Ovídio José Francisco

Aplicação de técnicas de Recuperação de Informação para Organização e Extração de Históricos de Decisões de Documentos de Reuniões

Sorocaba, SP

18 de maio de 2018

Lista de símbolos

D	Conjunto de documentos de uma coleção.
n	Número total de documentos em uma coleção.
m	Número total de termos em uma coleção de documentos.
T	Conjunto de termos de uma coleção.
t_i	i-ésimo termo do vocabulário da coleção de documentos.
w_i	Peso i-ésimo termo.
$\vec{d}$	representação vetorial do documento d.
q	consulta do usuário.
c_i	i-ésima sentença da coleção de documentos.
B	Lista de segmentos.
N	Total de sentenças do documento.
Z	Conjunto de tópicos.
z_i	i-ésimo tópico do conjunto de tópicos.

Sumário

1 Introdução

A popularização dos computadores possibilitou o armazenamento cada vez maior de conteúdos digitais, sendo bastante comum, o formato textual como livros, documentos, e-mails, redes sociais e páginas web. A produção de textos gera informações em volumes crescentes que superaram a capacidade humana de análise manual. Além disso, dados textuais quase sempre apresentam-se em formato não estruturado, caracterizado pela ausência de uma organização pré-definida que facilite a busca em sistemas computacionais. Contudo, dados textuais possuem uma organização linguística intrínseca o que possibilita a pesquisa de ferramentas automáticas para manipulação e consulta a bases dados textuais não estruturadas (CAO, 2017; MANNING; RAGHAVAN; SCHüTZE, 2008).

Assim, os processos de extração automática de conhecimento em coleções textuais são essenciais, e ao mesmo tempo, constituem um desafio, devido às suas características como o formato não estruturado e trechos com diferentes níveis de importância, desde informações essenciais até textos pouco informativos ou mesmo irrelevantes.

Além dos tipos de informações mais comuns que são armazenados no formato textual, como e-mails, relatórios, artigos e postagens em redes sociais, têm-se também o armazenamento das atas de reuniões, as quais permitem às organizações a documentação oficial de reuniões em arquivos digitais, facilitando a confecção, compartilhamento e consulta às decisões tomadas. Reuniões são tarefas presentes em ambientes de gestão e organizações de um modo geral, onde discute-se problemas, soluções, propostas, alterações de projetos e frequentemente são tomadas decisões importantes onde a comunicação entre os membros da reunião é feita de forma majoritariamente verbal. Para que seu conteúdo possa ser registrado e externalizado, adota-se a prática de escrever seu conteúdo em atas (SCHWARTZ-ZIV; WEISBACH, 2013; LEE et al., 2011).

Por exemplo, nas reuniões do conselho de um programa de pós-graduação de uma universidade, são decididos, quais são os critérios para credenciamento e permanência de docentes no programa. Ao longo do tempo, esse tema pode ser discutido e mencionado diversas vezes, podendo os critérios inclusive passar por significativas alterações, devido a diversos fatores. O coordenador do programa pode desejar recuperar qual foi a decisão mais recente, para poder aplicar os critérios a um potencial novo membro do programa, ou os membros do conselho podem desejar rever o histórico de tudo o que já foi discutido/decidido sobre o tema, para poder propor alterações nas regras, de forma mais adequada.

As atas de reunião possuem características particulares. Frequentemente apresentam um texto com poucas quebras de parágrafo e sem marcações de estrutura, como capítulos, seções ou quaisquer indicações sobre o tema do texto. Devido a fatores como a não

estruturação e volume dos textos, a localização de um assunto em uma coleção de atas é uma tarefa custosa, especialmente considerando o seu crescimento em uma instituição.

As organizações costumam manter seus documentos eletrônicos organizados em pastas e nomeá-los com informações básicas sobre a reunião a que se refere como a data e alguma referência cronológica, por exemplo *"37ª Reunião Ordinária do Conselho ..."*. Essa forma de organização facilita a localização dos arquivos com ferramentas que fazem buscas pelo nome dos arquivos e pastas, sem levar em conta o teor dos documentos. Outra forma comum é o uso de ferramentas que fazem buscas nos conteúdos dos documentos, buscando por ocorrências de palavras-chave nos textos. Essas ferramentas permitem buscas combinadas com operadores lógicos como *and*, *or* e *not* ou ainda suporte a expressões regulares. Esse recurso, conhecido como *grepping*[1], produz resultados satisfatórios em muitos casos. Por outro lado, traz algumas desvantagens como: 1) transfere certa complexidade da tarefa ao usuário 2) não há suporte a padrões mais flexíveis como a proximidade entre as palavras ou palavras que estejam na mesma sentença 3) informa apenas se um documento casa ou não com a consulta do usuário com base na presença ou ausência dos termos da consulta (AGGARWAL; ZHAI, 2012; MANNING; RAGHAVAN; SCHüTZE, 2008). Ainda nesse contexto, usa-se outras técnicas de Recuperação de Informação como o Modelo de Espaço Vetorial para ranquear documentos atribuindo pontuações para a similaridade de cada par documento/consulta. Com isso, é possível apresentar os documentos ordenados conforme a sua relevância com a consulta. Utiliza-se também dicionários de sinônimos (*thesaurus*) para expandir a consulta do usuário por meio de conhecimento externo adicionado ao sistema.

Contudo, essas técnicas baseiam-se na frequência de palavras, em que os documentos e consultas são vistos como conjuntos de termos sem levar em conta relações entre termos que compartilham um mesmo tópico dentro do domínio. Por exemplo, as consultas "*alunos bolsa CAPES*" e "*suporte financeiro a pesquisa*" tratam de um mesmo assunto, nesse caso, a transferência de valores monetários como apoio a carreira acadêmica. Utilizando-se das técnicas até agora mencionadas, obteria-se resultados distintos, uma vez que as consultas não compartilham termos e não há relação direta ente eles. Como efeito, os resultados de cada consulta limitariam-se a documentos que compartilham termos com a consulta. Essas técnicas produzem resultados melhores a medida que o usuário fornece termos mais acertados na consulta, o que por vezes é dependente de certo conhecimento e familiaridade com o domínio no qual a coleção de documentos está inserida. Além disso, o retorno ao usuário é uma lista documentos integrais, o que pode exigir uma segunda busca dentro de um documento para encontrar o trecho desejado.

Uma vez que a ata registra a sucessão de assuntos discutidos na reunião, um sistema de recuperação de informação idealmente deve retornar ao usuário apenas os trechos que

[1] O nome *grepping* é uma referência ao comando `grep` do Unix

tratem do assunto pesquisado ao invés de documentos inteiros. Assim, cada trecho com um assunto predominate pode ser considerado um subdocumento. Portanto, em primeiro lugar, há a necessidade de descobrir onde há mudanças de assunto no texto.

Técnicas de segmentação automática de textos (segmentação textual) podem ser aplicadas com esse propósito. A tarefa de segmentação automática de textos, ou segmentação textual consiste em dividir um texto em partes que contenham um significado relativamente independente. Em outras palavras, é identificar as posições nas quais há uma mudança significativa de assunto. É útil em aplicações que trabalham com textos sem indicações de quebras de assunto, ou seja, não apresentam seções ou capítulos, como transcrições automáticas de áudio, vídeos e grandes documentos que contêm vários assuntos como atas de reunião e notícias (BOKAEI; SAMETI; LIU, 2015; SAKAHARA; OKADA; NITTA, 2014; MISRA et al., 2009; EISENSTEIN; BARZILAY, 2008; CHOI, 2000). Pode ser usada para melhorar o acesso a informação solicitada por meio de uma consulta, onde é possível oferecer porções menores de texto mais relevantes ao invés de exibir um documento grande que pode conter informações menos pertinentes. Além disso, encontrar pontos onde o texto muda de assunto, pode ser útil como etapa de pré-processamento em aplicações voltadas ao entendimento do texto, principalmente em documentos longos (CHOI, 2000).

Contudo, a segmentação textual apenas indica as transições de assuntos ao longo do texto, sem indicações sobre o teor dos segmentos. O assunto de cada trecho pode ser estimado por meio de técnicas de extração de tópicos. Os modelos de extração de tópicos fornecem uma estratégia que visa encontrar nas relações entre documentos, padrões latentes que sejam significativos para o entendimento dessas relações. Tais modelos podem ranquear um conjunto de termos importantes para um ou mais assuntos, bem como ranquear documentos por sua relevância para determinado tema (FALEIROS, 2016). Atualmente, destacam-se os modelos probabilísticos de extração de tópicos como LDA (BLEI; NG; JORDAN, 2003) e PLSA (HOFMANN, 1999). São abordagens amplamente utilizadas (ZHU et al., 2012) e frequentemente referenciadas em trabalhos que buscam extrair conhecimento e organizar bases textuais (STEYVERS; GRIFFITHS, 2007; O'CALLAGHAN et al., 2015). Nesse trabalho, a expressão *tópico* é usada para designar um assunto considerando que o mesmo foi extraído por meio de técnicas automáticas, ficando a expressão *assunto* utilizada como seu teor popular.

O processo de extração de tópicos atribui um peso a cada documento-tópico e uma relação termo-tópico que pode representar a probabilidade de ocorrência de um termo em um documento dado que o tópico está presente. A partir dessas representações, é possível agrupar documentos que compartilham o mesmo tópico bem como os termos que melhor descrevem o tópico. Com isso, obtém-se uma organização da coleção de documentos que favorece técnicas para navegação e consulta à coleção de documentos.

Devido as características das atas como ausência de meta-informação e a multipli-

cidade de assuntos, as técnicas de segmentação podem ser empregadas em conjunto com a extração de tópicos para aprimorar a recuperação de informação em coleções de atas de reunião.

Nesse trabalho, a segmentação textual é empregada na identificação dos assuntos tratados em cada ata. Os segmentos de atas gerados, aqui considerados como sub-documentos, são entregues a um extrator de tópicos. A tarefa do extrator é agrupar os segmentos com um mesmo tópico e eleger palavras da coleção que melhor descrevem cada grupo. Os descritores gerados pelo extrator de tópicos bem com os grupos são aproveitados nas técnicas de recuperação de informação a fim de ranquear os resultados e auxiliar o usuário a encontrar segmentos semelhantes por meio da navegação pelos grupos. Essa abordagem visa, em primeiro lugar, identificar os assuntos tratados em cada ata e gerar uma coleção de subdocumentos derivados da coleção de atas originais e, a partir disso, encontrar relações latentes entre os subdocumentos e termos da coleção. Com isso, tem-se uma estrutura que permite às técnicas de recuperação de informação fazer buscas à coleção de atas valendo-se da estrutura criada a partir das técnicas de segmentação textual e extração de tópicos.

A segmentação das atas permite ao sistema criar uma base de dados interna mais simples, em que os assuntos estão isolados em segmentos, formando assim, uma base de dados mais adequada à métodos de aprendizado de máquina e recuperação de informação. Além disso, permite que ao sistema final exibir apenas os trechos onde o assunto pesquisado está presente ao invés de entregar documentos integrais. A busca por palavras-chave em descritores transfere processamento da varredura dos documentos para o processo de extração de tópicos. Esse processo evita processamento e lentidão no momento da pesquisa. Essa estratégia é semelhante a criação de índices para aumentar a eficiência em sistemas de recuperação de informação. Além disso, encontra relações entre termos e documentos sem necessidade de conhecimento externo sobre o domínio. Ao agrupar os segmentos por tópicos, tem-se uma organização dos documentos que permite a visualização de segmentos semelhantes permitindo a navegação e exploração aos grupos, além das consultas por palavras-chave.

1.1 Objetivos

A Busca por informação em dados textuais implica basicamente em consultar uma base de dados não estruturada. O que se objetiva nesse trabalho de mestrado é utilizar a estrutura latente de documentos segmentados em conjunto com técnicas tradicionais de recuperação de informação, tendo como proposta o desenvolvimento de uma ferramenta para identificar, organizar e apresentar assuntos registrados em atas de reunião. O projeto se divide em dois objetivos específicos: 1) Avaliar os métodos de segmentação topical, extração de tópicos e recuperação de informação no contexto de atas de reunião. 2) Analisar

a qualidade dos subdocumentos apresentados quanto ao agrupamento, e relevância das informações contidas. Dessa forma, busca-se ajudar a suprir a necessidade de ferramentas para para esse cenário e contribuir com uma metodologia de extração de informação em documentos com assuntos que pode ser útil em trabalhos relacionados a outros domínios como documentos jurídicos, notícias e transcrições de conversas.

1.2 Organização da Dissertação

2 Conceituação Teórica

A popularidade dos computares permite a criação e compartilhamento de textos. Com isso, a quantidade de informação disponível facilmente extrapola a capacidade de humana de leitura e análise das coleções de documentos, estejam elas disponíveis na Internet ou em computadores pessoais. A necessidade de simplificar e organizar grandes coleções de documentos criou uma demanda por técnicas que permitam ao usuário acessar informações de seu interesse. Para esse fim, foram desenvolvidas técnicas para descobrir, extrair e agrupar textos de grandes coleções, entre essas, a modelagem de tópicos (HOFMANN, 1999; DEERWESTER et al., 1990; LEE; SEUNG, 1999; BLEI, 2012) e técnicas para identificar e recuperar informações com base em buscas de usuários.

2.1 Conceitos Básicos

2.2 Representação de Textos

Uma das formas mais comuns para que a grande maioria dos algoritmos de aprendizado de máquina possa extrair padrões das coleções de textos é a representação no formato matricial conhecido como Modelo Espaço Vetorial (*Vectorial Space Model* - VSM) (REZENDE, 2003), onde os documentos são representados como vetores em um espaço Euclidiano m-dimensional em que cada termo extraído da coleção é representado por uma dimensão. Assim, cada componente de um vetor expressa a relação entre os documentos e as palavras. Essa estrutura é conhecida como *document-term matrix* ou matriz documento-termo. Uma das formas mais populares para representação de textos é conhecida como *Bag Of Words* a qual é detalhada a seguir.

2.2.1 *Bag Of Words*

Nessa representação, cada termo é transformado em um atributo (*feature*) (REZENDE, 2003), em que a_{ij} é o peso do termo j no documento d_i e indica a sua relevância dentro da base de documentos. As medidas mais tradicionais para o cálculo desses pesos são a binária, onde o termo recebe o valor 1 se ocorre em determinado documento ou 0 caso contrário; *document frequency*, que é o número de documentos no qual um termo ocorre; *term frequency - tf*, atribui-se ao peso a frequência do termo dentro de um determinado documento; *term frequency-inverse document frequency, tf-idf*, pondera a frequência do termo pelo inverso do número de documentos da coleção em que o termo ocorre. Essa representação é mostrada pela Tabela 1.

	t_1	t_2	t_j	...	t_n
d_1	a_{11}	a_{12}	a_{1j}	...	a_{1n}
d_2	a_{21}	a_{22}	a_{2j}	...	a_{2n}
d_i	a_{i1}	a_{i2}	a_{ij}	...	a_{in}
...	...	...	...	...	...
d_m	a_{m1}	a_{m2}	a_{mj}	...	a_{mn}

Tabela 1 – Coleção de documentos na representação *bag-of-words*

Essa forma de representação sintetiza a base de documentos em um contêiner de palavras, ignorando a ordem em que ocorrem, bem como pontuações e outros detalhes, preservando apenas o peso de determinada palavra nos documentos. É uma simplificação de toda diversidade de informações contidas na base de documentos sem o propósito de ser uma representação fiel do documento, mas oferecer a relação entre as palavras e os documentos a qual é suficiente para a maioria dos métodos de aprendizado de máquina (REZENDE, 2003).

2.3 Pré-processamento de Textos

2.3.1 Medidas de Proximidade

No modelo espaço vetorial, a similidade entre um documentos x e y é calculada pela correlação entre os vetores $\vec{x}$ e $\vec{y}$, a qual pode ser medida pelo cosseno do ângulo entre esses vetores. Dados dois documentos $x = (x_1, x_1, \ldots, x_t)$ e $y = (y_1, y_1, \ldots, y_t)$, calcula-se:

$$cosseno(x, y) = \frac{\vec{x} \bullet \vec{y}}{|\vec{x}| \times |\vec{y}|} = \frac{\sum_{i=1}^{t} x_i \cdot y_i}{\sqrt{\sum_{i=1}^{t} x_i^2} \times \sqrt{\sum_{i=1}^{t} y_i^2}} \tag{2.1}$$

Valores de cosseno próximos a 0 indicam um ângulo próximo a 90º entre $\vec{x}$ e $\vec{y}$, ou seja, o documento x compartilha poucos termos com a consulta y, enquanto valores próximos a 1 indicam um ângulo próximo a 0º, ou seja, x e y compartilham termos e são similares (TAN; STEINBACH; KUMAR, 2005; FELDMAN; SANGER, 2006).

2.4 Recuperação de Informação

Devido à popularização dos computadores e à grande disponibilidade de documentos em formato digital, em especial na Web, a área da Recuperação de Informação (RI) tem recebido atenção de pesquisadores nas últimas décadas. Recuperação de informação é área da computação que envolve a aplicação de métodos computacionais no tratamento e busca de informação em bases de dados não estruturados, usualmente grandes coleções de documentos textuais armazenados em dispositivos eletrônicos. De fato, não há

dados completamente não estruturados ao se considerar a estrutura linguística latente em documentos textuais. O termo 'não estruturado' se refere a dados que oferecem uma estrutura claramente estruturada para sistemas computadorizados, a exemplo de documentos textuais (MANNING; RAGHAVAN; SCHüTZE, 2008).

A tarefa central da recuperação de informação é encontrar informações de interesse dos usuários e exibí-las. Essa necessidade motiva o desenvolvimento de o desenvolvimento de sistemas de recuperação de informação (SRI). Nesses sistemas o usuário expressa sua necessidade por meio da formulação de uma consulta, usualmente composta por um conjunto de palavras-chave. Então, o sistema apresenta os resultados da busca, frequentemente documentos, em ordem de relevância com a consulta.

2.4.1 Modelos de Recuperação de Informação

Um modelo de recuperação de informação deve criar representações de documentos e consultas a fim de predizer a necessidade expressa nos termos da consulta. Com base na entrada do usuário esses modelos buscam por documentos similares aos termos da consulta. Segue abaixo a descrição dos três modelos clássicos para recuperação de informação.

2.4.1.1 Modelo Booleano

O modelo booleano ou modelo lógico foi um dos primeiros modelos aplicados a recuperação informação sendo utilizado a partir de 1960. Nesse modelo uma consulta é considerada uma sequencia de termos conectador por operadores lógicos como AND, OR e NOT. Como resultado, classifica cada documento como relevante ou não relevante à consulta, sem gradação de relevância. Esses operadores lógicos podem ser manipulados por usuários com algum conhecimento em álgebra booleana para aumentar a quantidade de resultados ou restringí-la.

