Samuel Naassom do Nascimento Porto

SUMARIZAÇÃO AUTOMÁTICA DE DOCUMENTOS JURÍDICOS DA LÍNGUA PORTUGUESA APLICANDO MODELOS DE LINGUAGEM NATURAL BASEADOS EM APRENDIZADO PROFUNDO

SÃO PAULO
2020

Samuel Naassom do Nascimento Porto

SUMARIZAÇÃO AUTOMÁTICA DE DOCUMENTOS JURÍDICOS DA LÍNGUA PORTUGUESA APLICANDO MODELOS DE LINGUAGEM NATURAL BASEADOS EM APRENDIZADO PROFUNDO

Iniciação Científica apresentado ao programa de Engenharia de Computação do Insper.

Orientador: Fábio José Ayres

SÃO PAULO
2020

RESUMO

Atualmente há gigantescas quantidades de informações presentes no mundo digital, à vista disso é imprescindível conseguir selecionar e utilizar os conteúdos relevantes nesse mar de informações. A sumarização automática consiste no emprego de técnicas a fim de obter um resumo coeso do texto original em poucas linhas, preservando seu contexto e com interferência humana mínima. Destarte, é possível observar que essa técnica é um grande objeto de estudo de *natural language processing* (NLP) e se mostra de grande utilidade quando aplicada em textos acadêmicos, pesquisa de informações na Web, em artigos de revistas e jornais, entre outros.

Neste trabalho é empregado recentes avanços das arquiteturas de *machine learning* (ML) na construção de sumários extrativos de documentos individuais da língua portuguesa, por meio da clusterização de *sentence embeddings* advindos de um modelo de *deep learning.* O modelo BERT foi treinado para compreender a língua portuguesa, posteriormente as *sentence embeddings* oriundas do processo de tokenização do BERT são identificadas e classificadas pelo K-Means, permitindo assim a seleção das sentenças mais próximas dos centroides para a formação de sumários dinâmicos, sendo avaliados em termos de desempenho por meio de comparações com métodos já estabelecidos de sumarização automática extrativa de textos que usam TextRank e TF-IDF.

O objetivo desse trabalho é aplicar técnicas do estado da arte de NLP em textos da língua portuguesa em contrapartida ao pouco número desse tipo de pesquisas realizadas no idioma. Após o levantamento de uma pipeline eficaz e escalável, a próxima etapa da pesquisa consiste em treinar esse modelo para compreender documentos jurídicos complexos da língua portuguesa e estruturar uma ferramenta desenvolvida em Python capaz de sumarizar de maneira coesa e coerente atas de reunião do Supremo Tribunal Federal (STF) e classificá-las por tópicos em um dashboard, possibilitando análises inovadoras.

SUMÁRIO

INTRODUÇÃO

Atualmente há gigantescas quantidades de informações presentes no mundo digital e esse número cresce de maneira acelerada, conforme a Visual Networking Index a Web está na era do zettabyte (Thomas Barnett Jr, 2020), em que 1 zettabyte corresponde a 36 mil anos de reprodução de vídeos de alta qualidade. Em 2022 a estimativa para o tráfico global da Web é de 4.8 zettabytes por ano, um crescimento de quase 11 vezes a quantidade de informação gerada em 2012 que foi cerca de 437 exabytes. Consequentemente é imprescindível conseguir selecionar e utilizar os conteúdos relevantes, de forma hábil, nesse mar de informações.

Assim sendo, a sumarização automática consiste numa ferramenta de grande utilidade em que os sumários podem ser considerados como autocontidos ou indexadores. Quando autocontidos a informação presente neles é vista como suficiente para o leitor, não havendo necessidade de recorrer ao texto de origem. E no caso de serem indexadores, se a informação cativar o leitor ele pode se dirigir ao texto original. Providenciando assim, análises e tomadas de decisões ágeis quando eles são aplicados em textos acadêmicos, pesquisas de informações na Web, artigos de revistas e jornais, e assim por diante.

