Pontifícia Universidade Católica de São Paulo
PUC-SP

KOICHI SANOKI

A ideia principal de um texto acadêmico: um meio para descrever tendências semânticas

Doutorado em Tecnologias da Inteligência e Design Digital

SÃO PAULO

2019

intencionalmente deixado em branco

Pontifícia Universidade Católica de São Paulo
PUC-SP

KOICHI SANOKI

A ideia principal de um texto acadêmico: um meio para descrever tendências semânticas

Doutorado em Tecnologias da Inteligência e Design Digital

Tese apresentada à Banca Examinadora da Pontifícia Universidade Católica de São Paulo, como exigência parcial para obtenção do título de Doutor em Tecnologias da Inteligência e Design Digital com concentração na área Processos Cognitivo e Ambientes Digitais sob a orientação do Prof. Dr. Ítalo Santiago Vega.

SÃO PAULO

2019

intencionalmente deixado em branco

Banca Examinadora

_____

_____

_____

_____

_____

Agradecimentos

"O presente trabalho foi realizado com apoio da Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil (CAPES) - Código de Financiamento 001."(cf. Art. 3º, do Ato do Pró-Reitor de Pós-Graduação nº 07 - 2018) nº do processo: 88887.197361/2018-00.

Agradecimentos

À minha mulher, Dona Luiza, a meu filho Leonardo e sua esposa Sarah. Principalmente à minha neta Alani, de dois anos, que foi o maior incentivo para minha participação no mundo acadêmico e para o término desta tese, na tentativa de deixar-lhe um futuro melhor.

Ao Prof. Dr. Ítalo Santiago Vega, orientador dedicado que me impulsionou a chegar neste momento.

À Sra. Edna Conti, secretária do programa TIDD da PUC-SP, que dedica sua atenção a todos, mas precisou aplicá-la ainda mais a mim nas atividades desse curso.

A Marcele Flores, a revisora que deixou esta tese apresentável.

RESUMO

Esta tese descreve como formular indagações do tipo paradoxal em palavras semânticas sobre a ideia principal de um artigo acadêmico, despertando curiosidade para efetuar a leitura completa de tal artigo. Ao localizar essa ideia principal de um texto que se encontra em uma mídia digital, aplica-se um algoritmo descrito para interpretá-lo com uma técnica estrutural orientada a objetos. A função primordial desse algoritmo é reorganizar as partes do texto para que façam sentido enquanto objeto computacional que contém a ideia principal. O processo de localização da ideia começa com pesquisas em bibliotecas virtuais, repositórios de periódicos ou de outras publicações onde parte do tema já tenha sido veiculada em uma mídia computacional com as suas referências bibliográficas. Para dar suporte a essa função, há a identificação dos subtítulos e grandes itens encontrados no texto, tais como Introdução, Metodologia, Discussão, Resultado, e Conclusão, e a há a produção de um paralelo entre o tema desse artigo e um objeto na vida real, entre as descrições da Introdução e características do objeto, e entre as metodologias descritas e os métodos de classe em estrutura orientada a objetos. Com os itens identificados, o passo seguinte é a detecção dos períodos gramaticais e a verificação de quais são os tipos de verbos, pois são eles que descrevem as ações, ou estáticas da sentença, que é o objeto desta tese. Sendo o texto, com valor semântico, uma costura e um entrelaçamento de palavras resultando em uma mensagem de informações que a princípio não obedecem a regras de composição acadêmica, nem a sequências lógicas, e nem a tipos de algoritmos heurísticos, torna-se necessário que haja uma sequência padronizada nas descrições das características do tema. Realizada a organização, é criada uma padronização de acordo com o que o algoritmo determina, para que assim o artigo seja tratado como uma estrutura orientada a objetos. Essa é a maneira para organizar quais são as ações e funções que o artigo descreve, e que serão reescritas no esquema computacional como: objeto, propriedade e funções. Uma vez identificada a ideia, serão geradas indagações com características do tipo paradoxais, tais como: “sair pela direita” e “entrar pela esquerda”, e com características semânticas, como: “ver” é o ato de enxergar, “montar” é estar sobre, afirmações que geram indagações polêmicas para quem não enxerga, ou para quem não consegue se locomover, e que possam despertar no pesquisador o interesse pela leitura de determinado texto acadêmico, que então questionarão qual é a ideia que essas palavras associadas a valores semânticos representam nesse artigo. As indagações estão relacionadas às palavras que compõem a ideia principal e as secundárias encontradas nos períodos gramaticais do texto. A fórmula da indagação sobre a ideia principal consiste em ter o pronome interrogativo questionando o valor semântico das palavras em tipo paradoxal, apoiada no algoritmo computacional que identifica o item principal e os secundários de um artigo acadêmico, e gera características com um tipo de ironia que o texto original não descreve explicitamente.