Uma desvantagem desse modelo é que não é possível medir a relevância de um documento em relação a consultado do usuário, devia a essa limitação não há informação que permita a ordenação dos resultados, que é uma característica relevante para muitos SRI.

As vantagens desse modelo são a facilidade de implementação e a possibilidade de usuários experientes usarem os operadores lógicos como uma forma de controle sobre os resultados da busca. Por outro lado, para usuários inexperientes isso pode ser considerado uma desvantagem, uma vez que o uso de expressões lógicas não é intuitivo. Apesar dos problemas apresentados, visto sua simplicidade, esse modelo foi largamente utilizado em sistemas comerciais.

2.4.1.2 Modelo Espaço Vetorial

Uma das formas mais comuns para representação textual é conhecida como Modelo Espaço Vetorial (*Vectorial Space Model* - VSM) (REZENDE, 2003), onde os documentos e consultas são representados como vetores em um espaço Euclidiano t-dimensional em que cada termo extraído da coleção é representado por uma dimensão. Considera-se que um documento pode ser representado pelo seu conjunto de termos, onde cada termo k_i de um documento d_j associa-se um peso $w_{ij} \geq 0$ que indica a importância desse termo no documento. De forma similar, para uma consulta q, associa-se um peso $w_{i,q}$ a cada termo consulta. Assim o vetor associado ao documento d_j é dado por $\vec{d_j} = (w_{1,j}, w_{2,j}, ..., w_{t,j})$ e o vetor associado a consulta q é dado por $\vec{q} = (w_{1,q}, w_{2,q}, ..., w_{t,q})$.

No modelo vetorial, a similidade entre um documento d_j e uma consulta q é calculada pela correlação entre os vetores $\vec{d_j}$ e $\vec{q}$, a qual pode ser medida pelo cosseno (Equação 2.1) do ângulo entre esses vetores, conforme já mostrado na Seção 2.3.1.

Avaliar a relevância de um documento sob uma consulta é fundamental para os modelos de RI. Para isso pode-se utilizar medidas estatísticas simples como a frequência do termo, conhecida como TF (do inglês *Term Frequency*) e a frequência de documentos, conhecida como DF (do inglês *Document Frequency*). A frequência do termo indica o número de vezes que um termo ocorre na coleção de documentos. A frequência de documentos, indica o número de documentos que contém ao menos uma ocorrência de um determinado termo. Considera-se que os termos que ocorrem frequentemente em muitos documentos, em geral, não trazem informações úteis para discriminar a relevância dos documentos, então, a fim de diminuir o peso de termos altamente frequentes, usa-se o fator IDF (*Inverted Document Frequency*), que é o inverso da número de documentos que contem um termo. O IDF é a medida de informação que um termo fornece com base em quão raro ou comum esse termo é para a coleção. Seja N o número de documentos de uma coleção e n_i o número de documentos onde o termo k_i ocorre, o cálculo de IDF é dado por:

$$IDF(k_i) = log\frac{N}{n_i} \tag{2.2}$$

Entre as medidas mais populares para ranqueamento de buscas está a TF-IDF (*Term Frequency-Inverted Document Frequency*) que pondera a frequência de um termo em um documento com sua frequência na coleção total de documentos. Assim, a relevância de um termo para um documento é dada por:

$$w_{i,j} = freq_{i,j} \cdot IDF(k_i) \tag{2.3}$$

Onde $freq_{i,j}$ é a frequência do termo k_i no documento d_j. A medida TF-IDF atribui valores altos para termos que ocorrem frequentemente em um documentos, e

valores menores para termos que ocorrem poucas vezes em um documento ou em muitos documentos da coleção. A ideia da medida tf.idf e quantificar a importância de um termo em um documento com base em sua frequência no próprio documento e sua distribuição ao longo da coleção de documentos (CROFT; METZLER; STROHMAN, 2009; SALTON; BUCKLEY, 1988; SHAMSINEJADBABKI; SARAEE, 2012; SALTON; ALLAN, 1994).

Uma vez que o sistema calcula os graus de similaridade entre os documentos e a busca por meio da equação 2.1, é possível ranquear os resultados por ordem de relevância. Além disso, sua relativa simplicidade e flexibilidade, favorecem a aplicação desse modelo em sistemas de recuperação de informação (TAN; STEINBACH; KUMAR, 2005; CROFT; METZLER; STROHMAN, 2009; MANNING; RAGHAVAN; SCHüTZE, 2008).

2.4.1.3 Modelo Probabilístico

O modelo probabilístico é baseado no princípio da ordenação probabilística (*Probability Ranking Principle*) onde dada um consulta q e um documento d_j relevante a q, o modelo tenta estimar a probabilidade do usuário encontrar o documento d_j. O modelo assumente que para uma consulta q há um conjunto de documentos R_q que contém exatamente os documentos relevantes e nenhum outro, sendo este um conjunto resposta ideal que maximiza a probabilidade do usuário encontrar um documento d_j relevante a q.

Seja $\overline{R_q}$ o complemento de R de forma que $\overline{R_q}$ contém todos os documentos não relevantes à consulta q. Seja $P(R_q|d_j)$ a probabilidade do documento d_j ser relevante à consulta q e $P(\overline{R_q}|d_j)$ a probabilidade de d_j não ser relevante à q. A similaridade entre um documento d_j e uma consulta q é definida por:

$$sim(d_j, q) = \frac{P(R_q|dj)}{P(\overline{R_q}|dj)} \tag{2.4}$$

A fim de obter-se uma estimativa numérica das probabilidades, o modelo assume o documento como uma combinação de palavras e seus pesos aos quais atribui-se valores binários que indicam a presença ou ausência de um termo, isto é, $w_{ij} \in \{0, 1\}$ e $w_{iq} \in \{0, 1\}$. Seja $p_i = P(t_i|R_q)$ a probabilidade do termo t_i ocorrer em um documento relevante à consulta q, e $s_i = P(t_i|\overline{R_q})$ a probabilidade do termo t_i estar presente em um documento não relevante. Seja ainda $\prod_{i:d_i=1}$ o produto dos termos com valor 1. Então, pode-se calcular:

$$sim(d_j, q) = \prod_{i:d_i=1} \frac{p_i}{s_i} \cdot \prod_{i:d_i=0} \frac{1 - p_i}{1 - s_i} \tag{2.5}$$

onde $\prod_{i:d_i=1}$ significa o produto dos termos com valor 1.

O modelo também supõe que os termos ocorrem independentemente no documento, ou seja, a ocorrência de um termo não influencia a ocorrência de outro. Partindo dessas

suposições, a Equação 2.5 passa por transformações que incluem aplicação da regra de Bayes e simplificações matemáticas, e chega-se a Equação 2.6 conhecida como equação de Robertson-Spark Jones a qual é considerada a expressão clássica para ranqueamento no modelo probabilístico. Detalhes da dedução dessa equação podem ser encontrados em (CROFT; METZLER; STROHMAN, 2009; MANNING; RAGHAVAN; SCHüTZE, 2008; RIJSBERGEN, 1979).

$$sim(d_j, q) = \sum_{i=1}^{t} w_{i,j} \cdot w_{i,q} \cdot \sigma_{i/R} \tag{2.6}$$

onde t é o número total de termos da coleção e

$$\sigma_{i/R} = \log \frac{p_i}{1 - p_i} + \log \frac{1 - s_i}{s_i} \tag{2.7}$$

Esse modelo tem com principal desvantagem a necessidade de estimar a separação inicial entre R_q e $\overline{R_q}$, pois não se conhece inicialmente o conjunto dos documentos relevantes a uma consulta, o qual deve ser aprimorado por meio de interações com o usuário. Além disso, o modelo não leva em consideração a frequência dos termos na indexação do documento. O modelo apresenta como vantagem a característica de atribuir probabilidades as similaridades entre documentos e consultas, o que permite ranquear dos resultados por ordem de relevância.

2.5 Segmentação Textual

A tarefa de segmentação textual consiste em dividir um texto em partes ou segmentos que contenham um significado relativamente independente. Em outras palavras, é identificar as posições nas quais há uma mudança significativa de assuntos. As técnicas de segmentação textual consideram um texto como uma sequência linear de unidades de informação que podem ser, por exemplo, cada termo presente no texto, os parágrafos ou as sentenças. Cada unidade de informação é um elemento do texto que não será dividido no processo de segmentação e cada ponto entre duas unidades é considerado um candidato a limite entre segmentos. Nesse sentido, um segmento pode ser visto como uma sucessão de unidades de informação que compartilham o mesmo assunto.

Os primeiros trabalhos dessa área se apoiam na ideia de que a mudança de assunto em um texto é acompanhada de uma proporcional mudança de vocabulário. Essa ideia, chamada de coesão léxica, sugere que a distribuição das palavras é um forte indicador da estrutura do texto, e demonstrou-se que há uma estreita correlação entre quedas na coesão léxica em janelas de texto e a transição de assuntos (KOZIMA, 1993). Em seu trabalho, Kozima calculou a coesão léxica de uma janela de palavras usando *spreading activation* em uma rede semântica especialmente elaborada para o idioma Inglês. Contudo,

a implementação de um algoritmo para outros domínios depende da construção de uma rede adequada.

O conceito de coesão léxica permite a aplicação da técnica de janelas deslizantes para encontrar os segmentos de um texto, em que se verifica a frequência dos termos em um fragmento do documento. Inicialmente, estabelece-se a partir do início do texto, um intervalo de t termos, chamado janela que em seguida é deslocada em passos de k termos adiante até o final do texto. A cada passo, analisa-se os termos contidos na janela.

O conceito de coesão léxica motivou a elaboração dos primeiros algoritmos para segmentação textual, entre eles o *TextTiling*. O *TextTiling* baseia-se na ideia de que um segmento pode ser identificado pela análise dos termos que o compõe. Inicialmente, o *TextTiling* recebe uma lista de candidatos a limite entre segmentos, usualmente finais de parágrafo ou finais de sentença. Utilizando a técnica de janelas deslizantes, para cada posição candidata são construídos 2 blocos, um contendo as sentenças que a precedem e outro com as que a sucedem. O tamanho desses blocos é um parâmetro a ser fornecido ao algoritmo e determina o tamanho mínimo de um segmento. Esse processo é ilustrado na Figura 1.

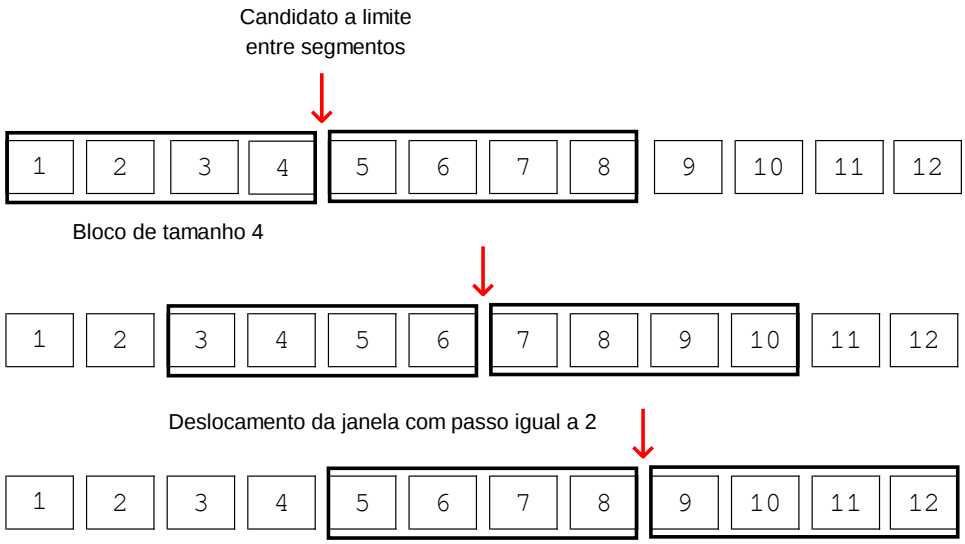

Figura 1 – Processo de deslocamento da janela deslizante. Os quadrados numerados representam as sentenças e os retângulos representam os blocos de texto a serem comparados. O deslocamento movimenta o candidato a limite e por consequência os blocos que o antecede e sucede.

Em seguida, os blocos de texto são representados por vetores que contém as frequências de suas palavras. Diferente da proposta de Kozima, o *TextTiling* utiliza cosseno (Equação 2.1) como medida para a similaridade entre os blocos adjacentes. Um limite ou transição entre segmentos é identificado sempre que a similaridade entre as unidades que antecedem e precedem o ponto candidato cai abaixo de um limiar, indicando uma diminuição da similaridade entre os blocos adjacentes. Ou seja, identifica-se uma transição

entre segmentos pelos vales na curva de dissimilaridades. Para cada final de sentença representada por c_i atribui-se uma profundidade dada por $(c_{i-1} - c_i) + (c_{i+1} - c_i)$ e será um limite entre segmentos caso a profundidade exceda $\overline{s} - \sigma$, onde $\overline{s}$ é a média da profundidade de todos os vales do documento e σ, o desvio padrão. Na Figura 2 é ilustrado a curva de dissimilaridade entre os blocos adjacentes.

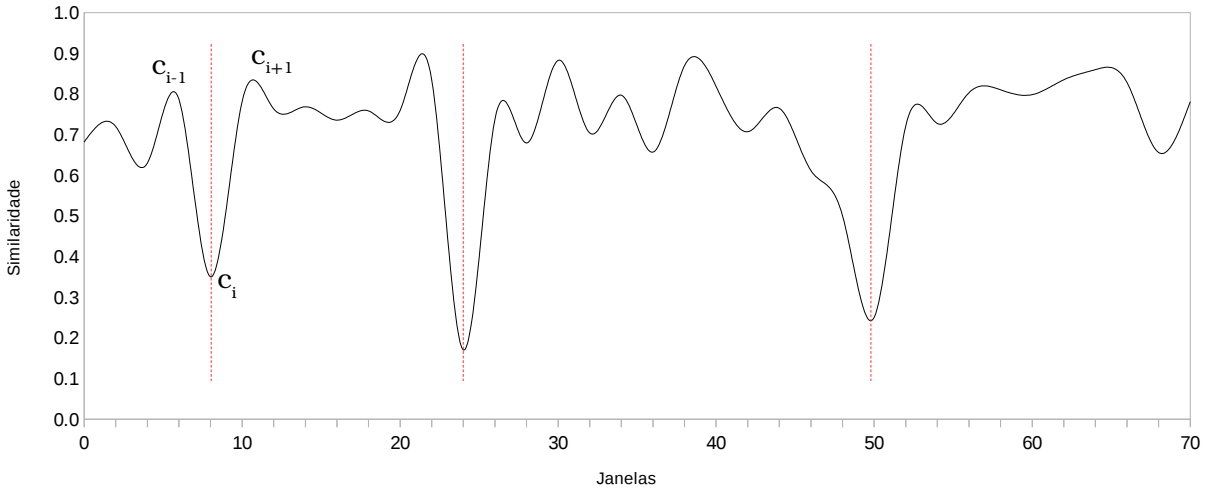

Figura 2 – Curva de dissimilaridades entre blocos de texto adjacentes. As linhas pontilhadas representam diminuições de similaridade que indicam limites entre segmentos.

O TextTiling apresenta como vantagens a facilidade de implementação e baixa complexidade computacional, favorecendo a implementação de trabalhos similares (NAILI; CHAIBI; GHEZALA, 2016; BOKAEI; SAMETI; LIU, 2015; CHAIBI; NAILI; SAMMOUD, 2014; KERN; GRANITZER, 2009; GALLEY et al., 2003), e sua utilização como base line em outros trabalhos (CARDOSO; PARDO; TABOADA, 2017; DIAS; ALVES; LOPES, 2007). Por outro lado, algoritmos mais complexos, como os baseados em matrizes de similaridade, apresentam acurácia relativamente superior como apresentado em (CHOI, 2000; KERN; GRANITZER, 2009; MISRA et al., 2009).

Outro algoritmo frequentemente referenciado na literatura é o C99 (CHOI, 2000) o qual é baseado em uma matriz de *ranking* das similaridades. A utilização de da coesão léxica pode não ser confiável para segmentos pequenos nessa abordagem, pois a ocorrência adicional de uma palavra pode causar certo impacto e alterar o cálculo da similaridade. Além disso, o estilo da escrita normalmente não é constante em todo o texto. Por exemplo, textos iniciais dedicados a introdução costumam apresentar menor coesão do que trechos dedicados a um tópico específico. Portanto, comparar a similaridade entre trechos de diferentes regiões não é apropriado. Devido a isso, as similaridades não podem ser comparadas em valores absolutos. Contorna-se esse problema fazendo uso de matrizes de similaridade para encontrar os segmentos de texto. Para isso, o *C99* constrói uma matriz que contém as similaridades de todas as unidades de informação (normalmente sentenças ou parágrafos).

Na Figura 3 é mostrado um exemplo de uma matriz de similaridade onde a intensidade do ponto(i, j) representa a similaridade entre as sentenças i e j. Observa-se que a matriz é simétrica, assim cada ponto na linha diagonal representa a similaridade quando $i = j$ (ou seja, com a mesma sentença) e revela quadrados com maior concentração de pontos ao longo da diagonal. A concentração de pontos ao longo da diagonal indica porções de texto com maior coesão léxica.

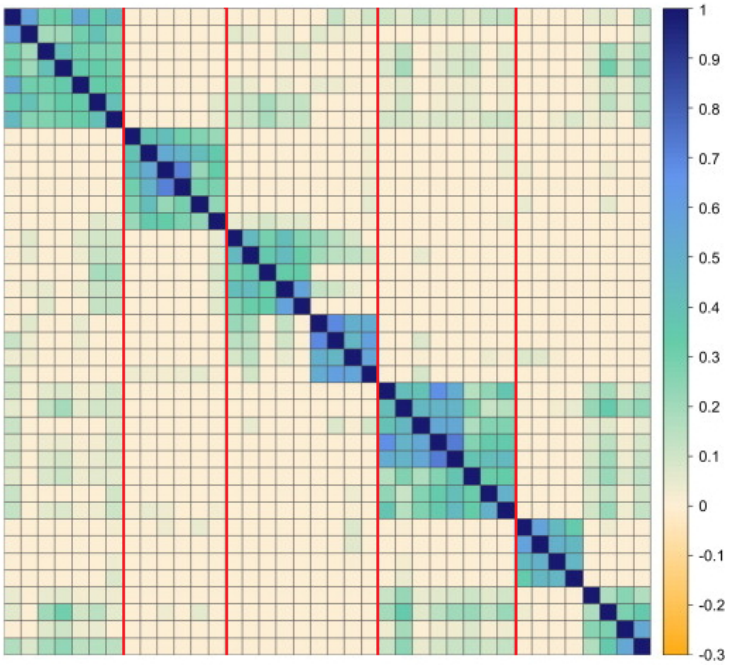

Figura 3 – *DotPlot* da similaridade entre sentenças onde as linha verticais representam segmentos reais (EISENSTEIN; BARZILAY, 2008).

