INSTITUTO FEDERAL DA PARAÍBA

MICAEL MARQUES RODRIGUES SILVA

RAYANNE KELLY MARCELINO BARROS ELIAS

SUMARIZAÇÃO AUTOMÁTICA: COMPARAÇÃO ENTRE O ALGORITMO DE LUHN E UM ALGORITMO ALTERNATIVO

(18 Páginas)

Campina Grande – PB

2019

Sumarização Automática: comparação entre o algoritmo de Luhn e um algoritmo alternativo

M. M. R. Silva[1], R. K. M. B. Elias[1]
[1]*Instituto Federal da Paraíba (IFPB), Brasil*
{micasmarques1132, rayannekmb}@gmail.com

Resumo

Este artigo compara dois algoritmos da área de mineração de textos e que são utilizados para realização da sumarização automática. Para tal, foram avaliados nos experimentos o algoritmo de Luhn e um algoritmo proposto pelos autores. A avaliação consistiu em gerar resumos e através da aplicação de um questionário para um grupo de pessoas indicar aquele que apresentasse melhor entendimento de acordo com a coesão e coerência. Após analisar os dados obtidos foi demonstrado que o algoritmo de Luhn possui melhor resultado para gerar resumos que mantenham a ideia original do texto e proporcione uma boa experiência ao leitor segundo as métricas analisadas.

Palavras-chave: Sumarização automática. Algoritmo de Luhn. Algoritmo proposto. Coesão. Coerência, Processamento de Linguagem Natural.

1. Introdução

Diante da necessidade de usar o computador para ler e editar textos, foi preciso “ensinar” o computador a identificar textos em uma linguagem natural, e codificá-los para uma linguagem computacional. Com o passar dos anos, houveram vários avanços através de pesquisas na área de Inteligência Artificial (IA), o que possibilitou, a partir de uma de suas subáreas, uma forma para ensinar o computador a identificar palavras, chamado de Processamento de Linguagem Natural, PLN.

O PLN busca soluções para questões computacionais, através de uma aprendizagem automática no processamento de linguagem e se dedica a propor e desenvolver modelos computacionais para a realização de tarefas que dependem da língua humana, escrita como objeto primário. Para isso, linguistas, cientistas da computação, buscam fundamentos em várias disciplinas: Filosofia da Linguagem, Psicologia, Lógica, Inteligência Artificial, Matemática, Ciência da Computação, Linguística Computacional e Linguística. Segundo Gariba [2005] et al [1]. o PLN busca facilitar a interação do software com o usuário, o que possibilita além do melhor entendimento, conseguir exatamente o que se está procurando.

A sumarização automática de texto [2] permite um contato acessível informacional ao leitor, sendo assim de fundamental importância. Para tal, esse artigo se dispõe a analisar técnicas de PLN, e o uso de dados estatísticos. O desafio

consiste em uma redução da dimensionalidade e condensação semântica sem gerar prejuízo ao entendimento. Através, de uma metodologia para o teor experimental que o artigo propõe, é avaliado de forma qualitativa os sumários produzidos a partir de um conjunto significativo de palavras, comparando assim dois algoritmos e mostrando as falhas de ambos. O primeiro destes, um algoritmo amplamente usado para sumarização automática de textos dentro da IA, chamado Algoritmo de Luhn [3], que será abordado e explicado mais à frente.

De acordo também com uma parte da literatura revisada [4,8,9], o processo de sumarização automática de textos adiantaria os estudos e trabalhos de diversos estudantes e pesquisadores. Uns para seus estudos, e outros para elaboração de artigos, dissertações, teses e pesquisas. Mas esse processo continuará sendo explicado ao longo deste artigo.

A pesquisa literária também mostra que o problema na sumarização automática se dá em identificar segmentos relevantes de um texto e compô-los, para produzir os sumários. Sumários, nesse caso, são os textos resumidos. O problema é importante, pois através dessa automatização, a sumarização transmitirá somente o que é importante de uma fonte textual de informação, tornando o resumo amplamente informativo, e com todas as informações que foram passadas no texto original [5,6,7]. Mas, também mostra que avaliar é um processo custoso, pois é imprescindível o uso do julgamento humano. Métodos avaliativos automáticos existem, mas não são tão satisfatórios quanto o julgamento humano, onde o mesmo julgamento humano pode ser problemático, dada a abstração desta tarefa.

Uma vez que não raros são os casos que se faz necessária a obtenção de resumos de determinados textos e o quanto pode ser um processo trabalhoso a realização dos mesmos, principalmente dependendo do tamanho, torna-se de total relevância um método que possibilite a sumarização automática. Para tal, existem alguns algoritmos de mineração de textos que proporcionam resumos automatizados, destaca-se o clássico algoritmo de Luhn. Além deste, é proposto um novo algoritmo, objetivando assim o artigo como análise do resultado dos resumos dele em relação aos do já conhecido algoritmo de Luhn.

2. Processamento de Linguagem Natural (PLN)

O Processamento de Linguagem Natural consiste no desenvolvimento de modelos computacionais para a realização de tarefas que dependem de informações expressas em alguma língua natural (*e.g.* tradução e interpretação de textos, busca de informações em documentos e interface homem-máquina) [10,12].
De acordo com Covington [11], a pesquisa em PLN está voltada, essencialmente, a três aspectos da comunicação em língua natural:

- *som:* fonologia;

- *estrutura:* morfologia e sintaxe;
- *significado:* semântica e pragmática.

