UNIVERSIDADE DE SÃO PAULO
Instituto de Ciências Matemáticas e de Computação

Segmentação em tópicos utilizando redes complexas

Thales Sinelli Lima

Dissertação de Mestrado do Programa de Pós-Graduação em Ciências de Computação e Matemática Computacional (PPG-CCMC)

ICMC USP
SÃO CARLOS

SERVIÇO DE PÓS-GRADUAÇÃO DO ICMC-USP

Data de Depósito:

Assinatura: _____

Thales Sinelli Lima

Segmentação em tópicos utilizando redes complexas

Dissertação apresentada ao Instituto de Ciências Matemáticas e de Computação – ICMC-USP, como parte dos requisitos para obtenção do título de Mestre em Ciências – Ciências de Computação e Matemática Computacional. *VERSÃO REVISADA*

Área de Concentração: Ciências de Computação e Matemática Computacional

Orientador: Prof. Dr. Diego Raphael Amancio

USP – São Carlos
Setembro de 2019

Ficha catalográfica elaborada pela Biblioteca Prof. Achille Bassi
e Seção Técnica de Informática, ICMC/USP,
com os dados inseridos pelo(a) autor(a)

S616s

Sinelli Lima, Thales
Segmentação em tópicos utilizando redes complexas / Thales Sinelli Lima; orientador Diego Raphael Amancio. -- São Carlos, 2019.
78 p.

Dissertação (Mestrado - Programa de Pós-Graduação em Ciências de Computação e Matemática Computacional) -- Instituto de Ciências Matemáticas e de Computação, Universidade de São Paulo, 2019.

1. Redes complexas. 2. Processamento de língua natural. 3. Segmentação de tópicos. I. Amancio, Diego Raphael, orient. II. Título.

Bibliotecários responsáveis pela estrutura de catalogação da publicação de acordo com a AACR2:
Gláucia Maria Saia Cristianini - CRB - 8/4938
Juliana de Souza Moraes - CRB - 8/6176

Thales Sinelli Lima

Complex network approach to topic segmentation

Dissertation submitted to the Institute of Mathematics and Computer Sciences – ICMC-USP – in accordance with the requirements of the Computer and Mathematical Sciences Graduate Program, for the degree of Master in Science. *FINAL VERSION*

Concentration Area: Computer Science and Computational Mathematics

Advisor: Prof. Dr. Diego Raphael Amancio

USP – São Carlos
September 2019

AGRADECIMENTOS

Gostaria de agradecer a todos que fizeram parte dessa minha jornada acadêmica e pessoal que culmina nessa dissertação. Seria um desserviço tentar listar todos que tiveram um impacto significativo nessa minha jornada, sejam amigos que trago desde a graduação, ou até mesmo antes, que compartilharam as diversas etapas da vida dividindo as mesmas angústias e alívios e alegrias, colegas que sempre me ajudaram com ideias e descobertas e aprimoramentos, os grupos de confraria, jogatinas e RPGs que propiciaram momentos extraordinários ou professores que me deram instrução e direcionamento fundamental para eu ser quem sou hoje.

Reservo um agradecimento especial aos meus pais por sempre estarem ao meu lado me apoiando e acreditando em mim.

Agradeço também à Capes pelo financiamento dos meus estudo.

RESUMO

LIMA, T. S. **Segmentação em tópicos utilizando redes complexas**. 2019. 78 p. Dissertação (Mestrado em Ciências – Ciências de Computação e Matemática Computacional) – Instituto de Ciências Matemáticas e de Computação, Universidade de São Paulo, São Carlos – SP, 2019.

Redes complexas têm sido utilizadas para representar sistemas reais em várias áreas. Um dos sistemas estudados utilizando redes são as línguas naturais, com ênfase em estudos sobre as estruturas textuais. Nessa dissertação introduzimos modelos baseados em relações semânticas de palavras ou sentenças para a segmentação de tópicos. Aqui são abordados os principais conceitos utilizados no desenvolvimento dessa dissertação, assim como os resultados principais obtidos pelos métodos propostos. Vemos nos resultados obtidos que utilizando uma combinação de técnicas baseadas em redes previamente propostas juntamente com abordagens mais modernas de NLP obtemos resultados melhores que os modelos atuais baseados em redes.

Palavras-chave: Redes Complexas, Processamento de Língua Natural, Segmentação de Tópic.

ABSTRACT

LIMA, T. S. **Complex network approach to topic segmentation**. 2019. 78 p. Dissertação (Mestrado em Ciências – Ciências de Computação e Matemática Computacional) – Instituto de Ciências Matemáticas e de Computação, Universidade de São Paulo, São Carlos – SP, 2019.

Complex Networks have been used to represent real systems of several areas. One of such system is natural language, with enpahsis on text structure. In this thesis, we introduce models based on semantic relations between words or sentences for topic segmentation. Here, we discuss the main concepts used during the development of this dissertation, as well as the main results achieved by the models proposed. The results obtained by combining previous complexnetwork based models with modern NLP methods perform considerably better than the current complex network baseline.

Keywords: Complex Networks, Natural Language Processing, Topic Segmentation.

LISTA DE ILUSTRAÇÕES

Figura 1 – Exemplo de rede onde ocorre a propagação de importância nula. 24
Figura 2 – Em **(a)** temos um exemplo de rede aleatória e em **(b)** temos a sua distribuição de grau. 29
Figura 3 – Processo de criação de uma rede de pequeno mundo. 30
Figura 4 – Em **(a)** temos um exemplo de rede de pequeno mundo e em **(b)** temos a sua distribuição de grau. 30
Figura 5 – Em **(a)** temos um exemplo de rede livre de escala e em **(b)** temos a sua distribuição de grau. 31
Figura 6 – Figura que ilustra o método de detecção de comunidade multi nível. 39
Figura 7 – Exemplo de rede onde as comunidades que apresentam o maior valor de modularidade não condizem com o que seria esperado intuitivamente. . . 40
Figura 8 – Exemplo da aplicação da detecção multi níveis sobre uma rede gerada a partir de dados telefônicos da bélgica. Cada nó é uma comunidade e a escala de cor indica a língua predominante da comunidade (verde para holandês e vermelho para francês). Podemos mudar de hierarquia para visualizar melhor os constituintes da comunidade intermediária. 40
Figura 9 – Exemplo de gráfico de coerência original ao longo de um texto. Os números indicam os parágrafos e as linhas verticais indicam os pontos de mínimo local de similaridade, cujo valor é inferior ao limiar sugerido. 43
Figura 10 – Exemplo de gráfico de similaridade do TopicTiling. Os traços segmentados representam mínimos que não ultrapassam o limiar e os traços cheios as divisões que ultrapassam o limiar de profundidade de $\mu - \sigma/2$ 44
Figura 11 – As figuras representam exemplos de redes geradas, nas quais foram detectadas comunidades. A figura (a) é um exemplo de rede gerada utilizando o modelo de co-ocorrência estendido. A figura (b) ilustra uma rede gerada pelo modelo baseado em parágrafos. É possível observar que cada palavra pertence a uma comunidade; desta forma, podemos contar qual comunidade está dominando cada parágrafo. 46
Figura 12 – Exemplo do método CBOW onde tenta-se predizer uma palavra dado as palavras adjacentes à ela. 48
Figura 13 – Exemplo do método *skip-gram* onde dada uma palavra tenta-se predizer as palavaras adjacentes. 48
Figura 14 – Figura que mostra a arquitetura do w2v. 48

Figura 15 – Visualização da rede gerada utilizando o método proposto. As cores indicam a posição no texto de cada bloco. 51
Figura 16 – Representação do método de construção de textos artificiais. Primeiro os t tópicos são sorteados e então para cada um s sub-tópicos são selecionados. Finalmente para cada texto p extrai-se p parágrafos consecutivos concatena-se todos para formar um texto artificial completo 60
Figura 17 – Exemplo de verificação entre subsequências. Nesse exemplo o valor de Pk seria $\frac{3}{7}$ 61
Figura 18 – Gráfico da performance do modelo para cada configuração de texto e para cada valor de limiar de similaridade. 63
Figura 19 – Agrupado dos melhores resultados selecionados no *dataset* de treino e testados no *dataset* de teste 63
Figura 20 – Agrupado dos melhores resultados selecionados no *dataset* de treino e testados no *dataset* de teste limitando o alcance de ligaçoes na rede a 3 parágrafos de distância 64
Figura 21 – Resultado do método adaptado de parágrafos de (ARRUDA; COSTA; AMANCIO, 2015b) sobre o *dataset* padronizado dessa dissertação 65
Figura 22 – Resultado do método estendido de parágrafos de (ARRUDA; COSTA; AMANCIO, 2015b) utilizando similaridade baseada em *word2vec* para as ligações na rede e gráfico de similaridade para identificar quebras de parágrafos . . 65
Figura 23 – Resultado do método estendido de parágrafos de (ARRUDA; COSTA; AMANCIO, 2015b) utilizando similaridade de *word2vec* para as ligações na rede . 66
Figura 24 – Resultado do método estendido de parágrafos de (ARRUDA; COSTA; AMANCIO, 2015b) utilizando um gráfico de similaridade para identificar quebras de parágrafos 66
Figura 25 – (a) Exemplo de rede com duas comunidades onde alguns vértices apresentam bastante ligações para comunidade que não a dele. (b) Contagem de quantas ligações com nós de cada comunidade cada nó apresenta representada por um vetor. (c) Normalização dos vetores para que o pertencimento de cada nó em cada comunidade seja comparável. 67
Figura 26 – Resultado para a segmentação baseada em sentenças utilizando métodos diferentes para a detecção de comunidades. Os resultados apresentados são todos considerando os textos somente com um um sub-tópico por tópico 68

LISTA DE TABELAS

Tabela 1 – Exemplo de um pré-processamento utilizado em (ARRUDA; COSTA; AMANCIO, 2015b) sobre sentenças hipotéticas. Aqui identifiocamos as classes gramaticais com as siglas: NN para substantivos, VBZ para verbos, DT para artigos, ADP para preposições e JJ para adjetivos 45
Tabela 2 – Acurácia para classificação de textos obtida por cada combinação de estratégia e método de AM. 50
Tabela 3 – Tabela com todos os resultados obtidos sem a remoção de *stopwords* ou a utilização de comunidades sobrepostas 69
Tabela 4 – Tabela com todos os resultados obtidos removendo *stopwords* 70
Tabela 5 – Tabela com todos os resultados utilizando o método descrito de comunidades sobrepostas 71

SUMÁRIO

1 INTRODUÇÃO 17
1.1 Introdução 17

2 REDES COMPLEXAS 21
2.1 Considerações Iniciais 21
2.2 Definições Básicas 21
2.3 Medidas de Centralidade 22
2.3.1 Centralidade de Grau 22
2.3.2 Centralidade de Autovalor 23
2.3.3 Centralidade de Katz 24
2.3.4 PageRank 25
2.4 Outras medidas 26
2.4.1 Menor caminho médio 26
2.4.2 Coeficiente de Clusterização 27
2.5 Modelos de redes 28
2.5.1 Modelo aleatório de Erdös-Rényi 28
2.5.2 Modelo de Rede de Pequeno Mundo de Watts-Strogatz 29
2.5.3 Redes Livres de Escala de Barabási-Albert 31
2.6 Detecção de Comunidades e Modularidade 32
2.6.1 Métodos de Detecção de Comunidades 33
2.6.1.1 Particionamento de Kernighan–Lin (KI) 33
2.6.1.2 Particionamento Espectral 34
2.6.1.3 Detecção de Comunidade Espectral 36
2.6.2 Detecção de Comunidades Multi-níveis 38

3 TRABALHOS RELACIONADOS 41
3.1 Considerações Iniciais 41
3.2 Segmentação de Tópicos 41
3.2.1 Trabalho de (HEARST, 1997) 42
3.2.2 Trabalho de (RIEDL; BIEMANN, 2012) 43
3.2.3 Trabalho de (ARRUDA; COSTA; AMANCIO, 2015b) 44
3.3 Outras trabalhos relacionados 47
3.3.1 Trabalho de (MIKOLOV et al., 2013) 47

3.3.2 Trabalho de (ARRUDA; COSTA; AMANCIO, 2015a) 49
3.3.3 Trabalho de (ARRUDA et al., 2016) 50

4 TRABALHOS REALIZADOS 53
4.1 Considerações Iniciais 53
4.2 Motivação e Objetivos 53
4.3 Metodologia 54
4.3.1 Avaliação e banco de dados 54
4.3.2 Atividades propostas 55
4.3.2.1 Uso de redes de similaridade semântica 55
4.3.2.2 Métodos de detecção de comunidade 56
4.3.2.3 Tratamento de palavras ambíguas 56
4.3.2.4 Combinação de métodos 57
4.3.2.5 Análise de dependência de parâmetros 57

5 MÉTODOS E RESULTADOS 59
5.1 Base de Dados 59
5.2 Métrica 60
5.3 Métodos e Resultados 61
5.3.1 Métodos baseados em redes de similaridade semântica 61
5.3.1.1 Redes baseadas em sentença 61
5.3.1.2 Redes baseadas em palavras 63
5.3.2 Variação dos métodos 64
5.3.3 Seleção de método de detecção de Comunidade 67

6 CONCLUSÃO 73
6.1 Composição de métodos 73
6.2 Redes baseadas em sentença 73
6.3 Variações dos métodos 74
6.4 Trabalhos futuros 74

REFERÊNCIAS 75

CAPÍTULO

1

INTRODUÇÃO

1.1 Introdução

Redes complexas têm sido utilizadas recentemente como um método eficaz para modelagem de problemas reais em diversos campos do conhecimento. Entre os exemplos mais notórios de aplicações de redes podemos citar o estudo de redes de proteínas e a análise de bancos de dados de redes sociais (COSTA *et al.*, 2007). Ao contrário do estudo tradicional de grafos, cujo foco estava no estudo de grafos com padrões aleatórios de conectividade, os recentes avanços na área têm focado no estudo topológico de redes reais. Isso se deu especialmente pela descoberta de que muitos sistemas reais, quando modelados por redes não apresentam características aleatórias, mas seguem padrões topológicos proeminentes (COSTA *et al.*, 2007). Para o presente projeto, é de especial interesse notar que redes já foram aplicadas para modelar sistemas linguísticos (ARRUDA; COSTA; AMANCIO, 2015b; SILVA; AMANCIO, 2013; AMANCIO, 2015a). Em especial, nos últimos anos, vários estudos mostraram que redes textuais são capazes de capturar informações semânticas (AMANCIO, 2015b; CANCHO; SOLÉ; KÖHLER, 2004; MARTINEZ-ROMO *et al.*, 2011).

Uma modelagem bem conhecida de textos por redes é o modelo de co-ocorrência de palavras (ARRUDA; COSTA; AMANCIO, 2015b; CHOI, 2000). Esse modelo cria redes cujos nós são as palavras de determinado texto e as ligações são feitas caso duas palavras apareçam próximas uma da outra, seja esta proximidade determinada pela árvore sintática ou pela simples distância entre elas no texto. Alguns métodos tradicionais de análise de textos já se utilizaram dessa abordagem de co-ocorrência (CHOI, 2000; RIEDL; BIEMANN, 2012). No entanto, grande parte destes métodos apenas contam as repetições da mesma palavra em uma determinada janela, ou seja, aparacem próximas uma da outra no texto. O estudo de co-ocorrência de palavras utilizando redes complexas traz a vantagem de armazenar a informação de contexto de uso de palavras de maneira simples, mas que ao mesmo tempo fornece uma análise informativa relativa à semântica de textos. Sucessos iniciais do estudo

de redes de co-ocorrência em aplicações que dependem da sintaxe/estrutura e semântica dos textos podem ser vistos por exemplo em (ARRUDA; COSTA; AMANCIO, 2015b; SILVA; AMANCIO, 2012).

Embora a modelagem tradicional de co-ocorrência tenha sido útil para vários problemas de natureza estrutural (i.e. que dependem da sintaxe e do estilo), sua aplicabilidade em problemas dependentes da semântica pode ser um pouco limitada, já que a maioria dos atributos extraídos destas redes pouco dependem do conteúdo dos documentos (AMANCIO *et al.*, 2013). Por este motivo, neste projeto, propomos modificar a modelagem de co-ocorrência tradicional para capturar aspectos semânticos de textos, com aplicação em um importante problema na área de processamento de línguas naturais (PLN): a segmentação de tópicos. Mais especificamente, a segmentação de tópicos tem como objetivo principal detectar parcelas coesas de um texto. Tal detecção é essencial para o desempenho de outras tarefas de processamento de línguas naturais (DU; BUNTINE; JOHNSON, 2013), incluindo a sumarização automática (ANGHELUTA; BUSSER; MOENS, 2002) e a análise discursiva (GALLEY *et al.*, 2003).

Nesta dissertação, é usada a abordagem de redes complexas, pois a organização das palavras em textos, que não é geralmente utilizada em modelos tradicionais para a tarefa, pode ser capturada pela modelagem de redes (SILVA; AMANCIO, 2012; ARRUDA; COSTA; AMANCIO, 2015b). Desta forma, dois avanços científicos são esperados após a execução deste projeto pesquisa: (i) um modelo de redes complexas para tratamento semântico de textos será proposto; e (ii) o modelo proposto tratará o problema de segmentação de tópicos. Embora o foco deste projeto seja na aplicação descrita em (ii), a execução da tarefa (i) abre um leque de investigação para outros desenvolvimentos teóricos, já que a análise de padrões de conteúdo em documentos escritos representa uma área de grande interesse (CANCHO; SOLÉ; KÖHLER, 2004).

Uma primeira abordagem para tratamento do problema de segmentação de tópicos com redes foi proposta em (ARRUDA; COSTA; AMANCIO, 2015b). No entanto, apenas modelos bastante simples de similaridade textual foram usados. Com os recentes avanços na área de PLN, em especial na representação vetorial da semântica das palavras, foi possível adicionar camadas extras de informação na rede para obter resultados melhores. O método mais utilizado atualmente para essa representação vetorial, o *word2vec*, se utiliza de uma rede neural treinada para prever palavras a partir de sua vizinhança (contexto) (MIKOLOV *et al.*, 2013). Como resultado do treinamento, *embeddings* – uma representação semântica e vetorial – são criados para cada palavra no vocabulário. A característica mais importante dos *embeddings* para este projeto é que palavras similares semanticamente apresentam vetores similares (MIKOLOV *et al.*, 2013), o que confirma a adequação do modelo para capturar as relações de similaridade entre subtópicos. Outra vantagem dos *embeddings* é que, por serem vetores, torna-se possível realizar operações entre estes para simplificar a análise de sentenças

ou demostrar a relação entre duas palavras automaticamente (MITCHELL; LAPATA, 2010).

Um estudo aprofundado sobre como introduzir mais informação em redes textuais utilizando tanto modelagens alternativas, como métodos adicionais que consigam elucidar correlações textuais, são de especial interesse para garantir melhores resultados na área de segmentação de tópicos. Embora melhores resultados para a segmentação em si não sejam de utilidade aparente, eles são de especial importância para auxiliar em aplicações de recuperação de informação, sumarização de texto e análise de discurso (CHOI, 2000; ANGHELUTA; BUSSER; MOENS, 2002; HEARST, 1997). A presente dissertação extende os métodos criados em (ARRUDA; COSTA; AMANCIO, 2015b). As extensões utilizam heurísticas diferentes para a criação da rede textual e para a detecção de quebra de tópicos, incluindo o uso de medidas mais avançadas de representação de similaridade textual. Outro ponto importante a ser investigado corresponde ao uso de métodos mais avançados de clusterização em redes complexas – os métodos de detecção de comunidades. O uso de métodos mais avançados do que os apresentados em (ARRUDA; COSTA; AMANCIO, 2015b) justifica-se pelo fato de que as redes geradas com métodos mais avançados de cálculo de similaridade necessitam de tratamento adequado. A presença de pesos em arestas representando o grau de similaridade entre sentenças requerem, por exemplo, o uso de métodos que contabilizem pesos nestas estruturas.

O presente documento segue a seguinte estrutura: No Capítulo 2 são introduzidos conceitos básicos de redes complexas, bem como medidas relevantes e métodos de detecção de comunidades. O Capítulo 3 trata de trabalhos relacionados, incluindo trabalhos sobre aplicações de redes complexas à tarefas de PLN e métodos existentes de segmentação de tópicos. O Capítulo 4 apresenta as atividades propostas para o projeto, assim como da motivação e objetivos. O Capítulo 5 trata de resultados obtidos seguindo a metodologia proposta. O Capítulo 6 sumariza os resultados mais importantes numa conclusão e expõe algumas ideias de trabalhos futuros.

CAPÍTULO

2

REDES COMPLEXAS

2.1 Considerações Iniciais

A origem de redes complexas advém do estudo de teoria de grafos, que tem como marco inicial o estudo de Euler sobre o problema das 7 pontes (EULER, 1741). No entanto, o estudo de redes complexas tem preocupações diferentes do que teoria de grafos. De maneira simples pode-se dizer que a área de redes complexas se preocupa em extrair informações relevantes para um problema real modelado por grafos usualmente grandes. Nesse capítulo são introduzidos conceitos e métodos básicos para o estudo de redes.

O capítulo está disposto da seguinte forma: na Seção 2.2 são introduzidas definições básicas de redes. Nas Seções 2.3 e 2.4 são demonstradas medidas existentes. Na Seção 2.5 são expostas modelagens comuns de redes complexas. Finalmente na Seção 2.6, alguns métodos de detecção de comunidade são abordados.