Em seguida, cada valor na matriz de similaridade é substituído por seu *ranking* local. Para cada elemento da matriz, seu *ranking* será o número de elementos vizinhos com valor de similaridade menor que o seu. Assim, para cada elemento determina-se uma região quadrada de tamanho l em que o elemento em questão será comparado com $l \times l - 1$ elementos vizinhos. Na Figura 4.4 é destacado um quadro 3 x 3 de uma matriz. Tomando como exemplo o elemento com valor $0,5$, a mesma posição na matriz de *rankings* terá o valor 4, pois esse é o número de vizinhos com valores inferiores a $0,5$ dentro do quadro analisado na matriz de similaridades. Da mesma forma, na Figura 4.4 para o valor $0,2$ a matriz de *rankings* conterá o valor 1 na mesma posição. Após a construção da matriz de ranking obtêm-se um maior contraste entre os pontos, o que facilita a detecção de limites quando a queda de similaridade entre sentenças é mais sutil.

Finalmente, com base na matriz de *ranking*, o C99 utiliza um método de *clustering* baseado no algoritmo *DotPloting* (REYNAR, 1998) que usa regiões com maior densidade em uma matriz de similaridades para determinar como os segmentos estão distribuídos. Um segmento é definido por duas sentenças c_i e c_j (respectivamente a primeira e última

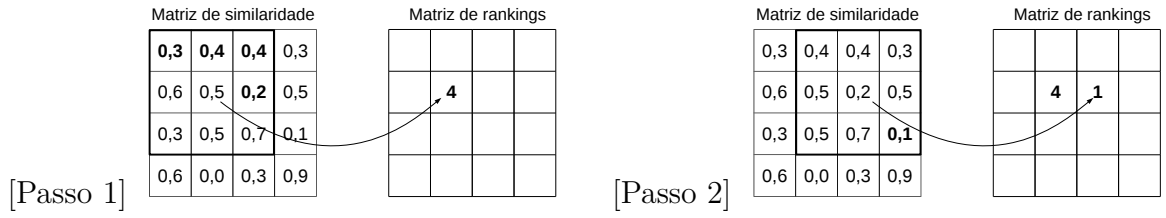

Figura 4 – Exemplo de construção de uma matriz de rankings.

sentença do segmento) que representam uma região quadrada ao longo da diagonal da matriz. Seja $B = \{b1, ..., b_m\}$ a lista de m segmentos, s_b a somatória dos valores dos *rankings* de um segmento $b \in B$ e a_b a sua área. Então, a densidade é computada por: Calcula-se a densidade dessa região como mostrado na Equação 2.8.

$$Den = \frac{\sum_{b=1}^{m} s_b}{\sum_{b=1}^{m} a_b} \tag{2.8}$$

O processo inicia com um único segmento formado por todas as sentenças do documento e o divide recursivamente em m segmentos. Cada passo divide B no ponto (i, j) que maximiza Den (Equação 2.8). O processo se repte até atingir o número de segmentos desejados ou um limiar de similaridade.

Desenvolveu-se também abordagens probabilísticas para segmentação textual, por exemplo, o método proposto por (UTIYAMA; ISAHARA, 2001) encontra a segmentação por meio de um modelo estatístico. Dado um texto representado por um conjunto de palavras $W = \{w_1, w_2, \ldots, w_n\}$ e um conjunto de segmentos $S = \{s_1, s_2, \ldots, s_m\}$ que segmenta W, a probabilidade da segmentação S é dada por:

$$P(S|W) = \frac{P(W|S)P(S)}{P(W)} \tag{2.9}$$

Com isso, é possível encontrar a sequência de segmentos mais provável $\hat{S} = argmax_S\ P(W|S)P(S)$. Nesse trabalho assume-se que os segmentos são estaticamente independentes entre si e as palavras nos segmentos são independentes dado o segmento que as contém. Essa simplificação permite decompor o termo $P(W|S)$ em um produtório de ocorrência de das palavras dado um segmento.

$$P(W|S) = \prod_{i=1}^{m} \prod_{j=1}^{n} P(w_j^i|S_i) \tag{2.10}$$

Onde $P(w_j^i|S_i)$ é a probabilidade da j-ésima palavra ocorrer no segmento S_i a qual é definida na Equação 2.11. Seja $f_i(w_j)$ a frequência da j-ésima palavra no i-ésimo segmento,

n_i é o número de palavras em S_i e k é o número de palavas diferentes em W. Calcula-se:

$$P(w_j^i|S_i) = \frac{f_i(w_j) + 1}{n_i + k} \tag{2.11}$$

A suposição de independência entre segmentos e as palavras neles contidas, são é verificada no mundo real. Para segmentos muito pequenos a estimativa das probabilidades das palavras pode ser afetada, além disso, o modelo não leva em conta a importância relativa das palavras (MALIOUTOV; BARZILAY, 2006).

Os métodos baseados em coesão léxica que utilizam métricas como cosseno quantificam a similaridade entre sentenças baseando-se apenas na frequência das palavras, Essa abordagem, ignora certas características do texto que podem dar pistas sobre a estrutura do texto. Por exemplo, frases como "Prosseguindo", "Dando continuidade", "Ao final da reunião" podem ajudar a detectar o inicio ou final de segmento. A fim de aproveitar esses indicadores, pode-se usar um framework bayesiano que permite incorporar fontes externas ao modelo. O método *BayesSeg* (EISENSTEIN; BARZILAY, 2008) aborda a coesão léxica em um contexto bayesiano onde as palavas de um segmento surgem de um modelo de linguagem multinomial o qual é associado a um assunto.

Essa abordagem é similar à métodos probabilísticos de extração de tópicos como o Latent Dirichlet Allocation (LDA) (BLEI; NG; JORDAN, 2003), com a diferença que ao invés de atribuir tópicos ocultos a cada palavra, esses são usados para segmentar o documento. Nesse sentido, detecta-se um limite entre sentenças quando a distribuição de tópicos entre elas for diferente. O *BayesSeg* baseia-se na ideia que alguns termos são usados em tópicos específicos enquanto outros são neutros em relação aos tópicos do documento e são usados para expressar uma estrutura do documento, ou seja, as frases-pista vem de um único modelo generativo. A fim de refletir essa ideia, o modelo é adaptado para influenciar a probabilidade da sentença de ser uma final ou início de segmento conforme a presença de frases pista.

O *MinCutSeg* (MALIOUTOV; BARZILAY, 2006) aborda a segmentação textual como um problema de particionamento de grafo, em que cada nó representa um sentença e os pesos das arestas representam a similaridade entre duas sentenças (Figura 5). Nessa abordagem, a segmentação textual corresponde ao particionamento do grafo que representa o texto.

Essa abordagem é inspirada no trabalho de (SHI; MALIK, 2000) que propõe um critério para particionamento de grafos chamado *normalized-cut criterion* inicialmente desenvolvido para segmentação de imagens estáticas a qual foi aproveita a restrição de linearidade dos textos para segmentação textual.

Seja $G = V, E$ um grafo ponderado, unidimensional em que V é o conjunto de vértices que correspondem às sentenças e E é o conjunto de arestas que correspondem

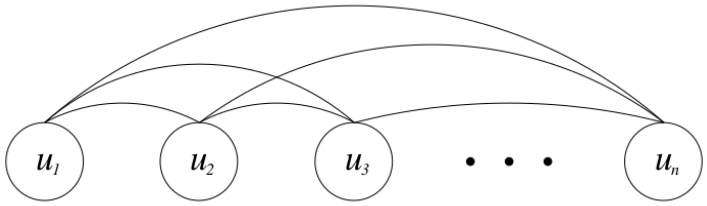

Figura 5 – Representação de texto baseada em grafo (MALIOUTOV; BARZILAY, 2006)

as similaridades entre as sentenças. Seja $w(u, v)$ o valor de similaridade entre o par de vértices u e v. O *MinCutSeg* visa particionar G em dois grafos disjuntos A e B de modo a minimizar o corte definido pela somatória das arestas que ligam u à v(Equação 2.12):

$$corte(A, B) = \sum_{u \in A, v \in B} w(u, v) \tag{2.12}$$

Além de maximizar a diferença entre as partições A e B, é necessário que essas seja homogenias em relação a similaridade de suas sentenças, conforme requerimento definido por (SHI; MALIK, 2000) em que o valor do corte deve ser normalizado pelo volume das partições dado por:

$$vol(A) = \sum_{u \in A, v \in V} w(u, v) \tag{2.13}$$

Em seguida, define-se o critério de corte normalizado (NCorte) como o resultado da normalização do corte pelo volume, conforme mostrado na Equação 2.14.

$$NCorte(A, B) = \frac{corte(A, B)}{vol(A} + \frac{corte(A, B)}{vol(B)} \tag{2.14}$$

Uma vez que um texto normalmente é dividido em mais que dois segmentos, é necessário extender o modelo para atender a essa necessidade. Seja A_{1k} uma partição e $V - A_k$ a diferença entre o grafo V e a partição k. O critério para múltiplos cortes normalizados é então estendido para:

$$NCorte_k(V) = \frac{corte(A_1, V - A_1)}{vol(A_1)} + \cdots + \frac{corte(A_k, V - A_k)}{vol(A_k)} \tag{2.15}$$

A decomposição do modelo em uma somatória de termos individuais permite empregar técnicas de programação dinâmica para o problema de cortes multidirecionais em grafos. Mais detalhes da formulação dessa solução estão disponíveis em (MALIOUTOV; BARZILAY, 2006).

Embora o problema minimizar cortes normalizados em grafos seja um problema do tipo NP-Completo[1], no contexto de segmentação textual esse problema é restrito a manter a linearidade dos vertices. A segmentação linear em um grafo implica que todos os vértices entre as extremidades esquerda e direitas de uma partição pertencem à essa partição, consequentemente o espaço de soluções possíveis é reduzido o que permite a execução do algoritmo em tempo polinomial.

2.5.1 Medidas de Avaliação em Segmentação Textual

As medidas de avaliação tradicionais como precisão e revocação permitem medir o desempenho de modelos de Recuperação de Informação e Aprendizado de Máquina por meio da comparação dos valores produzidos pelo modelo com os valores observados em uma referência. Usa-se uma tabela, chamada matriz de confusão, para visualizar o desempenho de um algoritmo. Na Tabela 2 é apresentada uma matriz de confusão para duas classes (Positivo e Negativo).

	Predição Positiva	Predição Negativa
Positivo real	VP (Verdadeiro Positivo)	FN (Falso Negativo)
Negativo real	FP (Falso Positivo)	VN (Verdadeiro Negativo)

Tabela 2 – Matriz de confusão.

No contexto de segmentação textual, um falso positivo é um limite identificado pelo algoritmo que não corresponde a nenhum limite na segmentação de referência, ou seja, o algoritmo indicou que em determinado ponto há uma quebra de segmento, mas na segmentação de referência, não há quebra no mesmo ponto. De maneira semelhante, um falso negativo é quando o algoritmo não identifica um limite existente na segmentação de referência, ou seja, em determinado ponto há, na segmentação de referência, um limite entre segmentos, contudo, o algoritmo não o identificou. Um verdadeiro positivo é um ponto no texto indicado pelo algoritmo e pela segmentação de referência como uma quebra de segmentos, ou seja, o algoritmo e a referência concordam que em determinado ponto há uma transição de assunto. Na avaliação de segmentadores, não há o conceito de verdadeiro negativo. Este seria um ponto no texto indicado pelo algoritmo e pela segmentação de referência onde não há uma quebra de segmentos. Uma vez que os algoritmos apenas indicam onde há um limite, essa medida não é necessária.

Nesse sentido, a precisão indica a proporção de limites corretamente identificados pelo algoritmo, ou seja, correspondem a um limite real na segmentação de referência. Porém, não diz nada sobre quantos limites reais existem. É calculada dividindo-se o

[1] NP-Completo configura um tipo de problema para o qual não se conhece uma solução determinística que possa ser computada em tempo polinomial. Papadimitriou provou que o problema de corte mínimo em grafos está incluso nessa categoria.

número de limites identificados automaticamente pelo número de candidatos a limite (Equação 2.16).

$$Precisão_{seg} = \frac{VP}{VP + FP} \tag{2.16}$$

A revocação, é a proporção de limites verdadeiros que foram identificados pelo algoritmo. Porém não diz nada sobre quantos limites foram identificados incorretamente. É calculada dividindo-se o número de limites identificados automaticamente pelo número limites verdadeiros (Equação 2.17).

$$Revocação_{seg} = \frac{VP}{VP + FN} \tag{2.17}$$

Existe uma relação inversa entre precisão e revocação. Conforme o algoritmo aponta mais segmentos no texto, este tende a melhorar a revocação e ao mesmo tempo, reduzir a precisão. Esse problema de avaliação pode ser contornado utilizado a medida F^1 que é a média harmônica entre precisão e revocação onde ambas tem o mesmo peso (Equação 2.18).

$$F^1{}_{seg} = \frac{2 \times Precisão \times Revocação}{Precisão + Revocação} \tag{2.18}$$

As medidas de avaliação tradicionais, precisão e revocação, podem não ser confiáveis, por não considerarem a distância entre os limites, mas penalizam o algoritmo sempre que um limite que não coincide perfeitamente com a referência. Essas medidas podem ser mais adequadas quando necessita-se de segmentações com maior exatidão. Em outras palavras, computam apenas os erros do algoritmo quando se detecta falsos positivos ou falsos negativos, o que nesse contexto de segmentação textual pode não ser suficiente, dado a subjetividade da tarefa. Além dessas medidas, que consideram apenas se um segmento foi perfeitamente definido conforme uma referência, pode-se também considerar a distância entre o segmento extraído automaticamente e o segmento de referência (KERN; GRANITZER, 2009). Chama-se *near misses* o caso em que um limite identificado automaticamente não coincide exatamente com a referência, mas é necessário considerar a proximidade entre eles.

Na Figura 6 é apresentado um exemplo com duas segmentações extraídas automaticamente e uma referência. Em ambos os casos não há nenhum verdadeiro positivo, o que implica em zero para os valores de precisão, acurácia, e revocação, embora o resultado do algoritmo A possa ser considerado superior ao primeiro se levado em conta a proximidade dos limites.

Considerando o conceito de *near misses*, algumas medidas de avaliação foram propostas. Proposta por (BEEFERMAN; BERGER; LAFFERTY, 1999), P_k atribui valores parciais a *near misses*, ou seja, limites sempre receberão um peso proporcional à

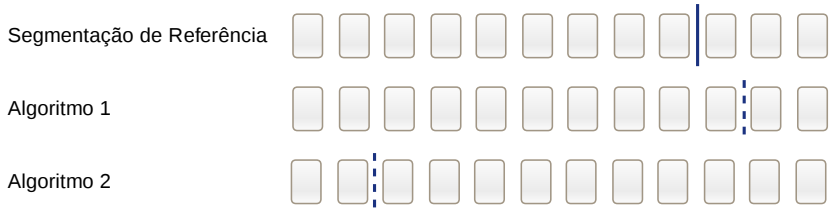

Figura 6 – Exemplos de *near missing* e falso positivo puro. Os blocos indicam uma unidade de informação e as linha verticais representam uma transição de assunto.

sua proximidade, desde que dentro de um janela de tamanho k. Para isso, esse método move uma janela de tamanho k ao longo do texto. A cada passo verifica, na referência e no algoritmo, se as extremidades (a primeira e última sentença) da janela estão ou não dentro do mesmo segmento, então, penaliza o algoritmo caso este não concorde com a referência. Ou seja, dado dois termos de distância k, P_k verifica se o algoritmo coloca os termos no mesmo segmento ou em segmentos distintos e o penaliza caso não concorde com a referência. Dadas uma segmentação de referência ref e uma segmentação automática hyp, ambas com N sentenças, P_k é computada como:

$$P_k(ref, hyp) = \frac{1}{N-k} \sum_{i=1}^{N-k} (\delta_{ref}(i, i+k) \bar{\oplus} \delta_{hyp}(i, i+k)) \tag{2.19}$$

Onde $\delta_S(i, j)$ é a função indicadora que retorna 1 se as sentenças c_i e c_j estão no mesmo segmento e 0 caso contrário, $\bar{\oplus}$ é o operador `XNOR` (ambos ou nenhum) que retorna 1 se ambos os argumentos forem diferentes. O valor de k é calculado como a metade da média dos comprimentos dos segmentos reais. Como resultado, é retornada a dissimilaridade entre a segmentação calculada pela contagem de discrepâncias divida pela quantidade de segmentos analisados. Essa medida pode ser interpretada como a probabilidade de duas sentenças extraídas aleatoriamente pertencerem ao mesmo segmento.

WindowDiff (PEVZNER; HEARST, 2002) é uma medida alternativa à P_k. De maneira semelhante, move uma janela pelo texto e penaliza o algoritmo sempre que o número de limites proposto pelo algoritmo não coincidir com o número de limites esperados para aquela janela. Ou seja, o algoritmo é penalizado quando não concordar com a segmentação de referência quanto ao número de segmentos na janela. Mais formalmente, para cada intervalo k, compara o número de segmentos obtidos pela referência r_i com o obtido pelo algoritmo a_i e penaliza o algoritmo se $r_i \neq a_i$. Na Equação 2.20 é mostrada a definição de WindowDiff onde $b(i, i+k)$ representa o número de limites entre as sentenças i e $i+k$ e N, o total de sentenças no texto.

$$WindowDiff(ref, hyp) = \frac{1}{N-k} \sum_{i=1}^{N-k} (|b(ref_i - ref_{i+k}) - b(hyp_i - hyp_{i+k})| > 0) \tag{2.20}$$

Assim, consegue manter a sensibilidade a *near misses* e além disso, considerar o tamanho das janelas. A fim de melhor equilibrar o peso dos falsos positivos em relação a *near misses*, dobra-se a penalidade para falsos positivos, evitando-se a supervalorização dessa medida.

As medidas *WindowDiff* e P_k, consideram a quantidade e proximidade entre os limites, sendo mais tolerantes a pequenas imprecisões. Essa é uma característica desejável, visto que as segmentações de referência possuem diferenças consideráveis. *WindowDiff* equilibra melhor os falsos positivos em relação a *near misses*, ao passo que P_k os penaliza com peso maior. Isso significa que segmentadores melhores avaliados em P_k ajudam a selecionar as configurações que erram menos ao separar trechos de texto com o mesmo assunto, enquanto *WindowDiff* é mais tolerante nesse aspecto. De maneira geral, observa-se melhores resultados de *WindowDiff* quando os algoritmos aproximam a quantidade de segmentos automáticos da quantidade de segmentos da referência. Por outro lado, P_k avalia melhor as configurações que retornam menos segmentos. Contudo, não é possível definir um valor adequado, uma vez que os segmentadores humanos frequentemente apontam segmentações diferentes.

