Projeto 3 - Processamento de Linguagem Natural

Introdução:

A sumarização de um texto é o mesmo que um resumo, um texto que sintetiza o assunto de outro, buscando captar seus pontos principais e comunicar, com um nível menor de detalhes o que é tratado no texto original.
Existem dois tipos principais de sumarização, a extrativa e a abstrata. A extrativa se resume em extrair partes do texto consideradas relevantes para o entendimento geral e uni-las de forma a gerarem um novo texto. Já a abstrata demanda de um conhecimento de PNL mais aguçado, pois ela gera um novo texto a partir de um entendimento mais profundo do conteúdo tratado. É possível inclusive que partes do resumo nem aparecem no texto original, dado que houve uma abstração do assunto, a partir da qual foi gerado novo conteúdo. A sumarização utilizada para a realização de ambos os algoritmos (descritos abaixo) foi a extrativa.
Para a realização deste projeto foram utilizadas principalmente as seguintes páginas web:

Introduction to Text Summarization using the TextRank Algorithm

A Sumarização Automática de Textos: Principais Características e Metodologias*

gensim: models.doc2vec – Doc2vec paragraph embeddings

1. Clustering:

É uma técnica para agrupar dados, sendo que os grupos são chamados de clusters, e este agrupamento é chamado de aprendizado não supervisionado. O algoritmo utilizado neste projeto foi o K-Means, um dos mais comuns e de mais fácil entendimento.
O objetivo da clusterização neste projeto foi separar o documento em diferentes assuntos, pegando as sentenças mais ao centro do cluster (uma sentença para cada cluster), ou seja as que mais sintetizam cada um dos assuntos tratados no texto e se as juntarmos, temos um texto que resume o original com tantas frases quanto forem o número de clusters.

Métodos:

O processo se resume em duas etapas, sendo a primeira bastante simples e a segunda um pouco mais complexa, com mais passos.

A primeira coisa é treinar o vetorizador de palavras, pois para que seja possível sumarizar um texto, a informação qualitativa das palavras precisa ser quantificada e vetorizada para depois poder ser correlacionada.

Existem várias opções de métodos e algoritmos utilizados para a vetorizar as sentenças de um dado documento, como: TF-IDF, CBOW, Doc2Vec (usado neste relatório) e LDA para citar alguns exemplos. Estes algoritmos transformam as palavras em vetores numéricos, permitindo o uso de operações matemáticas para estudar as frases, assim possibilitando o uso de um computador para automatizar a sumarização, esta etapa é essencial no processo de automatizar a sumarização de textos.

Para que fosse possível automatizar este processo, foi necessário escolher um corpus textual para ser trabalhado, no caso foi escolhido o corpus reuters da biblioteca nltk da linguagem python. Assim, foi possível trabalhar com este para desenvolver um processo de automatização de sumarização dos documentos contidos nele - foi importante para o processo de exploração especificar um corpus porque caso contrário os textos estudados poderiam variar muito de assunto, léxico, estilo entre outras variáveis.

O corpus de notícias foi selecionado porque notícias são em geral bastante regulares no seu formato, variando principalmente no tópico abordado - que no caso é exatamente o que foi explorado neste relatório. Mas, como não é suficiente escolher o corpus, foi necessário também separar as palavras, ou tokens como serão chamadas daqui adiante.

Uma vez que o modelo foi escolhido para o caso de uso e devidamente implementado, como na biblioteca gensim que foi usada para produzir este relatório, é necessário treinar o vetorizador para o corpus estudado. O vetorizador precisa treinar com todas as sentenças do corpus para ‘ser contextualizado’, personificando-o para fins didáticos, no tipo de texto que ele vai sumarizar. Isso torna possível que os assuntos no texto sejam melhor compreendidos pelo vetorizador, nessa etapa ele se torna um vetorizador especificamente de notícias. Finalizando essa etapa, é salvo o vetorizador treinado.

A segunda etapa utiliza o vetorizador treinado na primeira etapa para finalmente sumarizar um dado documento. Agora treinado, o vetorizador é capaz de quantificar sentenças de notícias do reuters com uma certa confiabilidade e até outros documentos, contanto que eles não sejam muito discrepantes do corpus de treino.

Agora, é selecionado um documento para ser sumarizado - neste caso foi escolhido um artigo que fala sobre o comércio entre os Estados Unidos e a Ásia. O artigo é destrinchado em uma lista contendo todas as suas sentenças, que são em si destrinchadas em todos os tokens que as compõem. O vetorizador, então, é utilizado na etapa 2 para vetorizar todas as sentenças do documento, tornando o documento em dados.

Tendo vetorizado todas as sentenças do documento, é usado um simples algoritmo de clusterização chamado k means, que usa segmentos de retas equidistantes a centros de clusters que são definidos pela densidade da distribuição de informação. Como agora as sentenças são vetores, todos com 200 dimensões, é possível definir alguns clusters de vetores para serem usados na sumarização - estes clusters servem para indicar quais sentenças estão relacionadas a quais, e também quais sentenças melhor se aproximam à média de um dado cluster.

