} \right) \tag{8}$$

***F_1*:** realiza uma média harmônica entre *Precision* e *Recall* para avaliar se o grupo g_j contém apenas documentos de uma única classe c_i e todos documentos da classe:

$$F_1 = 2 * \frac{\textit{Precision} * \textit{Recall}}{\textit{Precision} + \textit{Recall}}, \tag{9}$$

em que a *Precision* calcula a fração de pontos corretamente atribuídos a grupos rotulados com a classe c_i (majoritária) pelo total de pontos em tais grupos, e a *Recall* calcula a fração de pontos corretamente atribuídos a grupos rotulados com a classe c_i pelo total de pontos da classe c_i. Para sumarizar os resultados da *Precision* e *Recall* para todas as classes da coleção e consequentemente calcular o valor da medida F1, foram utilizadas as técnicas *micro-averaging*, que corresponde à soma dos termos individuais de cada uma das medidas, e *macro-averaging*, que corresponde à média dos valores das medidas calculadas para cada classe.

Como as inicializações dos algoritmos de agrupamento são aleatórias, foram executadas 10 repetições para cada algoritmo. Para as 10 repetições, foram coletados o melhor resultado, o pior resultado, a média e o desvio padrão dos resultados. Devido ao grande número de resultados, na próxima seção serão reportados os resultados correspondentes a melhor média obtida por um algoritmo em uma determinada coleção. Também por questões de espaço, será reportado apenas os resultados numéricos da medida *Macro* F_1 e uma análise comparativa por meio de número de vitórias considerando cada medida de avaliação. Porém, vale ressaltar que todos os resultados, bem como as implementações, estão disponíveis em `(URL omitida devido ao processo de revisão às cegas)`.

4. Resultados

Na Tabela 2 são apresentados os resultados considerando a medida *Macro*-F_1. Pode-se observar pela tabela que o algoritmo LP em conjunto com as redes kNN obteve os melhores

resultados para 20 das 30 coleções de textos. Pode-se observar também que quando o algoritmo LP-kNN não obteve os melhores resultados, obteve valores próximos aos melhores. Por fim, em algumas situações, o algoritmo LP-kNN presentou grandes diferenças em relações à outros algoritmos. Por exemplo, na bases do domínio TREC, as diferenças chegam a ser superiores à 20%.

Table 2. Melhores médias de resultados para *Macro*F_1 de todos algoritmos utilizados das coleções textuais utilizadas. O melhores resultados estão assinalados em negrito.

Collection	k-Means	Bk-Means	NMF	LP-kNN	LP-Bip.	Sim. Rank.	Rank. Clus.	GMM	LDA
Classic4	0,900	0,822	0,839	**0,931**	0,205	0,786	0,175	0,819	0,911
CSTR	0,623	0,509	0,752	**0,830**	0,297	0,639	0,319	0,623	0,826
Dmoz-Bus.	0,441	0,152	0,062	0,419	0,054	0,224	0,060	0,293	**0,456**
Dmoz-Comp.	0,486	0,270	0,380	0,460	0,104	0,319	0,098	0,344	**0,504**
Dmoz-Health	0,652	0,361	0,560	**0,679**	0,185	0,474	0,132	0,477	0,639
Dmoz-Science	0,475	0,260	0,408	0,475	0,159	0,327	0,139	0,351	**0,496**
Dmoz-Sports	**0,702**	0,274	0,163	0,659	0,074	0,381	0,075	0,454	0,524
FBIS	**0,595**	0,274	0,532	0,583	0,020	0,284	0,056	0,442	0,065
Hitech	0,526	0,402	0,484	**0,571**	0,069	0,500	0,207	0,478	0,493
Irish-Sentiment	0,552	0,471	0,541	0,538	0,188	0,522	0,529	0,529	**0,561**
La1s	0,627	0,409	0,562	**0,721**	0,075	0,592	0,175	0,606	0,691
La2s	**0,748**	0,438	0,740	0,739	0,075	0,593	0,176	0,591	0,708
LATimes	0,611	0,399	0,612	**0,656**	0,300	0,574	0,167	0,545	0,364
NSF	0,524	0,309	0,422	0,550	0,126	0,337	0,067	0,407	**0,685**
Oh0	0,584	0,328	0,690	**0,778**	0,032	0,512	0,132	0,508	0,690
Oh5	0,546	0,330	0,562	**0,675**	0,027	0,431	0,141	0,480	0,604
Oh10	0,513	0,321	0,563	**0,640**	0,029	0,428	0,149	0,500	0,580
Oh15	0,614	0,355	0,584	**0,730**	0,027	0,456	0,147	0,511	0,631
Opinosis	**0,480**	0,408	0,345	0,474	0,048	0,255	0,025	0,335	0,339
Re0	0,374	0,286	0,352	**0,411**	0,048	0,263	0,069	0,332	0,350
Re1	0,355	0,226	0,385	**0,440**	0,014	0,270	0,034	0,303	0,273
Re8	0,900	0,822	0,839	**0,931**	0,205	0,786	0,175	0,819	0,911
SyskillWebert	0,926	0,935	0,853	**0,961**	0,371	0,709	0,426	0,883	0,789
Tr11	0,470	0,406	0,356	**0,625**	0,053	0,401	0,190	0,492	0,417
Tr12	0,517	0,399	0,477	**0,624**	0,057	0,377	0,222	0,596	0,584
Tr21	0,327	0,355	0,355	0,433	0,183	0,301	0,188	**0,469**	0,213
Tr23	0,351	0,377	0,380	**0,716**	0,102	0,362	0,210	0,459	0,430
Tr31	0,559	0,509	0,478	**0,730**	0,167	0,446	0,148	0,608	0,613
Tr41	0,621	0,505	0,496	**0,807**	0,116	0,445	0,118	0,549	0,163
Tr45	0,576	0,467	0,405	**0,793**	0,037	0,461	0,107	0,597	0,597