2.6 Anotação de Subtópicos

A avaliação de segmentadores frequentemente requer uma segmentação de referência. Essa referência deve refletir um segmentação real sendo confiável para apoiar a avaliação da qualidade de técnicas de segmentação.

A construção de um corpus anotado demanda tempo e disponibilidade de anotadores humanos, o que a torna uma tarefa relativamente custosa. Assim, é necessário seguir procedimentos que assegurem que a tarefa seja concluída com o exito esperado que o resultado produzido seja válido, confiável e consistente para fins de pesquisas científicas. Para isso, (HOVY; LAVID, 2010) propuseram uma metodologia para anotação em corpus que pode ser resumida em sete passos: (1) Escolha do evento a ser anotado, (2) Seleção do corpus,(3) Selecionar e treinar os anotadores, (4) Especificar o processo de anotação, (5) Modelar uma interface para anotação, (6) Escolher e aplicar medidas de avaliação e (7) Disponibilizar e manter o produto. A seguir, serão descritos outros trabalhos relacionados a anotação de corpus e em seguida a os passos dessa metodologia.

Um dos primeiros trabalhos a produzir um corpus com anotações de segmentos foi (HEARST, 1997) no qual um corpus constituído por doze artigos de revistas foram anotados por sete técnicos pesquisadores. Cada artigo continha entre 1.800 e 2500 palavras. O autor considerou um limite entre segmentos real onde pelo menos três anotadores marcavam uma transição de tópico. No trabalho de (KAZANTSEVA; SZPAKOWICZ, 2012) utilizou-se um livro ficcional contendo vinte capítulos que foi segmentado por seis

alunos de graduação que além de marcar os pontos de transição entre segmentos, forneceram um descrição breve sobre cada segmento identificado.

Outros trabalhos abordaram corpus compostos pela transcrição de audios. Por exemplo, (PASSONNEAU; LITMAN, 1997) transcreveu vinte narrativas sobre um filme que foi segmentada e anotada por sete voluntários. Cada narrativa, continha cerca de 13.500 palavras. Os anotadores não receberam nenhum treinamento formal para a tarefa, mas apenas foram solicitados a usar suas noções de comunicabilidade para identificar as mudanças de tópicos. No trabalho de (GALLEY et al., 2003) investigou-se a transcrição de um conjunto de vinte e cinco reuniões obtidas do *ICSI Meeting corpus* (JANIN et al., 2003) em que pelo menos três anotadores analisaram os pontos onde ocorreu troca da pessoa que fala e apontaram como sendo ou não uma mudança de assunto.

Nesses trabalhos utilizou-se os anotadores como juízes para produzir uma referência em que decidiu-se sobre cada candidato a limite entre segmentos por meio da opinião da maioria. Além desses trabalhos, outros se valeram de segmentações produzidas artificialmente. Por exemplo, (CHOI, 2000) produziu um corpos formado por 700 documentos. As referências foram geradas pela concatenação de sentenças extraídas de documentos diferentes. De maneira semelhante, (CHAIBI; NAILI; SAMMOUD, 2014) utilizou a concatenação de artigos de noticias para produzir os documentos. Os autores consideram um limite real o ponto que divide dois artigos originais.

2.6.1 Metodologia para anotações em corpus

Os trabalhos citados anteriormente utilizaram procedimentos diferentes para produzir segmentações de referência para seus trabalhos. Como já citado, (HOVY; LAVID, 2010) propôs que o processo de anotação em corpus pode der sintetizado e dividido em sete passos.

2.6.1.1 Escolha do corpus

A criação de corpus raramente é restrita a um único propósito. O material original deve ser preferencialmente constituído de documentos disponíveis livremente à comunidade, a fim de facilitar a comparação, extensão e avaliação de trabalhos futuros. Devido a diversidade linguística de diferentes domínios e gêneros de textos, a escolha dos documentos de amostra deve procurar ser representativa ao domínio a ser abordado. O corpus é considerado representativo quando o assunto a abordado na amostra tem correspondência com a interpretação do público geral desse domínio.

2.6.1.2 Escolha da teoria a ser explicada

A anotação deve ajudar a explicar uma teoria, ou seja, fornecer informações úteis à sua compreensão. Essa teoria irá guiar a especificação do processo de anotação, quais informações deseja-se extrair e como interpretá-las. Quanto mais complexa for a teoria ser explicada, mais complexa será a tarefa de anotação bem como as instruções que os anotadores deverão seguir. Além disso, deve-se estabelecer de início o nível de detalhamento necessário. A complexidade da teoria e nível de detalhamento impactam na condução da anotação e da estabilidade da anotação.

2.6.1.3 Selecionar e treinar os anotadores

O treinamento e o nível de conhecimento dos anotadores ainda é uma questão em aberto. Alguns pesquisadores afirmam que estes devem ser especialistas no domínio do corpus. Outros afirmam que pessoas adequadamente treinadas podem produzir resultados satisfatórios. Considerando a necessidade de treinamento, tem-se a subjetividade das tarefas que dificulta a elaboração de instruções precisas. Tarefas que permitem a especificação de procedimentos que levam em conta a possibilidade de diferentes casos e variáveis, põem em dúvida a necessidade da criação de um corpus anotado. Por outro lado, a ausência de treinamento implica que as anotações terão como base o conhecimento prévio dos anotadores e sua preconcepção a cerca do domínio o que diminui o nível de concordância entre os anotadores e dificulta a replicação de outros trabalhos.

2.6.1.4 Especificar o procedimento de anotação

Alguns processos de anotações podem levar longos períodos, criando a necessidade de dividir a tarefa em fases. Nesses casos, frequentemente os anotadores fazem reuniões periódicas a fim de relatar eventuais problemas. Em caso de baixa concordância, pode-se abrir espaço para discussão de pontos com baixa concordância, a qual é chamada de fase de "reconciliação" que embora recomendada, em alguns casos pode ocasionar um enviesamento dos resultados. Outra estratégia para baixa concordância é que solicitar que os anotadores marquem o nível de certeza sobre as anotações.

2.6.1.5 Modelar uma interface para anotação

Um software com interface amigável, além de facilitar o trabalho, evita erros durante o processo. O ganho em tempo e a melhoria na qualidade dos resultados justifica a criação de uma interface. Exemplos softwares para anotação na área de Processamento de linguagem natural e Bioinformática podem ser encontrados em (GRUENSTEIN; NIEKRASZ; PURVER, 2007).

2.6.1.6 Escolher e aplicar medidas de avaliação

Quando observa-se baixa concordância entre os anotadores, entende-se que há uma falha no processo de anotação ou na teoria a ser explicada, o que implica que o dados produzidos não servem para a fins de pesquisa ou aplicações práticas. A medida dessa concordância deve determinar a confiabilidade dos resultados. A medida mais utilizada em Processamento de linguagem natural é o coeficiente *kappa* (CARLETTA, 1996) que retorna um valor no intervalo de 0 até 1, onde 1 significa uma concordância perfeita e 0 que não houve concordância. Seja, $P(A)$ a proporção de vezes que os anotadores concordam e $P(E)$ a proporção de concordância esperada ao acaso. O cálculo de *kappa* é dado por:

$$kappa = \frac{P(A) - P(E)}{1 - P(E)} \tag{2.21}$$

Essa medida, apresenta como limitação a entrada de apenas dois casos. Como alternativa, a medida conhecida como *Fleiss's k* (SHROUT; FLEISS, 1979) pode ser utilizada quando há mais que dois anotadores, porém restringe-se a anotações com apenas duas categorias. Na avaliação de segmentadores, as medidas P_k (Equação 2.19) e *WindowDiff* (Equação 2.20) podem ser utilizadas, uma vez que são medidas de similaridade, como visto em (KAZANTSEVA; SZPAKOWICZ, 2012; CARDOSO; PARDO; TABOADA, 2017).

2.6.1.7 Disponibilizar e manter o produto

Uma vez criado, o corpus anotado deve ser disponibilizado para uso em outros trabalhos. Recomenda-se fornecer o corpus original além dos resultados obtidos, observando-se desde o início e ao longo do tempo a propriedade e eventuais licenças sob o corpus original.

2.7 Modelos de Extração de Tópicos

Os modelos de extração de tópicos são abordagens não-supervisionadas que visam descobrir padrões latentes nas relações entre os documentos e seus termos. Baseiam-se na premissa de que um documento é produzido a partir de tópicos previamente definidos que determinam os termos a serem utilizados em um documento. Nesse contexto, um documento é uma mistura de tópicos onde cada termo presente no documento pode ser associado a um tópico. Um tópico por sua vez, é uma estrutura com valor semântico que é representada por um conjunto de termos e seus pesos que indicam o quão significante esses termos são para um assunto e pode ser útil para o entendimento do tema do tópico (STEYVERS; GRIFFITHS, 2007; BLEI, 2012).

Para descobrir esses tópicos, algumas técnicas foram propostas. Em termos de metodologia, a maioria dos trabalho enquadram-se em duas principais categorias, os

modelos não-probabilísticos e os modelos probabilísticos.

2.7.1 Modelos Não Probabilísticos

Os modelos não-probabilísticas baseiam-se em técnicas de fatoração de matrizes, onde a matrix documento-termo é projetada em um espaço com menor dimensionalidade chamado *Latent Semantic Space*. Seja $d \in D = \{d_1, \cdots, d_n\}$ o vetor que representa a coleção de documentos, $t \in T = \{t_1, \cdots, t_m\}$ seus termos distintos e $z \in Z = \{z_1, \cdots, z_k\}$ seus tópicos. Esses métodos aprendem decompondo a matriz documento-termo W, em duas matrizes Z e A, tal que a resultante de ZA seja uma aproximação da matriz W original. Mais formalmente tem-se:

$$Z \cdot A = \hat{W} \approx W \tag{2.22}$$

Sendo n o número de termos, m o número de documentos da coleção, k a quantidade de tópicos a serem extraídos, a matriz A corresponde a matriz documento-tópico e possui dimensão $k \times n$. Z corresponde a matriz termo-tópico e possui dimensão $m \times k$. Uma vez que $k \ll n, m$, então A e Z são menores que a matriz de entrada, o que resulta em uma versão comprimida da matriz original, pois $k \cdot n + m \cdot k \ll n \cdot m$. Ao final, obtém-se uma representação documento-tópico que atribui um peso para cada tópico em cada documento da coleção e uma representação termo-tópico que representa a probabilidade de ocorrência de um termo em um documento dado que o tópico está presente no documento.

Nesse sentido, o *Latente Semantic Indexing* (LSA) (DEERWESTER et al., 1990) usa a técnica chamada *Singular Value Decomposition* (SVD) para encontrar padrões no relacionamento entre assuntos e termos em uma coleção de texto não estruturada. Entretanto, esse método não fornece uma interpretação para elementos com valores negativos (DEERWESTER et al., 1990) (CHENG et al., 2013).

Outro modelo popular é o *Non-Negative Matrix Factorization* (NMF) (LEE; SEUNG, 1999). Diferente do LSA, no processo de fatoração apenas operações aditivas são permitidas, o que garante que as matrizes resultantes não possuem elementos negativos, permitindo uma interpretação mais intuitiva de seus valores. Além disso, o processo de fatoração proporciona a propriedade de *clustering*, ou seja, agrupar as colunas da matriz W, e dessa forma, oferece a característica interessante de agrupar os documentos da coleção.

2.7.2 Modelos Probabilísticos

Os modelos probabilísticos consideram os documentos como uma mistura de tópicos e um tópico como uma distribuição probabilística sobre os termos. O processo de elaboração do documento a partir desses tópicos é chamado de processo generativo ou modelo

generativo, o qual é desconhecido porém pode ser estimado com base nos termos presentes no documento, também chamados de variáveis observáveis. Assim, o processo de extração de tópicos consiste em estimar o modelo generativo que deu origem ao documento.

O PLSA (HOFMANN, 1999) foi um dos primeiros a estender o modelo LSA e formalizar a extração de tópicos probabilísticos. De maneira similar ao LSA, esse modelo decompõe uma matriz esparsa a fim de reduzir a dimensionalidade. O PLSA cria um modelo estatístico chamado *aspect model* que associa os tópicos às variáveis observáveis atribuindo probabilidades às ligações entre os tópicos e os documentos e entre as palavras e os tópicos. Assim, cada documento pode ser representado como a probabilidade de um tópico estar presente, $P(z|d)$. E a probabilidade de um termo ocorrer dado que um tópico esta presente, $P(t|z)$. Em comparação ao LSA, é considerado um método mais robusto por proporcionar uma interpretação probabilística. Por outro lado, esse modelo apresenta desvantagens como o número de parâmetros do modelo que cresce linearmente com o número de documentos da coleção, o que pode ocasionar *overfitting*.

A fim de contornar esses problemas, o LDA (BLEI; NG; JORDAN, 2003) estende o modelo PLSA incorporando um modelo generativo onde os cada tópico obedece à distribuição multivariada de *Dirichlet* o que o torna menos propenso ao *overfitting* e capaz de inferir tópicos a documentos ainda não observados. É referenciado na literatura como estado da arte sobre modelos probabilísticos de extração de tópicos e influencia uma grande quantidade de trabalhos, tornando-se base para novos modelos. No modelo LDA, o processo de geração de palavras se dá em duas etapas:

1. Atribui-se uma distribuição aleatória sobre os tópicos.
2. Para cada termo no documento:
    a) Atribui-se aleatoriamente a um tópico da distribuição obtida na etapa 1;
    b) Seleciona-se aleatoriamente uma palavra do tópico correspondente.

Assim cada documento é associado a múltiplos tópicos com proporções distintas (etapa 1). Cada palavra do documento é obtida de um tópico específico (etapa 2.b) que foi anteriormente obtido a partir da distribuição de tópicos do documento (etapa 2.a). Isso permite ao modelo LDA atribuir, para cada documento, múltiplos tópicos com proporções distintas (BLEI, 2012).

Os modelos de extração de tópicos foram inicialmente propostos para utilização em mineração texto onde são empregados na redução de dimensionalidade, extração de informações em textos, bem como na organização e recuperação de documentos, sendo utilizados para mensurar a relevância de um termo ou conjunto de termos para determinado assunto ou documento. Visto a popularidade nessas tarefas e flexibilidade dos modelos,

logo notou-se sua utilidade em outros tipos de dados com atributos discretos como imagens, grafos e genética.

2.7.3 Pós-processamento

2.8 Trabalhos Relacionados

A segment-based approach to clustering multi-topic documents (TAGARELLI; KARYPIS, 2013)

Multi-document Topic Segmentation (JEONG; TITOV, 2010)

Statistical Topic Models for Multi-label Document Classification (RUBIN et al., 2012)

O uso da Mineração de Textos para Extração e Organização não Supervisionada de Conhecimento (REZENDE; MARCACINI; MOURA, 2011)

Multi-topic Aspects in Clinical Text Classification (SASAKI; REA; ANANIADOU, 2007)

Multi-topic Multi-document Summarization (MASAO; KôITI, 2000)

3 Sistema de recuperação de informação em documentos multi-temáticos

O sistema proposto tem como objetivo recuperar informações em uma coleção de documentos em que cada documento contém assuntos diversos e relativamente independentes entre si. Esse sistema deve identificar os assuntos de cada documento e disponibilizá-los de forma que o usuário consiga consultar a coleção de documentos e obter todo o histórico de ocorrências de um determinado tema de forma que possa identificar onde esse tema foi mencionado, bem como se houve uma decisão relacionada ao tema.

A proposta original deste trabalho contempla funcionalidades de classificação para identificar automaticamente o tipo de ocorrência onde um assunto é mencionado, o qual pode ser classificado como uma decisão, informe ou menção. Contudo essas funcionalidades configuram trabalhos futuros para continuação do sistema como concebido inicialmente. Assim, esse trabalho de mestrado está focado na segmentação de atas de reunião, no agrupamento desses segmentos em tópicos e na recuperação de trechos de atas relacionados ao assunto da pesquisa.

Esse Capítulo está organizado da seguinte forma: primeiramente, na Seção 3.1 é apresentada uma visão geral do sistema proposto, seu funcionamento e como as técnicas de segmentação textual e extração de tópicos são empregadas para gerar uma base de dados que concentra as informações necessárias para identificar e agrupar os diversos assuntos distribuídos na coleção de documentos. Ainda nessa Seção, o são apresentadas as técnicas de recuperação de informação utilizadas para entregar os segmentos de acordo com a consulta do usuário bem como permitir a exploração de segmentos relacionados ao mesmo tema, os quais originalmente estão distribuídos na coleção de documentos.

Na Seção 3.2 é apresentada a aplicação do sistema proposto ao contexto das atas de reunião, ou seja, a base de dados original é uma coleção de atas de reunião. Uma vez que as atas configuram um corpus com documentos multi-temáticos, conforme a proposta desse trabalho, essa aplicação visa a validação do sistema e a análise da eficiência das técnicas empregadas em um corpus em conformidade com a proposta desse trabalho de mestrado. Será apresentada também a preparação das atas, a descrição dos algoritmos utilizados e suas configurações.

3.1 Sistema Proposto

Essa Seção apresenta as etapas de desenvolvimento do sistema de recuperação de atas proposto, bem como o seu funcionamento geral, desde a preparação dos documentos

até a entrega dos históricos de ocorrência ao usuário.

