Insper

Projeto: sumarizador de documentos e extração de palavras-chave.

Arthur Rizzo e Vitor Liu

14/Maio/2020

Resumo

O objetivo deste relatório é explicar as etapas envolvidas para a realização do projeto de sumarização e extração de palavras-chave de documentos. Foram propostos dois métodos de sumarização extrativa: Clustering e TextRank. Ambos os algoritmos tiveram como dataset os arquivos já "tokenizados" do conjunto "CNN daily mail". Além disso, serão abordados alguns conceitos discutidos em aula como "word embeddings", distância cosseno, similaridade, entre outros.

1. Introdução

Na presente seção será apresentada de forma geral o que é a sumarização automática de texto e, em seguida, uma breve explicação de cada algoritmo utilizado neste projeto. Na seção 2 são descritos os métodos utilizados. A seção 3 é destinada a um breve relato dos resultados do projeto. A seção 4 contempla as conclusões do projeto e, por fim, temos as referências essenciais ao projeto.

a. Sumarização automática de texto

A sumarização de texto é a tarefa de transformar um texto ou conjunto de textos em uma versão condensada. O objetivo desta tarefa é obter um subtexto de tamanho reduzido em comparação ao documento original sem que se perca a essência do conteúdo. Esta atividade é trabalhosa e custosa em termos de tempo e, portanto, há um grande interesse em automatizar este processo. Decorrente disto, muitos estudos sobre técnicas de processamento de linguagem natural são desenvolvidos para encontrar soluções que possam suprir esta demanda.

A sumarização automática de texto pode ser dividida em duas categorias: sumarização extrativa e sumarização abstrativa. Na sequência está resumido o conteúdo das referências [1][2] a respeito destas duas abordagens.

A extrativa, como o próprio nome sugere, seleciona frases do próprio documento objeto da sumarização. A seleção é feita por um algoritmo que utiliza alguma métrica que determina a relevância da frase no texto. Em seguida um conjunto de frases extraídas são agrupadas para compor um subtexto coerente que condense o conteúdo essencial em poucas linhas.

Já a abstrativa, visa produzir um sumário procurando interpretar o documento objeto através de técnicas avançadas de processamento de linguagem natural. De forma que, palavras e frases contidas no subtexto gerado não estão necessariamente contidas no texto original. Esta abordagem se assemelha mais com a forma humana de se elaborar resumos e é

naturalmente mais trabalhosa. Em decorrência disto, é comum a utilização de redes neurais complexas como “deep learning” na tentativa de criar um texto novo que sintetize o conteúdo presente em um ou mais documentos.

b. Algoritmo Clustering

O algoritmo Clustering consiste em agrupar objetos baseado na similaridade entre os objetos. O Clustering em si não é um algoritmo, mas utiliza algum algoritmo para fazer a clusterização.

O Clustering para documentos é famoso por ser usado em motores de busca na internet. Ele facilita a recuperação de documentos possivelmente satisfatórios baseados em em tópicos relevantes. Para isso, os algoritmos mais utilizados para a clusterização de documentos são o algoritmo baseado em hierarquia e o algoritmo k-means.

Para utilizar qualquer algoritmo na clusterização, é necessário primeiramente representar o documento como vetores. Assim, com a aplicação de mais algoritmos, como TF-IDF, Word2Vec/Doc2Vec, Latent Dirichlet Allocation, transforma-se documentos/sentenças/palavras em vetores, chamados de embeddings. Com vários vetores, é possível agrupá-los em clusters baseado nas distâncias entre vetores com os algoritmos de clusterização.

Um modelo de clusterização é capaz de, ao receber um novo vetor, indicar a qual cluster ele faz parte / a qual cluster está mais próximo. Também, é possível obter os documentos/sentenças que mais representam cada cluster.

c. Algoritmo TextRank

O algoritmo TextRank (TR) [1] é baseado no algoritmo PageRank (PR). Por isso, o PR será brevemente discutido com a finalidade de melhorar o entendimento do algoritmo TR.

Criado com o objetivo de ranquear páginas web, o PR, atribui a cada página um peso individual com relação às outras páginas de um conjunto. Esse peso representa uma probabilidade de uma pessoa visitar a página respectiva com base na interligação entre cada página dentro de um conjunto de páginas web e a quantidade n de páginas presentes no conjunto. Com a probabilidade relativa entre as n páginas cria-se uma matriz de probabilidades que, por sua vez, é utilizada no PR para se calcular o peso específico de cada página e assim obter uma ordem de relevância entre as páginas.

A essência do TR é similar à do PR, o escopo é ranquear as páginas com uma matriz de similaridade, no lugar de uma matriz de probabilidade. Além disso, é válido notar que o TR é um método de sumarização extrativa não supervisionado e pode ser implementado para sumarizar ou extrair palavras-chave de um texto. Portanto, ele deve resumir um documento com base na relevância (peso) de cada frase ou palavra presente no texto, ou conjunto de textos, e produzir um subtexto composto pelas sentenças de maior peso ordenadas de forma que possua alguma coerência. Sendo o peso determinado pelo algoritmo PR e é calculado através da matriz de similaridade entre sentenças ou palavras.

