Ovídio José Francisco

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

Sorocaba, SP

11 de fevereiro de 2018

Lista de símbolos

Γ	Letra grega Gama
Λ	Lambda
ζ	Letra grega minúscula zeta
$\in$	Pertence

Sumário

1 Introdução

2 Conceituação Teórica

A popularidade dos computares permite a criação e compartilhamento de textos onde a quantidade de informação facilmente extrapola a capacidade de humana de leitura e análise de coleções de documentos, estejam eles disponíveis na Internet ou em computadores pessoais. A necessidade de simplificar e organizar grandes coleções de documentos criou uma demanda por modelos de aprendizado de máquina para extração de conhecimento em bases textuais. 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).

2.1 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 (KOZIMA, 1993). O autor demonstrou que há uma estreita correlação entre quedas na coesão léxica em janelas de texto e a transição de assuntos. Em seu trabalho, 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 dependia da construção de uma rede adequada.

Para encontrar os segmentos de um texto, alguns dos primeiros algoritmos utilizam a técnica de janelas deslizantes, onde se verifica a frequência dos termos em um fragmento do documento. Inicialmente, estabelece-se a partir do início do texto, um *range* de w 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.

A partir desses conceitos, um dos primeiros algoritmos baseados na ideia que um

segmento pode ser identificado pela análise das palavras que o compõe foi o *TextTiling*. O *TextTiling* é um algoritmo baseado em janelas deslizantes, em que, para cada candidato a limite, analisa-se o texto circundante. O *TextTiling* recebe uma lista de candidatos a limite, usualmente finais de parágrafo ou finais de sentenças. Para cada posição candidata são construídos 2 blocos, um contendo 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 precede.

Em seguida, os blocos de texto são representados por vetores que contém as frequências de suas palavras. Diferente da proposta de Kosima, utiliza *cosine* como medida para a similaridade entre os blocos, conforme apresentada na Equação 2.1, onde dados dois blocos de texto, x e y, $f_{x,j}$ é a frequência do termo j em x e $f_{y,j}$ é a frequência do termo j em y.

$$Sim(x, y) = \frac{\Sigma_j f_{x,j} \times f_{y,j}}{\sqrt{\Sigma_j f_{x,j}^2 \times \Sigma f_{y,j}^2}} \tag{2.1}$$

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 y_i atribui-se uma profundidade dada por $(y_{i-1} - y_i) + (y_{i+1} - y_i)$ 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 os deslocamentos da janela deslizante e a curva de dissimilaridade entre os blocos adjacentes.

O TextTiling apresenta como vantagens a facilidade de implementação e baixa complexidade computacional, favorecendo a implementação de trabalhos similares (**??????????**), e usado com base line em outros trabalhos (**????**). Por outro lado, algoritmos mais complexos, como os baseados em matrizes de similaridade, apresentam acurácia relativamente superior como apresentado posteriormente em (**??????**).

Outro algoritmo frequentemente referenciado na literatura é o C99 (**??**) o qual é baseado em uma matriz de *ranking* das similaridades. Embora muitos trabalhos utilizem a coesão léxica do texto, para pequenos segmentos pode não ser confiável, 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. Então, 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 quanto $i = j$ (ou seja, com a mesma sentença) e revela quadrados com maior concentração de pontos ao longo da diagonal. Essas regiões indicam porções de texto com maior coesão léxica.

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, cada elemento é comparado com seus vizinhos dentro de uma região denominada máscara. Na Figura 2.1 é destacado um quadro 3 x 3 de uma matriz em que cada elemento é a similaridade entre duas unidades de informação. 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 2.1 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 facilitando 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* (**??**) 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 i e j que representam uma região quadrada ao longo da diagonal da matriz. Calcula-se a densidade dessa região como mostrado na Equação 2.2. Seja $s_{i,j}$ a somatória dos *rakings* de um segmento e $a_{i,j}$ sua área interior. Seja $B = \{b1, ..., b_m\}$ a lista de m segmentos e s_k e a_k são a somatória dos valores dos rankings e a área de um segmento k em B. Então, a densidade é computada por:

$$D = \frac{\sum_{k=1}^{m} s_k}{\sum_{k=1}^{m} a_k} \tag{2.2}$$

O processo incia com um único segmento formado por todas as sentenças do documento e o divide recursivamente em m segmentos. Cada passo divide um dos segmentos em B no ponto (i, j) que maximiza D (Equação 2.2). O processo se repte até atingir o

número de segmentos desejados ou um limiar de similaridade.