2.2 Definições Básicas

Formalmente podemos definir um grafo como $G = \{V, E\}$ onde $V = \{v_1, v_2, ..., v_N\}$ é o conjunto de N vértices, ou nós, e $E = \{e_1, e_2, ..., e_M\}$ é o conjunto de M arestas, sendo que cada aresta conecta dois vértices. Para a representação da conectividade usamos uma Matriz de adjacência A. Sobre essa matriz de adjacência podemos impor condições que vão servir para caracterizar o grafo. Se impusermos que A seja simétrica temos um grafo não direcionado, caso contrário temos um grafo direcionado. Podemos também definir que toda entrada do matriz de adjacência pode ser somente 0 ou 1, tendo assim um grafo não ponderado, e sem essa restrição temos um grafo ponderado. No caso do grafo ponderado a definição formal fica como $G = \{V, E, W\}$, onde $W = \{w_1, w_2, ..., w_m\}$ representa o peso de cada aresta.

Em um grafo não ponderado, definimos para um vértice v_i o seu grau k_i como

$$k_i = \sum_j A_{ij} \tag{2.1}$$

No caso de uma rede ponderada podemos definir analogamente a força s_i de uma dada aresta e_i

$$s_i = \sum_j A_{ij} \tag{2.2}$$

Quando não especificado, trabalharemos com grafos não direcionados e não ponderados.

Para grafos direcionados temos duas medidas de grau: o grau que entra e o grau que sai. No caso, o elemento (i, j) da matriz indica que exite uma aresta que sai do nó i e chega no nó j. Então definimos k^{in} e k^{out} como:

$$k_i^{out} = \sum_j A_{ij} \tag{2.3}$$

$$k_i^{in} = \sum_j A_{ji} \tag{2.4}$$

2.3 Medidas de Centralidade

Medidas de centralidade em redes complexas, como a maior parte das áreas de estudos de redes, tem seu princípio em teoria de grafo. No entanto, a maior parte das medidas de centralidade em redes foram propostas no campo de análise de redes sociais (NEWMAN, 2010).

2.3.1 Centralidade de Grau

Medidas de centralidade são um tema central para o estudo de rede complexas, nela almeja-se encontrar quais são os vértices mais importantes de uma rede. Parte da razão do interesse sobre esse tema é o fato de não haver somente uma maneira de interpretar o que é um vértice de fato importante. Uma definição comum argumenta que um vértice importante é um vértice que tem bastantes ligações dentro da rede. Dessa definição surge o modelo mais básico de centralidade: a centralidade de grau.

Seguindo a definição, temos que basta verificar o grau de cada vértice e encontrar aqueles que apresentam maior grau para encontrar os vértices mais importantes. Essa definição, embora seja interessante para redes sociais como *facebook*, onde o alcance social de uma pessoa é bem aproximado pelo número de pessoas com quem ela é amiga, a medida não é

tão interessante para redes de citação, por exemplo. Em redes de citação, embora o número de citações de um artigo seja interessante, é mais interessante saber se os artigos que o citam são importantes também. Não adianta um artigo ser citado por centenas de outros artigos que não receberam nenhuma citação.

2.3.2 Centralidade de Autovalor

Diversos métodos foram propostos para tentar medir a importância de um dado nó, considerando também a importância de seus vizinhos. O primeiro deles tenta utilizar explicitamente a ideia de que um dado vértice i é tão importante quanto os vértices que são conectados a ele. Para isso definimos um vetor de centralidade $\mathbf{x}$ de tamanho N, onde a entrada i dele representa a importancia do $i-ésimo$ nó. Tendo pelo menos um valor inicial para $\mathbf{x}$, podemos propagar a importância na rede usando a equação 2.5.

$$\mathbf{x}'_i = \sum_j A_{ij} x_j \Rightarrow \mathbf{x}' = A\mathbf{x} \tag{2.5}$$

Efetivamente o que essa definição diz é que a importância de um dado nó é a soma das importância dos vértices ligados a ele. Perceba que essa definição pode ser aplicada diversas vezes a medida de encontrar um valor "destilado" para a centralidade. Podemos escrever então:

$$\mathbf{x}(t) = A^t \mathbf{x}(0) \tag{2.6}$$

Da álgebra linear sabemos que podemos escrever um vetor qualquer na base dos autovetores de uma matriz. Aqui nós vamos escrever $\mathbf{x}$ na base $\mathbf{v}_1, \mathbf{v}_2, ..., \mathbf{v}_N$ de autovetores da matriz de adjacência. Os autovalores correspondentes são κ_i, onde $\kappa_i > \kappa_j$ se $i > j$.

$$\mathbf{x} = \sum_i c_i \mathbf{v}_i \tag{2.7}$$

temos então:

$$\mathbf{x}(t) = A^t \sum_i c_i \mathbf{v}_i = \sum_i c_i \kappa_i \mathbf{v}_i = \kappa_1^t \sum_i c_i \left(\frac{\kappa_i}{\kappa_1}\right)^t \mathbf{v}_i \tag{2.8}$$

Como κ_1 é o maior autovalor, temos que no limite de t indo para o infinito, o único valor que resta da somatória é o do autovetor correspondente ao maior autovalor. Ou seja, para um t suficientemente grande temos:

$$\mathbf{x}(t) = A^t \mathbf{x} = \kappa_1^t \mathbf{v}_1 \tag{2.9}$$

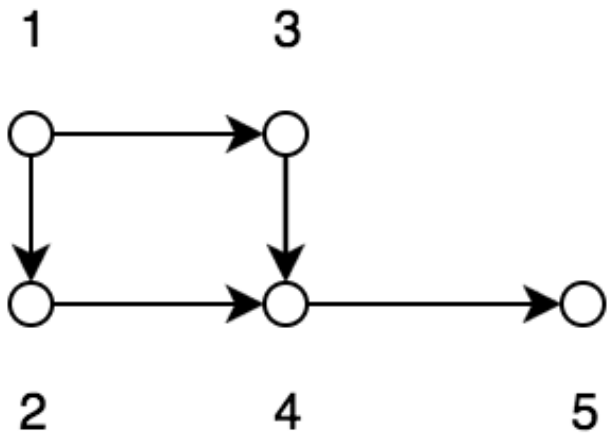
Esse processo iterativo provém um valor de $\mathbf{x}$ cada vez maior pois não existem autovalor negativos na matriz de adjacência e isso no entanto não é um problema, pois quando analisamos uma rede queremos saber somente quão central é um vértice em relação aos outros vértices da rede, não necessitando um valor absoluto. Podemos então só utilizar o valor de $\mathbf{v}_1$ como a centralidade para os vértices da rede.

2.3.3 Centralidade de Katz

A centralidade de autovetores não apresenta muitos problemas em redes não direcionadas. No entanto alguns problemas que aparecem em rede direcionas precisam ser solucionados.

O primeiro problema é a propagação de importância zero. Na Figura 1 temos um exemplo onde isso acontece. Note que um dos vértices da rede tem $k^{in} = 0$. É fácil perceber quem após a primeira iteração do método de autovetores a sua importância será zero, simplesmente por nenhum vértice apontar para ele. Em si isso não é um problema, porém quando existem outros vértices na rede que só são apontados por ele, as suas importâncias se tornarão zero. Isso pode acabar por se alastrar pela rede inteira.

Figura 1 – Exemplo de rede onde ocorre a propagação de importância nula.



Fonte: Elaborada pelo autor.

Para solucionar esse problema podemos modificar 2.5 para que a importância de um vértice nunca seja zero. No caso, vamos adicionar um valor de importância básica β. Também adicionamos um peso α na medida anterior para sermos capazes de variar esses parâmetros de acordo com a necessidade.

$$\mathbf{x}_i = \alpha \sum_j A_{ij} x_j + \beta \Rightarrow \mathbf{x} = \alpha A\mathbf{x} + \mathbb{1}\beta \tag{2.10}$$

Note que essa expressão pode ser solucionada para $\mathbf{x}$ utilizando inversão de matrizes. Temos então:

$$\mathbf{x} = (I - \alpha A)^{-1} \mathbb{1} \beta \tag{2.11}$$

Para realizar a inversão de matrizes é necessário tomar cuidados como verificar se ela é de fato inversível. Sendo I a matriz identidade, temos que o determinante a seguir não pode ser zero:

$$\det(I\alpha^1 - A)^{-1} \neq 0 \tag{2.12}$$

Podemos ver que basta $1/\alpha$ não ser um auto valor de A que ela é inversível. O α em geral é selecionado de modo que ele seja próximo de $1/\kappa_1$ para maximizar o termo que depende de A, fazendo com que os valores de centralidade obtidos se aproximem dos valores obtidos utilizando a centralidade de autovetores (em redes que não apresentam o problema de propagação de importância nula).

O valor de β nesse caso acaba servindo somente como normalização para o valor de $\mathbf{x}$. No entanto, se a importância básica para cada vértice for definida de modo a assumir diversos valores, temos então um vetor $\boldsymbol{\beta}$. A expressão final nesse caso será:

$$\mathbf{x} = (I - \alpha A)^{-1} \boldsymbol{\beta} \tag{2.13}$$

A escolha de valores diferentes para importância básica pode ser guiada por fatos externos a rede. Por exemplo, em uma rede social podemos ter que a idade de uma dada pessoa é um fator relevante para a sua importância na rede, portanto teríamos valores de β diferentes para vértices de pessoas com idades diferentes.

Embora a centralidade de Katz resolva o problema da propagação da importância nula, ela não resolve todos problemas que aparecem para redes direcionadas. Um problema que é especialmente importante para redes de *sites* é a propagação de importância positiva. No exemplo do *Google*, ele é um site extremamente importante pois ele é apontado por muitos sites, no entanto, ele aponta para muitos outros sites que acabam recebendo uma contribuição positiva significativa só por aparecerem no *Google*.

2.3.4 PageRank

Para evitar que a importância seja propagada indevidamente, foi proposto o método *PageRank*, que é similar com a centralidade de Katz. A única mudança feita é sobre a

propagação de importância 2.10, no caso ela é re-escrita como:

$$\mathbf{x}'_i = \alpha \sum_j \frac{A_{ij}}{k_j^{out}} x_j + \beta \tag{2.14}$$

Como podemos ver a propagação é diluida pelas arestas que saem de dado vértice. Para a notação matricial introduzimos a matriz de grau D definida como:

$$D_{ij} = \begin{cases} k_i^{out} & \text{se } i = j \\ 0 & \text{se } i \neq j \end{cases} \tag{2.15}$$

Se por um acaso existir $k_i^{out} = 0$ ele é artificialmente modificado para 1.

Incorporando D em 2.14 obtemos:

$$\mathbf{x} = \alpha A D^{-1} \mathbf{x} + \beta \mathbb{1} \tag{2.16}$$

Novamente podemos resolver a equação para $\mathbf{x}$. Como β nesse caso é constante vamos utilizar $\beta = 1$. Portanto:

$$\mathbf{x} = (1 - \alpha A D^{-1})^{-1} \mathbb{1} = D(D - \alpha A)^{-1} \mathbb{1} \tag{2.17}$$

Nesse caso queremos que α esteja próximo do inverso do maior autovalor da matriz AD^{-1}. Novamente podemos ter valores de $beta$ diferentes para cada vértice, podemos então escrever o *PageRank* utilizando um vetor $\boldsymbol{\beta}$:

$$\mathbf{x} = (I - \alpha A D^{-1})^{-1} \boldsymbol{\beta} \tag{2.18}$$

2.4 Outras medidas

As medidas em redes não se limitam a medidas de centralidade de nó. Várias medidas estão interessadas em extrair outros tipos de propriedades da rede para realizar uma análise topológica da rede.

2.4.1 Menor caminho médio

Chamado daqui em diante simplesmente de caminho médio, por simplicidade, este é um exemplo de medida que pode ser utilizada tanto localmente quanto globalmente. A variante local se preocupa com o caminho médio entre um dado nó e todos os outros

pertencentes a rede, enquanto a variante global é a média do caminho médio de todos os nós. O caminho médio de um nó i pode ser escrito como:

$$d_i = \sum_j d_{ij} \tag{2.19}$$

onde d_{ij} é a menor distância entre os nós i e j. Podemos então escrever a variante global como:

$$d = \sum \frac{d_i}{N} \tag{2.20}$$

Onde N é o número total de nós.

É importante notar que essa medida não pode ser aplicada para grafos desconexos, pois não é possível calcular a distância entre dois nós entre os quais não existe um caminho.

2.4.2 Coeficiente de Clusterização

Assim como o caminho médio essa medida apresenta uma variante local e uma global. A clusterização mede de fato a quantidade de triângulos existentes na rede (grupo de três vértices onde existem arestas entre todos os pares).

Para normalizar a medida é utilizado o número de pares de vizinho de dado vértice. Temos então informalmente que o coeficiente de clusterização local de um nó i é dado por:

$$C_i = \frac{\text{número de triângulos que } i \text{ faz parte}}{\text{número de pares de vizinhos de } i} \tag{2.21}$$

Formalmente escrevemos:

$$C_i = \frac{1}{2} \sum_{jk} \frac{A_{ij} A_{jk} A_{ik}}{\frac{k_i(k_i-1)}{2}} \tag{2.22}$$

Como estamos somando sobre todos pares de j e k, cada triângulo é contado duas vezes, por isso temos o termo $1/2$.

A variante global pode ser calculada como a média dos coeficientes locais:

$$C = \frac{1}{N} \sum_i C_i \tag{2.23}$$

Ela também pode ser calculada diretamente como:

$$C = \frac{\frac{\sum_{ijk} A_{ij} A_{jk} A_{ik}}{6}}{\frac{\sum_i k_i(k_i-1)}{2}} = \frac{\sum_{ijk} A_{ij} A_{ik} A_{ik}}{3 \sum_i k_i(k_i - 1)} \tag{2.24}$$

A medida de clusterização foi proposta em (WATTS; STROGATZ, 1998) para ser usada em conjunto com a distância média para classificar redes de pequeno mundo, que são descritas em 2.5.2.

2.5 Modelos de redes

O recente interesse pelo estudo de redes complexas se deve à descoberta que muitos sistemas reais, quando modelados utilizando redes, apresentam estruturas não aleatórias (COSTA *et al.*, 2007). Acoplado com o aumento da disponibilidade de dados das diversas áreas do conhecimento, muitos métodos foram desenvolvidos para conseguir lidar com tal tamanho e variedade de informação recentemente disponíveis. Computadores mais potentes também possibilitaram o estudo de redes anteriormente intratáveis.

A utilização de redes complexas em diversas áreas acarretou na identificação de características que acontecem nos diversos tipos de redes. Algumas dessas características são: presença de comunidades, distribuição do grau segundo uma lei de potência, propriedades de pequeno mundo, entre outras (COSTA *et al.*, 2007).

Para entender esses fenômenos, foram criados modelos matemáticos para a construção de redes que apresentassem as características presentes nas redes reais. Esses modelos tentam reproduzir os padrões que se repetem em redes reais para entender as implicações desses tipos de estrutura. Abaixo estão descritos os três modelos mais famosos de redes complexas.

2.5.1 Modelo aleatório de Erdös-Rényi

Inicialmente proposto em (ERDOS; RÉNYI, 1960), este é um modelo extremamente simples de redes. Ele pode ser descrito somente pela probabilidade p de haver uma ligação entre dados dois vértices e pelo número total de vértices N. Tendo esses dois valores, a criação de uma rede aleatória é bem simples: basta criar os N nós e para cada par de nó sortear, com probabilidade p, se existe uma aresta entre eles.

Alguns pontos importantes que devemos ressaltar são que: nesse tipo de rede há uma distribuição de Poisson dos graus dos vértices e o grau médio dos nós é dado por:

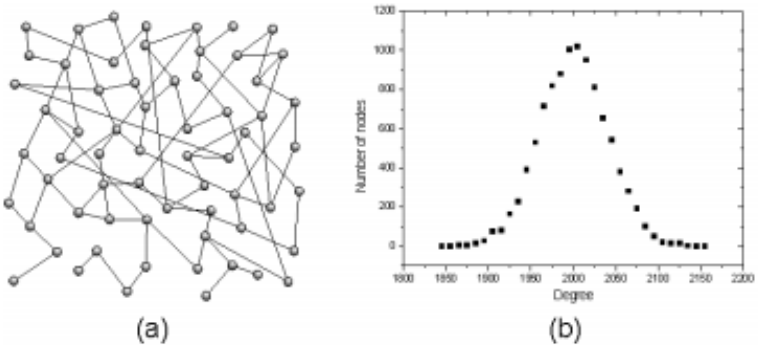
$$\langle k \rangle = p(N-1) \tag{2.25}$$

Interessante notar que o grau médio diverge para $N \to \infty$ e também que a estrutura da rede pode ser facilmente descrita pelo grau médio dela, bastando inverter 2.25 para obter:

$$p = \frac{\langle k \rangle}{(N-1)} \tag{2.26}$$

Na Figura 2 temos um exemplo de uma rede aleatória e a sua distribuição de grau.

Figura 2 – Em **(a)** temos um exemplo de rede aleatória e em **(b)** temos a sua distribuição de grau.



Fonte: Costa *et al.* (2007).

2.5.2 Modelo de Rede de Pequeno Mundo de Watts-Strogatz

Várias redes reais, especialmente redes sociais, exibem um tipo de comportamento muito específico: uma quantidade alta de triângulos na rede, ou seja, seu coeficiente de aglomeração alto, e seu caminho médio pequeno.

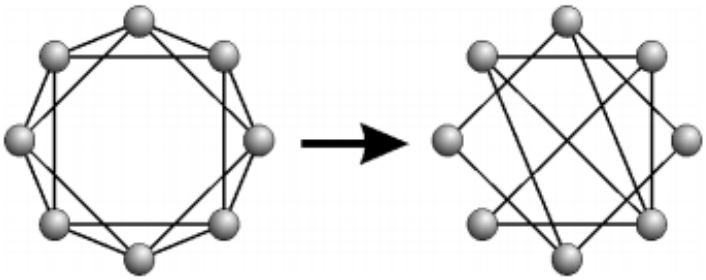
Redes com essas característica são estudadas desde o célebre experimento de Milgram (TRAVERS; MILGRAM, 1967), que concluiu que duas pessoas nos Estados Unidos estão separadas em média por seis conhecidos. Embora no estudo temos a conclusão sobre um caminho médio pequeno, é fácil notar que a quantidade de triângulos na rede é alta. Basta considerar uma pessoa A, amiga de B e de C. A probabilidade que B e C também sejam amigos é bem mais alta do que a probabilidade de duas pessoas arbitrárias serem amigas.

Segundo (WATTS; STROGATZ, 1998) essas características de pequeno mundo estão presentes em redes desde geração de energia até redes neurais de *C. elegans*. Para poder estudar redes com essas características sem precisar de redes reais, em (WATTS; STROGATZ, 1998) é proposto um modelo para aleatoriamente gerar redes do tipo pequeno mundo.

Os parâmetros necessários para construir a rede são o número de vértices N, o número de conexões iniciais κ e uma probabilidade de re-ligar uma aresta p. Dado o número de vértices e o número de conexões iniciais, é montada uma rede regular, onde cada vértice está ligado aos seus κ vizinhos mais próximos. É importante notar que nessa configuração inicial, a rede apresenta um coeficiente de aglomeração muito alto, porém a distância média é alta. Após ter a rede regular montada, cada aresta tem uma chance p de ser religada aleatoriamente a um outro vértice, como visto na Figura 3. Para se obter uma rede final com as características desejadas é necessário cumprir alguns requisitos.

Como pode-se ver facilmente, se $p \to 1$, todas as arestas serão religadas e não haverá resquícios da estrutura original. Sendo assim teremos uma rede aleatória, que exibe um

Figura 3 – Processo de criação de uma rede de pequeno mundo.



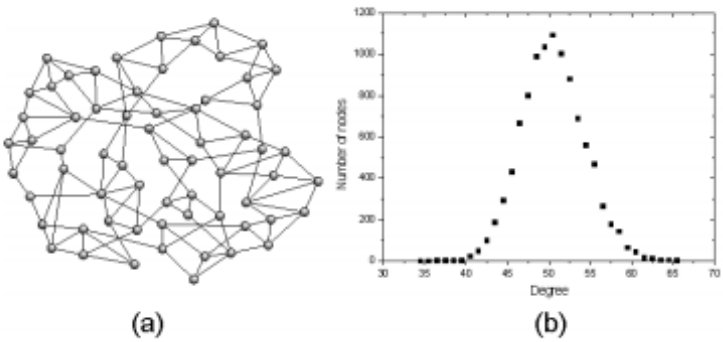
Fonte: Watts e Strogatz (1998).

.

comportamento oposto à rede regular inicial, ou seja, o menor caminho médio será baixo, assim como o coeficiente de aglomeração. Claramente o valor de p também não pode ser muito baixo pois a estrutura da rede se manterá ainda muito regular e o menor caminho médio continuará sendo grande. Queremos então um valor intermediário capaz de capturar as duas características desejadas. O comportamento desejado pode ser visto na Figura 4 (a). A distribuição de grau se assemelha a uma distribuição normal e exibe um pico em $\langle k \rangle = \kappa$, o número de conexões iniciais, como visto na Figura 4 (b).

Uma boa diretriz para montar a rede é utilizar um valor de κ não muito pequeno em comparação com o tamanho da rede e nem muito grande. Nota-se que quando o número de arestas se aproxima do número de vértices, embora tenhamos as qualidades propostas, não é uma rede interessante pois mesmo se $p = 0$ e se $p = 1$ não haverá uma variação significativa da rede. Por outro lado uma rede muito esparsa sempre terá um caminho médio elevado.

Figura 4 – Em **(a)** temos um exemplo de rede de pequeno mundo e em **(b)** temos a sua distribuição de grau.



Fonte: Costa *et al.* (2007).