Por outro lado, os algoritmos de agrupamento aplicados às redes bipartidas apresentaram resultados inferiores em relação as demais abordagens, principalmente o algoritmo LP aplicado as redes bipartidas. Isso se deve pelo fato de objetos do tipo termo, com alta frequência e que ocorrem em um alto número de documentos, influenciarem excessivamente a propagação de rótulos. Observa-se que o algoritmo *Simple Ranking* obteve melhores resultados que os demais algoritmos aplicados às redes bipartidas, *Label Propagation* e *Ranking Clustering*

Na Figura 3 são apresentados o número de vitórias de cada algoritmo considerando as diferentes medidas de avaliação. Pode-se observar o que algoritmo de propagação de rótulos em redes de similaridade (*LP-kNN*) obteve ampla maioria do número de vitórias para as todas medidas utilizadas. Pode-se observar também pelos resultados que o algoritmo *k-Means* e o algoritmo *GMM* obtiveram os melhores resultados para algumas coleções. Já os demais algoritmos não obtiveram a melhor qualidade de agrupamento para nenhum dos casos analisados.

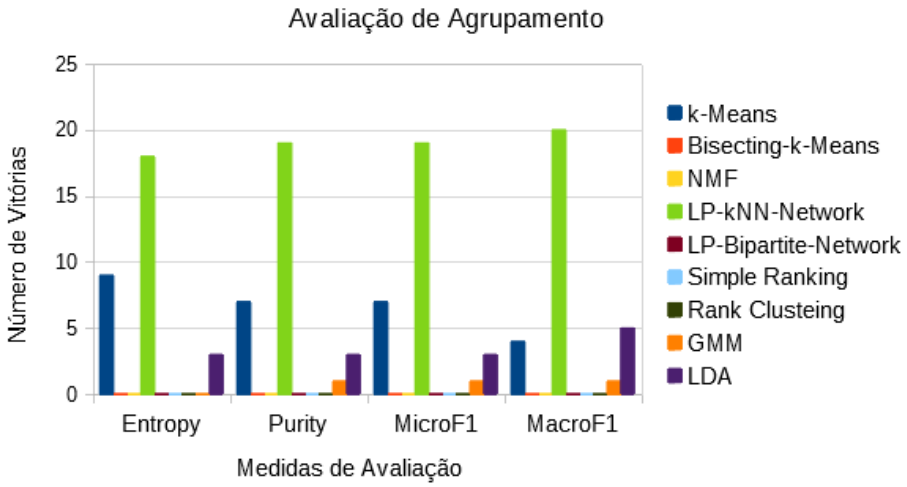

Figure 3. Gráfico do número de vitórias.