O *C99*, assim como outros métodos baseados em divisão, tem sua performance fortemente influenciado pela quantidade segmentos desejados. Esses métodos alcançam melhores resultados em contextos onde a quantidade de segmentos é conhecida (**????????**). Na Figura 5 é mostrado a variação de *WindowDiff* em função da quantidade de segmentos solicitada ao algoritmo. Observa-se que há um ponto ótimo próximo 0,45 indicando que para o conjunto de atas analisado o método dá melhores resultados com número de segmentos desejados próximo a 45% do número de sentenças.

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 dar "pistas"de inicio ou final de segmento. A fim de aproveitar esses indicadores, usa-se um framework bayesiano que permite incorporar fontes externas ao modelo. O método BayesSeg (**??**) 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) (**??**), 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.

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".

Ao final do processo de segmentação, são produzidos fragmentos de documentos, aqui chamados de subdocumentos. Esses subdocumentos contém um texto, assim como no documento original, em um estágio de processamento inicial, pois ainda não estão estruturados. Ocorre que as técnicas de aprendizado de máquina exigem uma representação estruturada dos textos conforme será visto na Seção **??**.


conteudo/capitulos/figs/curva-similaridade.pdf

Figura 3 – *DotPlot* da similaridade entre sentenças onde as linha verticais representam segmentos reais (**??**).

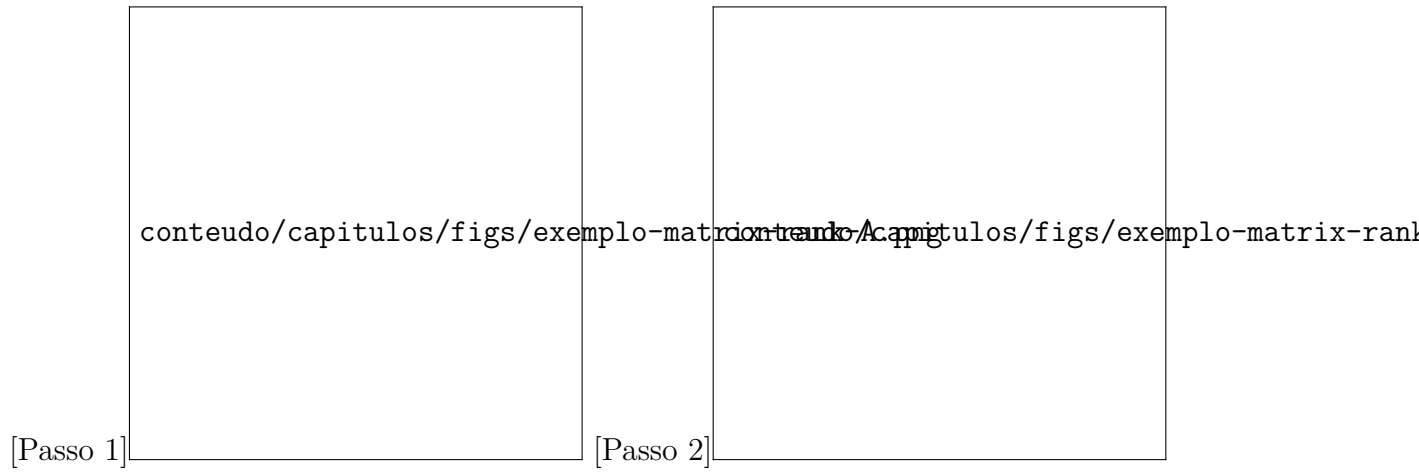

Figura 4 – Exemplo de construção de uma matriz de rankings.