A *fonologia* está relacionada ao conhecimento dos sons que compõem as palavras de uma língua. A *morfologia* reconhece as palavras em termos das unidades produtivas que a compõem (*e.g. caçou → caç + ou).* A *sintaxe* define a estrutura de uma frase, com base na forma como as palavras se relacionam nessa frase (figura 1). A *semântica* associa significado a uma estrutura sintática, em termos dos significados das palavras que a compõem (*e.g.* à estrutura da figura 1, podemos associar o significado *"um animal perseguiu/capturou outro animal"*). Finalmente, a pragmática verifica se o significado associado à uma estrutura sintática é realmente o significado mais apropriado no contexto considerado (*e.g.* no contexto predador-presa, *"perseguiu/capturou"* → *"comeu*").

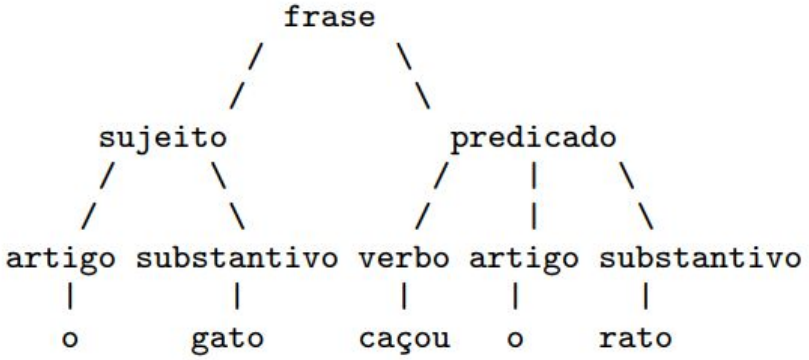

Figura 1. Árvore Sintática

Mesmo com o avanço no relacionamento homem-máquina, a comunicação via linguagem natural continua sendo um desafio: como criar programas capazes de interpretar mensagens codificadas em linguagem natural e decifrá-las para a linguagem de máquina? Com o passar dos anos, houve muitas pesquisas e desenvolvimentos nos mais diversos ramos do processamento de linguagem natural, destacando-se a tradução automática, considerada pela maioria como o marco inicial na utilização dos computadores para o estudo das línguas naturais.

Para modelar a língua e possibilitar que a máquina a entenda, são necessários pré-processamentos que abstraem e estruturam a língua, deixando apenas o que é informação relevante. Esse pré-processamento reduz o vocabulário e torna os dados menos esparsos, característica conveniente para o processamento computacional.

2.1 Técnicas de Processamento da Linguagem Natural

2.1.1 Normalização

A normalização abrange tratativas como a tokenização, transformação de letras maiúsculas para minúsculas, remoção de caracteres especiais, remoção de tags HTML/Javascript/CSS, dentre outras. O processo de tokenização tem como objetivo separar palavras ou sentenças em unidades. A tokenização lexical marca cada palavra como um *token* no texto, identificando-a mesmo se tiver encostada em alguma pontuação. Um exemplo de texto tokenizado lexicalmente seria:

Esta é uma sentença.

['esta', 'é', 'uma', sentença', '.']

A tokenização sentencial identifica e marca sentenças. Um exemplo seria:

Esta é a primeira sentença. Esta é a segunda. Esta é a terceira!

['Esta é a primeira sentença.', 'Esta é a segunda.', 'Esta é a terceira!']

A normalização é importante por começar a estruturar o texto, já que os processamentos seguintes atuam em cima de unidades sentenciais e lexicais.

2.1.2 Remoção de Stopwords

Uma das tarefas muito utilizadas no pré-processamento de textos é a remoção de *stopwords*. Esse método consiste em remover palavras muito frequentes, tais como "a", "de", "o", "da", "que", "e", "do" entre outras, pois na maioria das vezes não são informações relevantes para a construção do modelo.Esse processo só pode ser aplicado quando realmente as palavras não forem importantes para a compreensão do sentido do texto.

2.1.3 Remoção de Numerais

Outra remoção necessária é a dos numerais presentes no texto. Os numerais não agregam informação relevante por não trazerem carga semântica. O PLN remove também os símbolos que os acompanham, como "R$", "$", "US$", "km", "milhões", "bilhões", dentre outros.

2.1.4 Stemização e Lematização

O processo de stemização (do inglês, *stemming*) consiste em reduzir uma palavra ao seu radical. A palavra “meninas” se reduziria a “menin”, assim como “meninos” e “menininhos”. As palavras “gato”, “gata”, “gatos” e “gatas” reduziram-se para “gat”. A lematização reduz a palavra ao seu lema, que é a forma no masculino e singular. No caso de verbos, o lema é o infinitivo. Por exemplo, as palavras “gato”, “gata”, “gatos” e “gatas” são todas formas do mesmo lema: “gato”. Igualmente, as palavras “tiver”, “tenho”, “tinha”, “tem” são formas do mesmo lema “ter”. A vantagem de aplicar a stemização ou lematização é clara: redução de vocabulário e abstração de significado.

2.2 Compreensão da linguagem natural

Essa parte do processamento de linguagem natural é responsável por transformar sentenças de um texto em estruturas lógicas, ou seja, é compreender uma frase que carrega um valor que pode ser verdadeiro ou falso.

A compreensão da linguagem natural tem como objetivo facilitar a manipulação de texto por computadores, além de identificar instruções recebidas por humanos e até por outras máquinas. É o processo de construção de uma base semântica formal da linguagem. O significado atribuído a sentenças pode ser interpretado pelo computador, da mesma forma que os humanos o fazem.

3. Mineração de Textos

A mineração de textos é um conjunto de métodos utilizado para navegar, organizar, encontrar e descobrir informações em bases textuais [16].

A tecnologia de mineração de textos deriva das técnicas de recuperação de informações e da descoberta de informações estruturadas, por meio do uso de bancos de dados e de procedimentos estatísticos [16].É uma subárea da extração de informações, porém é utilizada somente para análise em textos.