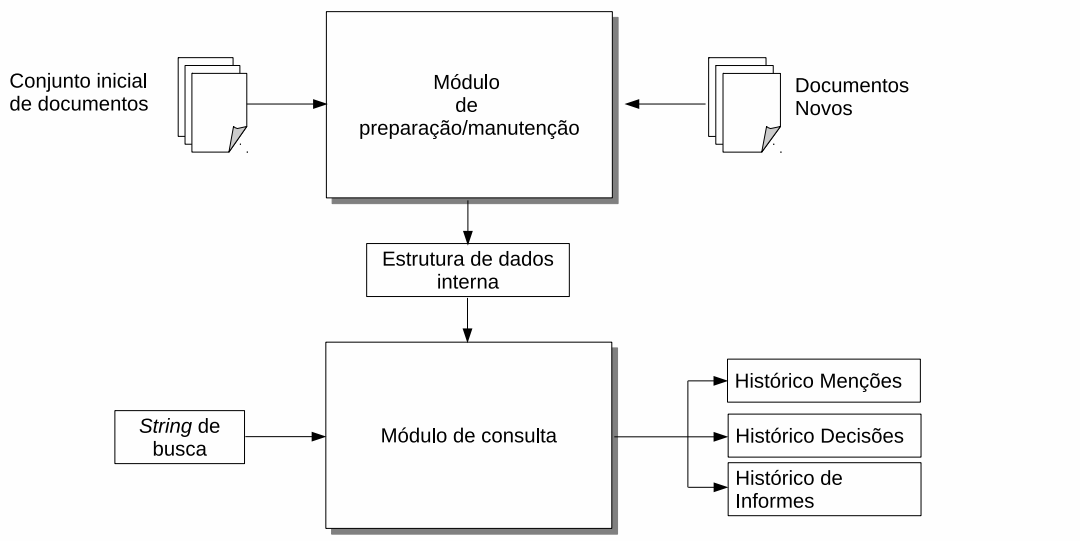

Figura 7 – Visão geral do sistema

A Figura 7 mostra a visão geral do sistema com suas principais entradas e saídas. Inicialmente o sistema recebe um conjunto inicial de documentos. A função de Módulo de preparação/manutenção é processar e manter esses textos para gerar uma base de dados interna que codifica os textos extraídos com seus respetivos tópicos. O Módulo de consulta recebe a consulta do usuário que expressa o assunto de interesse. Em seguida, os trechos de texto que fazem menção ao esse assunto são exibidos ao usuário.

3.1.1 Módulo de preparação e manutenção

O módulo de preparação e manutenção tem como função principal manter uma base de dados necessária para os processos de recuperação de informação. Inicialmente, esse módulo deve receber um conjunto inicial de documentos os quais devem ser divididos em segmentos de texto que contêm um assunto predominante, e agrupá-los em categorias por meio de técnicas de extração de tópicos. Além disso, considera-se o crescimento da bases de documentos, assim, o sistema deve receber novos documentos a medida que são gerados.

3.1.1.1 Preparação dos documentos

As atas são normalmente armazenadas em arquivos binários do tipo *pdf*, *doc*, *docx* ou *odt*. A partir desses arquivos, elas devem ser pré-processadas e estruturadas para que possam ser aplicados métodos de MI e RI. Para isso, o texto puro é extraído e passa por processos de transformação que incluem remoção de elementos considerados menos significativos e a identificação de sentenças. Esse processo é ilustrado na Figura 8 e descrito a seguir.

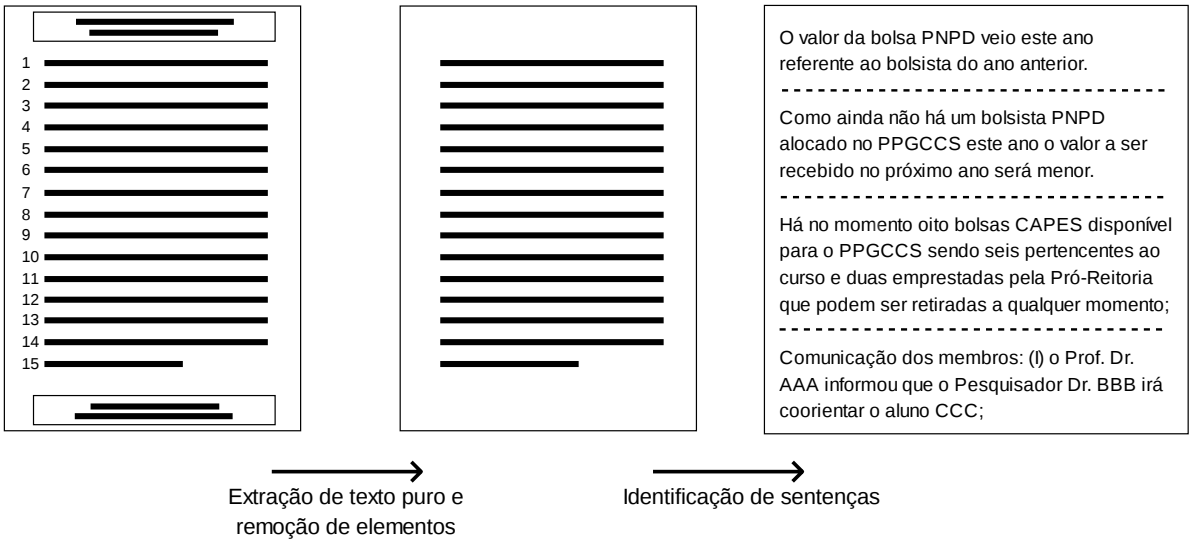

Figura 8 – Etapa de preparação de um documento que inclui da extração de texto puro, remoção de elementos menos significativos e a identificação de sentenças

As atas contém trechos que podem ser considerados pouco informativos e descartados durante o pré-processamento, como cabeçalhos e rodapés que se misturam aos tópicos tratados na reunião, podendo ser inseridos no meio de um tópico prejudicando tanto os algoritmos de MT e RI, quanto a leitura do texto pelo usuário. Um cabeçalho é a porção de texto que inicia cada página do documento e, de forma semelhante, um rodapé e a porção que as encerra. Detecta-se os cabeçalhos e os rodapés sempre que há uma repetição das primeiras e últimas palavras do documento.

Nesse trabalho considera-se as sentenças as menores unidades de informação a serem processadas pelos algoritmos de segmentação, por tanto, devem ser identificadas. Ao considerar intuitivamente que uma sentença seja uma sequência de palavras entre sinais de pontuação como “.”, “!” e “?”, alguns erros poderiam ocorrer quando esses tiverem outra função dentro do texto como em abreviações[1], endereços de internet e datas. Outro problema seriam frases curtas com poucas palavras e que não expressam um conceito completo, mas parte dele. Devido ao estilo de pontuação desses documentos, como encerrar sentenças usando um “;” e inserção de linhas extras, foram usadas as regras especiais para identificação de finais de sentença. No Algoritmo 1 é mostrado como cada *token* é identificado e como final de sentença.

[1] As abreviações são identificadas por meio de uma lista com 234 abreviações conhecidas.

Algoritmo 1: Identificação de finais de sentença.

Entrada : Texto
Saída : Texto com identificações de finais de sentença

para todo *token, marcá-lo como final de sentença se:*
  Terminar com um `!`
  Terminar com um `.` e não for uma abreviação
  Terminar em `.?;` e:
    For seguido de uma quebra de parágrafo ou tabulação
    O próximo *token* iniciar com `({["'`
    O próximo *token* iniciar com letra maiúscula
    O penúltimo caracter for `)}]"'`
fim

3.1.1.2 Pré-Processamento dos Documentos

Na etapa de pré-processamento, os documentos são pré-processados individualmente conforme são recebidos pelos algoritmos de segmentação extração de tópicos. Inicialmente, cada texto passou por um processo de transformação em que as letras foram convertidas em caixa baixa e eliminou-se sinais de pontuação, numerais e termos menores que três caracteres. Em seguida removeu-se os termos que não contribuem para a etapa de segmentação, as quais são chamadas de *stop words*, para identificá-las usou-se uma lista de 438 palavras. Em seguida, extraiu-se o radical de cada palavra por meio da técnica *stemming*.

Nesse trabalho, todos os termos restantes após a remoção de *stop words* e *stemming* foram mantidas independentemente de valores de *Document Frequency* devido a característica das atas de possuir múltiplos assuntos em um documento. Cada segmento pode ser considerado como um sub-documento quem contém um assunto relativamente independente.

3.1.1.3 Segmentação

Como já mencionado, uma ata registra a sucessão de assuntos discutidos em uma reunião, porém apresenta-se com poucas quebras de parágrafo e sem marcações de estrutura, como capítulos, seções ou quaisquer indicações sobre o assunto do texto. Portanto, faz-se necessário descobrir quando há uma mudança de assunto no texto da ata. Para essa tarefa, as técnicas de segmentação de texto recebem uma lista de sentenças, da qual considera cada ponto entre duas sentenças como candidato a limite, ou seja, um ponto onde há transição entre assuntos (BOKAEI; SAMETI; LIU, 2015; BOKAEI; SAMETI; LIU, 2016; MISRA et al., 2009; SAKAHARA; OKADA; NITTA, 2014).

As técnicas de segmentação abordadas na Subseção 2.5 dividem o texto de cada ata em segmentos que contêm um assunto relativamente independente. Esses segmentos serão processados por um extrator de tópicos que irá extrair descritores e agrupá-los por

tópicos.

3.1.1.4 Extração de Tópicos

Após a segmentação da coleção e identificação das transições de assuntos em cada documento da coleção, o sistema inicia a etapa de extração de padrões por meio das técnicas de extração de tópicos, ou seja, uma vez identificado onde há uma mudança de assunto, o próximo passo é obter conhecimento sobre os assuntos.

O resultado do processo de extração de tópicos é a representação dos documentos e seus tópicos em uma matriz documento-tópico que atribui um peso a cada tópico para cada documento e uma matriz termo-tópico que pode representar a probabilidade de ocorrência do termo quando um tópico ocorre em documento ou a frequência esperada desse termo. Nesse sistema, essas representações são utilizadas para melhorar as tarefas de recuperação de informação e agrupamento dos documentos. O agrupamento por tópicos e seus descritores são utilizados para ajudar o usuário a analisar e identificar os subdocumentos conforme sua consulta, bem como encontrar resultados similares.

3.1.1.5 Estrutura de Dados Interna

A estrutura de dados interna é o resultado dos processos de segmentação e extração de tópicos. Constituída por: (1) documentos originais para referência, (2) segmentos de texto contendo um assunto relativamente independente, (3) matrizes documento-tópico e termo-tópico. Na Figura 9 é apresentado a visão geral da estrutura de dados interna.

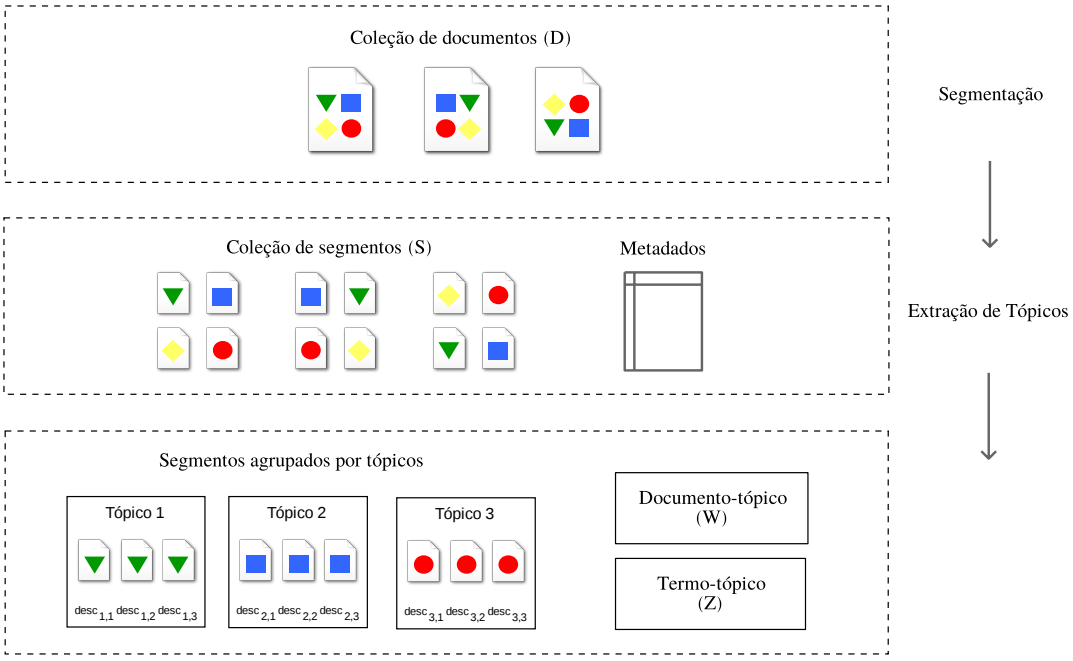

Figura 9 – Visão geral da estrutura de dados interna e seu processo de geração.

Na Figura 9 observa-se os arquivos da coleção de documentos (D) que contêm assuntos diversos (representados pelos círculos, quadrados e triângulos) fincando disponíveis para visualização e fonte original das informações. Os arquivos da originais (D) são mantidos para reverência aos textos integrais fincando disponíveis para visualização e fonte original das informações.

A partir do texto extraído dos documentos originais é gerado o conjunto de segmentos de texto (S) que contém os textos em formato legível a serem exibidos ao usuário pelo módulo de consulta. Para se conheça o arquivo que deu origem a cada segmento, o sistema gera metadados que mantém a ligação entre os documentos e seus segmentos. Esses segmentos de texto são armazenadas em arquivos de texto plano.

Os segmentos são tratados como sub-documentos pelo sistema sendo a partir deles que o extrator de tópicos constrói as matrizes documento-tópico e termo-tópico. Uma vez construídas, essas matrizes são armazenadas em arquivos e usadas para agrupar os segmentos com o mesmo tópico bem como encontrar os melhores termos da coleção para descrever cada tópico.

Deve-se ressaltar que o sistema recebe um conjunto inicial de documentos a partir do qual a estrutura de dados interna é gerada. A medida que novos documentos são gerados e adicionados ao sistema, serão processados pelo módulo de preparação e manutenção e incorporados a estrutura de dados interna bem como as matrizes documento-tópico e termo-tópicos serão atualizadas pelo extrator de tópicos.

3.1.2 Módulo de Consulta

Uma vez que a estrutura de dados interna contem os assuntos abordados na coleção de documentos, o tipo de ocorrência para cada assunto e o trecho onde se encontram, é responsabilidade do módulo de consulta receber a *string* de consulta do usuário, resgatar os dados desejados e apresentá-los em ordem cronológica, dando condições para o usuário acessar os segmentos encontrados bem como os documentos originais.

3.1.2.1 Ranqueamento

3.1.2.2 Visualização

O usuário final precisa de uma interface adequada para visualizar os resultados retornados pela busca considerando-se a relevância dos segmentos selecionados e a navegação pelos tópicos. Uma boa apresentação deve permitir ao usuário identificar a os resultados e ser relativante suficiente para compreensão do conteúdo, evitando a leitura do texto completo. Ou seja, o texto de cada segmento apresentado deve ser suficiente para compreensão do assunto mencionado, sem necessidade de visualizar o documento original.

Na Figura 10 é apresenta a tela principal do sistema. Observa-se a esquerda os segmentos agrupados por tópicos, os quais são representados por um componente para visualização em árvore onde as pastas, identificadas pelos descritores, representam um tópico. Uma pasta quando expandida revela os segmentos agrupados no tópico que representa. À direita, como resultado da consulta, são apresentados os segmentos selecionados pela relevância com a *string* de busca do usuário.

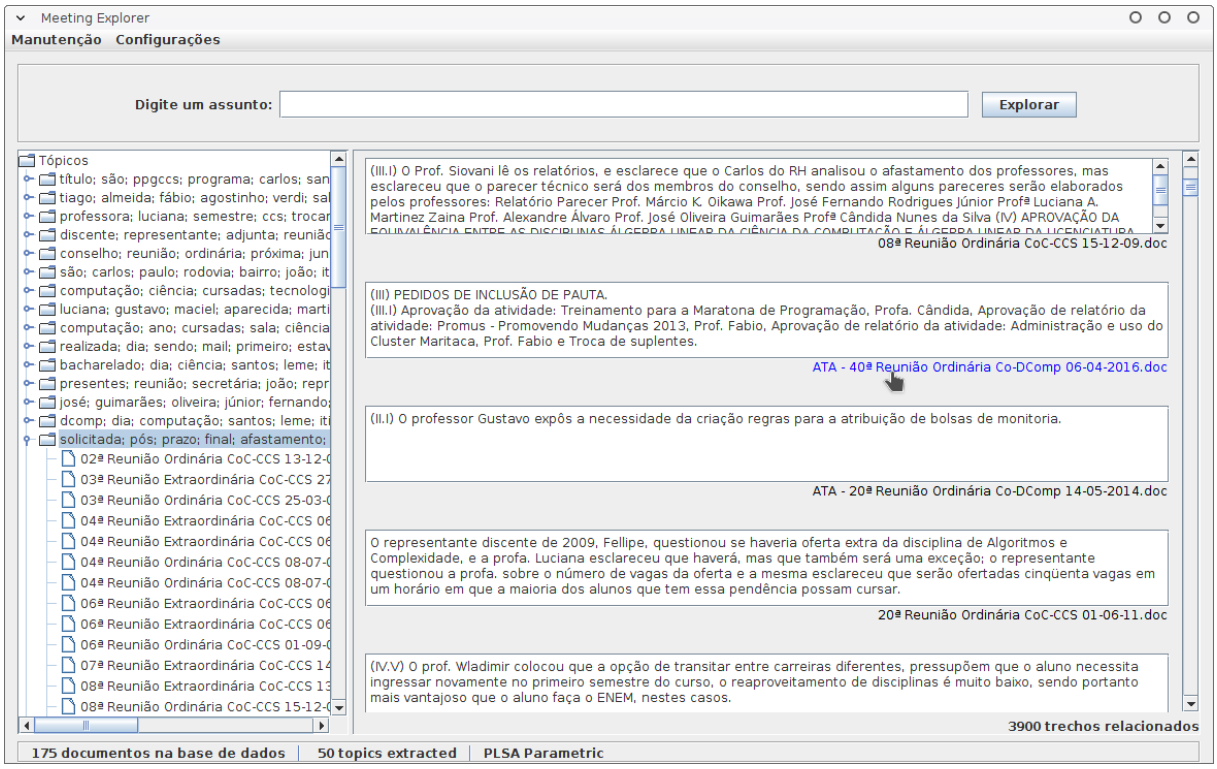

Figura 10 – Tela principal do sistema.

O sistema apresenta os cada resultado da busca exibindo o texto contido no segmento e um *link* para visualizar o arquivo original. Como parte da proposta, o sistema oferece como opção a ordenação cronológica dos resultados a fim de apresentar cada resultado dentro de um histórico de menções. Dessa forma o usuário tem acesso uma interface que lhe fornece uma visão temporal das menções ao tema pesquisado.

3.2 Validação em um copus de atas de reunião

O foco principal deste trabalho de mestrado é a exploração e recuperação de informação em atas de reunião. A escolha foi feita baseada nas características desses documentos como textos relativamente curtos, em comparação com outros documentos como notícias, artigos, sites da *web*; o estilo de escrita formal em que o redator evita repetições de temos e conceitos em benefício da estética do texto; Multiplicidade de

assuntos contidos em uma mesma ata, a qual é difícil determinar um assunto central, mas diversos assuntos independentes que foram tratados durante a reunião.

4 Avaliação dos Segmentadores

4.1 Preparação de um corpus de referência

A avaliação de um segmentador automático de textos exige uma referência, isto é, um conjunto de textos com os limites entre os segmentos conhecidos. Essa referência, deve ser confiável, sendo uma segmentação legítima que é capaz de dividir o texto em porções relativamente independentes, ou seja, uma segmentação ideal.