2.5.3 Redes Livres de Escala de Barabási-Albert

Redes livres de escala, diferentemente dos dois modelos vistos anteriormente, não possuem um valor de grau característico, mas sim uma distribuição peculiar de grau. Nessas redes há uma distribuição desigual da conectividade, ou seja, exitem poucos vértices com grau muito alto e muitos vértices com grau baixo. Especificamente a distribuição segue uma lei de potência do tipo:

$$P(k) \sim k^{-\gamma} \tag{2.27}$$

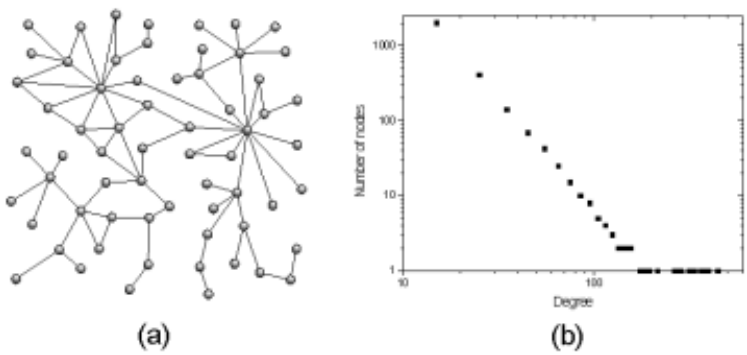
Um exemplo de uma rede livre de escala e sua distribuição de grau podem ser vistas na Figura 5.

Um aspecto interessante devido a essa distribuição é a existência dos *hubs*, os vértices com maior grau da rede. Tendo essa característica em vista, o modelo proposto em (BARABÁSI; ALBERT, 1999) propõe a construção da rede seguindo um padrão de crescimento dado pela ligação preferencial. Na prática, isso significa que a rede vai sendo construída vértice por vértice e para cada novo vértice criado as ligações a serem criadas vão preferencialmente ser feitas com vértices de maior grau.

Supondo um caso em que vamos colocar um novo vértice i em uma rede já existente, seguindo o modelo de (BARABÁSI; ALBERT, 1999) a probabilidade desse nó estar ligado a um dado vértice j é:

$$P(i \rightarrow j) = \frac{k_j}{\sum_u k_u} \tag{2.28}$$

Figura 5 – Em **(a)** temos um exemplo de rede livre de escala e em **(b)** temos a sua distribuição de grau.



Fonte: Costa *et al.* (2007).

2.6 Detecção de Comunidades e Modularidade

Uma das áreas mais estudadas em redes complexas é a área de detecção de comunidades. Muitos fatores contribuem para isso, mas a grande quantidade de problemas que podem ser resolvidos estudando estruturas de comunidades dentro de uma rede é o principal deles (NEWMAN, 2010). A detecção de comunidades em redes complexas se assemelha com o particionamento de grafos, inclusive compartilhando ao particionamento algumas ideias básicas, como a minimização de arestas entre duas partições ou comunidades.

O problema de detecção de comunidade pode ser definido informalmente como um problema de encontrar conjuntos de vértices que se ligam mais entre eles mesmos do que com vértices pertencentes a outros conjuntos. É importante notar que aqui surge a principal diferença em relação ao particionamento de grafos, pois para detecção de comunidades não é informado o número final de comunidades ou o número de membros de cada comunidade. Como veremos mais adiante esse relaxamento das condições do problema fez com que os métodos já existentes precisassem de algumas modificações para obter o resultado desejado.

Para definir formalmente o problema vamos utilizar o conceito de modularidade (NEWMAN; GIRVAN, 2004). A ideia de modularidade é calcular quão bem dividida uma rede está. A fórmula mais utilizada para calcular a modularidade Q é:

$$Q = \frac{1}{2m} \sum_{i,j} \left(A_{i,j} - \frac{k_i k_j}{2m} \right) \delta(c_i, c_j) \tag{2.29}$$

Onde m é o número de arestas da rede, A é a matriz de adjacência e k_i é o grau do nó i, δ é o delta de Kronecker e c_i é a comunidade à qual i pertence. A ideia dessa definição é que a modularidade pode ser definida como a diferença entre o número de arestas existentes dentro das comunidades dado por:

$$Q_1 = \frac{1}{2m} \sum_{i,j} A_{i,j} \delta(c_i, c_j) \tag{2.30}$$

e o número de arestas esperado, caso as arestas estivessem distribuídas aleatoriamente:

$$Q_2 = \frac{1}{2m} \sum_{i,j} \frac{k_i k_j}{2m} \delta(c_i, c_j) \tag{2.31}$$

A partir dessa definição podemos também encontrar um limite superior para a modularidade de uma rede: basta supor que para cada par de vértices, se eles estão conectados, então eles estão na mesma comunidade. Essa suposição claramente não é verdadeira a menos que a rede em questão não seja conexa, porém ela fornece um valor que pode ser utilizado para verificar quão boa são as comunidades propostas para a rede. A fórmula para essa

modularidade máxima é:

$$Q_{max} = \frac{1}{2m} \sum_{i,j} \left(A_{i,j} - \frac{k_i k_j}{2m} \right) A_{i,j} \tag{2.32}$$

Note que para obter o Q_{max}, basta substituir o δ de Kronecker pela entrada (i, j) da matriz de adjacência.

2.6.1 Métodos de Detecção de Comunidades

Historicamente o estudo de particionamento de grafos precede o de detecção de comunidades por algumas décadas, por isso é interessante analisar pelo menos alguns dos métodos mais tradicionais de particionamento e como eles podem ser adaptados para realizar detecção de comunidades.

2.6.1.1 Particionamento de Kernighan–Lin (KI)

O objetivo principal do particionamento de grafos não é a minimização da função de modularidade mas sim encontrar o corte mínimo do grafo, ou seja separar o grafo em dois grupos de modo a minimizar o número de arestas entre os dois grupos. Embora encontrar o corte mínimo não necessariamente maximize a modularidade do grafo, ela é uma aproximação boa para a separação de um grafo em somente duas comunidades.

Um dos métodos mais tradicionais para esse problema é o particionamento de Kernighan–Lin (KERNIGHAN; LIN, 1970). O método começa com a separação aleatória do grafo em dois grupos, onde a quantidade de nós pertencentes a cada um é pré-determinada. Depois de separados o método encontra o par vértice que, caso troquem de grupo, reduzem ao máximo o corte do grafo. A iteração se repete até que nenhuma a troca de grupo reduza o corte do grafo. O algoritmo pode ser descrito como:

1. Separar os vértices em dois grupos aleatoriamente com tamanhos determinados
2. Encontrar o par de vértices cuja mudança de grupo maximiza a diminuição do corte
3. Repetir 2 até não haver trocas que diminuam o corte

Como o número de elementos de cada grupo é pré-determinado, para encontrar a solução ótima para a modularidade, precisariam ser testadas todas as possibilidades de divisão. Embora seja possível realizar esse processo as $N/2$ vezes necessárias, isso acaba sendo custoso demais.

2.6.1.2 Particionamento Espectral

Um outro método comum para o particionamento é o método de particionamento espectral, ele recebe esse nome por se utilizar em parte da decomposição de uma matriz em seus autovalores e autovetores (FIEDLER, 1973).

Considerando que os nós de uma rede podem ser separados em dois grupos, podemos definir **s** como:

$$s_i = \begin{cases} 1 & \text{se } i \in g_1 \\ -1 & \text{se } i \in g_2 \end{cases} \tag{2.33}$$

Onde g_1 e g_2 são os dois grupos.

Sendo assim, podemos calcular o número de arestas entre os dois grupos como:

$$R = \frac{1}{4}\sum_{ij} A_{ij}(1 - s_i s_j) \tag{2.34}$$

É fácil notar que quando $s_i = s_j$ não há contribuição para R e o fator $1/4$ é necessário pois o par ij é o mesmo que ji e $1 - s_i s_j = 2$ quando $s_i \neq s_j$.

Abrindo a equação podemos escrever:

$$\sum_{ij} A_{ij} = \sum_{ij} k_i \delta_{ij} = 2m \tag{2.35}$$

Onde δ é o delta de Kronecker, k_i é o grau do vértice i e m é o número de arestas da rede. Sendo assim podemos defini L tal que:

$$L_{ij} = k_i \delta_{ij} - A_{ij} \tag{2.36}$$

Então:

$$R = \frac{1}{4}\sum_{ij} L_{ij} s_i s_j = \frac{1}{4}\mathbf{s}^T L \mathbf{s} \tag{2.37}$$

A partir de agora nós vamos tratar esse problema como um problema de otimização sobre **s**. As restrições sobre **s** são

$$\sum_i s_i^2 = n \tag{2.38}$$

$$\mathbb{1}\mathbf{s} = \sum_i s_i = n_1 - n_2 \tag{2.39}$$

Onde $\mathbb{1}$ é o vetor unitário, n_i é a quantidade de vértices pertencentes ao grupo i e $n = n_1 + n_2$ é o número total de vértices. Com essas condições podemos utilizar o método de multiplicadores de Lagrange para achar o valor extremo da função. Temos então para cada i:

$$\frac{\partial}{\partial s_i}\Big[\sum_{jk} L_{jk} s_j s_k + \lambda\Big(n - \sum_i s_i^2\Big) + 2\mu\Big((n_1 - n_2) - \sum_j s_j\Big)\Big] = 0 \tag{2.40}$$

Temos portanto para cada i:

$$\sum_j L_{ij} s_j = \lambda s_i + \mu \tag{2.41}$$

Ou em notação matricial:

$$L\mathbf{s} = \lambda\mathbf{s} - \mathbb{1}\mu \tag{2.42}$$

É fácil notar que pela definição de L temos:

$$\sum_j L_{ij} = 0 \tag{2.43}$$

Portanto, $\mathbb{1}$ é um auto vetor de L com autovalor 0. Sabendo que L é simétrica, podemos então encontrar uma relação entre λ e μ seguindo:

$$\mathbb{1}^T(L\mathbf{s}) = \mathbb{1}^T(\lambda\mathbf{s} - \mathbb{1}\mu) \tag{2.44}$$

$$(\mathbb{1}L)^T\mathbf{s} = \lambda\mathbb{1}^T\mathbf{s} - \mathbb{1}^T\mathbb{1}\mu \tag{2.45}$$

$$0\mathbf{s} = \lambda(n_1 - n_2) - n\mu \tag{2.46}$$

$$\lambda = \frac{n}{n_1 - n_2}\mu \tag{2.47}$$

Definindo:

$$x = s + \mathbb{1}\mu/\lambda \tag{2.48}$$

Como $L\mathbb{1} = 0$, temos:

$$L\mathbf{x} = L\mathbf{s} = \lambda\mathbf{s} - \mathbb{1}\mu = \lambda\mathbf{x} \tag{2.49}$$

Assim, $\mathbf{x}$ é auto vetor de L, cujo auto valor é λ. Além disso temos que :

$$\mathbf{x}^T L\mathbf{x} = \mathbf{s}^T L\mathbf{s} \tag{2.50}$$

Portanto:

$$R = \frac{1}{4}\mathbf{x}^T L\mathbf{x} = \frac{1}{4}\lambda\mathbf{x}^T\mathbf{x} = \lambda\frac{n_1 n_2}{n} \tag{2.51}$$

Para minimizar R basta selecionar o menor λ maior que 0. Como 0 é um auto valor de L precisamos selecionar o segundo menor auto valor pois o menor é 0. Note que ao selecionar um λ nós fixamos um $\mathbf{x}$, mas o que queremos de fato é o valor de $\mathbf{s}$ onde as entradas são 1 ou -1.

Primeiro precisamos encontrar $\mathbf{s}_x$ que corresponde ao $\mathbf{x}$. Para encontrar o $\mathbf{s}$ final, temos que maximizar o produto escalar $\mathbf{s}^T\mathbf{s}_x$. Queremos então:

$$\max(\mathbf{s}^T\mathbf{s}_x) = \max\left(\mathbf{s}^T\left(x - \mathbb{1}\frac{n1 - n2}{n}\right)\right) \tag{2.52}$$

Como $\mathbf{s}_i$ pode ser somente 1 ou -1 basta selecionar:

$$s_i = \begin{cases} 1 & \text{se } \left(\mathbf{x}_i - \frac{n1-n2}{n}\right) > 0 \\ -1 & \text{se } \left(\mathbf{x}_i - \frac{n1-n2}{n}\right) < 0 \end{cases} \tag{2.53}$$

Nesse método, assim como no método de particionamento de Kernighan–Lin, o número de elementos em cada grupo é imposto(dado pelo vínculo da 2.39) e o grafo é separado somente em dois grupos. No entanto, ideias básicas do particionamento espectral serão utilizadas no método a seguir para realizar a detecção de comunidades.

2.6.1.3 Detecção de Comunidade Espectral

A detecção de comunidades espectral, proposta em (NEWMAN, 2013), se utiliza do conceito de modularidade mas dispensa o vínculo 2.39. No entanto a normalização de $\mathbf{s}$ ainda é necessária.

$$\sum_i s_i^2 = n \tag{2.54}$$

Nesse caso ao invés de tentar minimizar o corte do grafo, vamos querer minimizar a modularidade Q dada por:

$$Q = \frac{1}{2m}\sum_{i,j}\left(A_{i,j} + \frac{k_i k_j}{2m}\right)\delta(c_i, c_j) \tag{2.55}$$

Usando $(1+s_i s_j)/2$ no lugar do δ temos:

$$Q = \frac{1}{2m}\sum_{i,j}\left(A_{i,j} - \frac{k_i k_j}{2m}\right)\frac{(1+\mathbf{s}_i\mathbf{s}_j)}{2} \tag{2.56}$$

Podemos então introduzir uma "matriz de modularidade" B como:

$$A_{i,j} - \frac{k_i k_j}{2m} = B_{ij} \Rightarrow Q = \frac{1}{2m}\sum_{i,j} B_{i,j}\frac{(1+s_i s_j)}{2} \tag{2.57}$$

Podemos reduzir utilizando:

$$\sum_i B_{i,j} = \sum_i A_{ij} - \frac{k_j}{2m}\sum_i k_i = k_j - k_j\frac{2m}{2m} = 0 \tag{2.58}$$

$$\Rightarrow Q = \frac{\mathbf{s}^T B\mathbf{s}}{4m} \tag{2.59}$$

Aplicamos então o método dos multiplicadores de Lagrange para encontrar os extremos da função modularidade:

$$\frac{\partial}{\partial s_i}\left[\sum_{jk} B_{jk}s_j s_k + \beta(n - \sum_j \mathbf{s}_j^2)\right] = 0 \tag{2.60}$$

E obtemos que $\mathbf{s}$ é autovetor de B:

$$2\sum_j B_j s_j - \beta s_j = 0 \Rightarrow B\mathbf{s} = \beta\mathbf{s} \tag{2.61}$$

Portanto:

$$Q = \frac{1}{4m}\mathbf{s}^T B\mathbf{s} = \frac{\beta\mathbf{s}^T\mathbf{s}}{4m} = \frac{\beta n}{4m} \tag{2.62}$$

Ao contrário do particionamento anterior queremos que $\mathbf{s}$ seja o auto vetor de B com o maior auto valor para obter um valor máximo para a modularidade. O procedimento para encontrar as comunidades é análogo ao do particionamento, tendo então que as entradas positivas de $\mathbf{s}$ são parte de uma comunidade e as negativas de outra.

Infelizmente esse método só é capaz de separar uma rede em duas comunidades. Podemos realizar esse método iterativamente até não haver uma quebra de comunidade que aumente a modularidade final. No entanto, cada separação é irreversível, ou seja uma vez que i e j são colocados em comunidades diferentes eles ficarão separados até o fim do método. Isso é um problema pois os erros vão acabar por se acumular a cada iteração.

Um segundo problema apresentado por esse método é o custo computacional, sendo $O(n^3)$ para matrizes de adjacência densas.

2.6.2 Detecção de Comunidades Multi-níveis

A detecção de comunidades multi-níveis se baseia em maximizar iterativamente a modularidade e no processo criar uma hierarquia de comunidades. Para isso é necessária uma fórmula simples para calcular a diferença de modularidade para o caso de um dado nó mudar de comunidade.

Para simplificar o problema, calculamos a diferença de modularidade quando adicionamos um nó i sem comunidade a uma comunidade existente c_k:

$$\Delta Q = \frac{1}{2m}\sum_j \left(A_{ij} - \frac{k_i k_j}{2m}\right)\delta(c_j, c_k) \tag{2.63}$$

Para facilitar, re-escrevemos a primeira parte do lado direito da equação como:

$$\sum_j A_{ij}\delta(c_j, c_k) = k_{i,in} \tag{2.64}$$

Onde $k_{i,in}$ é a soma arestas que ligam o nó i a nós pertencentes a comunidade c_j. Da segunda parte da somatória re-escrevemos:

$$\sum_j \left(\frac{k_i k_j}{2m}\right)\delta(c_j, c_k) = \frac{k_i \sum_{tot}}{2m} \tag{2.65}$$

Onde podemos interpretar $\sum_{tot}$ como a soma de arestas incidentes na comunidades c_k. No caso, se uma aresta conecta dois vértices dentro da comunidade c_k, essa aresta é contada duas vezes. Temos então:

$$\Delta Q = \frac{1}{2m}\left(k_{i,in} - \frac{k_i \sum_{tot}}{2m}\right) \tag{2.66}$$

Uma equação análoga pode ser escrita para a retirada de um nó da comunidade c_k. Sendo assim temos uma maneira fácil de calcular a variação de modularidade causada por uma troca de comunidade.

A intuição principal do método é a aglutinação de nós para formação de comunidades. A cada iteração do método são criados "super nós"que são a junção dos nós da comunidade detectada. Eles são chamados de nós pois eles guardam as informações relevantes de conexão i.e. o número de arestas internas a força de ligação com outros nós. Essas quantidades são importantes pois elas são utilizadas nas iterações posteriores. O algorítimo está descrito abaixo e ilustrado no Figura 6.

Agora pode-se descrever o algorítimo:

1. Inicializar cada nó como uma comunidade diferente.

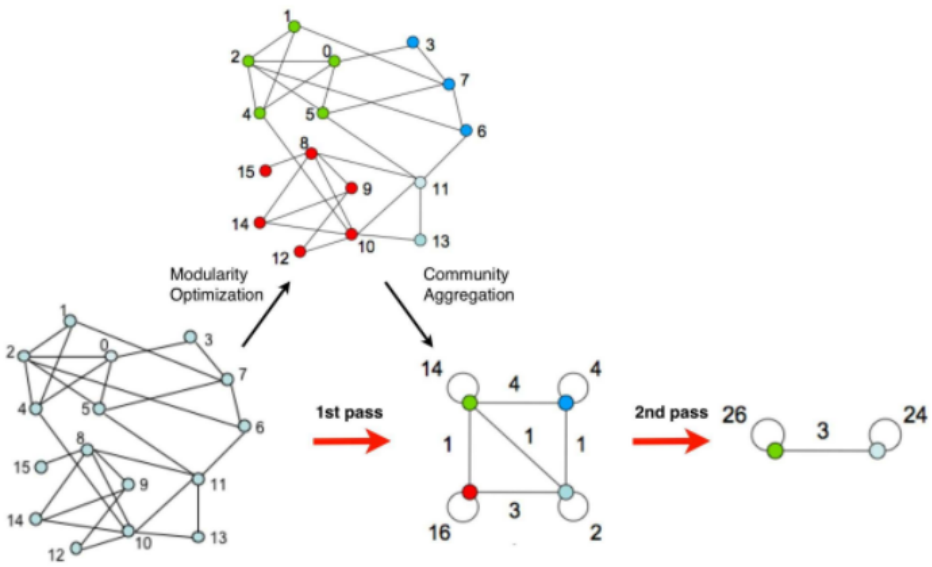
2. Para cada nó, verificar se alguma mudança possível de comunidade aumenta a modularidade. Se existir, a nova comunidade do nó analisado é a que maximiza a modularidade.

3. Repetir o passo 2 até não ser possível aumentar a modularidade. Cada nó pode mudar várias vezes de comunidade.

4. Aglutinar todos nós de mesma comunidade em um "super nó". Guardando as forças somadas de cada vértice, tanto para ligações internas quanto externas.

5. Repetir os passos anteriores até não ter aumento de modularidade.

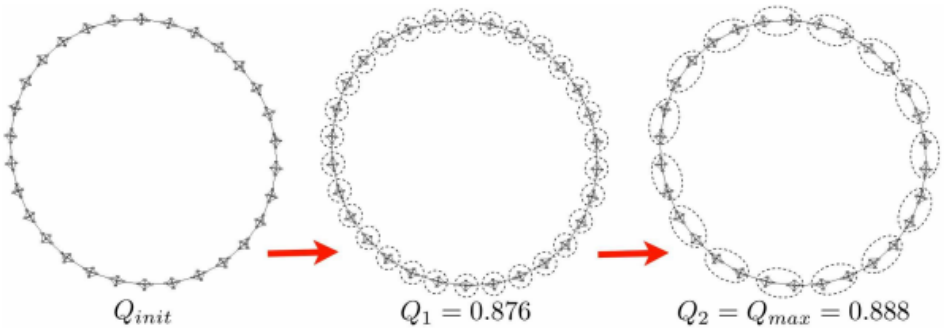
Figura 6 – Figura que ilustra o método de detecção de comunidade multi nível.



Fonte: Blondel *et al.* (2008).