Por mais que possa parecer similar, a mineração de textos é diferente de mecanismos de busca, uma vez que na busca o usuário já sabe o que quer encontrar; enquanto em mineração de textos o usuário descobrirá conhecimento e padrões até então por ele desconhecidos. Em suma, na busca o usuário pesquisa determinada informação e na mineração é realizada a coleta de novos conhecimentos “escondidos” nos textos.

Mineração de textos é o processo de extrair conhecimento não conhecido previamente a partir de fontes textuais, tais como correio, imprensa, transações,

websites, newsgroups, fóruns, listas de correspondência, redes sociais, dentre outros.

4. Sumarização Automática

A sumarização no contexto geral é uma atividade bastante comum. Quando se narra um evento a uma pessoa, costuma-se fazer um resumo do que aconteceu, e não uma narração completa e detalhada. Mesmo não sabendo, as pessoas estão sempre sumarizando. Muito frequentes também são os sumários escritos, como por exemplo, notícias em jornais, artigos de revistas, resumo de textos científicos, entre outros.

Computacionalmente explicando, existem duas formas de se abordar o problema da sumarização,a superficial e a profunda. Neste trabalho foi abordada a primeira, que utiliza métodos estatísticos e/ou empíricos para obter o sumário.Essa técnica é a mais simples de ser implementada e é utilizada por grande parte dos pesquisadores, porém, pode produzir sumários com problemas de coesão e principalmente de coerência, o que pode deixar o resumo sem um sentido lógico da ordem das frases, apresentando deficiências no sentido das frases. Por outro lado, a sumarização profunda realiza uma análise semântica frase a frase no texto, analisando a forma que as frases são construídas e o relacionamento de uma frase a outra.

A Figura 2 apresenta a ideia principal do funcionamento de um sumarizador automático de documentos.

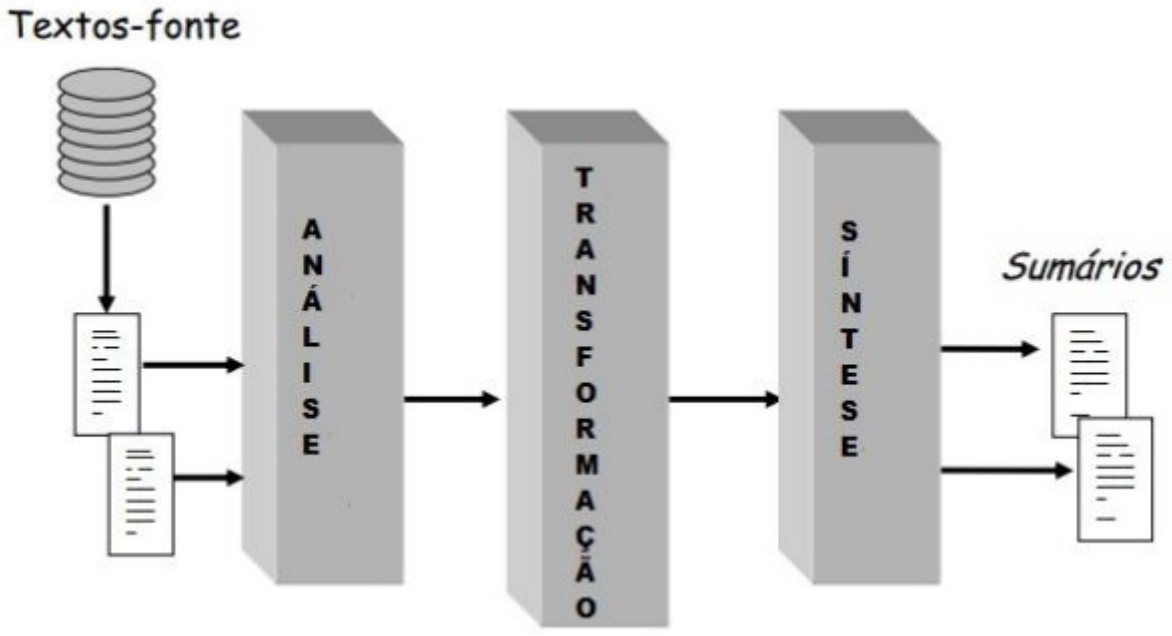

Figura 2. Arquitetura geral de um Sumarizador [13]

Na Figura 2, inicialmente pode-se perceber que o primeiro passo é quando o sistema recebe como entrada um arquivo fonte. Depois esse arquivo é passado para o módulo de análise, onde o algoritmo examina sua forma de estruturação, que é a forma em que o texto se encontra em relação a sua composição de frases. No procedimento seguinte ocorre a transformação e o cálculo da relevância das palavras chaves por meio de técnicas de extração.

Na transformação o algoritmo divide o texto em sentenças, que são as frases; então aplica seu método de extração das palavras chaves, por fim, a partir das palavras extraídas é feito o agrupamento das frases, o que irá resultar nos resumos.

Alguns algoritmos baseiam-se na geração automática de resumos, nos quais os assuntos principais de que trata um documento podem ser determinados por meio da análise dos termos que mais ocorrem no mesmo [14]. As sentenças mais importantes, que são usadas para compor o resumo do texto, são aquelas em que os termos mais frequentes aparecem em maior quantidade dentro de uma mesma frase.

No entanto, como muitos termos aparecem mais de uma vez dentro do texto, é adotado um critério de corte que consiste em considerar somente aqueles que se repetem mais que a média de repetições no documento inteiro. Caso o número de termos selecionados ainda seja muito grande para fins práticos, o valor de corte pode ser aumentado para média de repetições mais meia ou uma vez o desvio padrão (o desvio padrão aplica-se como a raiz quadrada do número de termos). Por exemplo, se o número de termos selecionados de uma frase é 09, aplicando a raiz resulta no número 3. Com isso, para todas as frases as raízes com maiores resultados serão as frases mais importantes.