Há duas maneiras usuais de abordarem sumarização automática de texto: sumarização abstrata e sumarização extrativa. A sumarização abstrata consiste em interpretar o texto e fazer inferências sobre seu conteúdo numa versão resumida, com palavras possivelmente diferentes das presentes no texto original e resultados similares a trabalhos humanos de sumarização. Contudo, a construção de sumários abstratos é desafiadora, pois, essa abordagem leva em consideração análises morfológicas, sintáxicas, semânticas e pragmáticas do texto; transformando a geração dos sumários numa atividade de alta complexidade ao ensinar a máquina a maneira de fazer inferências de um humano.

A sumarização extrativa consiste na justaposição e reordenação de sentenças coletadas diretamente do texto original, apoiada numa função score que tem como objetivo selecionar as sentenças de forma que haja manutenção do contexto, coesão e coerência. Porém, quando é selecionada uma sentença que contém pronomes que fazem referências a sentenças anteriores há uma dificuldade em garantir essas qualidades do sumário gerado. Mesmo assim a sumarização

extrativa se mostra vantajosa e de maior utilização, pois, ela é mais robusta, possui baixa complexidade e seu custo em relação a sumarização abstrata é significativamente menor.

Tendo isso em mente, neste projeto utilizaremos sumarização extrativa de texto. Todo o código e resultados podem ser encontrados no repositório a seguir: https://github.com/PortoSamuel/IC-NLP.git.

TRABALHOS RELACIONADOS

A melhor abordagem para a construção de um programa de sumarização extrativa consiste em 4 etapas segundo Kulkarni e Apte (Kulkarni, 2013): Pré-processamento, extração de *features*, seleção de sentenças e reordenação, por fim, formação do sumário. Com o intuito de otimizar a qualidade dos sumários gerados, diferentes técnicas de computação ou de inteligência artificial podem ser empregadas nas etapas da construção do sumário extrativo. Tal como o uso do algoritmo não supervisionado TextRank na formação do sumário ou o uso do TF-IDF (Term Frequency-Inverse Document Frequency) na extração das *features*, como será discutido a seguir.

O TextRank é uma ferramenta comumente usada para sumarização automática extrativa de texto baseada em grafos (Barrios, 2016), o algoritmo não supervisionado aplica uma variação do PageRank (PAGE, 1999) sobre um grafo feito especificamente para a sumarização usando as sentenças como nós e para definir suas arestas, uma função que computa a similaridade baseada nos conteúdos léxicos que compartilham em comum.

A função que computa similaridade entre sentenças serve para definir os pesos das arestas entres as sentenças, quanto maior a similaridade entre sentenças maior o peso de suas arestas. Dessa forma as sentenças são ranqueadas e as mais representativas são selecionadas para formarem o sumário. Entretanto, essa forma de análise possui grandes lacunas de contexto ao analisar textos orais.

Outra prática bastante utilizada na criação de sumários é a implementação do TF-IDF com a finalidade de extrair as palavras chaves e frases mais importantes a partir de suas frequências de aparições no corpus do documento (Christian, 2016),

todavia, no processo de formação do sumário final o autor utiliza algoritmos tradicionais de NLP.

Até recentemente *recurrent* neural network (RNN) serviu como aplicação padrão para diversas tarefas de NLP, utilizando imensas quantidades de dados, caros recursos computacionais e muitas horas de treinamento para obter resultados razoáveis, com baixas performances em sentenças extensas e estando sujeito a *overfit* (Vaswani, 2017).

Alicerçado nisso, Vaswani apresentou a definição da arquitetura Transformers (Vaswani, 2017) utilizando *feed forward networks* e mecanismos de *attention*, se mostrando superior às abordagens de NLP que envolviam RNN ou *convolutional neural network* (CNN) (Zhang, 2016). Mesmo aliviando alguns dos problemas presentes com RNN e CNN a arquitetura ainda possuía uma performance inferior à humana em diversos tópicos de NLP.

Em 2018 pesquisadores do Google foram inovadores ao definirem um modelo não supervisionado de aprendizado chamado BERT (Bidirectional Encoder Representations from Transformers) (Devlin, 2018) que possui resultados superiores a quase todos os outros modelos existentes. Os pesquisadores também publicaram uma série de modelos pré-treinados que podem ser utilizados para solução de problemas em diferentes aplicações de NLP, inclusive sumarização de texto.