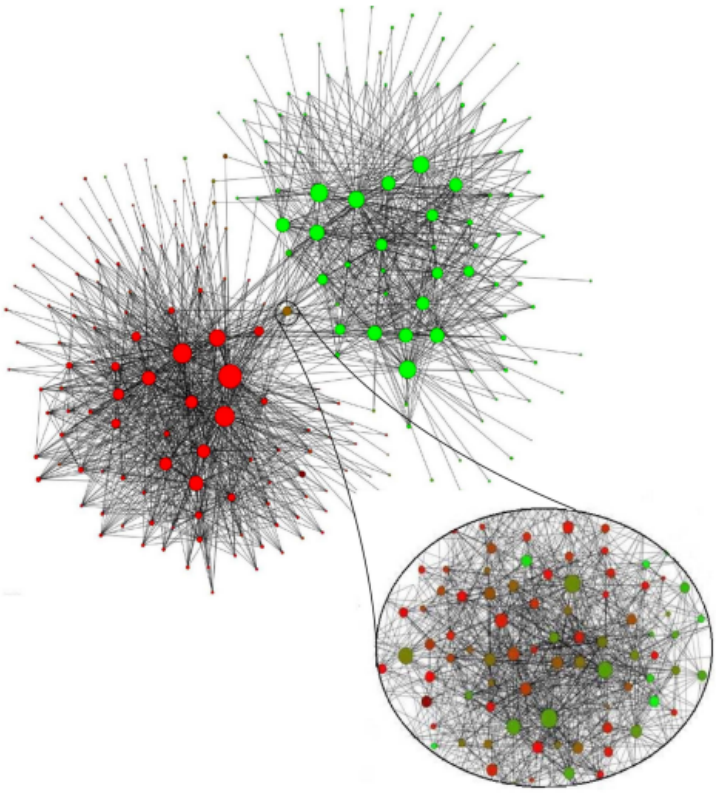
A cada passe do item 5 nota-se que temos uma nova ordem hierárquica. Essa propriedade é interessante pois ela pode ser útil para modificar a granularidade que queremos na nossa comunidade. Podemos ver nas Figura 7 que, pelo menos intuitivamente, as comunidades que não apresentam maior valor de modularidade seria uma solução melhor. A Figura 8 mostra um exemplo onde pode-se observar duas ordens hierárquicas para identificar comunidades menores que fornecem informações interessantes para análise.

Figura 7 – Exemplo de rede onde as comunidades que apresentam o maior valor de modularidade não condizem com o que seria esperado intuitivamente.



Fonte: Blondel *et al.* (2008).

Figura 8 – Exemplo da aplicação da detecção multi níveis sobre uma rede gerada a partir de dados telefônicos da bélgica. Cada nó é uma comunidade e a escala de cor indica a língua predominante da comunidade (verde para holandês e vermelho para francês). Podemos mudar de hierarquia para visualizar melhor os constituintes da comunidade intermediária.



Fonte: Blondel *et al.* (2008).

CAPÍTULO

3

TRABALHOS RELACIONADOS

3.1 Considerações Iniciais

Nessa seção serão apresentados trabalhos relacionados ao projeto. O principal trabalho (ARRUDA; COSTA; AMANCIO, 2015b) trata do uso de redes para a tarefa de segmentação, enquanto (HEARST, 1997) e (RIEDL; BIEMANN, 2012) tratam de outras abordagens para o problema. Também discutimos (ARRUDA *et al.*, 2016; ARRUDA; COSTA; AMANCIO, 2015a) que tratam de aplicação de redes complexas para problemas de PLN.

3.2 Segmentação de Tópicos

A segmentação em tópicos tem como objetivo identificar estruturas textuais coesas que tratam sobre o mesmo assunto. Em áreas como recuperação de informação e sumarização de textos (ANGHELUTA; BUSSER; MOENS, 2002; DIAS; ALVES; LOPES, 2007), é de grande interesse identificar se determinada parte do texto trata de um assunto único pois isso auxilia a identificação de partes que são de fato relevantes (DU; BUNTINE; JOHNSON, 2013).

O problema da segmentação de textos em tópicos já foi amplamente estudado (CHOI, 2000; RIEDL; BIEMANN, 2012; HEARST, 1997), porém muito pouco com o viés de redes complexas. Tradicionalmente, os métodos de segmentação estão divididos em duas categorias: linear e hierárquica (RIEDL; BIEMANN, 2012). Enquanto a segmentação linear investiga mudanças sequenciais em textos, ou seja, se duas sentenças consecutivas se encaixam no mesmo tópico ou não(RIEDL; BIEMANN, 2012), a segmentação hierárquica, se preocupa em criar uma hierarquia de tópicos dentro do texto, como visto em (EISENSTEIN, 2009).

3.2.1 Trabalho de (HEARST, 1997)

Introduzida em (HEARST, 1997), esse trabalho é um dos primeiros na área de segmentação de tópicos. Nele é introduzido o *TextTilling*, um algoritmo para a segmentação que é definido a partir de um índice de coerência. A ideia fundamental do índice é que quando há uma troca de tópico, há uma mudança abrupta de coerência. Dois métodos foram propostos para o cálculo da coerência: similaridade entre sentenças consecutivas e quantidade de vocabulário novo introduzido. O primeiro método em geral apresenta os melhores resultados e, por este motivo, a discussão da técnica *TextTiling* será baseada na similaridade de sentenças.

Antes de calcular a coerência, o texto é pré-processado por um método chamado de tokenização (de *token* do inglês), onde todas as palavras são transformadas para manter apenas os respectivos lemas. O lema de uma palavra é a forma básica dela. O lema de um verbo é seu infinitivo, para substantivos é a forma singular masculina do substantivo. Além disso, as pontuações são eliminadas e o texto é mapeado para letras minúsculas (esse processo também é discutido em 3.2.3). O texto então é separado em blocos (b), sentenças (s) e *tokens* (t). Mais especificamente, os *tokens* são os lemas que restaram das palavras (após passar pelo processo acima), sentenças são as sequências de *tokens* entre dois pontos e blocos são sequências de m sentenças consecutivas.

Tendo essas definições, é introduzido o índice de coerência de um dado ponto no texto. Esse índice tem o intuito de identificar se dois blocos consecutivos são sobre um mesmo tópico. Ele é calculado como:

$$I(j) = \frac{\sum_t w_{t,b_j} w_{t,b_{j+1}}}{\sqrt{\sum_t w_{t,b_j}^2 \sum_t w_{t,b_{j+1}}^2}}, \tag{3.1}$$

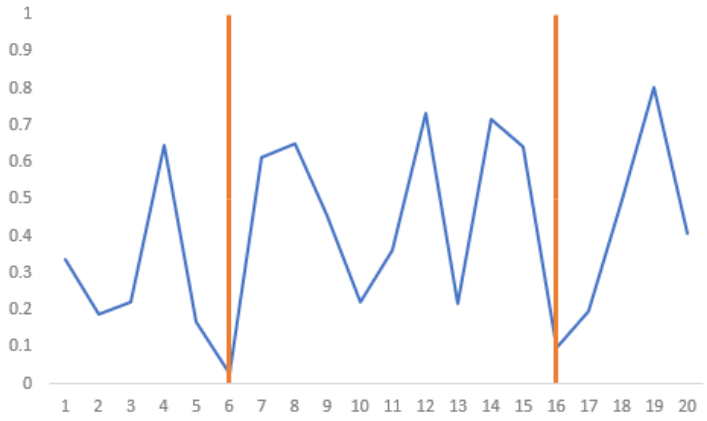
onde w_{t,b_j} é a frequência que o *token* t aparece no bloco b_j. Os valores de I variam entre $[0;1]$, sendo maior quando ambos os blocos apresentam frequências parecidas do mesmo $token$. A ideia do método é que se o mesmo *token* se repete em dois blocos consecutivos, é mais provável que ambos os blocos tratem do mesmo assunto, ou seja, eles são coerentes entre si. Tendo calculado todos os valores de I, pode-se desenhar um gráfico que mostra a variação do índice de coerência.

Assumindo que o índice de coerência é alto quando dois blocos compartilham o mesmo assunto, queremos identificar os pontos de mínimos locais do gráfico pois eles são possíveis quebras de tópico.

Na Figura 9, temos um exemplo do gráfico de I. É possível observar que nem todos pontos de mínimo local são possíveis quebras de tópico, já que esperamos que existam variações dentro do mesmo tópico, como indicado por barras verticais. Nesse método, são selecionados somente os pontos de mínimos menores que $\mu - \sigma/2$, onde μ é média e σ o desvio padrão da coerência. Note que, nesta abordagem, nenhuma informação sobre organização

do texto é utilizada, pois uma palavra pode estar em qualquer posição de uma sentença e o índice de coerência seria o mesmo.

Figura 9 – Exemplo de gráfico de coerência original ao longo de um texto. Os números indicam os parágrafos e as linhas verticais indicam os pontos de mínimo local de similaridade, cujo valor é inferior ao limiar sugerido.



Fonte: Elaborada pelo autor.

3.2.2 Trabalho de (RIEDL; BIEMANN, 2012)

O método *TopicTiling*, introduzido em (RIEDL; BIEMANN, 2012), se utiliza em partes do método *TextTiling* e também da técnica de Alocação Latente de Dirichlet (BLEI; NG; JORDAN, 2003) (do inglês LDA) para definir vetores que são comparados para calcular a similaridade entre duas sentenças. Como nos outros métodos, o texto é *tokenizado* e, adicionalmente, cada *token* recebe do LDA um ID numérico, que referencia um tópico (p.e. 1 - Política, 2 - Esportes). É importante notar que cada palavra pode receber diversos IDs, pois a distribuição de palavras no texto analisadas no documento influencia qual ID ela recebe. Neste método, as sentenças não precisam ser de mesmo tamanho, portanto são selecionadas de acordo com a pontuação. Blocos de sentença são então definidos como no método *TextTiling* e, para cada bloco, é definido um vetor de dimensão n, onde n é o número de IDs possíveis. A entrada i de um dado vetor é a quantidade de vezes que o ID i aparece no seu respectivo bloco. Finalmente, a similaridade entre dois blocos é calculada como o produto escalar normalizado entre seus respectivos vetores (similaridade cosseno). Com isso temos que a similaridade entre blocos varia de 0, para blocos pouco coerentes, até 1, para blocos muito coerentes.

Um gráfico da coerência calculado é então desenhado. Ele tem o mesmo intuito do gráfico do *TextTilling* ilustrado na Figura 9, porém a separação de tópicos é um pouco diferente.

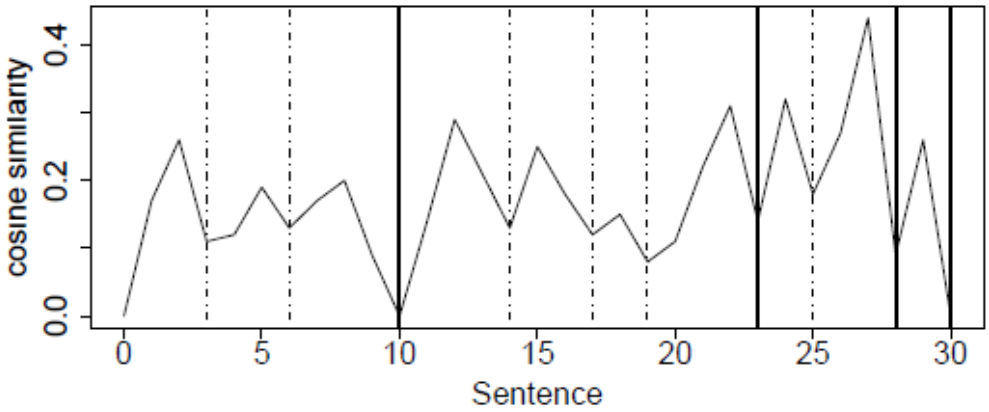
A intuição da separação proposta é que a quebra de tópico ocorre quando há uma

queda considerável na similaridade. Para determinar uma separação de tópicos, calcula-se a profundidade d_p de um mínimo levando em consideração as alturas do máximo à direita e à esquerda segundo a fórmula:

$$d_p = \frac{1}{2}(m_e(p) - c_p + m_d(p) - c_p), \tag{3.2}$$

onde m_e e m_d são os valores de máximo à esquerda e à direita respectivamente e c_p é o valor de coerência do ponto em estudo (essa é a média da diferença entre o mínimo estudado e os máximos mais próximos à esquerda e à direita). Para o ponto p ser considerado um ponto de mudança de tópico, o valor de d_p deve ser maior que $\mu - \sigma/2$, onde μ e σ são a média e o desvio padrão da similaridade do texto inteiro. Um exemplo do processo de segmentação é ilustrado na Figura 10. Note que a segmentação ocorre quando há uma queda brusca na similaridade de blocos vizinhos.

Figura 10 – Exemplo de gráfico de similaridade do TopicTiling. Os traços segmentados representam mínimos que não ultrapassam o limiar e os traços cheios as divisões que ultrapassam o limiar de profundidade de $\mu - \sigma/2$.



Fonte: Riedl e Biemann (2012).

3.2.3 Trabalho de (ARRUDA; COSTA; AMANCIO, 2015b)

Em (ARRUDA; COSTA; AMANCIO, 2015b) foram exploradas possibilidades de utilização de redes complexas para o problema da segmentação de tópicos. Aqui faremos uma revisão profunda do trabalho pois ele serve de base para o presente projeto.

Como visto em (RIEDL; BIEMANN, 2012; HEARST, 1997) a primeira etapa do método é o pré-processamento do texto, porém algumas etapa são adicionadas para montar redes mais descritivas. Além de retirar pontuações, deixar as palavras todas em minúscula e lematizar as palavras do texto, todas as palavras foram classificadas de acordo com as suas classes gramaticais utilizando um *tagger* do NLTK (BIRD; KLEIN; LOPER, 2009). A seguir todas as palavras que não eram substantivos ou verbos foram retiradas, pois elas podem ser utilizadas em vários contextos, independentemente do tópico. Também foram retiradas as *stopwords*

(palavras muito comuns com pouco conteúdo semântico). E finalmente cada palavra restante do texto, junto com a sua classe, formam uma única entidade, que se tornarão um nó das redes a serem criadas. As etapas desse processo podem ser vistas na Tabela 1

Tabela 1 – Exemplo de um pré-processamento utilizado em (ARRUDA; COSTA; AMANCIO, 2015b) sobre sentenças hipotéticas. Aqui identifiocamos as classes gramaticais com as siglas: NN para substantivos, VBZ para verbos, DT para artigos, ADP para preposições e JJ para adjetivos

Frase original	the boy plays with the leaves	this is not a good sentence
1- Lematização	the boy play with the leaf	this be not a good sentence
2 - Classificação	the:DT boy:NN be:VBZ with:ADP the:DT leaf:NN	This:DT be:VBZ not:ADV a:DT good:JJ sentence:NN
3 - Remoção por classe	boy:NN play:VBZ leaf:NN	be:VBZ sentence:NN
4 - Remoção de *stopwords*	boy:NN play:VBZ leaf:NN	sentence:NN

Fonte: Elaborada pelo autor.

Tendo o texto pré processado podemos começar a criar redes com eles. No artigo são propostas três maneiras para criar a rede:

- Modelo de co-ocorrência estendido: nesse modelo, as palavras do texto pré processado são ligadas se elas estão a uma distância menor ou igual a um determinado limite. Portanto, se temos a sequência w_1 w_2 w_3 w_4 e um limite igual a 2, teremos as ligações $w_1 \leftrightarrow w_2$, $w_1 \leftrightarrow w_3$, $w_2 \leftrightarrow w_3$, $w_2 \leftrightarrow w_4$ e $w_3 \leftrightarrow w_4$.

- Modelo baseado em parágrafos: nesse modelo, considera-se que um parágrafo é uma estrutura fundamental de um tópico. Portanto, somente palavras pertencentes a um mesmo parágrafo podem ser ligadas. No entanto ainda é utilizada uma distância limite para a ligação entre palavras. Note que esse modelo é igual ao anterior, porém sem ligações entre palavras de parágrafos diferentes.

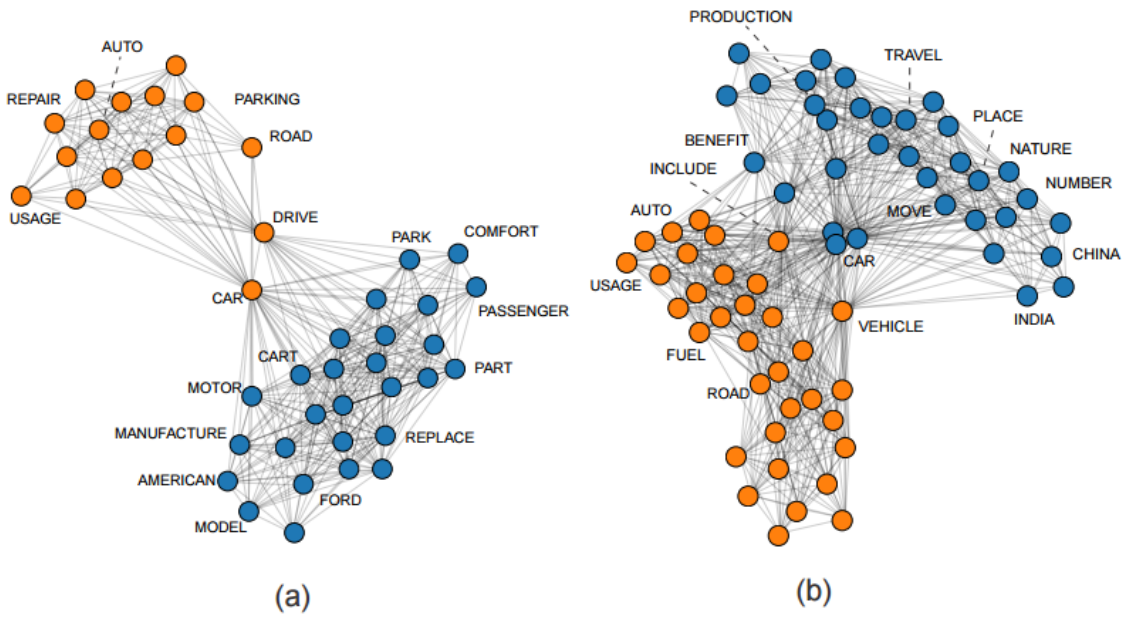
- Modelo adaptado de parágrafos: esse modelo tenta minimizar o efeito de ligações que não são estatisticamente relevantes. Dentro de um dado texto são consideradas conexões não relevantes aquelas que acontecem unicamente ao acaso. Este é o caso de pares de palavras muito frequentes: se elas ocorrem com alta frequência, existe uma alta chance de elas aparecerem juntas mesmo em textos sem significado (aleatórios). Por esse motivo, para cada par de palavras, o modelo apenas considera que a interação entre elas é relevante se elas co-ocorrem um número de vezes significativamente maior do que o esperado ao acaso, em textos aleatórios.

Em todos os modelos é necessário incluir um parâmetro d para a distância máxima entre palavras a serem conectadas. Em (ARRUDA; COSTA; AMANCIO, 2015b) é utilizado $d = 20$ para todos os modelos.

Tendo criado uma rede textual, é necessário então distinguir onde ocorrem as quebras de tópicos no texto. As redes propostas acima não tratam sentenças ou parágrafos explicitamente pois elas são constituídas somente por palavras isoladas. Supomos então que, caso seja possível determinar um grupo de palavras que aparece junto várias vezes, esse grupo de palavras vai formar uma comunidade na rede e essa comunidade estará ligada a um tópico do texto.

Usando técnicas de detecção de comunidade em redes complexas (Seção 2.6.2), torna-se possível encontrar o agrupamento de palavras. Supondo que temos as comunidades de determinada rede (ver Figura 11), o procedimento de atribuição é simples: inicialmente, o texto original é recuperado, parágrafo por parágrafo. A seguir, contamos quantas palavras pertencentes à cada comunidade existem dentro de cada parágrafo. Finalmente, para cada parágrafo, consideramos que ele é um representante do tópico equivalente à comunidade com maior número de ocorrência dentro dele. A segmentação então ocorre quando dois parágrafos consecutivos são dominados por representantes de duas comunidades diferentes.

Figura 11 – As figuras representam exemplos de redes geradas, nas quais foram detectadas comunidades. A figura (a) é um exemplo de rede gerada utilizando o modelo de co-ocorrência estendido. A figura (b) ilustra uma rede gerada pelo modelo baseado em parágrafos. É possível observar que cada palavra pertence a uma comunidade; desta forma, podemos contar qual comunidade está dominando cada parágrafo.



Fonte: Arruda, Costa e Amancio (2015b).

Algumas considerações importantes devem ser feitas sobre o artigo, no entanto. O *dataset* utilizado foi especialmente criado para o artigo. Este *dataset* é formado por 40 textos retirados da wikipédia sendo que cada texto tem o seu sub-tópico próprio, e pertence a um grande tópico que abrange diversos subtópicos. Foram selecionados oito tópicos diferentes, sendo que cada um contém 5 sub-tópicos. A utilização de um *dataset* próprio se deve ao fato dele não tentar competir com os métodos existentes e sim confirmar a possibilidade de utilização de métodos baseados em redes. O uso de um *dataset* próprio serviu especialmente para ter controle sobre o tipo de textos que seriam analisados. Era de especial importância ter controle sobre a quantidade de tópicos presentes nos textos e a quantidade de parágrafos para cada tópico.

Ter controle sobre os textos a serem analisados permite analisar em quais casos o segmentador obtém melhores performances. Também é possível ver qual estratégia de construção de rede conseguiu os melhores resultados para cada número de tópicos e parágrafos. Em métodos propostos vamos construir sobre os problemas apresentados pelo métodos aqui apresentados. Os resultados desses método estão dispostos na Seção 5 para facilitar a comparação com os resultados preliminares.

3.3 Outras trabalhos relacionados

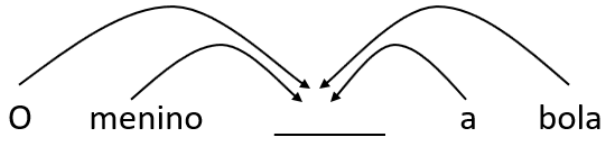
3.3.1 Trabalho de (MIKOLOV et al., 2013)

O trabalho de (MIKOLOV *et al.*, 2013) propõe um método chamado de word2vec que é baseado em uma rede neural de duas camadas capaz de gerar vetores densos para representar palavras. Uma das características mais interessantes dos resultados obtidos é que a composição de vetores é interpretável, embora seja mais evidente em casos simples. O exemplo clássico é que se somar os vetores que representam rei com mulher e subtrair do resultado o vetor de homem, o vetor resultante é similar ao de rainha. Esse tipo de resultado já havia sido observado utilizando outros métodos de representação vetorial de palavras (MITCHELL; LAPATA, 2010), porém os resultados apresentados para a representação semântica do word2vec quando publicado eram o estado da arte.