5. Algoritmos de Sumarização Avaliados

Neste trabalho foram analisados e avaliados os algoritmos de Luhn e um algoritmo criado pelos escritores deste artigo, ambos sumarizadores baseados na extração de palavras chaves do texto. O algoritmo de Luhn é um dos trabalhos mais importantes na área de PLN, em Sumarização de Documentos[3], é um algoritmo clássico que serviu como base para muitos algoritmos, seu método de extração é baseado na extração de palavras chaves.

O algoritmo proposto pelos autores tem embasamento no algoritmo chamado GistSumm (GistSumarizer), que é embasado no algoritmo de Luhn, mas é mais completo, e seu método de extração de palavras chaves baseado na sentença principal do texto é muito eficiente quando trata-se de gerar resumos com as ideias principais de um texto. Segundo Pardo [17], o GistSumm atualmente encontra-se como o estado da arte de sumarização automática de documentos, com uma função para sumarização multi-documentos.

Devido a essas características, acredita-se que o algoritmo proposto é melhor que o de Luhn, pois o mesmo é baseado no GistSumm, e acredita-se que este apresente resultados melhores que o algoritmo de Luhn, pois segundo o trabalho de Oliveira [15], a eficiência de um algoritmo de sumarização está ligada ao desempenho de seu método de extração de palavras chaves. Para verificar essa hipótese, serão utilizadas tabelas com os resultados, essas tabelas mostram todos os dados tanto em forma numérica e percentual, onde podemos perceber a eficiência e demais características dos algoritmos, tais como taxa de erros e acertos. Avaliações de forma semelhante a essas foram feitas nos trabalhos de Luhn [3], e Pardo [17], sendo que somente o trabalho de Pardo aplica-se ao português do Brasil.

5.1 Algoritmo de Luhn

O algoritmo de Luhn analisa as frases mais importantes de um documento, que são aquelas que mais aparecem no texto. Neste contexto, não são consideradas as stopwords, que são palavras como artigos, preposições, conjunções, entre outras que aparecem com frequência em um texto, porém são insignificantes em relação ao significado semântico do documento. Em suma, as stopwords são utilizadas apenas para dar um sentido gramatical correto na formação das frases. Como o algoritmo faz a sumarização baseada na frequência que as palavras ocorrem, elas são desconsideradas para não confundir o sumarizador.

O algoritmo não procura compreender os dados em um nível semântico, e simplesmente computa resumos com agrupamento de palavras que ocorrem com frequência no texto [3]. A Figura 3 apresenta os passos desde quando o algoritmo de Luhn recebe um texto como parâmetro até a geração final do resumo. A primeira tarefa é identificar as frases, calculando a similaridade entre todas as frases do texto.

Na segunda abordagem outra tarefa é fazer o cálculo da pontuação, levando em conta que o resumo nunca pode ter mais pontos e vírgulas que o texto original.Terminadas a primeira e segunda abordagens mostradas na Figura 3, o algoritmo finalmente extrai as sentenças escolhidas, as agrupa e mostra como resultado o resumo.

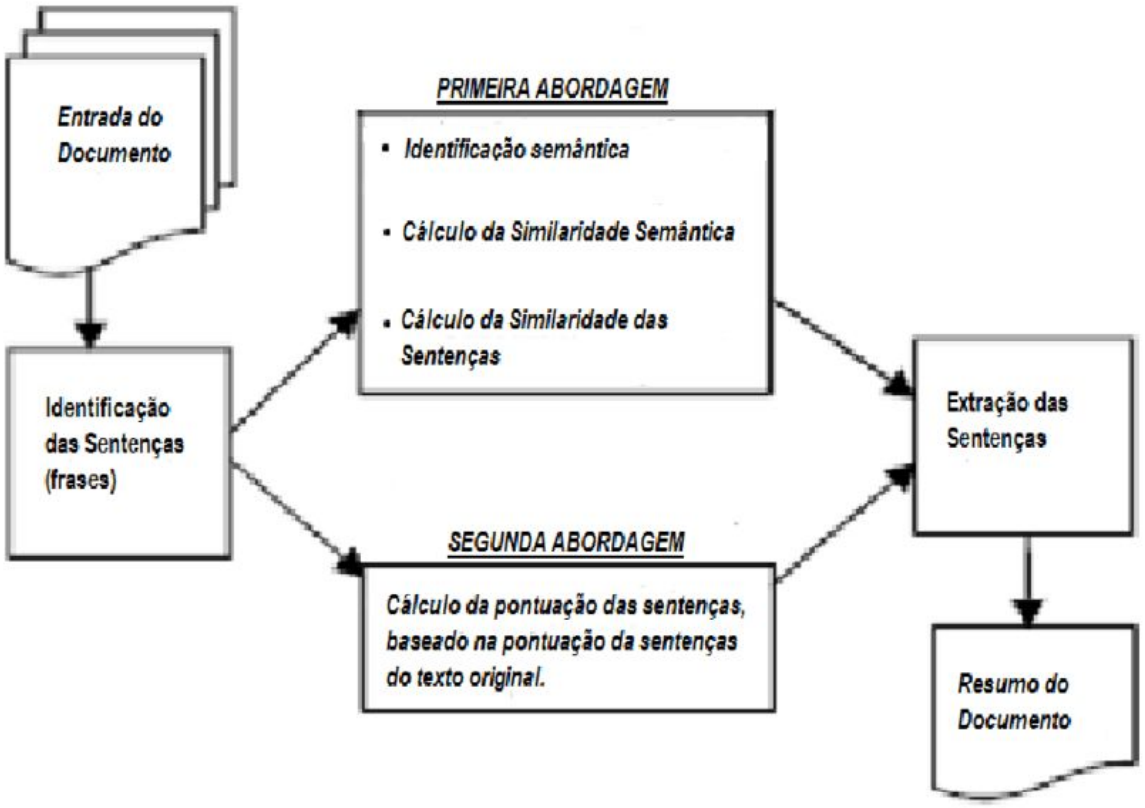

Figura 3. Modelo de Funcionamento do Algoritmo de Luhn [13]

5.2 Algoritmo Proposto

Para melhor compreensão, chamaremos o algoritmo proposto como Algoritmo de Marques, um dos escritores deste artigo. O algoritmo é baseado no algoritmo chamado GistSumm [13,16], um dos mais famosos e renomados algoritmos para sumarização automática na IA [12,17]. Este algoritmo consiste no método básico das técnicas de PLN.