Para compor a matriz calcula-se a distância cosseno de cada vetor de sentença ou palavra com relação às outras sentenças ou palavras presentes no documento objeto. Já a vetorização das palavras e sentenças, também conhecida como “embeddings”, pode ser determinada de diversas formas como, TF-IDF, LDA, Clustering, word2vec, GloVe, entre outras. No presente projeto de TR, utilizou-se o GloVe, uma ferramenta de “word embeddings” pré-determinados. É importante ressaltar que é uma boa prática pré-processar o dataset de forma a remover ruídos para se construir uma vetorização enxuta e coesa.

Por fim, ocorre a extração do sumário, ou seja, seleciona-se as n sentenças ou palavras ditas mais relevantes de acordo com o método e as fornece na ordem de aparição no texto original.

2. Métodos

Nesta seção será apresentada uma visão geral de como o método de sumarização extrativa foi utilizado e em seguida uma breve descrição específica dos métodos Clustering e TextRank.

a. Visão geral do experimento

O presente experimento concentrou-se na manipulação e processamento do dataset fornecido “CNN daily mail” que já está tokenizado mas ainda assim passou por algum pré-processamento, removendo palavras irrelevantes (stopwords) e padronizando um pouco mais as palavras e os documentos. Após o devido tratamento do dataset, então, aplicou-se algum método de vetorização para as sentenças contidas nos textos do dataset. Com os “embeddings” prontos, utilizou-se algum método para determinar a relevância de cada sentença para o eventual ranqueamento e seleção. Selecionados e agrupados as n sentenças de maior importância de acordo com cada método, formou-se o sumário apresentando-as preservando a ordem original do documento de onde foram retirados.

b. Experimento Clustering

Neste método, a vetorização foi feita com o algoritmo Doc2Vec, da biblioteca gensim do Python. Então, utilizou-se o algoritmo de clusterização k-means, usando a biblioteca sklearn. Como a clusterização foi sobre sentenças de documentos, foi definido 2 clusters por documento. Assim, bastou extrair as sentenças mais próximas do centro de cada cluster para que estas sejam o sumário do documento em questão.

c. Experimento TextRank

Neste método, a vetorização foi realizada com o auxílio da ferramenta GloVe. Os vetores resultantes foram fornecidos para a composição da matriz similaridade através do

cálculo da distância cosseno entre os elementos. Na sequência utilizou-se o algoritmo PageRank para determinar o peso de cada elemento e assim, poder ranquear e selecionar os ditos elementos mais relevantes do documento. Para fornecer os elementos selecionados na ordem original utilizou-se uma função que comparasse os elementos do texto com os elementos selecionados de forma que o primeiro elemento sempre fosse escolhido antes. Por fim, o desempenho foi medido por inspeção visual com o sumário oficial fornecido no próprio dataset.

3. Resultados

A presente seção tem o objetivo de relatar como cada método performou.

a. Comparação qualitativa

i. Clustering

A performance da implementação do Clustering neste projeto não foi tão satisfatória. Em poucos casos, as sentenças obtidas como sumário representavam a ideia central do documento; a maioria indicava um pouco alguma ideia do documento, mas não a central; e; em uma quantidade considerável de casos, eram sentenças completamente aleatórias.

ii. TextRank

A performance da implementação do TextRank neste projeto foi minimamente satisfatória. Sendo que para sua avaliação comparou-se os resultados do sumarizador que continham 10 frases com o sumário oficial para alguns documentos. Foram testados 4 textos, dos quais, o primeiro obteve um sumário de contendo 3 das 4 sentenças do sumário oficial. O segundo obteve um sumário contendo 3 das 4 frases oficiais. O terceiro obteve um sumário contendo apenas 1 das 4 sentenças oficiais. O quarto obteve um sumário contendo 2 das 4 frases do sumário oficial.

Quanto à performance das palavras-chaves obtidas, analisou-se as frases do sumário oficial e buscou-se as palavras-chaves visualmente para comparar com o algoritmo TR. Neste, observou-se um resultado bem ruim. Apenas os resultados dos textos 1 e 2 possuíam palavras que pareciam ter mais relevância nos documentos originais.

4. Conclusão

Ambas técnicas de sumarização implementadas não foram tão satisfatórias. Isso mostra que, por envolver várias etapas, desde o pré-processamento até a extração de sentenças para o sumário, e outras diversas variáveis, é um grande desafio realizar uma boa implementação destas técnicas. Porém, isso não descarta as suas utilizações pois foi possível notar que em alguns casos a sumarização foi correta. Portanto, é necessário realizar vários outros testes para aperfeiçoar as técnicas aplicadas.

Para projetos futuros, é interessante considerar também outros métodos de vetorização/clusterização e outros algoritmos.

5. Referências

[1] https://www.analyticsvidhya.com/blog/2018/11/introduction-text-summarization-textrank-python/

[2] https://medium.com/luisfredgs/automatic-text-summarization-with-machine-learning-an-overview-68ded5717a25