Com tal identidade surge a curiosidade que torna necessária a leitura completa do texto, e a aceitação, ou não, das formas de indagações descritas por esta tese.

Palavras Chave: Formulação de indagações. Algoritmo computacional. Tipo paradoxal. Valores semânticos.

ABSTRACT

This thesis describes how to formulate questions of the paradoxical type in semantic words about the main idea of an academic article, arousing curiosity to perform the reading Complete of such an article. By locating this main idea of a text that is on a digital media, an algorithm described to interpret it with an object-oriented structural technique is applied. The primary function of this algorithm is to rearrange the parts of the text to make sense As a computational object that contains the main idea. The process of locating the Idea begins with searches in virtual libraries, journal repositories, or other Publications where part of the theme has already been carried on a computational media with the Its bibliog raphical references. To support this function, there is the identification of the Subheadings and large items found in the text, such as introduction, methodology, Discussion, outcome, and conclusion, and the production of a parallel between the theme of this article and an object in real life, between the descriptions of the introduction and characteristics of the object, and between the methodologies described and the class methods in structure oriented to objects. With the items identified, the next step is the detection of grammatical periods and the verification of which are the types of verbs, because they describe the actions, or static of the sentence, which is the object of this thesis. Being the text, with semantic value, a seam and an interweaving of words resulting in a message of information that at first do not obey the rules of academic composition, neither the logical sequences, nor the types of heuristic algorithms, becomes A standardized sequence is required in the descriptions of the theme characteristics. When the organization is organized, a standardization is created according to what the algorithm determines, so that the article is treated as an object-oriented structure. This is the way to organize what actions and functions The article describes, and which will be rewritten in the computational schema as: Object, property, and functions. Once the idea is identified, questions will be generated with characteristics of the paradoxal type, such as: "Exit right" and "enter from the left", and with semantic characteristics, such as: "See" is the act of seeing, "assembling" is to be on, affirmations that generate Polemic inquiries for those who do not see, or for those who cannot get around, and that may awaken in the researcher the interest in reading a certain academic text, which will then question what is the idea that these words associated with semantic values represent in this article. The questions are related to the words that compose the main idea and the secondary ideas found in the grammatical periods of the text. The formula of the question about the main idea is to have the interrogative pronoun questioning the semantic value of the words in paradoxical type, supported by the computational algorithm that identifies the main item and the secondary items of an academic article, and generates Characteristics with a type of irony that the original text does not explicitly describe. With such identity arises the

curiosity that makes necessary the complete reading of the text, and the acceptance, or not, of the forms of inquiries described by this thesis.

Key-words: Formulation of inquiries. Computational algorithm. Paradoxical type. Semantic values.

SUMÁRIO

1 INTRODUÇÃO

1.1 Introdução

Atualmente, com uma grande quantidade de material acadêmico escrito disponível, muitas vezes quem realiza pesquisas se depara com dificuldades em encontrar o apoio ideal para o seu trabalho. Esta tese tem como objetivo estimular nos pesquisadores o interesse pela leitura de textos acadêmicos e facilitar a montagem das referências bibliográficas que darão suporte às suas monografias, dissertações e teses. Para tal, analisa o desenvolvimento de uma alternativa para a otimização da busca por referências, na forma de um algoritmo computacional.

A pesquisa em textos acadêmicos requer muito tempo, inclusive com investigação em inúmeros livros físicos, o que não será enfocado nesta tese, que trabalhará somente com textos presentes em repositórios digitais. Entre toda a leitura realizada, há muito conteúdo que acaba não oferecendo contribuição, o que pode provocar grande desestímulo. Com a ideia de evitar essa situação e fomentar a leitura completa, porém realmente útil, de um texto, este trabalho apresenta um algoritmo computacional capaz de ajudar muito numa localização mais precisa das ideias presentes num texto antes da sua leitura, fazendo com que o pesquisador possa identificar mais rapidamente, além do resumo e palavras-chave, o que precisa ser lido aprofundada e integralmente, otimizando assim o tempo de pesquisa.

Este algoritmo adquire informações de um texto em um repositório e as particiona em palavras, rotulando cada uma delas conforme sua classe gramatical para então identificar os tipos de ações existentes. Como quase todos os textos acadêmicos apresentam aspectos conceituais em partes como Introdução, Metodologia e Resultado, pode-se conhecer a ligação entre a parte Resumo e as outras partes do texto, assim como a frequência das palavras-chave ao longo do mesmo. É parte de um processo que ampliará e agilizará a melhor identificação dos conteúdos.