5.2.1 Funcionamento do algoritmo proposto

O algoritmo é um sumarizador extrativo que usa técnicas estatísticas para determinar a ideia central dos textos por ele sumarizados, que implementa bibliotecas da linguagem Python 3.7 [20]. Baseia-se na simulação da sumarização humana, primeiro identificando a ideia principal do texto e, então, acrescenta informações adicionais ou complementares [13]. Essas informações adicionais podem ser a segunda ou terceira frase mais importante do texto, seguindo em ordem

crescente de acordo com a quantidade de frases que se deseja extrair do texto.

Dessa forma, o sumarizador primeiro procura a sentença que melhor expressa a ideia principal do texto, e baseado nela são escolhidas as demais sentenças que vão compor o extrato textual. O GistSumm trabalha da mesma forma, só que mais bem implementado, que é a seguinte: primeiro o GistSumm realiza a identificação da sentença principal com o uso de métodos estatísticos simples, e por segundo conhecendo-se as sentenças principais é possível produzir extratos coerentes [3]. Mesmo quando a sentença escolhida não for a sentença principal e há uma aproximação significativa da mesma, o extrato já pode ser gerado [19].

6.0 Instrumentação

A instrumentação dos experimentos é agrupada por hardware e software. Utilizaremos um computador (Intel Core i5-7200U Dual Core 2.5 GHz com Turbo Max até 3.1 GHz) para executar uma coleção de códigos-fonte escritos em Python 3 que implementam os algoritmos propostos de serem avaliados (Luhn e Marques).

7.0. Design de Experimento

O experimento consiste nessas três etapas apresentadas na figura 4

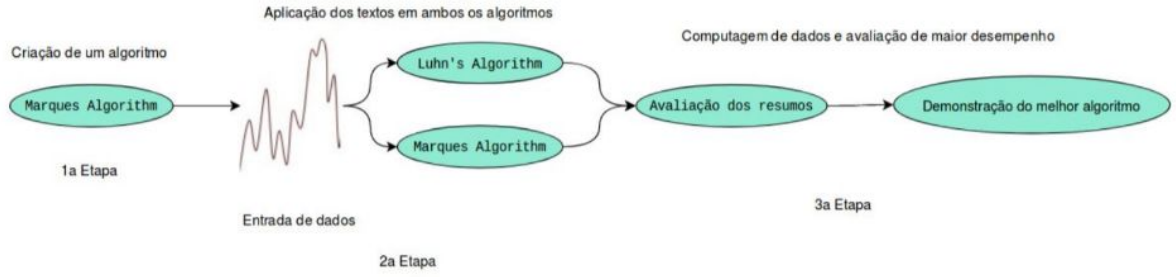

Figura 4 - Modelo Experimento

A primeira etapa (Figura 4) consiste na criação do algoritmo de Marques.

A segunda etapa (Figura 4) têm como objetivo realizar a sumarização automática de 10 artigos de diferentes áreas de estudos através de dois algoritmos que utilizam métodos distintos.

Na terceira etapa (Figura 4), o conjunto de resumos resultantes da segunda etapa serão utilizados para fins avaliativos por um grupo de avaliadores. E por conseguinte é computado a avaliação de cada pessoa e estatisticamente demonstrar

o algoritmo que realizou melhor desempenho.

O cenário a ser considerado para os experimentos é a simulação de resumos por uma máquina virtual, com capacidade de processamento e de memória RAM de 8 GB. E posteriormente a análise deles, dentro de um contexto educacional.

7.1 O Experimento da Primeira Etapa

A primeira etapa consiste na criação do algoritmo de Marques, que inicia criando uma matriz do texto, separando-o em sentenças, e dentro das sentenças, em palavras. Depois disso, ele identifica palavras que possuem apenas significado sintático dentro da sentença, mas não é relevante para o sentido do texto, como “ou”, “e”, “para”, e as retira da matriz, pois, como essas palavras são frequentes, o algoritmo acabaria dando importância para as mesmas, e isso dificultaria a análise textual. Também retira as pontuações do texto, pois o algoritmo as trata como sendo uma palavra, e isso também atrapalha a sumarização.

Após essa limpeza de texto, o algoritmo cria uma distribuição de frequência para a matriz de palavras, para que seja descoberta quais são as mais importantes.

A partir dessa distribuição de frequência, o algoritmo seleciona, a partir de um valor inserido, o quão comum a palavra se apresenta no texto geral para posteriormente classificá-las como significantes ou não, no texto final.

7.2 O Experimento da Segunda Etapa

A segunda etapa consiste em selecionar 10 artigos no google acadêmico, que incorporam diversas áreas do conhecimento e com complexidades diferentes, para realização de sumarização automática através do algoritmo de Luhn e o algoritmo de Marques.

Foram realizados testes nas configurações dos parâmetros comuns modificáveis nos algoritmos, isto é, na quantidade de sentenças importantes, a fim de equipará-los para a etapa seguinte. Dessa forma, neste experimento foi utilizado o parâmetro de 8 sentenças importantes.