O modelo em si foi estruturado como um problema de predição de palavras vizinhas, ou seja, dado um conjunto de palavras se quer predizer um outro conjunto de palavras que aparecem na vizinhança. No artigo dois métodos diferentes foram propostos: O *Continuous bag-of-word* (CBOW) e o *Skip-gram*. O CBOW, como ilustrado na Figura 12 recebe um conjunto de palavras até n de distância da palavra que quer ser predita. O modelo Skip-gram faz essencialmente o oposto, recebendo uma palavra como entrada e tentando predizer as palavras a até uma janela de n, como visto na Figura 13.

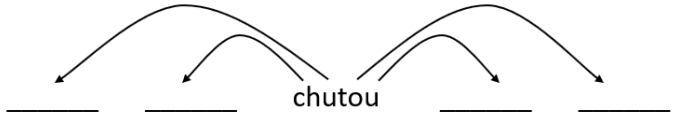
Para ter acesso aos vetores que são o resultado de interesse do artigo é preciso anali-

Figura 12 – Exemplo do método CBOW onde tenta-se predizer uma palavra dado as palavras adjacentes à ela.



Fonte: Elaborada pelo autor.

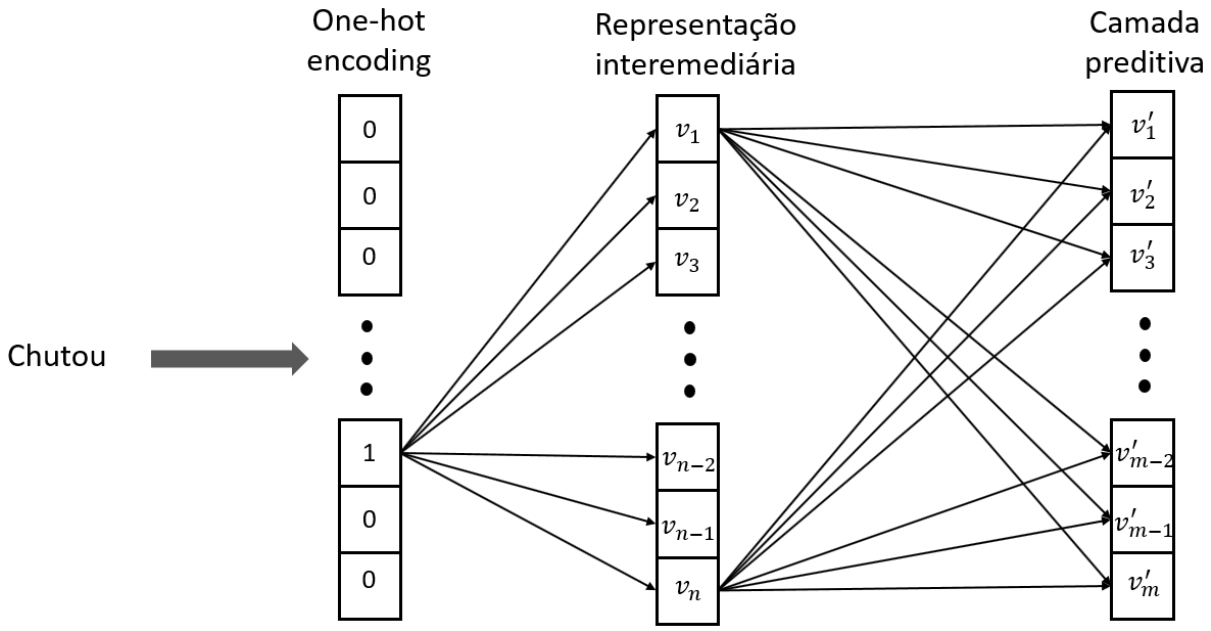
Figura 13 – Exemplo do método *skip-gram* onde dada uma palavra tenta-se predizer as palavaras adjacentes.



Fonte: Elaborada pelo autor.

sar o que ocorre dentro da rede neural. Como mencionado anteriormente, a rede neural é composta por duas camadas, uma camada escondida e uma camada de saída. A camada de saída tem tamanho do vocabulário que está sendo predito e a camada escondida pode ter um tamanho definido pelo usuário, como mostra a Figura 14

Figura 14 – Figura que mostra a arquitetura do w2v.



Fonte: Elaborada pelo autor.

Imaginando essa rede neural como uma composição de duas tarefas diferentes: a primeira, uma codificação da palavra de entras e a segunda, uma decodificação, podemos

ver que a saída da primeira camada é de fato uma representação para a palavra de entrada e pelo jeito que o treinamento é estruturado, palavras que aparecem em contextos similares e apresentam cargas semânticas similares terão codificações, na primeira camada, similares.

É importante notar que esse tipo de representação não é uma panaceia. Por exemplo, ela não lida bem com palavras que comumente são utilizadas com significados diferentes, como banco e manga. Outro problema é que é difícil extrair significado da composição de palavras de uma sentença inteira. Finalmente, outro ponto que vale se notar sobre o trabalho é que a ideia de utilizar a representação intermediária não é original dele O que fez com que esse método tivesse um impacto tão grande foi a eficiência do modelo que mesmo quando proposto podia ser treinado em menos de um dia em um corpus de mais de 1 bilhão de palavras (em inglês).

3.3.2 Trabalho de (ARRUDA; COSTA; AMANCIO, 2015a)

Em (ARRUDA; COSTA; AMANCIO, 2015a) são propostos métodos para diferenciar textos imaginativos de textos informativos. Para realizar a classificação dos textos, foram criadas redes textuais para cada um e sobre elas foram realizadas diversas medidas, incluindo o número de vértices na rede, modularidade e diversas medidas de centralidade.

As redes criadas foram redes de co-ocorrência simples, ou seja, cada palavra na rede só se conecta com palavras que aparecem adjacentes a ela no texto. o pré processamento do texto utilizado consistiu em classificar gramaticalmente as palavras, lematizá-las e em alguns casos retirar as *stopwords*.

Foram então propostas três estratégias para gerar características para a rede:

- **Estratégia global sem *stopwords*(EG)**: nela foram utilizadas redes sem *stopwords*. A estratégia consiste em realizar todas as medidas propostas e utilizar os valores globais delas como característica i.e. as características utilizadas foram as médias dos valores obtidos para cada palavra da rede. Para cada medida também foram utilizadas sua mediana, seu desvio padrão, seu máximo e seu mínimo.

- **Estratégia local sem *stopwords*(EL)**: como na EG as medidas foram realizadas sobre redes sem *stopwords*, porém cada valor de medida obtido para cada nó foi considerado uma característica.

- **Estratégia local com *stopwords*(ELS)**: Igual a EL, porém as redes analisadas contém *stopwords*.

Para a classificação dos textos foram utilizadas técnicas de seleção de características a fim de identificar quais medidas são mais discriminativas. Sobre as medias selecionadas e para cada estratégia foram aplicados métodos de aprendizado de máquina para classificar

os textos. Os métodos utilizados foram: KNN, Naive-Bayes e Árvore de Regração (CART). Os resultados obtidos estão dispostos na Tabela 2.

Tabela 2 – Acurácia para classificação de textos obtida por cada combinação de estratégia e método de AM.

Estratágia utilizada	KNN	CART	NB
EG	72%	78%	75%
EL	92%	92%	92%
ELS	98%	98%	98%

Fonte: Adaptada de Arruda, Costa e Amancio (2015a).

Os melhores resultados foram obtidos utilizando a estratégia ELS, independente do método de aprendizado de máquina utilizado. Isso é interessante pois mostra como em alguns tipo de classificação a utilização de *stopwords* melhora os resultados obtidos. Também é interessante notar que os resultados obtidos utilizando a EG foram consideravelmente piores que utilizando as outras estratégias, o que indica que as características de palavras individuais sejam importantes para a classificação.

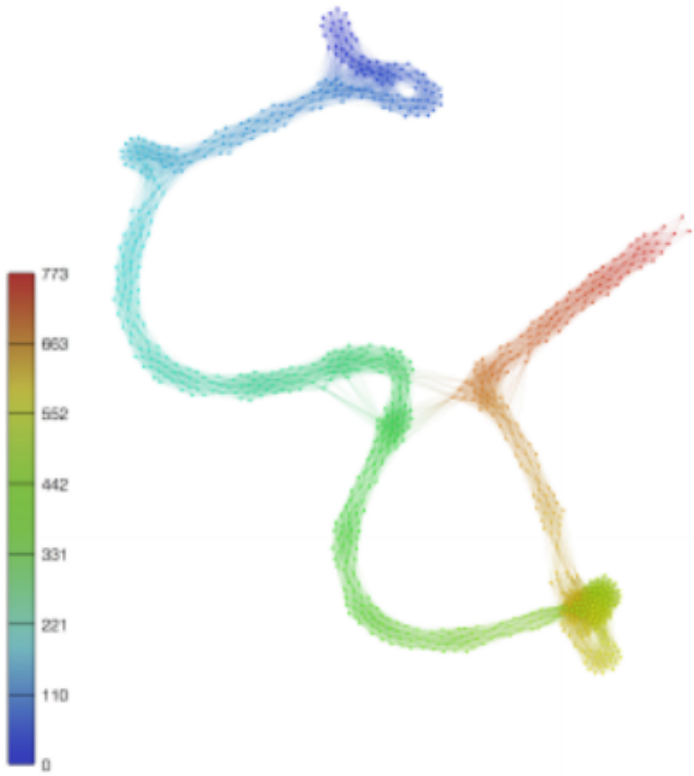
3.3.3 Trabalho de (ARRUDA et al., 2016)

Em (ARRUDA *et al.*, 2016) é proposta uma análise cujo resultado é interessante para o projeto. Nele é estudado a topologia de redes geradas a partir do livro Alice no País das Maravilhas. Para a construção da rede o texto é pré-processado, como em (ARRUDA; COSTA; AMANCIO, 2015b). No entanto a rede é criada utilizando sequências de parágrafos como nós e os nós são conectados de acordo com a similaridade entre eles.

O cálculo da similaridade entre sequências de parágrafos (blocos) é feita utilizando como base o tf-idf, do inglês *term frequency inverse document frequency*. O cálculo da similaridade consiste em primeiramente calcular, para cada palavra pertencente aos blocos a serem analisados, quantas vezes dada palavra apareceu no bloco em comparação com o total de vezes que ela apareceu no documento inteiro i.e. calcular o tf-idf das palavras no bloco. Tendo o tf-idf de todas as palavras, é então montado um vetor para cada bloco em que cada entrada é o tf-idf de determinada palavra. Naturalmente a $n-ésima$ entrada dos vetores de ambos blocos são baseados na mesma palavra. Sobre esses vetores podemos calcular a semelhança cosseno entre deles para termos uma aproximação da similaridade. Note que esse processo é similar aquele apresentado em (HEARST, 1997).

Para a montagem da rede, dois parâmetros foram definidos: o tamanho dos blocos e o limiar de similaridade para considerar a conexão de dois parágrafos como válida. No artigo, o tamanho dos blocos foi selecionado como 20. Dois limiares de similaridade foram utilizados no artigo: 0.24 e 0.46. Na Figura 15 podemos ver a rede gerada com o limiar mais baixo.

Figura 15 – Visualização da rede gerada utilizando o método proposto. As cores indicam a posição no texto de cada bloco.



Fonte: Adaptada de Arruda *et al.* (2016).

É interessante notar que é possível distinguir um caminho que segue o progresso do texto. Segundo (ARRUDA *et al.*, 2016) os pontos onde a conexão entre partes distantes acontecem quando há uma retomada de tópicos. Essa conclusão nos leva acreditar que esse tipo de método pode ser aplicado também a segmentação de tópicos.

CAPÍTULO

4

TRABALHOS REALIZADOS

4.1 Considerações Iniciais

No capítulo anterior foram discutidos os métodos já existentes para a tarefa de segmentação de tópicos e trabalhos sobre aplicação de redes complexas em tarefas relacionadas a PLN. Nesse capítulo dispomos as motivações e objetivos do mestrado, assim como a metodologia proposta para alcançar esses objetivos.

4.2 Motivação e Objetivos

A segmentação de textos em tópicos está relacionada a diversas aplicações muito importantes. Algumas das principais aplicações são: recuperação de informação, detecção de mudança de tópico e sumarização de textos (ANGHELUTA; BUSSER; MOENS, 2002; DIAS; ALVES; LOPES, 2007; HEARST, 1997). Na recuperação de informações, a segmentação auxilia na busca de palavras relacionadas àquelas utilizadas como palavras-chave. Esta tarefa também é relevante por ser capaz de selecionar quais partes de um documento são de fato relevantes para a busca, minimizando assim o esforço custoso e demorado de humanos. Para sumarização automática, a tarefa de segmentação é importante pois ela pode ajudar a garantir que o sumário final abranja todos os tópicos do texto original (ANGHELUTA; BUSSER; MOENS, 2002). Desta forma, permite-se que a sumarização possa ser aplicada em pontos individuais, tópico a tópico.

Tradicionalmente, a classificação de textos utiliza principalmente a informação das palavras presentes no texto, sem levar em consideração a disposição das palavras no texto. Durtante o mestrado utilizamos a organização das palavras ao longo do texto a partir de modelagens por redes, tendo como motivação o fato de que este tipo de informação tem sido relevante em várias tarefas relacionadas de análise textual, principalmente através do *framework* de redes complexas. A vantagem da modelagem por redes em relação à modela-

gens tradicionais, que não utilizam esse tipo de informação, é a sua capacidade de encontrar relações semânticas não triviais (ARRUDA; COSTA; AMANCIO, 2015b); um fator essencial para a captura de conhecimento em textos.

Uma segunda abordagem que ainda não tinha sido estudada é a utilização de conhecimento semântico raso para auxiliar na construção das redes. A existência de métodos computacionalmente baratos para medir a similaridade de palavras ou sentenças faz com que as redes a serem construídas sejam capazes de capturar mais informações por um custo computacional pequeno. Uma vez que redes complexas não foram exploradas em termos dos modelos baseados em *word embeddings*.

Em suma, o objetivo principal desse projeto foi o desenvolvimento de novas técnicas para a segmentação de textos em tópicos, usando métodos baseados em redes complexas. Como a informação introduzida por redes é complementar à informação tradicional (trazendo principalmente informação de conectividade), foi possível obter resultados otimizados através da combinação de técnicas tradicionais com as propostas para particionar textos em tópicos distintos de forma mais eficaz.

4.3 Metodologia

4.3.1 Avaliação e banco de dados

Os *datasets* tradicionais para a segmentação de tópicos foram propostos em (CHOI, 2000) e (GALLEY *et al.*, 2003). A principal diferença entre os dois é o grau de separabilidade dos subtópicos, sendo o *dataset* de Choi mais modularizado do que o dataset de *Galley*. Um problema que ambos *datasets* apresentam para o uso em métodos baseados em redes de sentenças é a presença de textos muito curtos, o que impossibilita a criação de uma rede capaz de apresentar comunidades bem definidas. Por essa razão esses datasets não foram considerados nesse trabalho.

O *dataset* utilizado em (ARRUDA; COSTA; AMANCIO, 2015b) é composto por 40 textos retirados da Wikipedia com 10 parágrafos cada. Para a avaliação, são criados textos artificiais que usam parágrafos dos textos originais. A quantidade de tópicos e a quantidade de parágrafos de cada tópico são escolhidos pelo usuário. A avaliação sobre esse *dataset* será utilizada principalmente como um *baseline* para comparações entre métodos baseados em rede. Por ser um *dataset* customizável pela variação da quantidade de tópicos e parágrafos, é possível obter informações sobre quão bem os métodos propostos se comportam sobre textos pequenos em comparação com textos maiores.

A métrica que será utilizada para avaliar a qualidade da segmentação é a P_k, proposta em (BEEFERMAN; BERGER; LAFFERTY, 1999). Essa medida se utiliza da ideia de verificar se para uma dada janela de sentenças o método identifica a quebra ou a continuidade de tópico

corretamente. Outra medida similar a ela é a *WindowDiff*, proposta em (PEVZNER; HEARST, 2002), sendo a diferença entre as duas que a *WindowDiff* tenta minimizar o problema do P_k que em determinadas situações apresenta um viés positivo para classificadores que consideram mais quebras de tópicos que o correto. Os resultados apresentados não levam ela em consideração pois dos trabalhos na literatura poucos utilizam ela como base.

4.3.2 Atividades propostas

Os trabalhos desse projeto estudaram modelos diferentes para representar textos por redes complexas. As modelagens propostas tiveram como objetivo facilitar a extração de informação semântica do texto. Também foram estudados métodos de detecção de comunidades para encontrar aqueles que produzem os melhores resultados. Os detalhes são fornecidos nas próximas seções.

4.3.2.1 Uso de redes de similaridade semântica

As modelagens sugeridas em (ARRUDA; COSTA; AMANCIO, 2015b) quantificam a similaridade semântica entre dois trechos de textos caso esses trechos compartilhem uma quantidade considerável de palavras. Embora essa abordagem seja interessante no cenário mais geral onde não existem recursos linguísticos para identificar relacionamentos mais complexos entre palavras, a ausência de conexões entre palavras similares pode levar a um desempenho inferior na tarefa estudada. Por esse motivo, um dos principais trabalhos realizados refere-se à utilização de métodos mais avançados de cálculo de similaridade entre trechos de textos. Uma das formas propostas utiliza o conceito de *word embeddings*, que é explicado a seguir.

Recentemente, trabalhos relacionadas a *word embedding* têm chamado muita atenção pelos resultados excelentes obtidos especialmente pelo método *word2vec* (MIKOLOV *et al.*, 2013). *Word embeddings* são representações vetoriais para palavras, que armazenam a semântica das mesmas. Uma das características mais interessantes dos *embeddings* é que é possível realizar operações com os vetores e obter resultados que são intuitivos para humanos. O mais clássico exemplo é $E(king) - E(man) + E(woman) \simeq E(queen)$ (MIKOLOV *et al.*, 2013), onde $E(palavra)$ é o *embedding* de determinada palavra. É interessante notar que os *embeddings* gerados pelo *word2vec* são calculados utilizando a co-ocorrência de palavras, geralmente a partir de grandes quantidades de textos. Em outras palavras, os vetores que representam a semântica de cada uma das palavras é obtido através de algoritmos de redes neurais que aprendem o contexto em que uma palavra tende a aparecer (MIKOLOV *et al.*, 2013). A dependência dos vetores gerados com o contexto pode ser visto através dos modos de treinamento geralmente utilizados, como discutido na seção 3.3.1. Na teoria, *embeddings* proporcionam uma maneira diferente e versátil para a modelagem de textos. Das possíveis aplicações desse método, foram estudadas:

- Redes de palavras conectadas pela similaridade dos *embeddings*: A alternativa mais direta para o modelo definido em (ARRUDA; COSTA; AMANCIO, 2015b) é a substituição das conexões geradas pela co-ocorrência de palavras em um texto por similaridade entre os *embeddings* das mesmas. Essencialmente, isso substitui as aproximações feitas em (ARRUDA; COSTA; AMANCIO, 2015b) por um valor de similaridade, que ao menos tem demostrado resultados bastante satisfatórios em comparação com medidas tradicionais (MIKOLOV *et al.*, 2013). Esta estratégia exigiu, como em (ARRUDA; COSTA; AMANCIO, 2015b), a escolha de um método para determinar ligações possíveis.

- Nós que representam sentenças: em (MITCHELL; LAPATA, 2010), são estudas maneiras para criar representações para composição de *embbedings* para representar uma sequência de palavras. No projeto realizado, foi utilizado o método mais direto para representar sentenças inteiras como nós em uma rede e criar ligações entre nós baseados na similaridade das composições geradas. Note que, desta forma, cada sentença foi associada a um tema após o processo de detecção de comunidades.

4.3.2.2 Métodos de detecção de comunidade

Existem diversos métodos para a detecção de comunidades em redes complexas, cada um com vantagens e desvantagens próprias (FORTUNATO, 2010). Parte desse projeto destinou-se a verificar qual (ou quais) dos métodos existentes alcançam os melhores resultados para a segmentação de tópicos. Os fatores estudados foram o custo computacional e a a performance final alcançada por cada método. Esse estudo foi realizado juntamente com os primeiros experimentos para definir rapidamente qual método seria utilizado para os experimentos seguintes.

4.3.2.3 Tratamento de palavras ambíguas

Na construção de redes de palavras, um problema recorrente é a presença de palavras ambíguas, como manga ou banco. Isso significa que essas palavras podem e devem fazer partes de comunidades diferentes na rede. Os métodos mais comuns de detecção de comunidades não são capazes de classificar um nó como pertencente a duas comunidades distintas. A ambiguidade também pode ocorrer no sentido de que uma mesma palavra pode aparecer em trechos distintos, mesmo que ela esteja relacionada a um mesmo sentido. Um exemplo seria um texto sobre carros ressaltando aspectos distintos de automóveis, por exemplo, peças mecânicas e marcas de luxo. Em ambos os casos, espera-se encontrar a palavra “carro”, que não deve ser tratada como sendo específica a um dos dois subtópicos apenas.

Para resolver o problema de ambiguidade mencionados, foi utilizado um método simples para atribuição de uma mesma paralvra para diversas comunidades. Esperava-se obter um ganho pois, nesse caso, uma palavra pode pertencer a mais de um tópico/comunidade,

com pesos distintos. Note que o uso de métodos de detecção de comunidades sobrepostas pôde ser aplicado tanto para redes de palavras convencionais, como em redes de sentenças.

4.3.2.4 Combinação de métodos

A abordagem de redes complexas para o problema de segmentação de tópicos introduz uma nova maneira de medir coerência textual, utilizada nos métodos apresentados no Capítulo 3. Por isso é interessante estudar como as diferentes metodologias se relacionam. Com isso pretendemos encontrar maneiras de combinar os métodos propostos no projeto com aqueles já existentes de modo a criar um segmentador melhor. Como visto no método *TopicTilling*, onde são combinados o LDA e o método *TextTilling*, temos resultados melhores utilizando métodos complementares (RIEDL; BIEMANN, 2012).