5. Considerações Finais

Uma das etapas fundamentais para o sucesso do agrupamento de textos é a representação estruturada da coleção de textos. Nos últimos anos têm ganhado destaque as representações baseadas em redes [Rossi et al. 2016, Mihalcea and Radev 2011], as quais são capazes de modelar relações entre entidades que compõem uma coleção de textos e extrair padrões que dificilmente podem ser extraídos por representações baseadas no modelo espaço-vetorial. Apesar dos benefícios das representações em redes, ainda é pouco explorado na literatura o seu impacto no agrupamento de textos. Com isso, esse trabalho visou a exploração de técnicas baseadas em redes, o uso de diferentes tipos de redes, e uma extensa comparação com algoritmos baseados no modelo espaço vetorial. Os resultados obtidos neste trabalho, demonstram a potencialidade da técnica de propagação de rótulos em conjunto com as redes de similaridade. O fato desta técnica ser capaz de capturar grupos com diferentes formatos e densidades demonstrou-se útil para o agrupamento de textos.

Os resultados obtidos neste trabalho estimulam o avanço em outras áreas que fazem uso de agrupamento na análise de textos, como aprendizado baseado em uma única classe [Golo and Rossi 2019] e o aprendizado ativo [Ienco et al. 2013]. Com isso, os trabalhos futuros se darão nas aplicações e análise do impacto dos algoritmos de agrupamento baseados em redes em tais linhas.

References

Aggarwal, C. C. (2018). *Machine Learning for Text*. Springer Publishing Company, Incorporated, 1st edition.

Angelova, R. and Weikum, G. (2006). Graph-based text classification: learn from your neighbors. In *Proc. Conf. Special Interest Group on Information Retrieval*, pages 485–492. ACM.

Breve, F., Zhao, L., Quiles, M., Pedrycz, W., and Liu, J. (2011). Particle competition and cooperation in networks for semi-supervised learning. *IEEE Transactions on Knowledge and Data Engineering*, 24(9):1686–1698.

Cao, J., Wang, S., Wen, D., Peng, Z., Philip, S. Y., and Wang, F.-y. (2020). Mutual clustering on comparative texts via heterogeneous information networks. *Knowledge and Information Systems*, 62(1):175–202.

de Sousa, C. A. R., Rezende, S. O., and Batista, G. E. A. P. A. (2013). Influence of graph construction on semi-supervised learning. In *Proc. Eur. Conf. Machine Learning and Knowledge Discovery in Databases*, pages 160–175.

Golo, M. P. S. and Rossi, R. G. (2019). An extensive empirical evaluation of preprocessing techniques and supervised one class learning algorithms for text classification (in press). In *Proceeding of the National Meeting on Artificial and Computational Intelligence (ENIAC)*, pages 1–12.

Ienco, D., Bifet, A., Žliobaitė, I., and Pfahringer, B. (2013). Clustering based active learning for evolving data streams. In *Int. Conf. Discovery Science*, pages 79–93. Springer.

Khennak, I., Drias, H., Kechid, A., and Moulai, H. (2019). Clustering algorithms for query expansion based information retrieval. In *Int. Conf Computational Collective Intelligence*, pages 261–272. Springer.

Marcacini, R. M., Hruschka, E. R., and Rezende, S. O. (2012). On the use of consensus clustering for incremental learning of topic hierarchies. In *Lecture Notes in Computer Science*, Alemanha. Springer Verlag.

Mei, J.-P., Lv, H., Yang, L., and Li, Y. (2019). Clustering for heterogeneous information networks with extended star-structure. *Data Mining and Knowledge Discovery*, 33(4):1059–1087.

Mihalcea, R. and Radev, D. (2011). *Graph-based natural language processing and information retrieval*. Cambridge University Press.

Newman, M. (2018). *Networks*. OUP Oxford.

Rossi, R. G., de Andrade Lopes, A., and Rezende, S. O. (2016). Optimization and label propagation in bipartite heterogeneous networks to improve transductive classification of texts (in press). *Information Processing & Management*.

Rossi, R. G., Marcacini, R. M., and Rezende, S. O. (2013). Benchmarking text collections for classification and clustering tasks. Technical Report 395, Institute of Mathematics and Computer Sciences, University of Sao Paulo.

Rossi, R. G., Rezende, S. O., and de Andrade Lopes, A. (2015). Term network approach for transductive classification. In *Int. Conf. Intelligent Text Processing and Computational Linguistics*, pages 497–515.

Šubelj, L. (2019). Label propagation for clustering. *Advances in Network Clustering and Blockmodeling*, pages 121–150.

Sun, Y., Han, J., Zhao, P., Yin, Z., Cheng, H., and Wu, T. (2009). Rankclus: integrating clustering with ranking for heterogeneous information network analysis. In *Proc. Int. Conf. Extending Database Technology: Advances in Database Technology*, pages 565–576. ACM.

Tan, P., Steinbach, M., Karpatne, A., and Kumar, V. (2019). *Introduction to Data Mining*. What's New in Computer Science Series. Pearson.