Segundo Pardo [18], para gerar um resumo eficiente deve ser extraído entre 20 a 50% do texto e 80% na taxa de compressão.

7.3 O Experimento da Terceira Etapa

Com o intuito de avaliar a qualidade e coesão dos resumos gerados na segunda etapa, serão escolhidas 30 pessoas com diferentes graus de instrução acadêmica e será distribuído os dois resumos resultantes da inserção de um artigo

em ambos os algoritmos para que um grupo de 3 pessoas que não possuem relação, escolhidas através de critérios como: conhecimento e preferência com o tema do texto, possam indicar o resumo que julgou possuir melhor coesão e coerência.

É de fundamental importância destacar que foi enviado para cada pessoa um formulário no google docs do seguinte formato

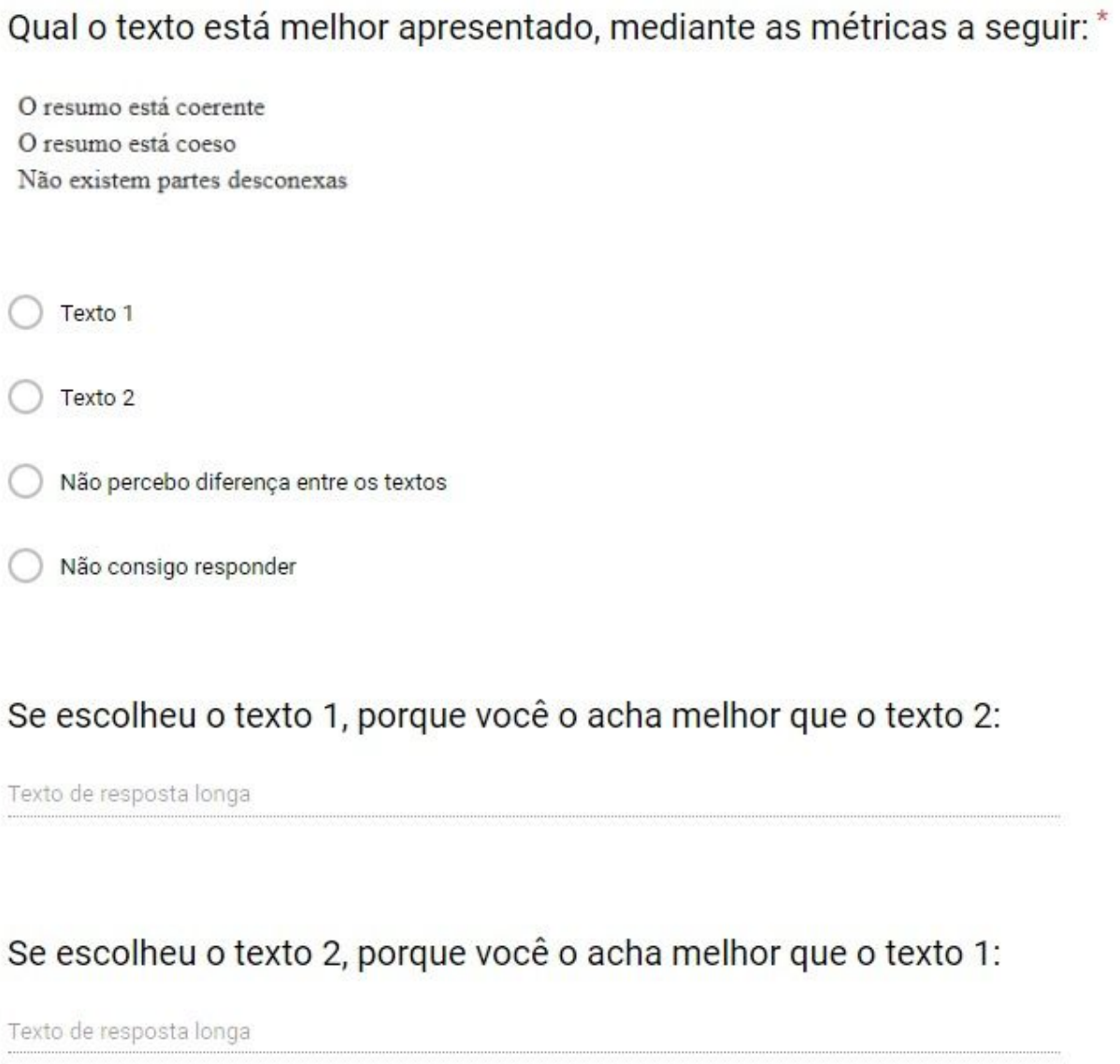

Figura 5 - Modelo Questionário

Vale ressaltar que não foi revelado os algoritmos utilizados, logo o texto 1 em todos os casos representa o resumo proveniente do algoritmo de Marques, enquanto o texto 2 o algoritmo de Luhn.

8. Resultados e Discussão

Primeiramente, como o objetivo de estudo está relacionado aos algoritmos foram coletadas informações sobre o modo de execução dos mesmos