Durante o projeto serão estudados os problemas presentes tanto nos métodos existentes, quanto naqueles que serão propostos. Com isso serão combinados métodos que apresentam problemas que não se sobrepõem, para obter segmentadores melhores. Em específico, a classificação híbrida proposta em (AMANCIO, 2015b) será utilizada para compor a classificação advinda de fontes distintas.

4.3.2.5 Análise de dependência de parâmetros

Em todas as modificações sugeridas na formação de rede, é importante que as redes sejam capazes de se organizar de maneira modular. Se isso não for verdade, as técnicas de detecção de modularidade serão ineficientes. Por este motivo, diversos aspectos da criação de redes foram ser estudados. É de especial interesse para esse projeto verificar como se comportam as redes geradas a partir dos *embeddings* propostos, uma vez que por si só eles não apresentam nenhuma característica inteligível (i.e., os vetores só fazem sentido quando comparados).

Os parâmetros estudados foram:

- Limiar de similaridade: em todas as redes geradas por similaridade textual (de palavras ou sentenças, com ou sem *word embeddings*), é necessário estabelecer algum limiar para a montagem da rede. Este passo é fundamental no algoritmo e pode impactar fortemente o desempenho da tarefa, já que um limiar excessivamente baixo pode deixar a rede muito densa. Por outro lado, um limiar muito alto pode deixar a rede desconexa, pois muitas conexões informativas podem ser ignoradas.

  O problema do limiar foi tratado de algumas formas específicas. Primeiramente: estudamos como o desempenho da tarefa depende do limiar, a fim de verificar se esta quantidade de fato gera um diferença significativa na qualidade do segmentador de tópicos.

- Relevância das palavras: em geral, palavras muito comuns não carregam muito significado, porém como serão estudadas diversas formas de composição de palavras é interessante verificar a influência que elas exercem na qualidade do sistema. Palavras que aparecem em vários contextos poderão ser retiradas, em um passo de pré-processamento. Dessa forma, foi avaliado como a exclusão dessas palavras não informativas da rede (antes da construção da rede) pode contribuir para o desempenho da tarefa.

CAPÍTULO

5

MÉTODOS E RESULTADOS

5.1 Base de Dados

Como mencionado na Seção 4.3.1, a base de dados utilizada para os experimentos foi a mesma descrita em (ARRUDA; COSTA; AMANCIO, 2015b). A base em si é composta por 40 textos de 10 parágrafos, sendo agrupados em 8 tópicos diferentes, cada um com 5 textos representando subtópicos diferentes. Com esse *dataset* podemos montar textos artificiais com características muito bem determinadas de forma que temos no final textos padronizados pela quantidade de tópicos diferentes, quantidade de sub-tópicos por tópico escolhido, e quantidade de parágrafos para cada sub-tópico.

A montagem dos textos artificiais foi feita sempre seguindo o mesmo procedimento. Primeiro são definidas as quantidades de tópicos t, sub-tópicos s e parágrafos p. Tendo a quantidade t de tópicos diferentes, são sorteados t tópicos entre os 8 possíveis e, para cada um dos tópicos, são selecionados s subtópicos que estarão presentes no texto final. Para cada subtópico é então selecionada uma sequência de parágrafos de tamanho p. Finalmente, todas as sequências são dispostas de forma aleatória e todas palavras são lematizadas para obter o texto artificial que será estudado. Esse procedimento está ilustrado na Figura 16.

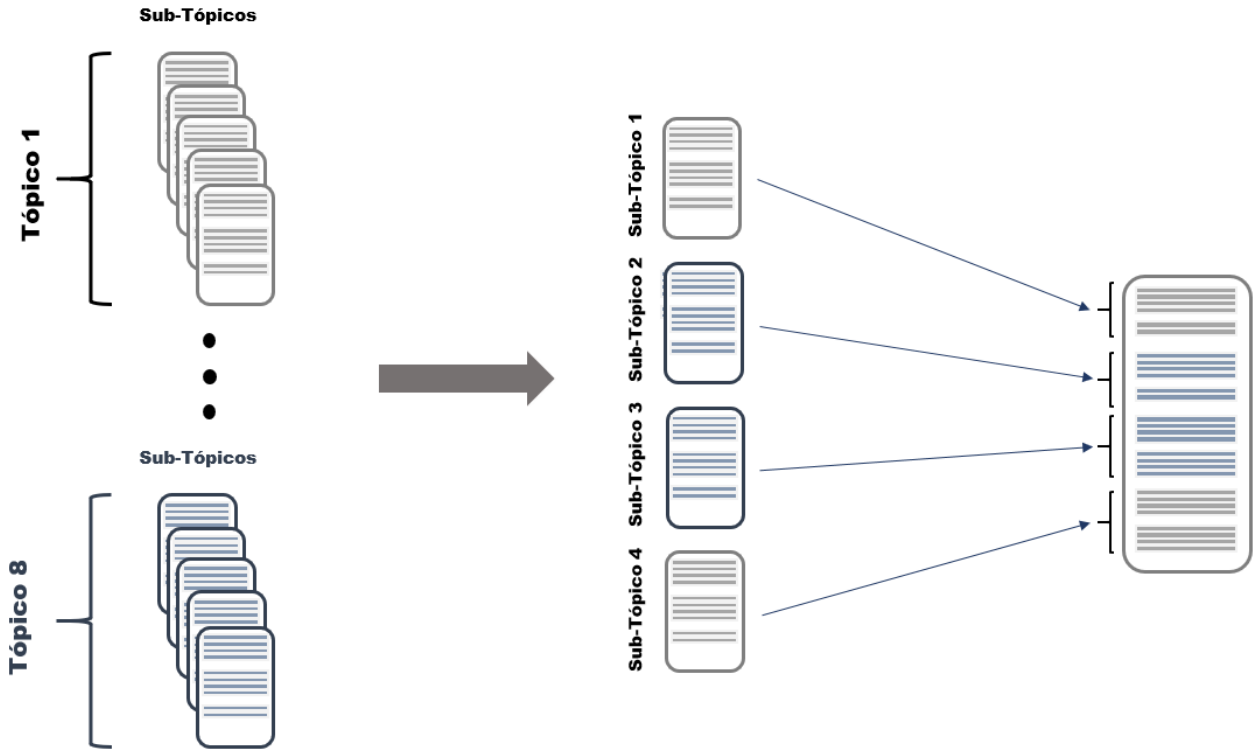
Os parâmetros que utilizados para a montagem dos textos artificiais foram os seguintes:

- Número de Tópicos: 3, 4, 5, 6
- Número de Sub-Tópicos: 1, 2
- Número de Parágrafos: 3, 4, 5

Todos os resultados que serão apresentados aqui foram realizados sobre o *dataset* gerado com esses parâmetros, sendo que para cada combinação de parâmetros foram feitos

100 textos artificiais.

Figura 16 – Representação do método de construção de textos artificiais. Primeiro os t tópicos são sorteados e então para cada um s sub-tópicos são selecionados. Finalmente para cada texto p extrai-se p parágrafos consecutivos concatena-se todos para formar um texto artificial completo



Fonte: Elaborada pelo autor.

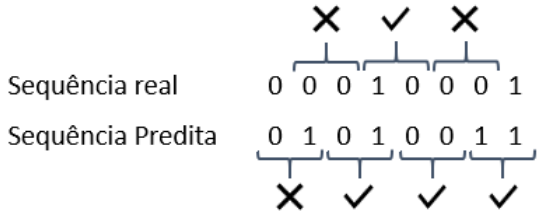
Para questões de avaliação, sempre que dois parágrafos consecutivos não pertencerem a um mesmo sub-tópico é considerado que entre eles n houve uma quebra de tópico.

5.2 Métrica

Para a avaliação de todos os resultados foi utilizado a medida Pk (BEEFERMAN; BERGER; LAFFERTY, 1999). O Pk mede efetivamente o erro normalizado de uma determinada segmentação. Ou seja, dada uma sequência de marcações de quebra ou não quebra predita, o Pk calcula um custo quando comparada com a sequência real. Para fazer isso, primeiro é definido um tamanho de janela. O valor da janela é calculado como metade do tamanho médio de sequências que não apresentam quebras, ou seja o número total de possíveis quebras dividido por duas vezes o número de quebras. Tendo esse valor, são comparadas as subsequências preditas e real definidas por essa janela que desliza por todo texto. Um acerto é definido como a comparação de duas sub-sequências que não apresentam nenhuma quebra ou duas sub-sequências em que ambas apresentam pelo menos uma quebra. Um

erro é quando essa condição não é satisfeita. A Figura 17 ilustra essa contagem. O valor final de Pk é a quantidade de erros dividido pela quantidade de acertos.

Figura 17 – Exemplo de verificação entre subsequências. Nesse exemplo o valor de Pk seria $\frac{3}{7}$



Fonte: Elaborada pelo autor.

5.3 Métodos e Resultados

Nessa seção serão descritos os temas que foram estudados durante os trabalhos de mestrado bem como os resultados. Para cada método abordado serão apresentadas figuras que demonstram a performance do método nas diferentes configurações de texto, as Tabelas 3, 4 e 5 dispõe os resultados compilados no final desse capítulo.

5.3.1 Métodos baseados em redes de similaridade semântica

Como discutido no Capitulo 4, podemos utilizar *word embddings* para a representação semântica de palavras. Duas ideias foram propostas para a utilização de *embeddings*. A primeira consiste em tentar representar sentenças inteiras como vetores e a segunda em utilizar a representação das palavras em si para aprimorar outros métodos de segmentação baseados em redes complexas.

É importante notar que em todos os momentos que um método de detecção de comunidade é utilizado, o método em questão é o multi-nível (BLONDEL *et al.*, 2008). Ele foi escolhido por ser um dos métodos mais eficientes e obter os melhores resultados. Os experimentos que foram utilizados para fazer a decisão estão descritos posteriormente na Seção 5.3.3.

5.3.1.1 Redes baseadas em sentença

A primeira possibilidade explorada foi a da utilização de *embeddings* pré-treinados utilizando *word2vec* (MIKOLOV *et al.*, 2013) para a representação de palavras e a representação de sentenças foi montada como simplesmente a soma dos vetores das palavras presentes na sentença. A identificação de sentenças e a lematização das mesmas foram realizadas utilizando o pacote NLTK para Python (BIRD; KLEIN; LOPER, 2009). Os *embeddings* utilizados

são os originários do artigo original (MIKOLOV *et al.*, 2013), sendo que utilizamos os vetores de tamanho 300. A biblioteca utilizada para fazer a leitura e operações sobre os *embeddings* pré-treinado foi o *gensim* (ŘEHŮŘEK; SOJKA, 2010). Tendo os vetores de todas as sentenças podemos montar uma rede de similaridade onde cada nó representa uma sentença do texto e as ligações são a similaridade entre duas sentenças. A ideia de representar o texto como uma rede permite que identifiquemos comunidades que possam aparecer espontaneamente, identificando assim tópicos no texto, sem a necessidade de passar como parâmetro a quantidade de tópicos no texto.

Uma vez que temos a rede e as comunidades detectadas, podemos adaptar o método proposto em (ARRUDA; COSTA; AMANCIO, 2015b), que foi discutido na Seção 3.2.3. No caso, ao invés de atribuir um tópico para cada palavra, atribuímos uma comunidade para cada sentença e montamos o nosso gráfico de similaridade entre parágrafos. Ou seja, cada ponto i do gráfico é a similaridade cosseno entre o vetor de contagem das comunidade presentes no parágrafo i com o vetor análogo do parágrafo $i+1$. Sobre o gráfico podemos aplicar a mesma heurística proposta em (RIEDL; BIEMANN, 2012) para identificar as quebras de tópicos.

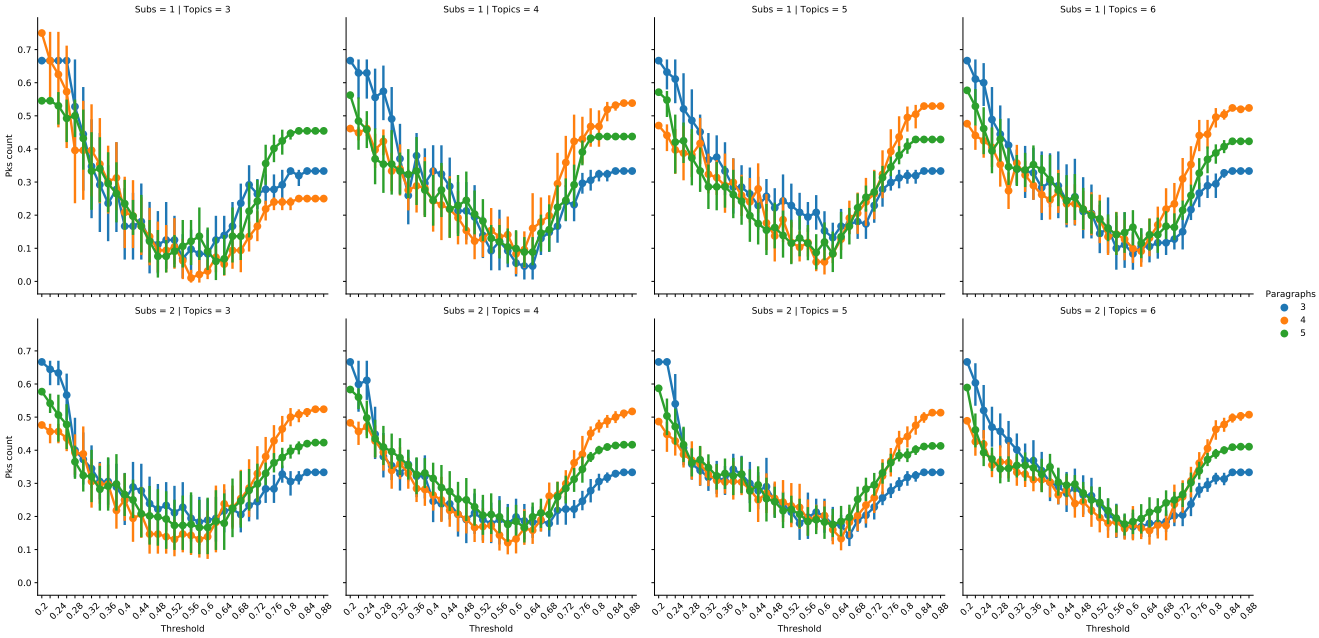
Um problema que surge quando se trabalha com esse tipo de rede de similaridade é que as comunidades não são muito claras, já que os valores de semelhança podem não variar muito. Para minimizar esse problema,foram aplicados diversos limiares de similaridade de modo que links com similaridade maior que o limiar eram mantidos. Depois de selecionar as ligações mais relevantes, retiramos o peso das ligações para aumentar o contraste.

Como a seleção de qual limiar deve ser utilizado impacta fortemente o resultado final, para cada configuração de número de tópicos, sub-tópicos e parágrafos, foi selecionado um limiar ótimo utilizando um *dataset* de treino, que foi construído exatamente como o *dataset* original, porém tomando cuidado para que as mesmas configurações presentes no original se repetisse nele. Os resultados obtidos no treino podem ser vistos no Figura 18. Os resultados obtidos no teste estão dispostos na Figura 19

Nos resultados podemos ver que quando o texto apresenta dois sub-tópicos diferentes os resultados são consideravelmente piores. Uma possibilidade de melhora para o método foi de limitar a distância entre sentenças conectadas na rede de similaridade. A ideia era que com mais subtópicos, a quantidade de ligações fica muito maior, sendo que muitas delas são só ruído, uma vez que não faz sentido sentenças muito distantes fazerem parte de um mesmo tópico.

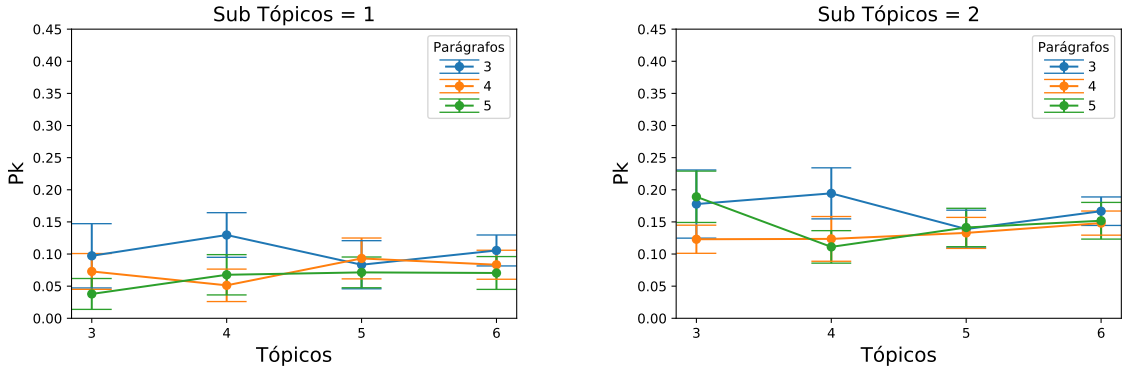
Fazendo o mesmo experimento que anteriormente, mas limitando ligações na rede para sentenças que estão a no máximo 3 parágrafos de distância, foram obtidos os resultados na Figura 20.

Figura 18 – Gráfico da performance do modelo para cada configuração de texto e para cada valor de limiar de similaridade.



Fonte: Elaborada pelo autor.

Figura 19 – Agrupado dos melhores resultados selecionados no *dataset* de treino e testados no *dataset* de teste



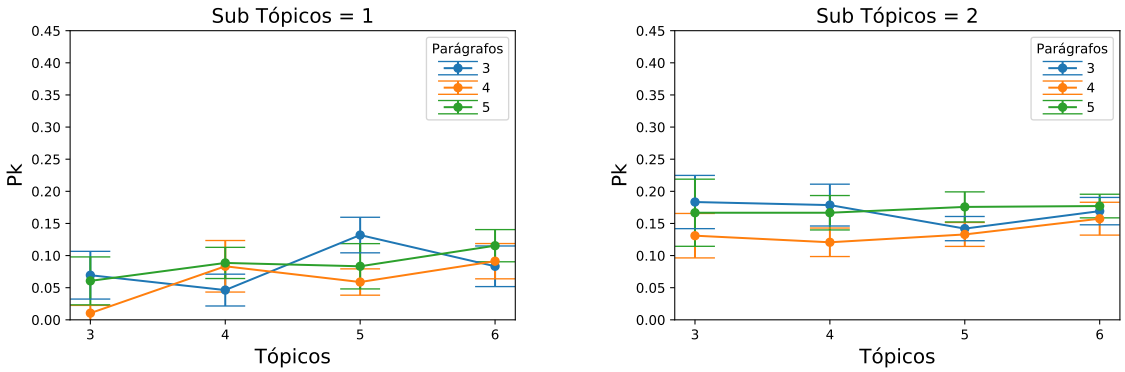
(a) (b)

Fonte: Elaborada pelo autor.

5.3.1.2 Redes baseadas em palavras

A criação de redes baseadas em sentenças, embora forneça um jeito simples para a criação de um valor de similaridade entre sentenças consecutivas muita informação pode ser perdida na somatória dos vetores. Para tentar minimizar isso foram desenvolvidos métodos para a utilização direta da similaridade entre palavras. Para isso foram feitas adaptações simples sobre os métodos propostos em (ARRUDA; COSTA; AMANCIO, 2015b) e discutidos na 3.2.3.

Figura 20 – Agrupado dos melhores resultados selecionados no *dataset* de treino e testados no *dataset* de teste limitando o alcance de ligaçoes na rede a 3 parágrafos de distância



(a) (b)

Fonte: Elaborada pelo autor.

A primeira mudança proposta consiste em transformar as ligações das redes do modelo baseado em parágrafos e do modelo adaptado de parágrafos para a similaridade segundo *word2vec* entre elas. Esses dois modelos foram escolhidos por serem aqueles que apresentaram os melhores resultados no artigo original. A janela de distância máxima entre palavras para que exista uma ligação entre elas foi considerada 20. Diferentemente do caso da rede se similaridade de sentença proposta, essas redes não são densas. Sendo assim, não foi necessário aplicar um limiar sobre a similaridade.

A segunda adaptação foi referente a identificação de quebras de tópicos. Originalmente, em (ARRUDA; COSTA; AMANCIO, 2015b), cada parágrafo era associado a um tópico que era o tópico mais comum nele. A ideia é utilizar a heurística do gráfico de similaridade. Onde novamente a similaridade entre um parágrafo e outro será a similaridade cosseno entre os vetores de contagem de aparição de cada tópico dentro do parágrafo

As Figuras 21, 22, 23 e 24 ilustram os resultado obtidos pelo método original e com as adaptações.

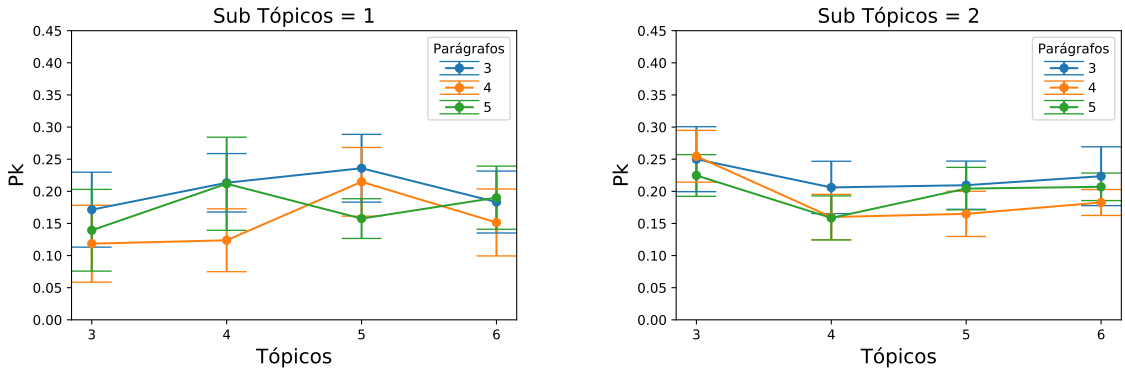
Para efeitos de comparação, foi estudado qual seria o impacto de utilizar cada adaptação separadamente. Os resultados estão dispostos nas figuras

5.3.2 Variação dos métodos

Sobre todos métodos propostos nessa seção, duas variações foram feitas. A primeira variação foi a remoção de *stopwords* dos textos. Como discutido na seção 3.2.3, a presença de palavras comuns mas sem carga semântica pode estar introduzindo ruído na identificação de quebra de tópicos. Os resultados estão dispostos na Tabela 4.