METODOLOGIA

Para fins didáticos as etapas iniciais de pré-processamento e extração de *features* foram abordadas como um único passo, levando a construção do modelo de sumarização extrativa em 3 etapas:

- Encontrar uma representação para as palavras mais representativas do texto original.
- Definir pontuação para as sequências com base nas palavras mais representativas.
- Produzir um sumário com base nas melhores K sentenças.

O texto original é tokenizado em sentenças limpas que são passadas ao BERT, produzindo como outputs *sentence embeddings* que podem ser agrupadas

em clusters de diferentes tamanhos, permitindo assim a criação de sumários de tamanhos dinâmicos ao escolher as n sentenças mais próximas aos centros dos clusters. A escolha das principais sentenças está sendo realizada com K-means, mas, poderia ser feita com outros modelos não supervisionados de clusterização como *Gaussian Mixture Models*.

- Text tokenization:

O texto é submetido a técnicas de tokenização que consiste num tratamento inicial para deixá-lo mais preparado para os modelos de *embeddings*, exemplo de aplicação são remoção de *stopwords* e outras palavras que não contribuem para o sentido geral, as ferramentas da biblioteca NLTK do python foram utilizadas para um tratamento inicial. E tanto a tokenização do texto, quanto a utilização dos modelos de *embedding* foram inspirados na biblioteca Transformers providenciada pela *huggingface* em: https://huggingface.co/transformers/index.html.

- Text Embedding (BERT):

O treinamento de modelos como BERT demoram dias para concluir, mesmo com quantidades substanciais de GPUs sendo utilizadas. Por conta disto a Google liberou para uso público modelos BERT pré-treinados com mais de 100 milhões de parâmetros, contudo, esses modelos são pré-treinados somente com palavras e textos em inglês. Para o processamento dos textos da língua portuguesa o modelo utilizado foi o portuguese-bert disponibilizado pela NeuralMind em: https://github.com/neuralmind-ai/portuguese-bert.

A implementação do modelo BERT pré-treinado utilizou a biblioteca transformers, da organização *huggingface*. O modelo produz como resultado uma matriz N x E que é o necessário para a clusterização, sendo N o número de sentenças e E a dimensão dos *embeddings* (Miller, 2019).

- Clustering Embeddings:

Dada a matriz N x E gerada, um parâmetro K pode ser utilizado como o número de clusters, dessa forma, com base num *ratio* as sentenças são solicitadas para formarem o sumário. O K-means da biblioteca Sci-kit Learn foi implementado para selecionar as sentenças mais próximas dos centróides e montá-lo ordenando as sentenças conforme suas aparições no texto original. Na figura a seguir estão representados 4 *clusters* e suas sentenças, advindas de um texto retirado da BBC como objeto de estudo (Brazil, 2020).