1.2 Estado da arte

No artigo *"Reading Wikipedia to answer open-domain questions"*, Danqi Chen aborda o ponto da utilização da *Wikipedia* como fonte única para respostas a qualquer questão factóide de um trecho de texto em estudo. Esta tarefa combina os desafios da recuperação de documentos (encontrando os artigos relevantes) com os da compreensão da leitura de um texto pelo algoritmo (identificando a resposta a partir desses artigos) (CHEN et al., 2017).

A compreensão da leitura, ou a capacidade de ler um texto e, em seguida, responder a perguntas sobre ele é uma tarefa desafiadora para máquinas. Exige tanto a compreensão da linguagem natural, quanto conhecimento sobre o mundo. O artigo *"SQuAD: 100,000 + Questions for machine comprehension of text"* mostra que um banco de dados denominado SQuAD, com mais de cem mil questões, faz a máquina buscar um texto especifico em determinadas áreas predefinidas pelos desenvolvedores deste aplicativo (RAJPURKAR et al., 2016).

O artigo *"Sentence similarity based on semantic nets and corpus statistics"* propõe uma estratégia de agrupamento semântico de sentenças para medir semelhanças entre textos com a finalidade de resolver o problema da perda de informações sobre a relação de dependência e informações estruturais (LI et al., 2006).

1.3 Contextualização

Existe um conjunto de regras, as macrorregras, que podem ser consideradas como estratégias através das quais o leitor resume um texto, retendo as informações que considera centrais. Isso envolve estratégias de apagamento, em que há a seleção de proposições relevantes, e se dá a generalização (a substituição de um conjunto de nomes de seres, de propriedades e de ações por um nome, propriedade ou ação mais geral) e a construção, ou seja, a substituição de uma sequência de proposições por uma proposição que dela é deduzida (MACHADO; BEZERRA, 2002).

Quando a ação de resumir um texto, além de implicar sua leitura, envolve também a consequente elaboração de um novo texto, tem-se a produção de um gênero, o resumo. Depreender as características desse gênero não é tarefa fácil, pois seus traços mais relevantes dependem das funções atribuídas à atividade de textualização (MATENCIO, 2002).

Na busca de como fazer automaticamente o resumo de livros e artigos, em análises realizadas por Sousa e Silva a partir de experimentos, foram observadas três fases no processamento da escrita: planejamento, tradução e revisão (conforme modelo de Hayes e Flower, apud Sousa e Silva) (FARIAS, 2000b). A ver:

• Fase de planejamento: – Pela seleção de ideias relevantes do texto-base e por sua organização em uma superestrutura. – O sujeito de seu experimento apenas reproduziu no seu resumo a ordem original do texto-base, por vezes apagando a informação básica ou mantendo-a redundante, ou ainda criando trechos incoerentes, demonstrando não ter reconhecido adequadamente sua superestrutura, nem estabelecido uma superestrutura para seu próprio texto.

• Fase de tradução: – Pelo uso de mecanismos formais de coesão, pela manutenção

do tópico discursivo, pelo uso de estruturas sintáticas complexas, etc. – O sujeito utilizou ou estruturas sintáticas muito semelhantes às do texto original, ou outras estruturas deficientes.

• Fase de revisão: – Pela reconsideração atenta do texto-resumo quanto aos objetivos da revisão (adequação à audiência, clareza da superestrutura, e marcas formais adequadas). – O sujeito limitou-se a ajustes superficiais (pontuação, concordância, por exemplo) (FARIAS, 2000b).

1.4 Hipótese

O propósito é desenvolver um algoritmo que gere indagações a partir de um texto acadêmico, e cujo resultado a otimização do tempo em uma atividade de leitura extremamente longa.

É preciso utilizar um algoritmo computacional que possa identificar a sequência lógica da ideia principal de um artigo científico ou acadêmico. Essa identificação é produzida pela decomposição e posterior reconstrução do conteúdo de um texto num modelo de lógica de programação estruturada.

Antes de fazer a desconstrução, qualquer que seja o meio, é preciso primeiro reconhecer se é referente à escrita de um objeto virtual, para assim poder classificá-lo no que pode ser uma definição, um processo, ou um algoritmo. Enfim, essa classificação definirá os métodos e atributos a serem utilizados na descrição do mesmo.

O aplicativo que contem esse algoritmo, consultando os repositórios científicos disponíveis e acessíveis, buscará em mídias materiais como textos, artigos, livros, periódicos e publicações, a partir de suas palavras-chave. Tais materiais consultados serão guardados em um banco de dados que será denominado de Corpus Resumos. Encerradas as consultas e realizado o armazenamento em corpus (este pode ser opcional), parte-se para a segunda fase, que será a geração de indagações acerca de cada texto armazenado.