A segunda variação implementada foi a modificação da atribuição de tópicos para

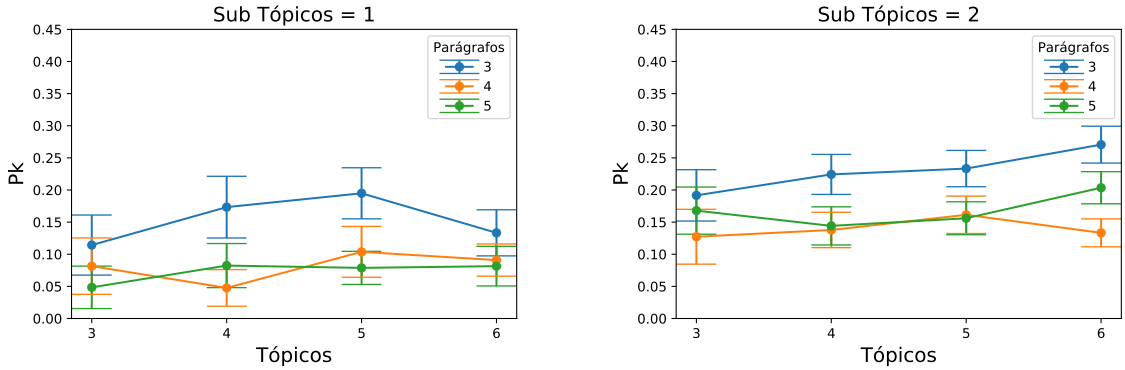
Figura 21 – Resultado do método adaptado de parágrafos de (ARRUDA; COSTA; AMANCIO, 2015b) sobre o *dataset* padronizado dessa dissertação



(a) (b)

Fonte: Elaborada pelo autor.

Figura 22 – Resultado do método estendido de parágrafos de (ARRUDA; COSTA; AMANCIO, 2015b) utilizando similaridade baseada em *word2vec* para as ligações na rede e gráfico de similaridade para identificar quebras de parágrafos



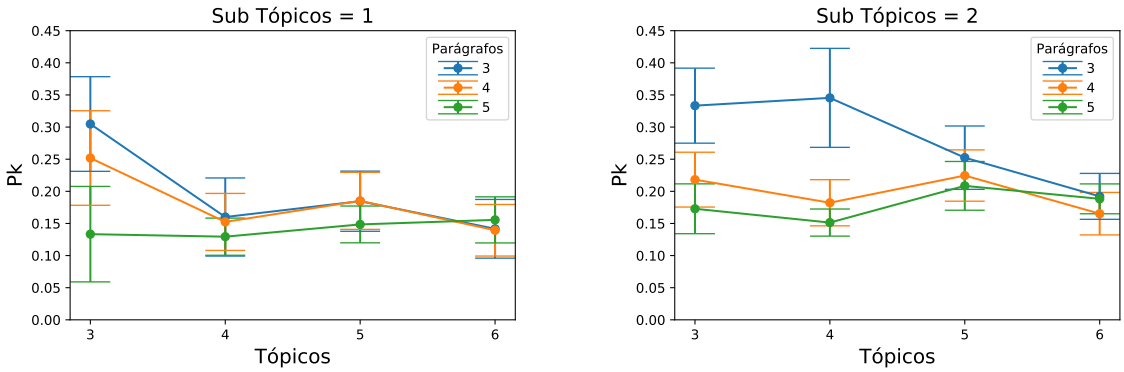
(a) (b)

Fonte: Elaborada pelo autor.

cada nó da rede, seja ela baseada em sentenças ou palavras. Quando a detecção de comunidades é realizada, cada nó da rede é atribuído a uma comunidade. Pode-se então ter uma rede como na Figura 26a. Nela é possível ver que existe um nós que intuitivamente poderia pertencer a qualquer uma das duas comunidades que só visíveis na imagem, porém os métodos de detecção vai atribuir ele a uma ou outra. E isso é o que espera-se ver com palavras ambíguas nos textos estudados. Se uma dada palavra aparece em dois contextos diferentes no mesmo texto, espera-se que ela pertença a ambas comunidades.

Como esse comportamento não é capturado pelo método de detecção utilizado, propomos o seguinte método de atribuição de comunidade: Ao invés de pertencer a uma comunidade cada nó terá um vetor de participação em cada comunidade. Ou seja se temos n comunidade em uma rede, cada nó tera um valor associado a cada uma das n comunidades.

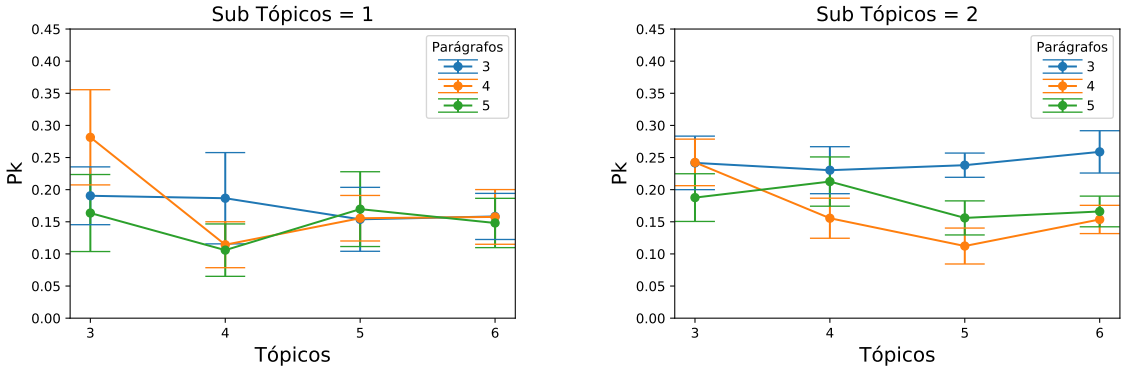
Figura 23 – Resultado do método estendido de parágrafos de (ARRUDA; COSTA; AMANCIO, 2015b) utilizando similaridade de *word2vec* para as ligações na rede



(a) (b)

Fonte: Elaborada pelo autor.

Figura 24 – Resultado do método estendido de parágrafos de (ARRUDA; COSTA; AMANCIO, 2015b) utilizando um gráfico de similaridade para identificar quebras de parágrafos



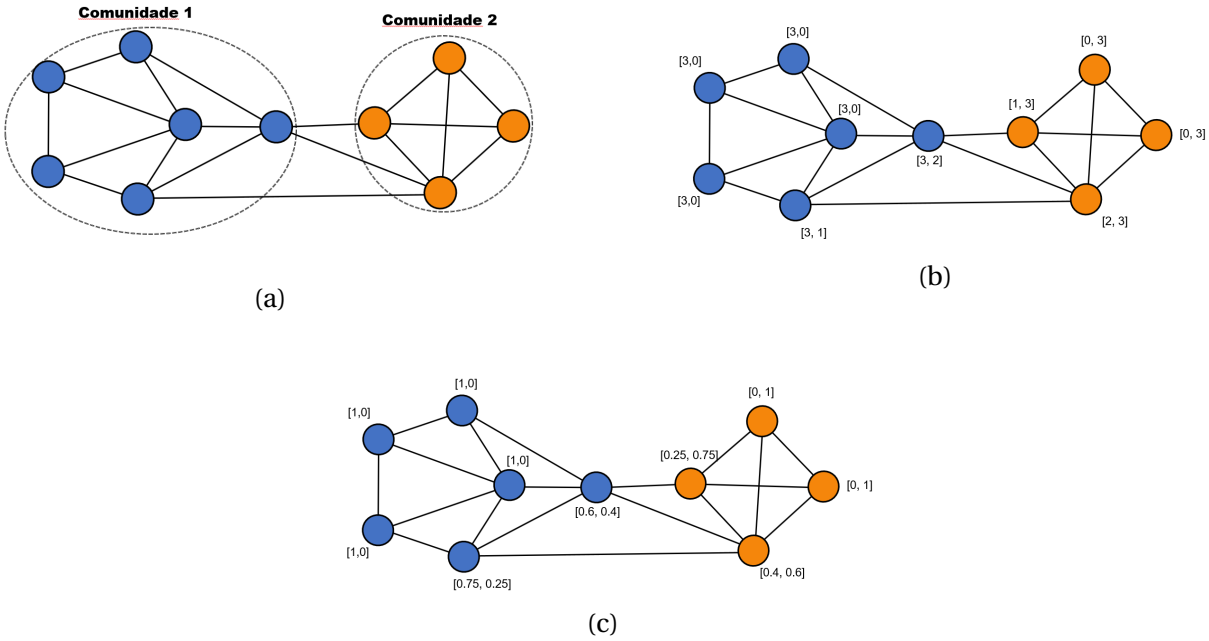
(a) (b)

Fonte: Elaborada pelo autor.

Esse valor é calculado como mostram as Figuras 26b e 26c: o valor de participação de um dado nó i em uma comunidade j é a quantidade de ligações que dado nó tem com nós que foram atribuídos da maneira tradicional à comunidade j, dividido pelo grau do nó i.

Sendo assim, para montar o vetor de cada parágrafo e posteriormente calcular a similaridade com o próximo parágrafo, ao invés de montar um vetor de contagem de ocorrência de cada comunidade dentro dele, é utilizada a soma simples dos vetores de comunidade de cada palavra do parágrafo. Os resultados obtido quando utilizamos as comunidades sobrepostas estão presentes na tabela 5

Figura 25 – (a) Exemplo de rede com duas comunidades onde alguns vértices apresentam bastante ligações para comunidade que não a dele. (b) Contagem de quantas ligações com nós de cada comunidade cada nó apresenta representada por um vetor. (c) Normalização dos vetores para que o pertencimento de cada nó em cada comunidade seja comparável.



Fonte: Elaborada pelo autor.

5.3.3 Seleção de método de detecção de Comunidade

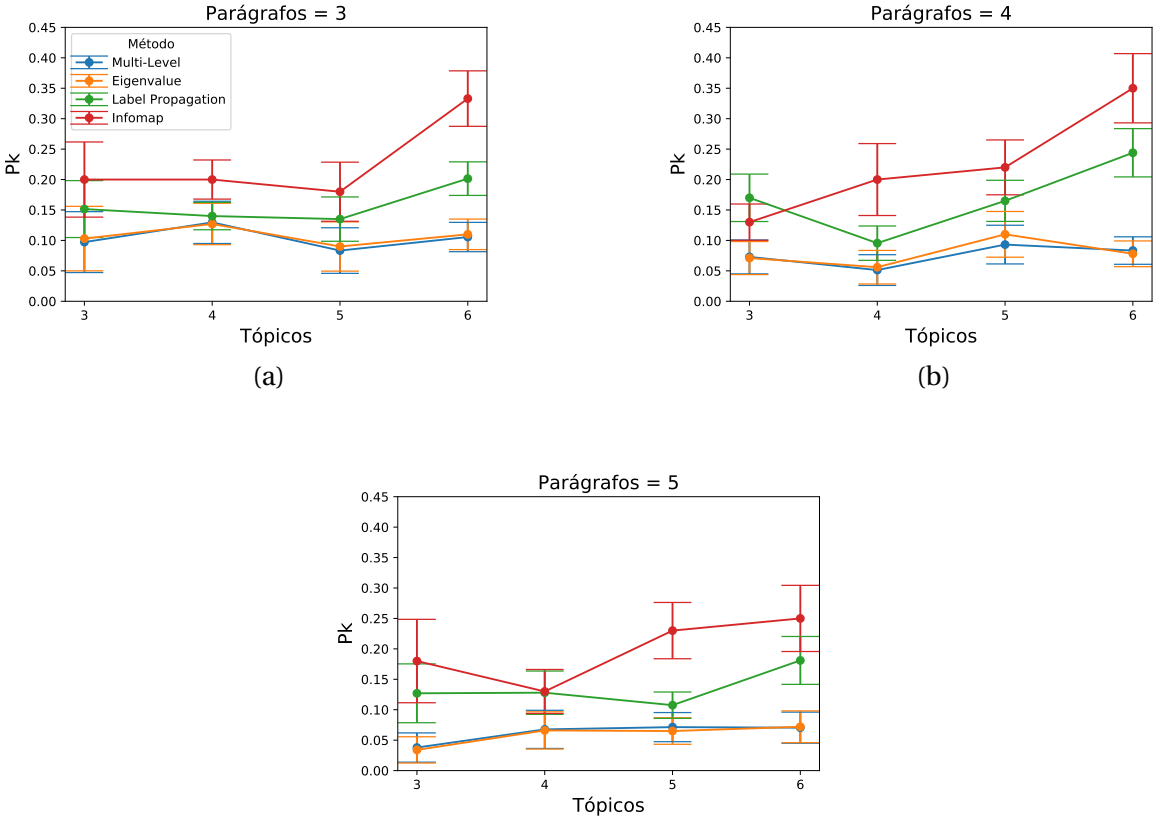
Como dito anteriormente foi necessário identificar um método de detecção de comunidade para ser utilizado em todos os métodos de segmentação propostos. Para isso, quando estudado o método de segmentação baseado em rede de sentenças sem limite de distância, foram testados 4 métodos de detecção diferentes:

- *Multi level* (BLONDEL *et al.*, 2008)
- *Eigenvalues* (NEWMAN, 2013)
- *Label propagation* (RAGHAVAN; ALBERT; KUMARA, 2007)
- *Info map* (ROSVALL; BERGSTROM, 2008)

Os resultados estão ilustrados na Figura 26:

Pode-se ver que os métodos multi-level e de *Eigenvalue* conseguem obter uma resposta final parecia e claramente superior aos outros dois métodos. A razão da escolha do multi-level para os testes é o fato dele ser consideravelmente mais rápido para executar.

Figura 26 – Resultado para a segmentação baseada em sentenças utilizando métodos diferentes para a detecção de comunidades. Os resultados apresentados são todos considerando os textos somente com um um sub-tópico por tópico



(a) (b) (c)

Fonte: Elaborada pelo autor.

Tabela 3 – Tabela com todos os resultados obtidos sem a remoção de *stopwords* ou a utilização de comunidades sobrepostas

Tópicos	3						4						5						6					
Sub-Tópicos	1			2			1			2			1			2			1			2		
Parágrafos	3	4	5	3	4	5	3	4	5	3	4	5	3	4	5	3	4	5	3	4	5	3	4	5
TextTilling	0.27 ± 0.24	0.23 ± 0.25	0.18 ± 0.23	0.34 ± 0.18	0.21 ± 0.11	0.18 ± 0.14	0.27 ± 0.22	0.18 ± 0.14	0.14 ± 0.15	0.33 ± 0.10	0.18 ± 0.09	0.19 ± 0.13	0.33 ± 0.23	0.18 ± 0.13	0.15 ± 0.17	0.33 ± 0.11	0.21 ± 0.14	0.23 ± 0.12	0.36 ± 0.19	0.15 ± 0.13	0.19 ± 0.13	0.32 ± 0.12	0.19 ± 0.08	0.19 ± 0.12
Extended paragraph	0.17 ± 0.17	0.12 ± 0.18	0.14 ± 0.19	0.25 ± 0.15	0.25 ± 0.12	0.22 ± 0.10	0.21 ± 0.14	0.12 ± 0.15	0.21 ± 0.22	0.21 ± 0.12	0.16 ± 0.11	0.16 ± 0.10	0.24 ± 0.16	0.21 ± 0.16	0.16 ± 0.09	0.21 ± 0.11	0.16 ± 0.11	0.20 ± 0.10	0.18 ± 0.14	0.15 ± 0.16	0.19 ± 0.15	0.22 ± 0.14	0.18 ± 0.06	0.21 ± 0.06
Extended paragraph coherence	0.19 ± 0.13	0.28 ± 0.22	0.16 ± 0.18	0.24 ± 0.12	0.24 ± 0.11	0.19 ± 0.11	0.19 ± 0.21	0.11 ± 0.11	0.11 ± 0.12	0.23 ± 0.11	0.16 ± 0.09	0.21 ± 0.11	0.15 ± 0.15	0.16 ± 0.11	0.17 ± 0.17	0.24 ± 0.06	0.11 ± 0.08	0.16 ± 0.08	0.16 ± 0.11	0.16 ± 0.13	0.15 ± 0.12	0.26 ± 0.10	0.15 ± 0.07	0.17 ± 0.07
Extended paragraph w2v	0.30 ± 0.22	0.25 ± 0.22	0.13 ± 0.22	0.33 ± 0.18	0.22 ± 0.13	0.17 ± 0.12	0.16 ± 0.18	0.15 ± 0.13	0.13 ± 0.09	0.35 ± 0.23	0.18 ± 0.11	0.15 ± 0.06	0.18 ± 0.14	0.19 ± 0.13	0.15 ± 0.09	0.25 ± 0.15	0.22 ± 0.12	0.21 ± 0.11	0.14 ± 0.14	0.14 ± 0.12	0.16 ± 0.11	0.19 ± 0.11	0.17 ± 0.10	0.19 ± 0.07
Extended paragraph coherence + w2v	0.11 ± 0.14	0.08 ± 0.13	0.05 ± 0.10	0.19 ± 0.12	0.13 ± 0.13	0.17 ± 0.11	0.17 ± 0.14	0.05 ± 0.09	0.08 ± 0.10	0.22 ± 0.09	0.14 ± 0.08	0.14 ± 0.09	0.19 ± 0.12	0.10 ± 0.12	0.08 ± 0.08	0.23 ± 0.08	0.16 ± 0.09	0.16 ± 0.08	0.13 ± 0.11	0.09 ± 0.08	0.08 ± 0.09	0.27 ± 0.09	0.13 ± 0.07	0.20 ± 0.08
Sentence Based Coherence	0.10 ± 0.15	0.07 ± 0.08	0.04 ± 0.07	0.18 ± 0.16	0.12 ± 0.07	0.19 ± 0.12	0.13 ± 0.10	0.05 ± 0.08	0.07 ± 0.09	0.19 ± 0.12	0.12 ± 0.10	0.11 ± 0.08	0.08 ± 0.11	0.09 ± 0.10	0.07 ± 0.07	0.14 ± 0.09	0.13 ± 0.07	0.14 ± 0.09	0.11 ± 0.07	0.08 ± 0.07	0.07 ± 0.08	0.17 ± 0.07	0.15 ± 0.06	0.15 ± 0.09
Sentence Based Coherence - limit 3	0.07 ± 0.11	0.01 ± 0.04	0.06 ± 0.11	0.18 ± 0.12	0.13 ± 0.10	0.17 ± 0.16	0.05 ± 0.07	0.08 ± 0.12	0.09 ± 0.07	0.18 ± 0.10	0.12 ± 0.07	0.17 ± 0.08	0.13 ± 0.08	0.06 ± 0.06	0.08 ± 0.11	0.14 ± 0.06	0.13 ± 0.06	0.18 ± 0.07	0.08 ± 0.09	0.09 ± 0.08	0.12 ± 0.08	0.17 ± 0.06	0.16 ± 0.08	0.18 ± 0.06

Fonte: Elaborada pelo autor.