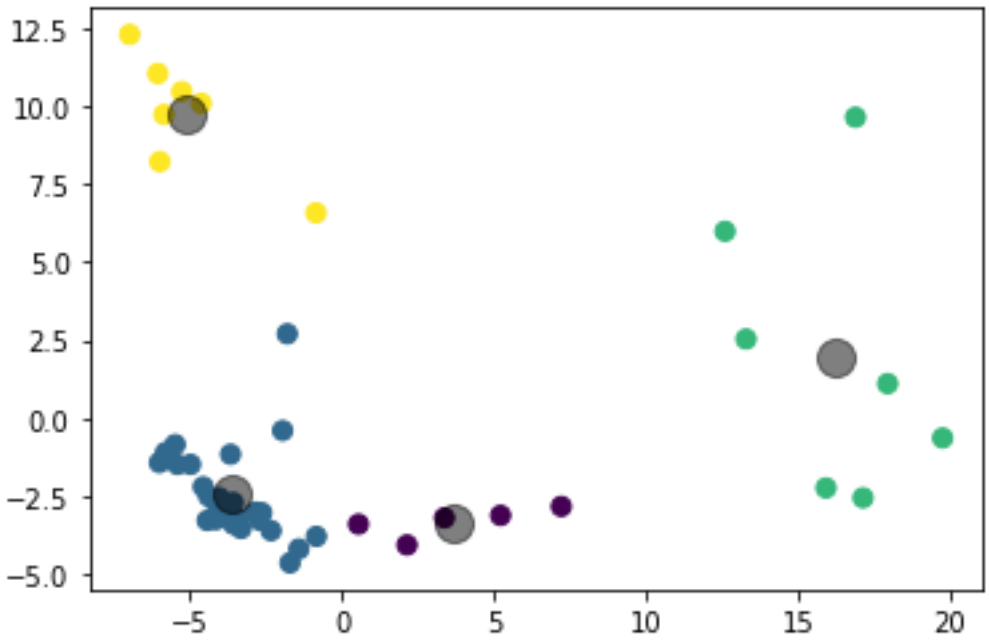

Figura 1: ***K-means classification over BERT N-2 layer embeddings***

RESULTADOS E DISCUSSÕES

Com o modelo em mão foi possível criar sumários dinâmicos e coerentes de textos de diferentes tamanhos da língua portuguesa, numa maneira muito eficiente e de qualidade superior aos sumários gerados utilizando as ferramentas citadas anteriormente. No README do repositório do projeto estão localizados os sumários extraídos do texto da BBC para comparação.

Levando em consideração que não há um sumário perfeito, a métrica de qualidade mais usada foi a observação humana, métrica bem razoável dado que não existe algo consolidado para mensurar aplicações deste tipo.

Falhas desta aplicação, comum à outras de sumarização, são: realizar pequenos sumários em grandes textos, lidar com palavras de múltiplos sentidos e transcrição da linguagem conversacional.

Em textos grandes com mais de 100 sentenças a dificuldade é criar um sumário representativo com uma pequena quantidade de sentenças. Com o incremento de mais sentenças o contexto é mantido, deixando o sumário mais claro ao leitor. Isso poderia acontecer pegando mais sentenças próximas a centroides dos clusters, incrementando a qualidade da saída, contudo, isso pode conflitar com a liberdade do usuário na escolha do tamanho do resultado. Outra melhoria poderia

acontecer na determinação do melhor número de sentenças para um texto específico.

CONCLUSÃO

A criação adequada de sumários é uma ferramenta muito útil que será usada futuramente neste projeto para classificação de grandes quantidades de textos jurídicos da língua portuguesa, de forma a produzir insights inovadores.

A próxima iteração será treinar esse modelo para compreender documentos jurídicos complexos e estruturar uma ferramenta desenvolvida em Python capaz de sumarizar de maneira coesa e coerente atas de reunião do STF e classificá-las por tópicos em um dashboard, possibilitando análises inovadoras.

REFERÊNCIAS

Barrios, F. e. (2016). Variations of the similarity function of textrank for automated summarization. *arXiv preprint arXiv:1602.03606*.

Brazil, B. |. (2020, 06 02). *Movimentos anti-Bolsonaro: o que é o 'Somos 70%' e outras iniciativas da sociedade civil contra o governo?* Retrieved from BBC | News | Brazil: https://www.bbc.com/portuguese/brasil-52898476

Christian, H. M. (2016). Single document automatic text summarization using term frequency-inverse document frequency (TF-IDF). *ComTech: Computer, Mathematics and Engineering Applications 7.4*, pp. 285-294.

Devlin, J. e. (2018). Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Kulkarni, A. R. (2013). A domain-specific automatic text summarization using Fuzzy Logic. *International Journal of Computer Engineering and Technology (IJCET) 4.4* , pp. 449-461.

Miller, D. (2019). Leveraging BERT for extractive text summarization on lectures. *arXiv preprint arXiv:1906.04165*.

PAGE, L. e. (1999). The PageRank citation ranking: Bringing order to the web. *Stanford InfoLab*.

Thomas Barnett Jr, e. a. (2020, 08 31). *Cisco Visual Networking Index (VNI)*. Retrieved from Complete Forecast Update, 2017–2022, APJC Cisco

Knowledge Network (CKN) Presentation: https://www.cisco.com/c/dam/m/en_us/network-intelligence/service-provider/digital-transformation/knowledge-network-webinars/pdfs/1213-business-services-ckn.pdf

Vaswani, A. e. (2017). Attention is all you need. *Advances in neural information processing systems*.

Wu, Y. e. (2016). Google's neural machine translation system: Bridging the gap between human and machine translation. *arXiv preprint arXiv:1609.08144*.

Zhang, Y. M. (2016). Extractive document summarization based on convolutional neural networks. *IECON 2016-42nd Annual Conference of the IEEE Industrial Electronics Society.* IEEE.