Selecionou-se um grupo de anotadores para analisar e coletar dados referentes a segmentação de cada ata. O grupo de anotadores foi formado por profissionais com alguma afinidade com atas de reunião, como profissionais administrativos, professores e coordenadores de curso. Optou-se por desenvolver um *software* como ferramenta para a coleta dos dados a fim de facilitar o trabalho de anotação e diminuir eventuais erros, conforme sugerido por (HOVY; LAVID, 2010). Essa ferramenta foi modelada para permitir aos anotadores visualizar os documentos e indicar livremente as divisões entre segmentos, bem como rotulá-los em classes e indicar palavras que melhor descrevem o assunto central do segmento. Os anotadores receberam informações básicas sobre o objetivo da pesquisa e instruções de como operar o *software*. Contudo, nenhum critério foi estabelecido para o procedimento ficando os anotadores livres para segmentar e rotular as atas orientados apenas pela interface da ferramenta. Na Figura 11 é mostrada a interface da ferramenta utilizada para as anotações.

Na Tabela 4 é mostrado um exemplo em que 6 dos 9 anotadores concordaram a respeito de um segmento. O trecho mostra quatro segmentos extraídos da segmentação de referência. Cada linha contém um sentença os números a esquerda indicam seu índice e os segmentos estão separados por uma linha horizontal.

Anexo a cada segmento é mostrado a classe uma amostra das classes e descritores rotulados por um dos anotadores. Esses rótulos não foram utilizados no processo de segmentação e não têm nenhuma influência sobre a segmentação de referência. Nesse trabalho, essas anotações são utilizadas na avaliação dos extratores de tópicos e do Módulo de consulta.

Após o processo de anotação, os dados coletados foram analisados para gerar as segmentações de referência. A segmentação de referência foi gerada utilizando o critério de maior concordância, como já relatado em outros trabalhos (HEARST, 1997; CARDOSO; PARDO; TABOADA, 2017; KAZANTSEVA; SZPAKOWICZ, 2012; PASSONNEAU; LITMAN, 1997; GALLEY et al., 2003). Considerou-se que ocorre um limite entre segmentos quando a maioria dos anotadores (metade mais um) concordaram que a mesma sentença é

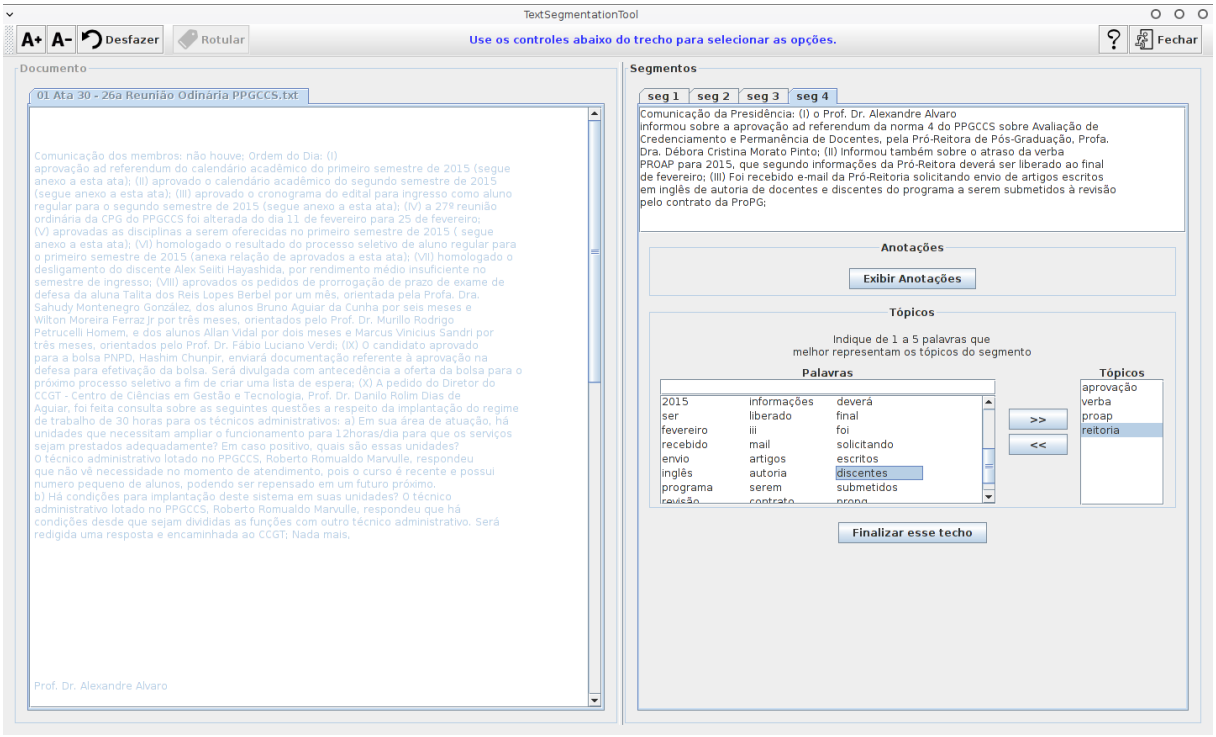

Figura 11 – Interface da ferramenta utilizada para anotações onde o texto a ser segmentado é exibido no painel a esquerda e os controles para anotação estão disponíveis a direita.

[7]	(II) Encerrada a etapa de inscrição para o processo seletivo como aluno regular para o segundo semestre de 2015: foram quarenta e nove inscrições on-line e dezoito candidatos entregaram a documentação; *<informe> <processo;seletivo>*
[8]	(III) O Prof. Dr. AAA informou que a Pró-Reitora comunicou a oferta de mais uma bolsa pela cota da Pró-Reitoria, mas não havia aluno disponível para alocação da bolsa.
[9]	(III) O Prof. Dr. AAA informou que a Pró-Reitora comunicou a oferta de mais uma bolsa pela cota da Pró-Reitoria, mas não havia aluno disponível para alocação da bolsa.
[10]	O Prof. Dr. BBB informou que havia uma aluna interessada, mas não informada durante o processo de elaboração do ranking no início do semestre.
[11]	Ficou decidido enviar e-mail aos docentes solicitando que comuniquem permanentemente interesse de alunos em bolsa pra atualização do ranking; *<informe> <solicitação;bolsa;cota;ranking;alunos>*
[12]	(IV) Com a mudança do Prof. Dr. DDD para o campus de São Carlos, o Prof. Dr. BBB assume o posto de suplente da linha Teoria Aplicada à Computação na CPG; *<informe> <mudança;suplente;teoria;aplicada;computação>*
[13]	Comunicação dos membros: Não houve; *<irrelevante>*

Tabela 3 – Exemplo de segmentação de referência com rotulação de um anotador

um final de segmento. A concordância entre os anotadores é uma medida importante que mostra como os anotadores compreendem os textos analisados e o nível de confiabilidade da segmentação de referência. Na Figura 12 é mostrado um exemplo de criação de uma segmentação de referência por meio da concordância entre anotadores. As primeiras linhas representam segmentações fornecidas por anotadores e a última linha representa a segmentação resultante da concordância entre a maioria dos segmentadores e portanto mais confiável.

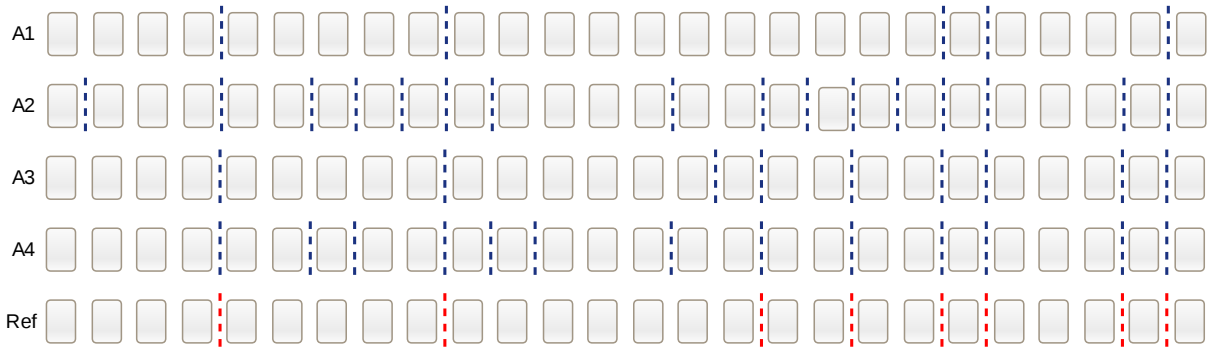

Figura 12 – Exemplo uma segmentação de referência criada a partir da concordância entre segmentações manuais.

Para mensurar a concordância entre anotadores, a medida *kappa (k)* (CARLETTA, 1996) é frequentemente utilizada (GRUENSTEIN; NIEKRASZ; PURVER, 2007; CARDOSO; PARDO; TABOADA, 2017; HEARST, 1997). Embora (CARLETTA, 1996) afirme que valores de k $>$ 0.8 indicam que os dados são confiáveis, visto a subjetividade da tarefa de segmentação textual, medidas menores podem ser aceitaveis, como reportado em (HEARST, 1997) que alcançou k = 0,64 e (CARDOSO; PARDO; TABOADA, 2017), k = 0,56.

A Tabela 4 contém, para cada ata, a quantidade de sentenças e a quantidade de segmentos identificadas pelos participantes.

Ata	**Sent.**	**A1**	**A2**	**A3**	**A4**	**A5**	**A6**	**A7**	**A8**	**A9**	**K**	**P_k**	**WD**
Ata 1	25	7	4	11	6	16	8	8	15	16	0.344	0.433	0.631
Ata 2	17	4	4	8	6	11	6	6	15	14	0.266	0.439	0.565
Ata 3	26	6	6	8	4	15	9	10	18	14	0.328	0.442	0.590
Ata 4	26	5	5	10	6	14	17	7	11	12	0.364	0.364	0.562
Ata 5	33	4	4	6	5	17	22	9	18	16	0.315	0.458	0.889
Ata 6	11	3	4	6	4	9	9	4	7	5	0.314	0.404	0.463
Ata 7	20	3	7	5	4	11	14	5	5	4	0.235	0.343	0.507
Ata 8	35	4	8	3	8	12	17	5	11	9	0.211	0.421	0.629
Ata 9	24	3	5	3	6	11	11	3	9	9	0.234	0.472	0.660
Ata 10	50	4	5	4	7	31	29	5	9	8	0.170	0.428	0.937
Ata 11	43	4	7	5	7	29	19	5	9	12	0.209	0.368	0.704
Ata 12	56	3	10	4	16	33	25	4	13	11	0.222	0.452	0.913

Tabela 4 – Descrição dos resultados obtidos pelos anotadores. Na segunda coluna, Sent., é mostrado a quantidade de sentenças de cada ata. Nas colunas A1-A9 é mostrado as quantidades de segmentos informados pelos anotadores. As colunas K, P_k e WD indicam respectivamente as medidas Kappa, P_k e WindDiff calculadas.

4.2 Configuração experimental

O *TextTiling* permite ajustarmos dois parâmetros, sendo o tamanho da janela e o passo. Por meio de testes empíricos escolheu-se os valores os valores 20, 40 e 60 para o tamanho da janela e 3, 6, 9 e 12 para o passo. Gerando ao final 20 configurações.

O *C99* permite o ajuste de três parâmetros, sendo, o primeiro a quantidade segmentos desejados, uma vez que, não se conhece o número ideal de segmentos e os documentos não apresentam muitos candidatos, calculou-se uma proporção dos candidatos a limite. Para isso atribuiu-se os valores 0,2; 0,4; 0,6; 0,8. Para o segundo parâmetro, o tamanho do quadro utilizado para gerar a matriz de ranking, atribuiu-se os valores 9 e 11, sendo 11 o valor padrão da apresentado pelo autor. O algoritmo permite ainda indicar se as sentenças serão representados por vetores contendo a frequência ou o peso de cada termo. Ambas as representações foram utilizadas. Considerando todos os parâmetros, foram geradas 16 configurações para o algoritmo *C99*.

Os algoritmos tradicionais baseados em coesão léxica como o *TextTiling* e *C99* são fortemente afetados pela distribuição das palavras no texto, pois a maioria das medidas de similaridade baseiam-se na frequência das palavras. Para esses, a remoção de termos menos significativo na etapa de pré-processamento pode influenciar o desempenho. Para outras abordagens como *MinCutSeg* e *BayesSeg* usou-se as configurações fornecidas por (EISENSTEIN; BARZILAY, 2008), onde essas técnicas foram utilizadas como *base line*. Para *TextSeg* não requer configuração de parâmetros. Há ainda outras estratégias passíveis de aplicação, como a utilização de fontes externas, por exemplo *thesaurus* e palavras pista, como discutido em (NAILI; CHAIBI; GHEZALA, 2016; GUTIERREZ et al., 2016; FERRET, 2009). Nesse trabalho, essas estratégias não são utilizadas para manter uma abordagem não supervisionada e independente de domínio.

4.3 Critérios de avaliação

Para fins de avaliação desse trabalho, um bom método de segmentação é aquele cujo resultado melhor se aproxima de uma segmentação de referência, sem a obrigatoriedade de estar perfeitamente alinhado com tal. Ou seja, visto o contexto das atas de reunião, e a subjetividade da tarefa, não é necessário que os limites entre os segmentos (real e hipótese) sejam idênticos, mas que se assemelhem em localização e quantidade.

Os algoritmos foram comparados com a segmentação de referência obtida e calculou-se as medidas mais aplicadas à segmentação textual, P_k e *WindowDiff*. Além dessas, computou-se também as medidas tradicionais acurácia, precisão, revocação e F^1 para comparação com outros trabalhos que as utilizam.

Calculou-se as medidas configurando cada algoritmo conforme mostrado na Sub-

seção 4.2. A fim de conhecer o impacto do pré-processamento nos algoritmos *TextTiling* e *C99*. Esses foram testados em duas etapas: com o texto integral, e com o texto pré-processado em que elementos menos significativos foram removidos, conforme mencionado na Seção 3.1.1. O teste de Friedman foi utilizado para gerar um ranking das melhores configurações para cada medida calculada. Com isso, foi possível descobrir quais valores otimizam um algoritmo para cada medida, considerando seus parâmetros e a influência do pré-processamento.

Como já mencionado, os algoritmos *MinCutSeg*, *TextSeg* e *BayesSeg* aplicou-se a etapa de pré-processamento e foram testados com as configurações apresentadas por (EISENSTEIN; BARZILAY, 2008).

4.4 Resultados

Obteve-se, por meio dos testes apresentados, as melhores configurações para as principais medidas de avaliação de segmentadores. Com essas configurações calculou-se a média de cada medida considerando o conjunto de documentos.

A seguir são apresentados os resultados obtidos com os algoritmos baseados em coesão léxica, considerando seus principais parâmetros e a aplicação do pré-processamento. Em seguida, são apresentados os resultados da avaliação final dos algoritmos abordados nesse trabalho.

Na Tabela 5 são apresentadas, as médias obtidas com o *TextTiling* bem como as configurações utilizadas, onde **J** é o tamanho da janela e **P** é o passo.

	Sem Pré-processamento			Com Pré-processamento		
Medida	**J**	**P**	**Média**	**J**	**P**	**Média**
P_k	50	9	0,142	50	9	0,144
WindowDiff	50	6	0,387	40	9	0,396
Acurácia	50	6	0,612	40	9	0,603
Precisão	40	9	0,611	50	12	0,613
Revocação	20	3	0,886	20	3	0,917
F^1	30	6	0,605	40	3	0,648

Tabela 5 – Resultados obtidos com o *TextTiling*

Uma vez que a coesão léxica é pressuposto de muitas abordagens em segmentação textual, fez-se uma análise desses documentos quanto a similaridade dos termos ao longo do texto. Verificou-se que a técnica de janelas deslizantes empregada pelo TextTiling encontra os vales que indicam transições entre segmentos, contudo ao comparar esses vales com a segmentação de referência, nota-se que a maioria dos limites coincide ou estão próximos aos vales, porém há casos onde a referência indica limites em trechos com alta coesão

léxica e outros onde a queda da coesão, indicada por vales, não coincide com nenhum limite de referência.

Na Figura 13 é apresentado a variação da coesão léxica ao longo de uma ata e a segmentação obtida pelo *TextTiling* usando tamanho de janela igual a 50 e passo 9. A linha horizontal representa a variação da coesão léxica e as linha verticais azuis e vermelhas representam os limites entre segmentos atribuídos pela referência e pelo algoritmo respectivamente.

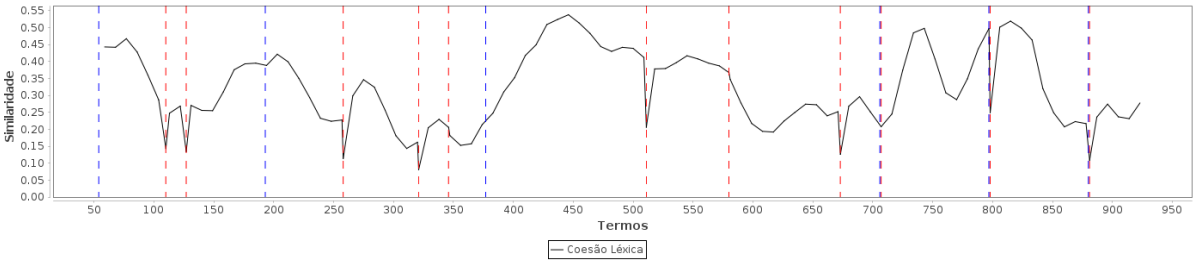

Figura 13 – Variação da coesão léxica ao longo de uma ata junto a uma segmentação automática em contraste com uma segmentação de referência.

Na Tabela 6 são apresentadas, as médias obtidas com o *C99* bem como as configurações utilizadas, onde **S** é a proporção de segmentos em relação a quantidade de candidatos, **M** é o tamanho do quadro utilizado para criar a matriz de *rankings* e **W** indica se os segmentos são representados por vetores contendo a frequência ou um peso das palavras.

	Sem Pré-processamento				Com Pré-processamento			
Medida	**S**	**M**	**W**	**Média**	**S**	**M**	**W**	**Média**
P_k	20	9	Sim	0,134	20	11	False	0,116
WindowDiff	60	9	Sim	0,411	60	9	Sim	0,390
Acurácia	60	9	Sim	0,588	60	9	Sim	0,609
Precisão	40	9	Sim	0,645	20	11	False	0,720
Revocação	80	9	Sim	0,869	80	11	Sim	0,897
F^1	80	9	Sim	0,638	80	11	Sim	0,655

Tabela 6 – Resultados obtidos com o *C99*

Verificou-se que, entre os métodos baseados em coesão léxica, o *C99* obteve melhor desempenho em acurácia, precisão, F^1, P_k e *WindowDiff*, em relação ao *TextTiling*, enquanto este obteve o melhor desempenho em revocação. De maneira geral, o algoritmo *C99* apresenta melhores resultados em relação ao *TextTiling*, contudo testes estatísticos realizados indicaram que não houve diferença significativa entre os métodos.