Tabela 4 – Tabela com todos os resultados obtidos removendo *stopwords*

Tópicos	3						4						5						6					
Sub-Tópicos	1			2			1			2			1			2			1			2		
Parágrafos	3	4	5	3	4	5	3	4	5	3	4	5	3	4	5	3	4	5	3	4	5	3	4	5
TextTilling	0.27 ± 0.24	0.23 ± 0.25	0.18 ± 0.23	0.34 ± 0.18	0.21 ± 0.11	0.18 ± 0.14	0.27 ± 0.22	0.18 ± 0.14	0.14 ± 0.15	0.33 ± 0.10	0.18 ± 0.09	0.19 ± 0.13	0.33 ± 0.23	0.18 ± 0.13	0.15 ± 0.17	0.33 ± 0.11	0.21 ± 0.14	0.23 ± 0.12	0.36 ± 0.19	0.15 ± 0.13	0.19 ± 0.13	0.32 ± 0.12	0.19 ± 0.08	0.19 ± 0.12
Paragraph extended coherence	0.17 ± 0.17	0.12 ± 0.18	0.14 ± 0.19	0.25 ± 0.15	0.25 ± 0.12	0.22 ± 0.10	0.21 ± 0.14	0.12 ± 0.15	0.21 ± 0.22	0.21 ± 0.12	0.16 ± 0.11	0.16 ± 0.10	0.24 ± 0.16	0.21 ± 0.16	0.16 ± 0.09	0.21 ± 0.11	0.16 ± 0.11	0.20 ± 0.10	0.18 ± 0.14	0.15 ± 0.16	0.19 ± 0.15	0.22 ± 0.14	0.18 ± 0.06	0.21 ± 0.06
Extended paragraph coherence	0.19 ± 0.13	0.28 ± 0.22	0.16 ± 0.18	0.24 ± 0.12	0.24 ± 0.11	0.19 ± 0.11	0.19 ± 0.21	0.11 ± 0.11	0.11 ± 0.12	0.23 ± 0.11	0.16 ± 0.09	0.21 ± 0.11	0.15 ± 0.15	0.16 ± 0.11	0.17 ± 0.17	0.24 ± 0.06	0.11 ± 0.08	0.16 ± 0.08	0.16 ± 0.11	0.16 ± 0.13	0.15 ± 0.12	0.26 ± 0.10	0.15 ± 0.07	0.17 ± 0.07
Paragraph extended coherence + w2v	0.30 ± 0.22	0.25 ± 0.22	0.13 ± 0.22	0.33 ± 0.18	0.22 ± 0.13	0.17 ± 0.12	0.16 ± 0.18	0.15 ± 0.13	0.13 ± 0.09	0.35 ± 0.23	0.18 ± 0.11	0.15 ± 0.06	0.18 ± 0.14	0.19 ± 0.13	0.15 ± 0.09	0.25 ± 0.15	0.22 ± 0.12	0.21 ± 0.11	0.14 ± 0.14	0.14 ± 0.12	0.16 ± 0.11	0.19 ± 0.11	0.17 ± 0.10	0.19 ± 0.07
Extended paragraph coherence + w2v	0.11 ± 0.14	0.08 ± 0.13	0.05 ± 0.10	0.19 ± 0.12	0.13 ± 0.13	0.17 ± 0.11	0.17 ± 0.14	0.05 ± 0.09	0.08 ± 0.10	0.22 ± 0.09	0.14 ± 0.08	0.14 ± 0.09	0.19 ± 0.12	0.10 ± 0.12	0.08 ± 0.08	0.23 ± 0.08	0.16 ± 0.09	0.16 ± 0.08	0.13 ± 0.11	0.09 ± 0.08	0.08 ± 0.09	0.27 ± 0.09	0.13 ± 0.07	0.20 ± 0.08
Sentence Based Coherence	0.10 ± 0.15	0.07 ± 0.08	0.04 ± 0.07	0.18 ± 0.16	0.12 ± 0.07	0.19 ± 0.12	0.13 ± 0.10	0.05 ± 0.08	0.07 ± 0.09	0.19 ± 0.12	0.12 ± 0.10	0.11 ± 0.08	0.08 ± 0.11	0.09 ± 0.10	0.07 ± 0.07	0.14 ± 0.09	0.13 ± 0.07	0.14 ± 0.09	0.11 ± 0.07	0.08 ± 0.07	0.07 ± 0.08	0.17 ± 0.07	0.15 ± 0.06	0.15 ± 0.09
Sentence Based Coherence - limit 3	0.07 ± 0.11	0.01 ± 0.04	0.06 ± 0.11	0.18 ± 0.12	0.13 ± 0.10	0.17 ± 0.16	0.05 ± 0.07	0.08 ± 0.12	0.09 ± 0.07	0.18 ± 0.10	0.12 ± 0.07	0.17 ± 0.08	0.13 ± 0.08	0.06 ± 0.06	0.08 ± 0.11	0.14 ± 0.06	0.13 ± 0.06	0.18 ± 0.07	0.08 ± 0.09	0.09 ± 0.08	0.12 ± 0.08	0.17 ± 0.06	0.16 ± 0.08	0.18 ± 0.06

Fonte: Elaborada pelo autor.

Tabela 5 – Tabela com todos os resultados utilizando o método descrito de comunidades sobrepostas

Tópicos	3						4						5						6					
Sub-Tópicos	1			2			1			2			1			2			1			2		
Parágrafos	3	4	5	3	4	5	3	4	5	3	4	5	3	4	5	3	4	5	3	4	5	3	4	5
TextTilling	0.27 ± 0.24	0.23 ± 0.25	0.18 ± 0.23	0.34 ± 0.18	0.21 ± 0.11	0.18 ± 0.14	0.27 ± 0.22	0.18 ± 0.14	0.14 ± 0.15	0.33 ± 0.10	0.18 ± 0.09	0.19 ± 0.13	0.33 ± 0.23	0.18 ± 0.13	0.15 ± 0.17	0.33 ± 0.11	0.21 ± 0.14	0.23 ± 0.12	0.36 ± 0.19	0.15 ± 0.13	0.19 ± 0.13	0.32 ± 0.12	0.19 ± 0.08	0.19 ± 0.12
Paragraph extended coherence	0.17 ± 0.17	0.12 ± 0.18	0.14 ± 0.19	0.25 ± 0.15	0.25 ± 0.12	0.22 ± 0.10	0.21 ± 0.14	0.12 ± 0.15	0.21 ± 0.22	0.21 ± 0.12	0.16 ± 0.11	0.16 ± 0.10	0.24 ± 0.16	0.21 ± 0.16	0.16 ± 0.09	0.21 ± 0.11	0.16 ± 0.11	0.20 ± 0.10	0.18 ± 0.14	0.15 ± 0.16	0.19 ± 0.15	0.22 ± 0.14	0.18 ± 0.06	0.21 ± 0.06
Extended paragraph coherence	0.19 ± 0.13	0.28 ± 0.22	0.16 ± 0.18	0.24 ± 0.12	0.24 ± 0.11	0.19 ± 0.11	0.19 ± 0.21	0.11 ± 0.11	0.11 ± 0.12	0.23 ± 0.11	0.16 ± 0.09	0.21 ± 0.11	0.15 ± 0.15	0.16 ± 0.11	0.17 ± 0.17	0.24 ± 0.06	0.11 ± 0.08	0.16 ± 0.08	0.16 ± 0.11	0.16 ± 0.13	0.15 ± 0.12	0.26 ± 0.10	0.15 ± 0.07	0.17 ± 0.07
Paragraph extended coherence + w2v	0.30 ± 0.22	0.25 ± 0.22	0.13 ± 0.22	0.33 ± 0.18	0.22 ± 0.13	0.17 ± 0.12	0.16 ± 0.18	0.15 ± 0.13	0.13 ± 0.09	0.35 ± 0.23	0.18 ± 0.11	0.15 ± 0.06	0.18 ± 0.14	0.19 ± 0.13	0.15 ± 0.09	0.25 ± 0.15	0.22 ± 0.12	0.21 ± 0.11	0.14 ± 0.14	0.14 ± 0.12	0.16 ± 0.11	0.19 ± 0.11	0.17 ± 0.10	0.19 ± 0.07
Extended paragraph coherence + w2v	0.11 ± 0.14	0.08 ± 0.13	0.05 ± 0.10	0.19 ± 0.12	0.13 ± 0.13	0.17 ± 0.11	0.17 ± 0.14	0.05 ± 0.09	0.08 ± 0.10	0.22 ± 0.09	0.14 ± 0.08	0.14 ± 0.09	0.19 ± 0.12	0.10 ± 0.12	0.08 ± 0.08	0.23 ± 0.08	0.16 ± 0.09	0.16 ± 0.08	0.13 ± 0.11	0.09 ± 0.08	0.08 ± 0.09	0.27 ± 0.09	0.13 ± 0.07	0.20 ± 0.08
Sentence Based Coherence	0.10 ± 0.15	0.07 ± 0.08	0.04 ± 0.07	0.18 ± 0.16	0.12 ± 0.07	0.19 ± 0.12	0.13 ± 0.10	0.05 ± 0.08	0.07 ± 0.09	0.19 ± 0.12	0.12 ± 0.10	0.11 ± 0.08	0.08 ± 0.11	0.09 ± 0.10	0.07 ± 0.07	0.14 ± 0.09	0.13 ± 0.07	0.14 ± 0.09	0.11 ± 0.07	0.08 ± 0.07	0.07 ± 0.08	0.17 ± 0.07	0.15 ± 0.06	0.15 ± 0.09
Sentence Based Coherence - limit 3	0.07 ± 0.11	0.01 ± 0.04	0.06 ± 0.11	0.18 ± 0.12	0.13 ± 0.10	0.17 ± 0.16	0.05 ± 0.07	0.08 ± 0.12	0.09 ± 0.07	0.18 ± 0.10	0.12 ± 0.07	0.17 ± 0.08	0.13 ± 0.08	0.06 ± 0.06	0.08 ± 0.11	0.14 ± 0.06	0.13 ± 0.06	0.18 ± 0.07	0.08 ± 0.09	0.09 ± 0.08	0.12 ± 0.08	0.17 ± 0.06	0.16 ± 0.08	0.18 ± 0.06

Fonte: Elaborada pelo autor.

CAPÍTULO

6

CONCLUSÃO

Sobre os resultados apresentados no Capítulo 5 podemos tirar algumas conclusões interessantes sobre os métodos propostos e como eles se comportam em diferentes configurações de texto.

Primeiro vale ressaltar que nenhum método consegue lidar bem com textos mais longos, em especial aqueles com mais do que um sub-tópico por tópico escolhido. Pode-se no entanto argumentar que há uma melhora sobre o *baseline* em especial quando comparado com os métodos baseados em sentenças. Porém, dada a alta variância dos resultados não é possível extrair nenhuma conclusão concreta sobre o comportamento de cada método especificamente.

Focando na análise dos textos com somente um sub-tópico podemos ver alguns padrões interessantes.

6.1 Composição de métodos

Nota-se primeiramente que a composição de métodos foi um sucesso. Os resultados onde o método estendido de parágrafo é enriquecido tanto com o método de (RIEDL; BIEMANN, 2012) quanto com os *embeddings* do *word2vec* teve os melhores resultados em quase todas as configurações de texto, superando consideravelmente os resultados do método original e do método original em conjunto com cada modificação separadamente

6.2 Redes baseadas em sentença

Dos resultados apresentados nessa dissertação, aqueles obtidos pelos métodos baseados em sentenças são de fato os destaques, tendo os melhores resultados em quase todas as configurações. Esse fato é um pouco surpreendente pois esperava-se que parte importnate da

informação semântica da frase se perdesse pois utilizamos o método simples para a criação dos *embeddings* de cada sentença.

Sobre a limitação de alcance de ligações podemos dizer que embora ela melhore consideravelmente os resultados em alguns casos, ela gera uma variância muito grande a medida que mudamos a quantidade de parágrafos de cada tópico, fazendo dela uma metodologia pouco robusta quando não se sabe qual é o tamanho médio de cada tópico.

6.3 Variações dos métodos

Infelizmente não é possível afirmar que houve nenhuma mudança significativa quando se varia tanto a presença ou não de *stopwords* dentro dos textos, quanto o pertencimento à diversas comunidades de cada nó.

Podemos no entanto postular que palavras muito comuns ou ambíguas não sejam os principais problemas desses métodos, já que os resultados ficaram dentro do esperado dado a variância observada nos resultados. Seria interessante no entanto realizar um estudo mais cuidadoso sobre onde há uma sobreposição maior de palavras entre tópicos diferentes. Não foi analisado nessa dissertação, por exemplo, se esse é um fator que levou à perda de performance quando há mais de um sub-tópico para cada tópico.

6.4 Trabalhos futuros

Com esses resultados e alguns desenvolvimentos interessantes na área de NLP, conseguimos ver algumas claras linhas de pesquisa que podem trazer melhorias futuras.

Dado que os melhores resultados obtidos nesse trabalho foram utilizando uma representação de *embedding* que atualmente é até considerada como rudimentar, uma clara extensão dessa dissertação é o estudo de como os métodos propostos aqui podem ser melhorados com as técnicas mais recentes de representação semântica de sentenças. Duas dessa novas técnicas pode ser vistas em (PETERS *et al.*, 2018) e (DEVLIN *et al.*, 2018), que recentemente conseguiram melhorar o estado da arte em diversas tarefas complexas de NLP.

Nelas, dois dos principais problemas encontrados podem ter sido solucionados. Em ambos modelos consegue-se um *embedding* específico para cada palavra, dado o contexto, minimizando assim problemas de ambiguidade de palavra. Além disso é possivel utilizar o método proposto em (KIROS *et al.*, 2015), onde a rede neural proposta é treinada para, dada uma sentença, predizer a próxima e a anterior, parecido com o *word2vec*. Isso pode gerar então uma representação capaz de distinguir entre sentenças que pertencem ao mesmo tópico ou não.

REFERÊNCIAS

AMANCIO, D. R. Authorship recognition via fluctuation analysis of network topology and word intermittency. **Journal of Statistical Mechanics: Theory and Experiment**, IOP Publishing, v. 2015, n. 3, p. P03005, 2015. Citado na página 17.

____. A complex network approach to stylometry. **PloS one**, Public Library of Science, v. 10, n. 8, p. e0136076, 2015. Citado nas páginas 17 e 57.

AMANCIO, D. R.; ALTMANN, E. G.; RYBSKI, D.; JR, O. N. O.; COSTA, L. d. F. Probing the statistical properties of unknown texts: application to the voynich manuscript. **PLoS One**, Public Library of Science, v. 8, n. 7, p. e67310, 2013. Citado na página 18.

ANGHELUTA, R.; BUSSER, R. D.; MOENS, M.-F. The use of topic segmentation for automatic summarization. In: **Proceedings of the ACL-2002 Workshop on Automatic Summarization**. [S.l.: s.n.], 2002. p. 11–12. Citado nas páginas 18, 19, 41 e 53.

ARRUDA, H. F. de; COSTA, L. d. F.; AMANCIO, D. R. Classifying informative and imaginative prose using complex networks. **arXiv preprint arXiv:1507.07826**, 2015. Citado nas páginas 16, 41, 49 e 50.

____. Topic segmentation via community detection in complex networks. **arXiv preprint arXiv:1512.01384**, 2015. Citado nas páginas 12, 13, 15, 17, 18, 19, 41, 44, 45, 46, 50, 54, 55, 56, 59, 62, 63, 64, 65 e 66.

ARRUDA, H. F. de; SILVA, F. N.; MARINHO, V. Q.; AMANCIO, D. R.; COSTA, L. d. F. Mesoscopic representation of texts as complex networks. **arXiv preprint arXiv:1606.09636**, 2016. Citado nas páginas 16, 41, 50 e 51.

BARABÁSI, A.-L.; ALBERT, R. Emergence of scaling in random networks. **science**, American Association for the Advancement of Science, v. 286, n. 5439, p. 509–512, 1999. Citado na página 31.

BEEFERMAN, D.; BERGER, A.; LAFFERTY, J. Statistical models for text segmentation. **Machine learning**, Springer, v. 34, n. 1-3, p. 177–210, 1999. Citado nas páginas 54 e 60.

BIRD, S.; KLEIN, E.; LOPER, E. **Natural Language Processing with Python**. [S.l.]: O'Reilly Media, 2009. Citado nas páginas 44 e 61.

BLEI, D. M.; NG, A. Y.; JORDAN, M. I. Latent dirichlet allocation. **Journal of machine Learning research**, v. 3, n. Jan, p. 993–1022, 2003. Citado na página 43.

BLONDEL, V. D.; GUILLAUME, J.-L.; LAMBIOTTE, R.; LEFEBVRE, E. Fast unfolding of communities in large networks. **Journal of statistical mechanics: theory and experiment**, IOP Publishing, v. 2008, n. 10, p. P10008, 2008. Citado nas páginas 39, 40, 61 e 67.

CANCHO, R. F. i; SOLÉ, R. V.; KÖHLER, R. Patterns in syntactic dependency networks. **Physical Review E**, APS, v. 69, n. 5, p. 051915, 2004. Citado nas páginas 17 e 18.

CHOI, F. Y. Advances in domain independent linear text segmentation. In: ASSOCIATION FOR COMPUTATIONAL LINGUISTICS. **Proceedings of the 1st North American chapter of the Association for Computational Linguistics conference**. [S.l.], 2000. p. 26–33. Citado nas páginas 17, 19, 41 e 54.

COSTA, L. d. F.; RODRIGUES, F. A.; TRAVIESO, G.; BOAS, P. R. V. Characterization of complex networks: A survey of measurements. **Advances in Physics**, Taylor & Francis, v. 56, n. 1, p. 167–242, 2007. Citado nas páginas 17, 28, 29, 30 e 31.

DEVLIN, J.; CHANG, M.-W.; LEE, K.; TOUTANOVA, K. Bert: Pre-training of deep bidirectional transformers for language understanding. **arXiv preprint arXiv:1810.04805**, 2018. Citado na página 74.

DIAS, G.; ALVES, E.; LOPES, J. G. P. Topic segmentation algorithms for text summarization and passage retrieval: An exhaustive evaluation. In: **AAAI**. [S.l.: s.n.], 2007. v. 7, p. 1334–1340. Citado nas páginas 41 e 53.

DU, L.; BUNTINE, W. L.; JOHNSON, M. Topic segmentation with a structured topic model. In: **HLT-NAACL**. [S.l.: s.n.], 2013. p. 190–200. Citado nas páginas 18 e 41.

EISENSTEIN, J. Hierarchical text segmentation from multi-scale lexical cohesion. In: ASSOCIATION FOR COMPUTATIONAL LINGUISTICS. **Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics**. [S.l.], 2009. p. 353–361. Citado na página 41.

ERDOS, P.; RÉNYI, A. On the evolution of random graphs. **Publ. Math. Inst. Hung. Acad. Sci**, Citeseer, v. 5, n. 1, p. 17–60, 1960. Citado na página 28.

EULER, L. Solutio problematis ad geometriam situs pertinentis. **Eneström 53**, 1741. Citado na página 21.

FIEDLER, M. Algebraic connectivity of graphs. **Czechoslovak mathematical journal**, Institute of Mathematics, Academy of Sciences of the Czech Republic, v. 23, n. 2, p. 298–305, 1973. Citado na página 34.

FORTUNATO, S. Community detection in graphs. **Physics reports**, Elsevier, v. 486, n. 3, p. 75–174, 2010. Citado na página 56.

GALLEY, M.; MCKEOWN, K.; FOSLER-LUSSIER, E.; JING, H. Discourse segmentation of multi-party conversation. In: ASSOCIATION FOR COMPUTATIONAL LINGUISTICS. **Proceedings of the 41st Annual Meeting on Association for Computational Linguistics-Volume 1**. [S.l.], 2003. p. 562–569. Citado nas páginas 18 e 54.

HEARST, M. A. Texttiling: Segmenting text into multi-paragraph subtopic passages. **Computational linguistics**, MIT Press, v. 23, n. 1, p. 33–64, 1997. Citado nas páginas 15, 19, 41, 42, 44, 50 e 53.

KERNIGHAN, B. W.; LIN, S. An efficient heuristic procedure for partitioning graphs. **The Bell system technical journal**, Alcatel-Lucent, v. 49, n. 2, p. 291–307, 1970. Citado na página 33.

KIROS, R.; ZHU, Y.; SALAKHUTDINOV, R. R.; ZEMEL, R.; URTASUN, R.; TORRALBA, A.; FIDLER, S. Skip-thought vectors. In: **Advances in neural information processing systems**. [S.l.: s.n.], 2015. p. 3294–3302. Citado na página 74.

MARTINEZ-ROMO, J.; ARAUJO, L.; BORGE-HOLTHOEFER, J.; ARENAS, A.; CAPITÁN, J. A.; CUESTA, J. A. Disentangling categorical relationships through a graph of co-occurrences. **Physical Review E**, APS, v. 84, n. 4, p. 046108, 2011. Citado na página 17.

MIKOLOV, T.; SUTSKEVER, I.; CHEN, K.; CORRADO, G. S.; DEAN, J. Distributed representations of words and phrases and their compositionality. In: **Advances in neural information processing systems**. [S.l.: s.n.], 2013. p. 3111–3119. Citado nas páginas 15, 18, 47, 55, 56, 61 e 62.

MITCHELL, J.; LAPATA, M. Composition in distributional models of semantics. **Cognitive science**, Wiley Online Library, v. 34, n. 8, p. 1388–1429, 2010. Citado nas páginas 19, 47 e 56.

NEWMAN, M. **Networks: An Introduction**. [S.l.]: Cambridge Univ Press, 2010. Citado nas páginas 22 e 32.

NEWMAN, M. E. Spectral methods for community detection and graph partitioning. **Physical Review E**, APS, v. 88, n. 4, p. 042822, 2013. Citado nas páginas 36 e 67.

NEWMAN, M. E.; GIRVAN, M. Finding and evaluating community structure in networks. **Physical review E**, APS, v. 69, n. 2, p. 026113, 2004. Citado na página 32.

PETERS, M. E.; NEUMANN, M.; IYYER, M.; GARDNER, M.; CLARK, C.; LEE, K.; ZETTLEMOYER, L. Deep contextualized word representations. **arXiv preprint arXiv:1802.05365**, 2018. Citado na página 74.

PEVZNER, L.; HEARST, M. A. A critique and improvement of an evaluation metric for text segmentation. **Computational Linguistics**, MIT Press, v. 28, n. 1, p. 19–36, 2002. Citado na página 55.

RAGHAVAN, U. N.; ALBERT, R.; KUMARA, S. Near linear time algorithm to detect community structures in large-scale networks. **Physical review E**, APS, v. 76, n. 3, p. 036106, 2007. Citado na página 67.

ŘEHŮŘEK, R.; SOJKA, P. Software Framework for Topic Modelling with Large Corpora. In: **Proceedings of the LREC 2010 Workshop on New Challenges for NLP Frameworks**. Valletta, Malta: ELRA, 2010. p. 45–50. <http://is.muni.cz/publication/884893/en>. Citado na página 62.

RIEDL, M.; BIEMANN, C. Text segmentation with topic models. **Journal for Language Technology and Computational Linguistics**, v. 27, n. 1, p. 47–69, 2012. Citado nas páginas 15, 17, 41, 43, 44, 57, 62 e 73.

ROSVALL, M.; BERGSTROM, C. T. Maps of random walks on complex networks reveal community structure. **Proceedings of the National Academy of Sciences**, National Acad Sciences, v. 105, n. 4, p. 1118–1123, 2008. Citado na página 67.

SILVA, T. C.; AMANCIO, D. R. Word sense disambiguation via high order of learning in complex networks. **EPL (Europhysics Letters)**, IOP Publishing, v. 98, n. 5, p. 58001, 2012. Citado na página 18.

____. Discriminating word senses with tourist walks in complex networks. **arXiv preprint arXiv:1306.3920**, 2013. Citado na página 17.

TRAVERS, J.; MILGRAM, S. The small world problem. **Phychology Today**, JSTOR, v. 1, p. 61–67, 1967. Citado na página 29.

WATTS, D. J.; STROGATZ, S. H. Collective dynamics of 'small-world'networks. **nature**, Nature Publishing Group, v. 393, n. 6684, p. 440–442, 1998. Citado nas páginas 28, 29 e 30.