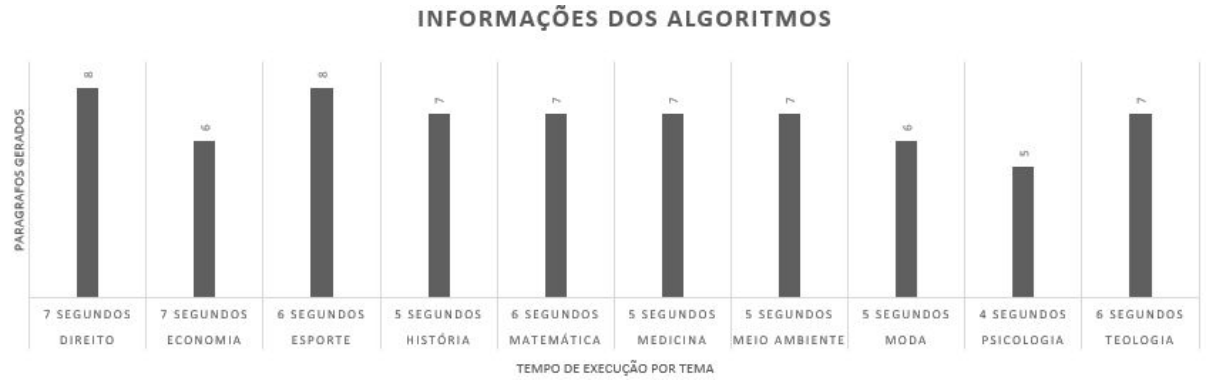

Gráfico 1 - Informações dos Algoritmos

Sendo assim, o algoritmo de Marques possui uma média de execução de 5,6 segundos, média igual ao de Luhn. É notório que em relação ao tempo de execução os dois algoritmos não tem diferença significativa, não podendo tal parâmetro ser utilizado como critério avaliativo. No gráfico, é informada a quantidade de parágrafos gerados, que como dito anteriormente, pode ser alterada no próprio algoritmo. Nesse caso, tal quantidade representa o número de sentenças importantes encontradas durante a mineração do texto [16], que ao encontrar estas sentenças gera o resumo com esta quantidade de parágrafos.

Após o período de 15 dias de junção da avaliação de um grupo de leitores em relação aos dois resumos gerados pelo algoritmo de Luhn e o de Marques, foram coletados os seguintes resultados apresentados.

Tabela 2 - Algoritmo de Luhn x Algoritmo de Marques

	Texto 1	Texto 2	Não há diferença entre os textos	Não consigo opinar
Teologia	0	2	1	0
Psicologia	0	3	0	0
Moda	1	2	0	0
Meio Ambiente	1	2	0	0
Medicina	2	1	0	0
Matemática	1	1	1	0

História	0	3	0	0
Esporte	0	3	0	0
Economia	0	3	0	0
Direito	1	0	2	0

De acordo com cada tema, é computada a quantidade de votos em cada algoritmo ou até mesmo nas opções neutras em que não há uma diferença significativa entre os textos ou o leitor não consegue opinar por determinados motivos. Observamos que assim como foi explicado no método avaliativo cada temática possui um total de 3 pessoas avaliando. Sendo assim, o algoritmo de Marques obteve 6 votos no total, enquanto o de Luhn, 20. Os outros 4 votos estão relacionados aos que não encontraram diferença entre os dois,

Graficamente (Gráfico 2), podemos observar de forma ainda mais clara a discrepância na preferência dos leitores em relação aos resumos obtidos pelos algoritmos.

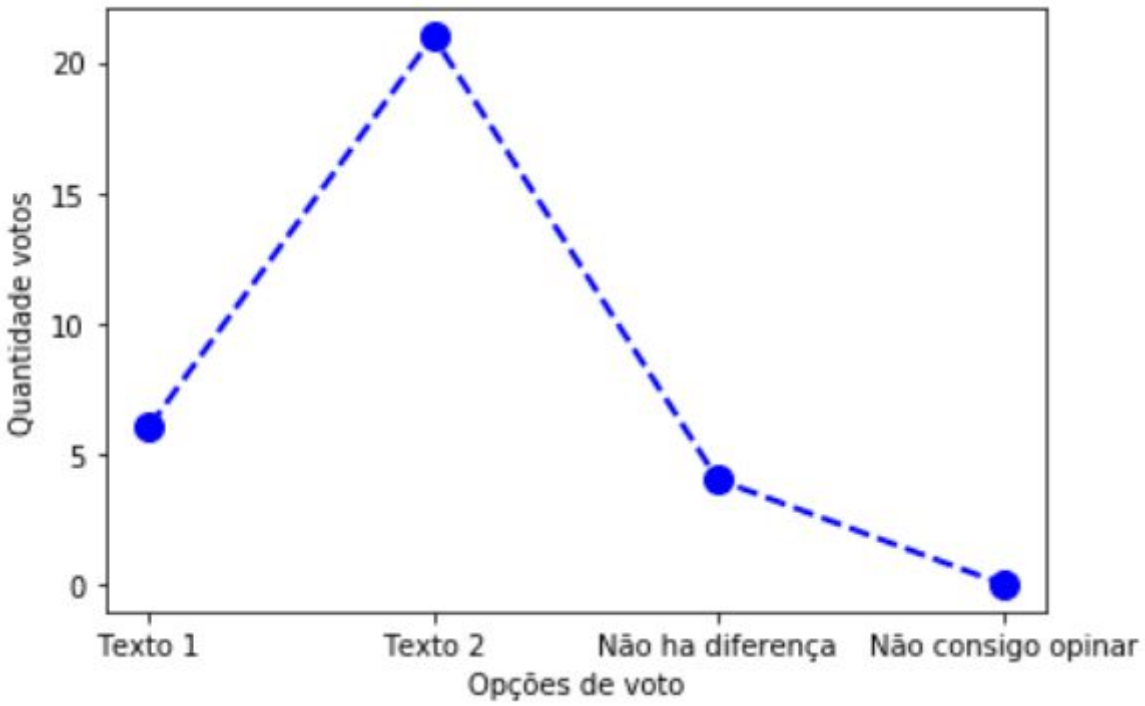

Gráfico 2. Quantidade de votos

9. Conclusão

De acordo com os testes realizados os algoritmos possuem desempenho significativamente diferentes, uma vez que o algoritmo de Luhn apresenta 66,66% dos votos em contraponto a 20% do algoritmo proposto. O resultado destoa do esperado inicialmente pelo viés dos pesquisadores, entretanto não implica em uma resultado incontestável, uma vez que pode-se citar uma série de fatores limitantes, principalmente em relação ao método avaliativo, iniciando pela dependência da avaliação humana, uma vez que não é controlável o grau de interesse e seriedade ao ler os textos propostos, a experiência que cada um possui com o assunto, o momento e o tempo que foi realizada a leitura. Além disso, a ordem em si dos textos apresentados no questionário, que tendenciou o resultado. Em relação, a construção do algoritmo proposto. Uma proposta de trabalho futuro é modificar o método de abordagem avaliativa, procurando ao máximo evitar a influência de fatores externos nos resultados.