Com os testes anteriores obteve-se, para cada medida, 4 configurações levando em conta ambos os algoritmos e a presença ou ausência do pré-processamento. Por meio do teste de Friedman e Nemenyi e verificou-se que não há diferença crítica entre os métodos

TextTiling e *C99*. Na Figura 14 é mostrado os Diagramas para as medidas *WindowDiff*, P_k, Acurácia, Precisão, Revocação e F^1.

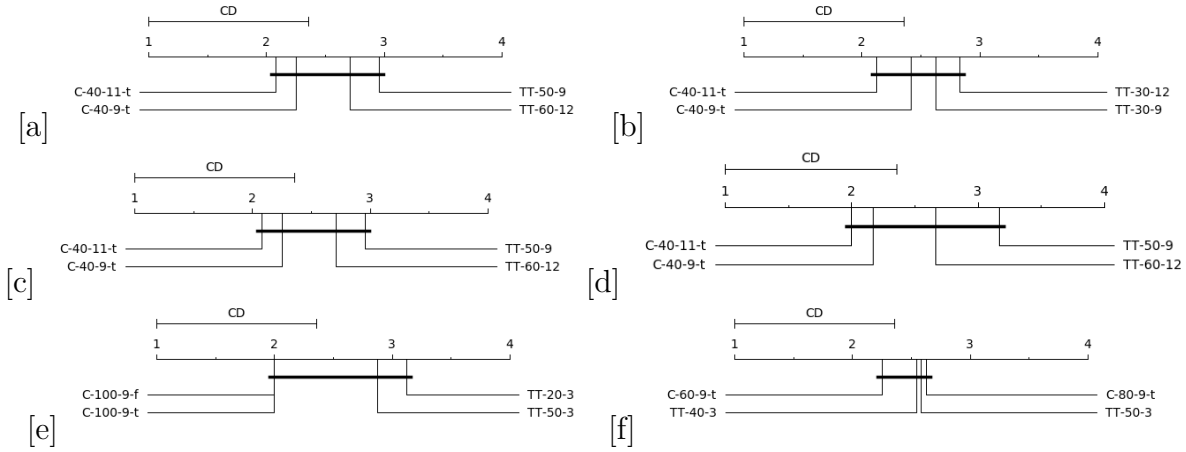

Figura 14 – Diagramas de Diferença Crítica sobre *ranking* dos algoritmos de segmentação baseados em coesão léxica de acordo com valores de *WindowDiff*, P_k, Acurácia, Precisão, Revocação e F^1.

A avaliação final foi feita pela comparação dos algoritmos usando as medidas *Pk* e *WindowDiff*. É apresentada também, para fins de comparação, as medidas tradicionais acurácia, precisão, revocação e F^1, entretanto, nesse contexto, essas medidas são menos significativa que Pk e WindowDiff, conforme já mencionado na Seção 2.5.1. A Tabela **??** contém as médias com cada algoritmo. Vale lembrar que P_k e *WindowDiff* são medidas de dissimilaridade, ou seja, os valores menores significam melhores resultados.

Na Figura 20 é apresentada a performance dos algoritmos nas medidas tradicionais. Observa-se valores altos de revocação para a segmentação por sentenças, pois é atribuído um limite a todo candidato a final de segmento, o que resulta no valor máximo para revocação. De maneira semelhante, o comportamento do *TextTiling* gera mais segmentos em relação aos demais, e com isso tem-se valores maiores de revocação, o que pode ser contornado configurando o algoritmo com passos maiores, ou ainda, sobre-escrevendo a função que calcula os *depth scores* para reconhecer vales mais largos.

Após a identificação dos segmentos, o algoritmo retorna uma lista com fragmentos do texto original. Cada segmento é incorporado à estrutura de dados interna como subdocumentos com um tema relativamente independente. Em seguida, esses subdocumentos serão analisados pelo extrator de tópicos para identificação de descritores e agrupamento.

Algoritmo		**Step**	**Win**	**P_k**	**WD**	**Ac**	**Pr**	**Re**	**F^1**	**#Segs**
TextTiling		20	30	0.461	0.444	0.581	0.560	**0.336**	**0.411**	8.833
TextTiling		30	45	**0.450**	**0.435**	**0.596**	**0.696**	0.275	0.373	6.417
Algoritmo	**RS**	**W**	**SRate**	**P_k**	**WD**	**Ac**	**Pr**	**Re**	**F^1**	**#Segs**
C99	3	true	0.300	**0.434**	**0.407**	0.607	0.655	0.376	0.457	9.250
C99	3	true	0.700	0.485	0.431	0.602	0.553	**0.797**	**0.633**	21.417
C99	5	true	0.500	0.460	0.421	**0.609**	0.580	0.600	0.571	15.500
C99	3	false	0.200	0.448	0.427	0.596	**0.719**	0.257	0.362	6.083
Algoritmo		**Cut**	**SRate**	**P_k**	**WD**	**Ac**	**Pr**	**Re**	**F^1**	**#Segs**
MinCutSeg		13	0.300	0.457	0.427	0.594	**0.638**	0.353	0.433	8.667
MinCutSeg		9	0.400	**0.444**	0.408	**0.614**	0.629	0.494	0.526	11.917
MinCutSeg		11	0.500	0.459	**0.407**	0.603	0.588	0.590	0.563	15.000
MinCutSeg		5	0.700	0.528	0.438	0.567	0.536	**0.746**	**0.599**	21.000
Algoritmo	**Prior**	**Disp.**	**SRate**	**P_k**	**WD**	**Ac**	**Pr**	**Re**	**F^1**	**#Segs**
BayesSeg	0.0800	0.5000	Auto	**0.380**	**0.361**	**0.655**	0.662	0.479	0.551	10.000
BayesSeg	0.1100	0.5000	Auto	0.388	0.370	0.649	**0.672**	0.433	0.523	9.000
BayesSeg	0.1100	0.1000	0.600	0.462	0.399	0.615	0.574	0.724	**0.619**	18.417
BayesSeg	0.0800	0.1000	0.900	0.645	0.517	0.490	0.478	**0.878**	0.600	27.500
Algoritmo			**SRate**	**P_k**	**WD**	**Ac**	**Pr**	**Re**	**F^1**	**#Segs**
TextSeg			Auto	**0.455**	0.439	0.585	**0.618**	0.266	0.368	6.417
TextSeg			0.500	0.475	**0.417**	**0.594**	0.565	0.608	0.566	15.500
TextSeg			0.900	0.604	0.484	0.524	0.498	**0.922**	**0.627**	27.500
Algoritmo			**SRate**	**P_k**	**WD**	**Ac**	**Pr**	**Re**	**F^1**	**#Segs**
Sentenças			1.000	**0.640**	**0.490**	**0.506**	**0.488**	**1.000**	**0.638**	30.500

Tabela 7 – Resultados

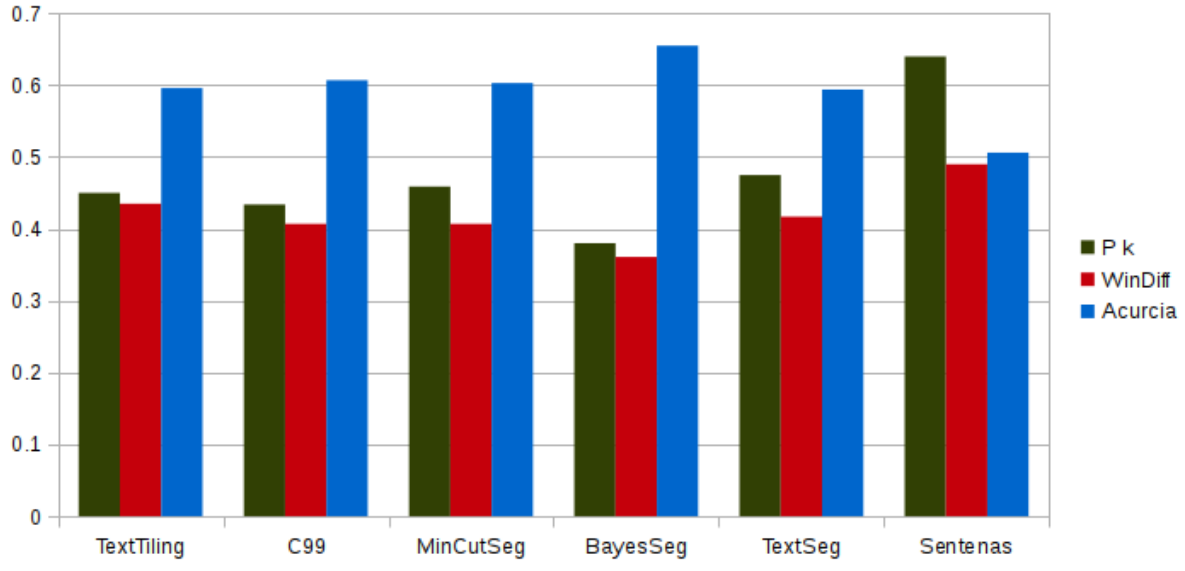

Figura 15 – Performance dos algoritmos de segmentação textual

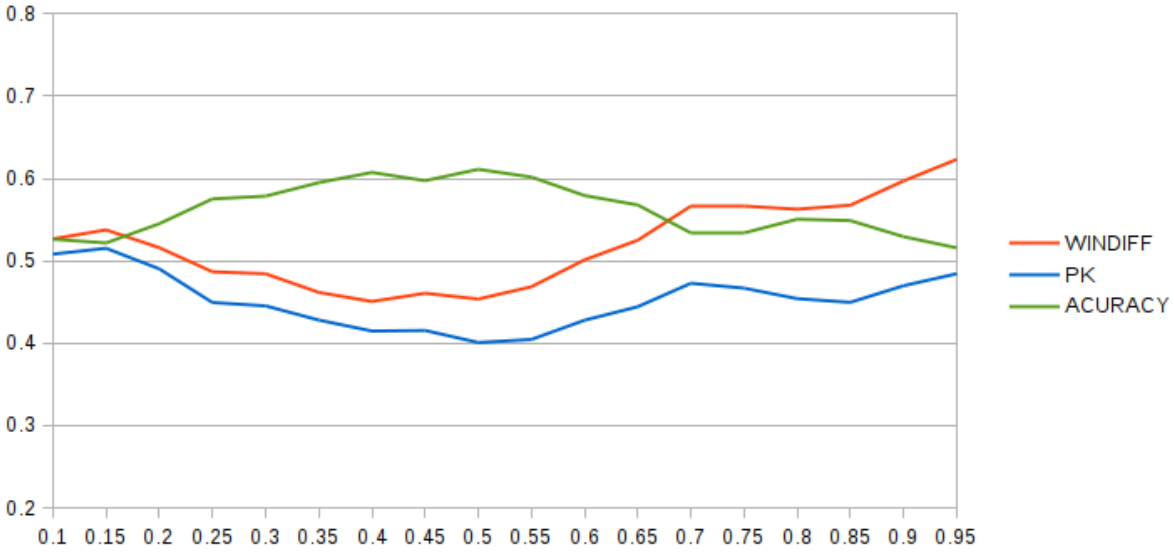

Figura 16 – Performance dos algoritmos de segmentação textual

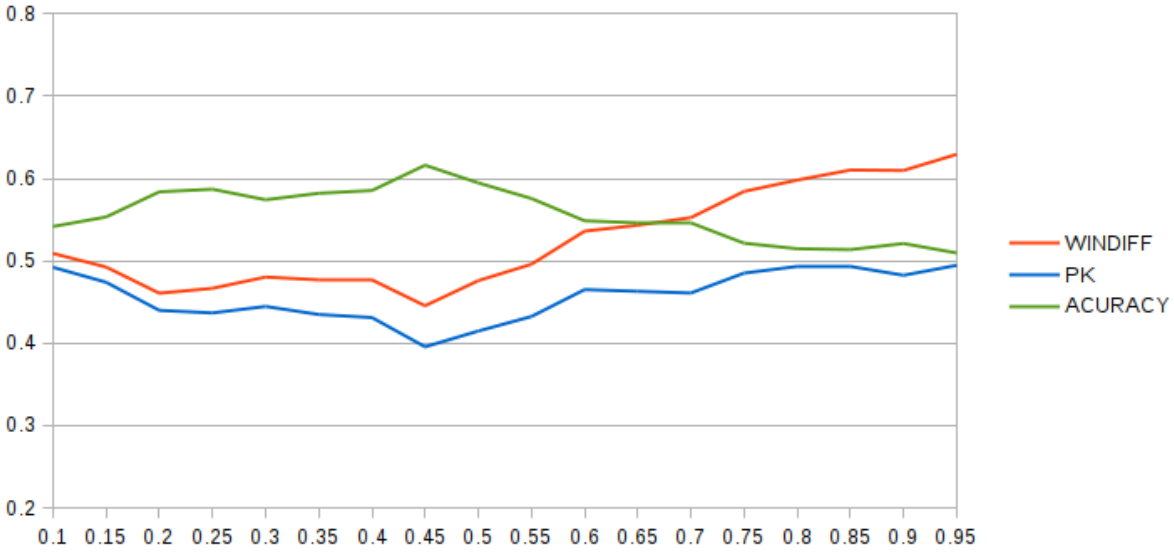

Figura 17 – Performance dos algoritmos de segmentação textual

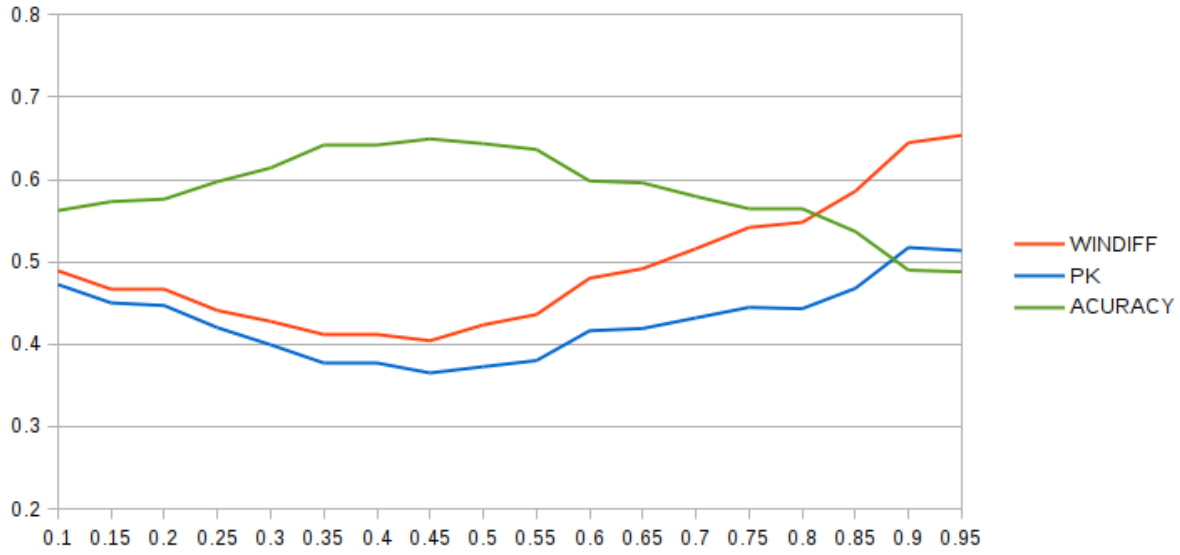

Figura 18 – Performance dos algoritmos de segmentação textual

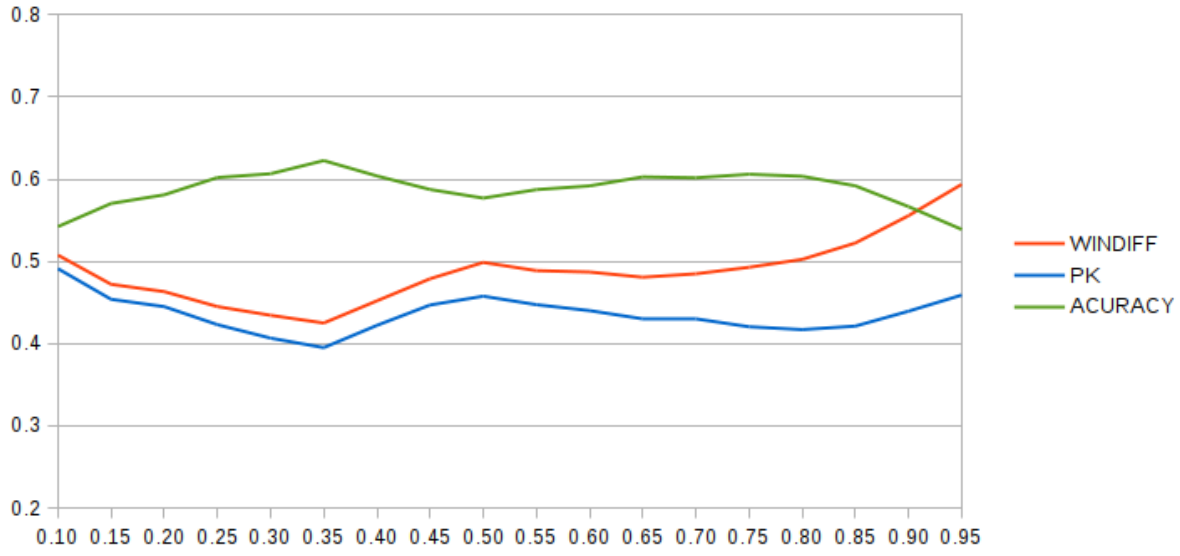

Figura 19 – Performance dos algoritmos de segmentação textual

Figura 20 – Performance dos algoritmos de segmentação textual

5 Avaliação dos Extratores de Tópicos

Neste capítulo, o processo de extração de tópicos em segmentos de atas de reunião é avaliado. A avaliação experimental neste trabalho é focada em três pontos principais: (1) Comparar algoritmos de extração de tópicos na tarefa de extração de padrões no contexto das atas de reunião. (2) Analisar a qualidade dos agrupamentos no que tange a navegação por grupos com mesmo tópico, (3) Analisar a qualidade dos descritores extraídos para recuperar os documentos dos grupos.

5.1 Configuração experimental

5.2 Critérios de avaliação

5.3 Resultados

6 Avaliação do Módulo de Consulta

Referências

AGGARWAL, C. C.; ZHAI, C. X. *Mining Text Data*. [S.l.]: Springer Publishing Company, Incorporated, 2012. ISBN 1461432227, 9781461432227. Citado na página 6.