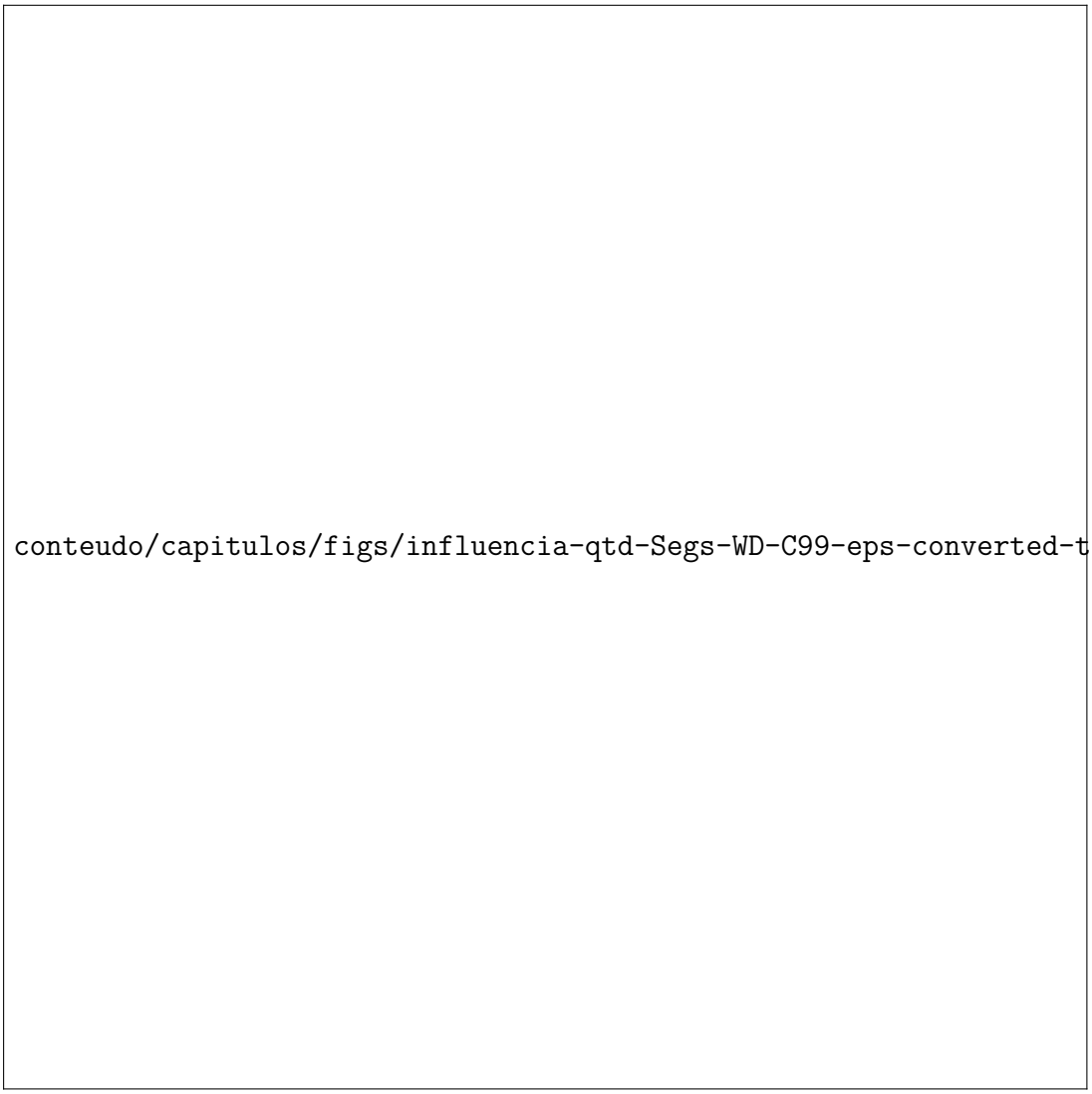

Figura 5 – Influência da quantidade de segmentos em *WindowDiff*

3 Sistema Proposto

Referências

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 na página 7.

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 na página 7.

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 na página 7.

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 7.

LEE, D.; SEUNG, H. S. Learning the parts of objects by non-negative matrix factorization. v. 401, p. 788–91, 11 1999. Citado na página 7.