Esse resultado não quer dizer que o algoritmo de Luhn é melhor que o de Marques, visto que, como foi explicado no item 5, Algoritmos de Sumarização Avaliados, o algoritmo de Marques foi criado com base no de Luhn, mas foi implementado tentando melhorar os defeitos que o algoritmo de Luhn tem, como não se aprofundar no texto, tentar sempre pegar qualquer tipo de sentença para gerar o resumo. O algoritmo de Marques tenta fazer uma análise de sentimento do texto, uma das funcionalidades do PLN, para a partir daí criar um resumo inteligente.

10. Referências

[1] Gariba, M. Jr., Schneider, M. C. K., Rosa, A. E., Casagrande, J. B., Santos, C. S. **"Reconhecimento de Fala e Processamento da Linguagem Natural"**.

[2] GONZALEZ, M.; LIMA, V. L. S. **Recuperação de informação e processamento da linguagem natural.** 2003.

[3] LUHN, H. P. **A Statistical Approach to Mechanized Encoding and Searching of Literary Information.** IBM Journal of Research and Development. N. 1. V. 4. p. 309-317. 1957.

[4] Balage Filho, P.P.; Uzêda, V.R.; Pardo, T.A.S.; Nunes, M.G.V. (2006). **Estrutura Textual e Multiplicidade de Tópicos na Sumarização Automática: o Caso do Sistema GistSumm**. Série de Relatórios Técnicos do Instituto de Ciências Matemáticas e de Computação, Universidade de São Paulo, no. 283. São Carlos-SP, Novembro, 18p.

[5] Barzilay, R.; Elhadad, M. (1997). **Using Lexical Chains for Text Summarization**. In the Proc. of the Intelligent Scalable Text Summarization Workshop, Madri, Spain. Also In I. Mani and M.T. Maybury (eds.), Advances in Automatic Text Summarization. MIT Press, pp. 111-121.

[6] Aretoulaki, M. (1996). COSY-MATS: **A Hybrid Connectionist-Symbolic Approach To The Pragmatic Analysis of Texts For Their Automatic Summarisation.** PhD. Thesis. University of Manchester.

[7] Black, W.J.; Johnson, F.C. (1988). **A Practical Evaluation of Two Rule-Based automatic Abstraction Techniques.** Expert Systems for Information Management, Vol. 1, No. 3. Department of Computation. University of Manchester Institute of Science and Technology.

[8] Carlson, L.; Marcu, D.; Okurowski, M.E. (2003). **Building a Discourse-Tagged Corpus in the Framework of Rhetorical Structure Theory.** In J. van Kuppevelt and R. Smith (eds.), Current Directions in Discourse and Dialogue, pp. 85-112. Kluwer Academic Publishers.

[9] Lin, C.Y. and Hovy, E. (2003). **Automatic Evaluation of Summaries Using N-gram Co Occurrence Statistics.** In Proceedings of 2003 Language Technology Conference (HLTNAACL 2003), Edmonton, Canada.

[10] Covington, M. **NLP for Prolog Programmers**, Prentice-Hall, 1994.

[11] Covington, M., Nute, D. and Vellino, A. **Prolog Programming in Depth**, Prentice-Hall, 1997.

[12] Russell, S. and Norvig, P. **Artificial Intelligence - a modern approach**, PrenticeHall, 1995.

[13] Russel, Matthew A.. **Mineração de dados da Web Social.** Primeira edição, Novatec. São Paulo, 2011.

[14] Silla Jr., Carlos N., Kaestner, Celso A.A.. **kNNSumm: Um Sumarizador Automático de Documentos Utilizando Aprendizado Baseado em Instâncias.** Pontifícia Universidade Católica do Paraná (PUC-PR). Curitiba, Paraná. 2004.

[15] Oliveira, Marcelo Arrantes; Guelpeli, Marcos Vinicius. **BLMSumm – Métodos de Busca Local e Metaheurísticas na Sumarização de Textos.** Centro Universitário de Barra Mansa (UBM). Barra Mansa, Rio de Janeiro. 2011.

[16] Aranha, Christian; Passos, Emmanuel. **A Tecnologia de Mineração de Textos.** RESI-Revista Eletrônica de Sistemas de Informação. 2006

[17] Rino, Lucia Helena Machado; Pardo, Thiago Alexandre Salgueiro. **A Sumarização Automática de Textos: Principais Características e Metodologias.** NILC/Departamento de Computação. Universidade Federal de São Carlos, São Paulo. 2003.

[18] MARGARIDO, P. R. A.; PARDO, T. A. S.; ALUÍSIO, S. M. **Sumarização Automática para Simplificação de Textos: Experimentos e Lições Aprendidas.** Avaliando a Qualidade Afetiva de Sistemas Computacionais, v. 1, n. 1, p. 3, 2009.

[19] Filho, Pedro Paulo Balage; Pardo, Thiago Alexandre Salgueiro; Nunes, Maria das Graças Volpe. **Sumarização Automática de Textos Científicos: Estudo de Caso com o Sistema GistSumm.** NILC - ICMC-USP. São Carlos, São Paulo. 2007

[20] **NLTK** = Bird, Steven, Edward Loper and Ewan Klein (2009), *Natural Language Processing with Python*. O'Reilly Media Inc.