BEEFERMAN, D.; BERGER, A.; LAFFERTY, J. Statistical models for text segmentation. *Machine Learning*, v. 34, n. 1, p. 177–210, 1999. ISSN 1573-0565. Disponível em: <http://dx.doi.org/10.1023/A:1007506220214>. Citado na página 24.

BLEI, D. M. Probabilistic topic models. *Commun. ACM*, ACM, New York, NY, USA, v. 55, n. 4, p. 77–84, abr. 2012. ISSN 0001-0782. Disponível em: <http://doi.acm.org/10.1145/2133806.2133826>. Citado 3 vezes nas páginas 11, 29 e 31.

BLEI, D. M.; NG, A. Y.; JORDAN, M. I. Latent dirichlet allocation. *J. Mach. Learn. Res.*, JMLR.org, v. 3, p. 993–1022, mar. 2003. ISSN 1532-4435. Disponível em: <http://dl.acm.org/citation.cfm?id=944919.944937>. Citado 3 vezes nas páginas 7, 21 e 31.

BOKAEI, M. H.; SAMETI, H.; LIU, Y. Linear discourse segmentation of multi-party meetings based on local and global information. *IEEE/ACM Trans. Audio, Speech and Lang. Proc.*, IEEE Press, Piscataway, NJ, USA, v. 23, n. 11, p. 1879–1891, nov. 2015. ISSN 2329-9290. Disponível em: <http://dx.doi.org/10.1109/TASLP.2015.2456430>. Citado 3 vezes nas páginas 7, 18 e 36.

BOKAEI, M. H.; SAMETI, H.; LIU, Y. Extractive summarization of multiparty meetings through discourse segmentation. *Natural Language Engineering*, Cambridge University Press, v. 22, n. 1, p. 41–72, 2016. Citado na página 36.

CAO, L. Data science: A comprehensive overview. *ACM Comput. Surv.*, ACM, New York, NY, USA, v. 50, n. 3, p. 43:1–43:42, jun. 2017. ISSN 0360-0300. Disponível em: <http://doi.acm.org/10.1145/3076253>. Citado na página 5.

CARDOSO, P.; PARDO, T.; TABOADA, M. Subtopic annotation and automatic segmentation for news texts in brazilian portuguese. *Corpora*, Edinburgh University Press, v. 12, n. 1, p. 23–54, 2017. Citado 4 vezes nas páginas 18, 29, 41 e 43.

CARLETTA, J. Assessing agreement on classification tasks: The kappa statistic. *Comput. Linguist.*, MIT Press, Cambridge, MA, USA, v. 22, n. 2, p. 249–254, jun. 1996. ISSN 0891-2017. Disponível em: <http://dl.acm.org/citation.cfm?id=230386.230390>. Citado 2 vezes nas páginas 29 e 43.

CHAIBI, A. H.; NAILI, M.; SAMMOUD, S. Topic segmentation for textual document written in arabic language. *Procedia Computer Science*, v. 35, p. 437 – 446, 2014. ISSN 1877-0509. Disponível em: <http://www.sciencedirect.com/science/article/pii/S1877050914010898>. Citado 2 vezes nas páginas 18 e 27.

CHENG, X. et al. Learning topics in short texts by non-negative matrix factorization on term correlation matrix. In: *SDM*. SIAM, 2013. p. 749–757. ISBN 978-1-61197-283-2. Disponível em: <http://dblp.uni-trier.de/db/conf/sdm/sdm2013.html#ChengGLWY13>. Citado na página 30.

CHOI, F. Y. Y. Advances in domain independent linear text segmentation. In: *Proceedings of the 1st North American Chapter of the Association for Computational Linguistics Conference*. Stroudsburg, PA, USA: Association for Computational Linguistics, 2000. (NAACL 2000), p. 26–33. Disponível em: <http://dl.acm.org/citation.cfm?id=974305.974309>. Citado 3 vezes nas páginas 7, 18 e 27.

CROFT, B.; METZLER, D.; STROHMAN, T. *Search Engines: Information Retrieval in Practice*. 1st. ed. USA: Addison-Wesley Publishing Company, 2009. ISBN 0136072240, 9780136072249. Citado 2 vezes nas páginas 15 e 16.

DEERWESTER, S. et al. Indexing by latent semantic analysis. *Journal of the American Society for Information Science*, v. 41, n. 6, p. 391–407, 1990. Citado 2 vezes nas páginas 11 e 30.

DIAS, G.; ALVES, E.; LOPES, J. G. P. Topic segmentation algorithms for text summarization and passage retrieval: An exhaustive evaluation. In: *Proceedings of the 22Nd National Conference on Artificial Intelligence - Volume 2*. AAAI Press, 2007. (AAAI'07), p. 1334–1339. ISBN 978-1-57735-323-2. Disponível em: <http://dl.acm.org/citation.cfm?id=1619797.1619859>. Citado na página 18.

EISENSTEIN, J.; BARZILAY, R. Bayesian unsupervised topic segmentation. In: *Proceedings of the Conference on Empirical Methods in Natural Language Processing*. Stroudsburg, PA, USA: Association for Computational Linguistics, 2008. (EMNLP '08), p. 334–343. Disponível em: <http://dl.acm.org/citation.cfm?id=1613715.1613760>. Citado 5 vezes nas páginas 7, 19, 21, 44 e 45.

FALEIROS, T. d. P. *Propagação em grafos bipartidos para extração de tópicos em fluxo de documentos textuais*. Tese (Doutorado), 2016. Citado na página 7.

FELDMAN, R.; SANGER, J. *Text Mining Handbook: Advanced Approaches in Analyzing Unstructured Data*. New York, NY, USA: Cambridge University Press, 2006. ISBN 0521836573, 9780521836579. Citado na página 12.

FERRET, O. Improving text segmentation by combining endogenous and exogenous methods. In: *International Conference Recent Advances in Natural Language Processing, RANLP*. [S.l.: s.n.], 2009. p. 88–93. Citado na página 44.

GALLEY, M. et al. Discourse segmentation of multi-party conversation. In: *Proceedings of the 41st Annual Meeting on Association for Computational Linguistics - Volume 1*. Stroudsburg, PA, USA: Association for Computational Linguistics, 2003. (ACL '03), p. 562–569. Disponível em: <http://dx.doi.org/10.3115/1075096.1075167>. Citado 3 vezes nas páginas 18, 27 e 41.

GRUENSTEIN, A.; NIEKRASZ, J.; PURVER, M. *MEETING STRUCTURE ANNOTATION – Annotations Collected with a General Purpose Toolkit*. [S.l.]: Springer, Dordrecht, 2007. Citado 2 vezes nas páginas 28 e 43.

GUTIERREZ, F. et al. A hybrid ontology-based information extraction system. *J. Inf. Sci.*, Sage Publications, Inc., Thousand Oaks, CA, USA, v. 42, n. 6, p. 798–820, dez. 2016. ISSN 0165-5515. Disponível em: <https://doi.org/10.1177/0165551515610989>. Citado na página 44.

HEARST, M. A. Texttiling: Segmenting text into multi-paragraph subtopic passages. *Comput. Linguist.*, MIT Press, Cambridge, MA, USA, v. 23, n. 1, p. 33–64, mar. 1997. ISSN 0891-2017. Disponível em: <http://dl.acm.org/citation.cfm?id=972684.972687>. Citado 3 vezes nas páginas 26, 41 e 43.

HOFMANN, T. Probabilistic latent semantic indexing. In: *Proceedings of the 22Nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*. New York, NY, USA: ACM, 1999. (SIGIR '99), p. 50–57. ISBN 1-58113-096-1. Disponível em: <http://doi.acm.org/10.1145/312624.312649>. Citado 3 vezes nas páginas 7, 11 e 31.

HOVY, E.; LAVID, J. Towards a 'science' of corpus annotation: A new methodological challenge for corpus linguistics. v. 22, p. 13–36, 01 2010. Citado 3 vezes nas páginas 26, 27 e 41.

JANIN, A. et al. The icsi meeting corpus. In: *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*. [S.l.: s.n.], 2003. p. 364–367. Citado na página 27.

JEONG, M.; TITOV, I. Multi-document topic segmentation. In: *Proceedings of the 19th ACM International Conference on Information and Knowledge Management*. New York, NY, USA: ACM, 2010. (CIKM '10), p. 1119–1128. ISBN 978-1-4503-0099-5. Disponível em: <http://doi.acm.org/10.1145/1871437.1871579>. Citado na página 32.

KAZANTSEVA, A.; SZPAKOWICZ, S. Topical segmentation: A study of human performance and a new measure of quality. In: *Proceedings of the 2012 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. Stroudsburg, PA, USA: Association for Computational Linguistics, 2012. (NAACL HLT '12), p. 211–220. ISBN 978-1-937284-20-6. Disponível em: <http://dl.acm.org/citation.cfm?id=2382029.2382058>. Citado 3 vezes nas páginas 26, 29 e 41.

KERN, R.; GRANITZER, M. Efficient linear text segmentation based on information retrieval techniques. *Proceedings of the International Conference on Management of Emergent Digital EcoSystems, MEDES '09*, p. 167–171, 2009. Disponível em: <https://www.scopus.com/inward/record.uri?eid=2-s2.0-74549147972&doi=10.1145%2f1643823.1643854&partnerID=40&md5=1c6f73bc0e07446fcc178440e48bbc40>. Citado 2 vezes nas páginas 18 e 24.

KOZIMA, H. Text segmentation based on similarity between words. In: *Proceedings of the 31st Annual Meeting on Association for Computational Linguistics*. Stroudsburg, PA, USA: Association for Computational Linguistics, 1993. (ACL '93), p. 286–288. Disponível em: <http://dx.doi.org/10.3115/981574.981616>. Citado na página 16.

LEE, D.; SEUNG, H. S. Learning the parts of objects by non-negative matrix factorization. v. 401, p. 788–91, 11 1999. Citado 2 vezes nas páginas 11 e 30.

LEE, J.-K. et al. Two-step sentence extraction for summarization of meeting minutes. In: . [S.l.: s.n.], 2011. v. 39, p. 614–619. Citado na página 5.

MALIOUTOV, I.; BARZILAY, R. Minimum cut model for spoken lecture segmentation. In: *Proceedings of the 21st International Conference on Computational Linguistics and the*

44th Annual Meeting of the Association for Computational Linguistics. Stroudsburg, PA, USA: Association for Computational Linguistics, 2006. (ACL-44), p. 25–32. Disponível em: <https://doi.org/10.3115/1220175.1220179>. Citado 2 vezes nas páginas 21 e 22.

MANNING, C. D.; RAGHAVAN, P.; SCHüTZE, H. *Introduction to Information Retrieval*. New York, NY, USA: Cambridge University Press, 2008. ISBN 0521865719, 9780521865715. Citado 5 vezes nas páginas 5, 6, 13, 15 e 16.

MASAO, U.; KôITI, H. Multi-topic multi-document summarization. In: *Proceedings of the 18th Conference on Computational Linguistics - Volume 2*. Stroudsburg, PA, USA: Association for Computational Linguistics, 2000. (COLING '00), p. 892–898. Disponível em: <https://doi.org/10.3115/992730.992775>. Citado na página 32.

MISRA, H. et al. Text segmentation via topic modeling: An analytical study. In: *Proceedings of the 18th ACM Conference on Information and Knowledge Management*. New York, NY, USA: ACM, 2009. (CIKM '09), p. 1553–1556. ISBN 978-1-60558-512-3. Disponível em: <http://doi.acm.org/10.1145/1645953.1646170>. Citado 3 vezes nas páginas 7, 18 e 36.

NAILI, M.; CHAIBI, A. H.; GHEZALA, H. H. B. Exogenous approach to improve topic segmentation. *International Journal of Intelligent Computing and Cybernetics*, v. 9, n. 2, p. 165–178, 2016. Disponível em: <https://doi.org/10.1108/IJICC-01-2016-0001>. Citado 2 vezes nas páginas 18 e 44.

O'CALLAGHAN, D. et al. An analysis of the coherence of descriptors in topic modeling. *Expert Syst. Appl.*, Pergamon Press, Inc., Tarrytown, NY, USA, v. 42, n. 13, p. 5645–5657, ago. 2015. ISSN 0957-4174. Disponível em: <http://dx.doi.org/10.1016/j.eswa.2015.02.055>. Citado na página 7.

PASSONNEAU, R. J.; LITMAN, D. J. Discourse segmentation by human and automated means. *Comput. Linguist.*, MIT Press, Cambridge, MA, USA, v. 23, n. 1, p. 103–139, mar. 1997. ISSN 0891-2017. Disponível em: <http://dl.acm.org/citation.cfm?id=972684.972689>. Citado 2 vezes nas páginas 27 e 41.

PEVZNER, L.; HEARST, M. A critique and improvement of an evaluation metric for text segmentation. *Computational Linguistics*, v. 28, n. 1, p. 19–36, 2002. Disponível em: <https://www.scopus.com/inward/record.uri?eid=2-s2.0-0037870455&doi=10.1162%2f089120102317341756&partnerID=40&md5=279abc4e76fcfc2c4a1896e76a245034>. Citado na página 25.

REYNAR, J. C. *Topic Segmentation: Algorithms and Applications*. Tese (Doutorado), Philadelphia, PA, USA, 1998. AAI9829978. Citado na página 19.

REZENDE, S. O. *Sistemas Inteligentes*. Barueri, SP: Manole, 2003. 337 - 270 p. Citado 3 vezes nas páginas 11, 12 e 14.

REZENDE, S. O.; MARCACINI, R. M.; MOURA, M. F. O uso da mineração de textos para extração e organização não supervisionada de conhecimento. *Revista de Sistemas de Informação da FSMA*, v. 7, p. 7–21, 2011. ISSN 19835604. Disponível em: <http://www.fsma.edu.br/si/7edicao.html>. Citado na página 32.

RIJSBERGEN, C. J. V. *Information Retrieval*. 2nd. ed. Newton, MA, USA: Butterworth-Heinemann, 1979. ISBN 0408709294. Citado na página 16.

RUBIN, T. N. et al. Statistical topic models for multi-label document classification. *Mach. Learn.*, Kluwer Academic Publishers, Hingham, MA, USA, v. 88, n. 1-2, p. 157–208, jul. 2012. ISSN 0885-6125. Disponível em: <https://doi.org/10.1007/s10994-011-5272-5>. Citado na página 32.

SAKAHARA, M.; OKADA, S.; NITTA, K. Domain-independent unsupervised text segmentation for data management. In: *2014 IEEE International Conference on Data Mining Workshop*. [S.l.: s.n.], 2014. p. 481–487. ISSN 2375-9232. Citado 2 vezes nas páginas 7 e 36.

SALTON, G.; ALLAN, J. Automatic text decomposition and structuring. In: *Intelligent Multimedia Information Retrieval Systems and Management - Volume 1*. Paris, France, France: LE CENTRE DE HAUTES ETUDES INTERNATIONALES D'INFORMATIQUE DOCUMENTAIRE, 1994. (RIAO '94), p. 6–20. Disponível em: <http://dl.acm.org/citation.cfm?id=2856823.2856826>. Citado na página 15.

SALTON, G.; BUCKLEY, C. Term-weighting approaches in automatic text retrieval. *Inf. Process. Manage.*, Pergamon Press, Inc., Tarrytown, NY, USA, v. 24, n. 5, p. 513–523, ago. 1988. ISSN 0306-4573. Disponível em: <http://dx.doi.org/10.1016/0306-4573(88)90021-0>. Citado na página 15.

SASAKI, Y.; REA, B.; ANANIADOU, S. Multi-topic aspects in clinical text classification. In: *Proceedings of the 2007 IEEE International Conference on Bioinformatics and Biomedicine*. Washington, DC, USA: IEEE Computer Society, 2007. (BIBM '07), p. 62–70. ISBN 0-7695-3031-1. Disponível em: <http://dx.doi.org/10.1109/BIBM.2007.53>. Citado na página 32.

SCHWARTZ-ZIV, M.; WEISBACH, M. S. What do boards really do? evidence from minutes of board meetings. *Journal of Financial Economics*, v. 108, n. 13, p. 349–366, 2013. Citado na página 5.

SHAMSINEJADBABKI, P.; SARAEE, M. A new unsupervised feature selection method for text clustering based on genetic algorithms. *J. Intell. Inf. Syst.*, Kluwer Academic Publishers, Hingham, MA, USA, v. 38, n. 3, p. 669–684, jun. 2012. ISSN 0925-9902. Disponível em: <http://dx.doi.org/10.1007/s10844-011-0172-5>. Citado na página 15.

SHI, J.; MALIK, J. Normalized cuts and image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, v. 22, n. 8, p. 888–905, Aug 2000. ISSN 0162-8828. Citado 2 vezes nas páginas 21 e 22.

SHROUT, P. E.; FLEISS, J. L. Intraclass correlations: Uses in assessing rater reliability. v. 86, p. 420–8, 04 1979. Citado na página 29.

STEYVERS, M.; GRIFFITHS, T. Probabilistic topic models. In: LANDAUER, T.; MCNAMARA, S. D.; KINTSCH, W. (Ed.). *Latent Semantic Analysis: A Road to Meaning*. Laurence Erlbaum, 2007. cap. Probabilistic topic models. Disponível em: <http://psiexp.ss.uci.edu/research/papers/SteyversGriffithsLSABookFormatted.pdf>. Citado 2 vezes nas páginas 7 e 29.

TAGARELLI, A.; KARYPIS, G. A segment-based approach to clustering multi-topic documents. *Knowledge and Information Systems*, v. 34, n. 3, p. 563–595, Mar 2013. ISSN 0219-3116. Disponível em: <https://doi.org/10.1007/s10115-012-0556-z>. Citado na página 32.

TAN, P.-N.; STEINBACH, M.; KUMAR, V. *Introduction to Data Mining, (First Edition)*. Boston, MA, USA: Addison-Wesley Longman Publishing Co., Inc., 2005. ISBN 0321321367. Citado 2 vezes nas páginas 12 e 15.

UTIYAMA, M.; ISAHARA, H. A statistical model for domain-independent text segmentation. In: *Proceedings of the 39th Annual Meeting on Association for Computational Linguistics*. Stroudsburg, PA, USA: Association for Computational Linguistics, 2001. (ACL '01), p. 499–506. Disponível em: <https://doi.org/10.3115/1073012.1073076>. Citado na página 20.

ZHU, D. et al. Intuitive topic discovery by incorporating word-pair's connection into lda. In: *2012 IEEE/WIC/ACM International Conferences on Web Intelligence and Intelligent Agent Technology*. [S.l.: s.n.], 2012. v. 1, p. 303–310. Citado na página 7.