Trabalhando com a suposição que a sentença ‘média’ é a que melhor sumariza um cluster de sentenças, são escolhidas as 5 mais próximas dos 6 clusters, assim descartando o cluster que não seja tão relacionado com o resto do documento; possivelmente comentários do repórter, como anedotas e coisas do tipo. Havendo escolhido as 5 sentenças mais ‘macias’, elas são então convertidas de volta para linguagem natural, organizadas na ordem em que aparecem no documento original e concatenadas, resultando em um parágrafo ‘médio’ do documento, ou seja, sua sumarização.

Resultados:

Exemplo da sentenca de um documento selecionado aleatoriamente com os tokens (palavras) separados, que foi usado para treinar o modelo Doc2Vec:

```
['ASIAN', 'EXPORTERS', 'FEAR', 'DAMAGE', 'FROM', 'U', '.', 'S', '.-',
'JAPAN', 'RIFT', 'Mounting', 'trade', 'friction', 'between', 'the', 'U', '.', 'S', '.',
'And', 'Japan', 'has', 'raised', 'fears', 'among', 'many', 'of', 'Asia', '"', 's',
'exporting', 'nations', 'that', 'the', 'row', 'could', 'inflict', 'far', '-', 'reaching',
'economic', 'damage', ',', 'businessmen', 'and', 'officials', 'said', '.']
```

Vetorização da sentença acima, com 200 dimensões (integralmente disponíveis no jupyter que acompanha este relatório):

```
[-5.42353955e-04  4.12977226e-02  4.10420373e-02 -5.72566539e-02
  1.46874199e-02 -2.35935976e-03  1.89877935e-02 -5.65520488e-02
  4.29281127e-03  4.94865440e-02 ... 7.29559362e-02 -2.83149239e-02
  6.58344617e-03 -2.70591527e-02  5.65542541e-02  4.20770124e-02]
```

Distância de algumas sentenças de exemplo a cada um dos 6 cluster, respectivamente (a sentença acima é a primeira):

```
[[4.18972918e-01 9.43791263e-01 1.16219615e+00 6.08483492e-01
  7.53776051e-01 1.03997684e+00]
 [9.09257071e-01 6.76677163e-01 1.62850313e+00 7.91070367e-01
  1.26591256e+00 1.04170832e+00]
 [6.45877696e-01 6.31479938e-01 1.31975458e+00 5.00145022e-01
  1.02801770e+00 9.83478731e-01]
 [1.02337094e+00 1.44240993e+00 3.65002415e-08 1.14167648e+00
  1.28812172e+00 1.37766441e+00]
 [8.37920020e-01 3.73051431e-01 1.42816536e+00 5.72514079e-01
  1.13763354e+00 8.04945473e-01]]
```

Qual 'categoria' se encaixa cada sentença de exemplo acima, qual o índice do cluster mais próximo da sentença respectiva:

```
[0 1 3 2 1]
```

Com o índice do cluster mais próximo de cada sentença exemplo, é possível relacionar a distância desta sentença ao seu cluster mais próximo, assim resultando na lista de distâncias das sentenças a seus devidos clusters, a seguir:

[0.41897291775203066, 0.6766771630009544, 0.5001450221680627, 3.650024149988857e-08, 0.37305143097197735]

Foram escolhidas as 5 sentenças com a menor distância a um cluster, assim assegurando que estas são as que melhor resumem o documento dado este modelo. Mesmo que duas sentenças pertençam a assuntos, ou clusters, diferentes, se pegarmos as 5 com menor distância, temos as 5 que melhor se relacionam a 5 dos 6 clusters do modelo; descartando um dos clusters, que nenhuma sentença chegou tão perto.

[0, 2, 10, 11, 12]

Com as sentenças mais próximas dos clusters listadas, faltou só ordená-las na ordem em que aparecem no documento original e assim é possível resumir o texto original grosseiramente nas suas sentenças mais importantes, como vemos no parágrafo a seguir:

ASIAN EXPORTERS FEAR DAMAGE FROM U . S .- JAPAN RIFT
Mounting trade friction between the U.S. And Japan has raised fears among many of Asia's exporting nations that the row could inflict far - reaching economic damage, businessmen and officials said. But some exporters said that while the conflict would hurt them in the long-run, in the short-term Tokyo's loss might be their gain. Taiwan had a trade trade surplus of 15.6 billion USD last year, 95% of it with the U.S. The surplus helped swell Taiwan's foreign exchange reserves to 53 billion USD, among the world's largest . " We must quickly open our markets, remove trade barriers and cut import tariffs to allow imports of U.S. Products, if we want to defuse problems from possible U.S.

Conclusão:

O processo de sumarizar textos com certeza é um dos grandes desafios da humanidade. Como um gari que limpa a rua para que os carros não fiquem presos no lixo é necessario tambem que nós busquemos livrarmo nos de toda a informação que congestiona nosso juízo e capacidade de racionalizar situacoes, processo árduo de ser feito manualmente e que é bastante ajudado pela automatização explorada neste relatório.

Apesar do resumo ter sido bastante convincente, é claro que existe ainda bastante progresso a ser feito na área de processamento de linguagem natural. No caso deste estudo, o que prejudicou majoritariamente foi o fato do algoritmo não ter treinado tanto e de nós não termos acesso a máquinas mais potentes, que possibilitaram treinar um modelo mais parrudo.