Uma vez disponibilizados os resumos, haverá a opção de fazer combinações ou junções dos mesmos, o que poderá mostrar outra forma de entender o que as palavras-chave estão nos indicando. Uma das alternativas é ter um algoritmo computacional que automaticamente gere indagações sobre os textos de livros, artigos, ou outros.

Outra possibilidade de busca da ideia principal é que o algoritmo recupere as informações (QUARESMA, 2006) do aplicativo no engendramento de indagações a partir de argumentos (SANOKI; VEGA, 2018a), e para isso é necessária a verificação das consistências das partes Introdução, Metodologia, Discussão, Resultados e Conclusão (CHEMIN, 2015) para que se estabeleça uma relação com as indagações a serem geradas a

partir da parte Resumo do texto (CAMARGO; BELLOTTO, 1996).

Por fim, todo esse processo localizará materiais úteis ao pesquisador, ajudando-o também a decidir se faz a leitura completa de determinado texto, ou não, estimulando consultas proficientes.

Faz-se necessária a divulgação dessa nova ideia, ou forma de indagar os textos acadêmicos, agregando força ao movimento de produção intelectual e somando competências, ou mudando o processo que se julga atual, acrescentando um passo a conhecimentos lidos ou adquiridos anteriormente.

1.5 Objetivos

A motivação para escrever este trabalho foi a vontade de expor que existem formas de incentivar a leitura de textos acadêmicos que ainda não foram exploradas. Nesta tese, escolheu-se como foco o desenvolvimento um algoritmo computacional para tal objetivo.

1.5.1 Objetivo geral

O objetivo desta tese é o estímulo à leitura de textos acadêmicos, oferecendo, para isso, meios de otimização da identificação das ideias dos seus conteúdos antes da leitura em si.

1.5.2 Objetivos específicos

- Demonstrar como um algoritmo computacional pode, através da decomposição e reconstrução de textos, e do armazenamento de dados e geração de questionários, agilizar a localização das ideias contidas nos textos.
- Explicar e fundamentar esse processo.
- Apresentar o algoritmo.

1.6 Justificativa

Esta tese apresenta uma contribuição para qualquer meio, mas principalmente para o acadêmico, que pode otimizar o tempo do pesquisador na seleção de referências, sem que ele precise efetuar leituras completas e exaustivas de um texto, que às vezes não correspondem à suas necessidades. Conforme Gil, ter resumos de livros, artigos, jornais, auxilia na tomada de decisão de leitura (GIL, 2000).

É possível conhecer os verdadeiros argumentos e justificativas de textos, o que vai além de conhecer somente seus títulos, resumos, e palavras-chave, sem ter de fazer uma leitura completa dos mesmos.

Segundo Lúcia Rino, para assegurar uma ideia central há de se ter um modelo de discurso com o aspecto inovador que está na associação da intencionalidade, da coerência e da semântica (RINO, 1996). Estudando e utilizando esses elementos, este trabalho visa uma ampliação significativa na busca das ideias principais de textos, através de um algoritmo. O título de um texto tratado como objeto, assim como as descrições das ações de funções no item denominado Metodologia, entre outros, alavancam o desenvolvimento ou a melhoria de uma ferramenta de apoio revolucionária, que ajuda a impulsionar conhecimentos inertes, estagnados ou ultrapassados.

Quando um pesquisador escreve, descreve ou propõe um algoritmo, normalmente utiliza um processo já existente, mas o que deve mesmo mostrar é que sempre há uma alternativa para atingir o objetivo além das que se conhece, pois existem muitas variáveis que não se consegue ver de forma sequencial ou lógica, uma vez que as ideias ficam num mundo externo ao nosso conhecimento habitual.

A aplicação desse algoritmo no mundo acadêmico pode trazer consideráveis contribuições ao desenvolvimento de monografias, dissertações e teses, visto que o aplicativo fará buscas de textos de forma repetitiva, denominadas "robôs", facilitando a localização dos mais variados tipos de informações disponíveis em redes de repositórios acadêmicos, antes impossíveis de serem acessados remotamente (SANOKI, 2017).

1.7 Linhas de pesquisa

Dentro da área de concentração Processos Cognitivos e Ambientes Digitais, das três linhas de pesquisa interdisciplinares que o programa de doutoramento oferece a escolhida é Inteligência em sistemas. São realizadas atividades de levantamento de pré-requisitos, estudo do objeto da pesquisa, e conclui-se com a produção de um algoritmo computacional para definição de um aplicativo que reproduz esse processo.

Para o desenvolvimento desse algoritmo, nessa linha de pesquisa, além da engenharia de modelagem de dados, outros conteúdos são necessários, como bancos de dados e mídias auxiliares, sem o que não há solução adequada para o desenvolvimento desse aplicativo.

1.8 Metodologia

O ponto inicial desta pesquisa, em ordem de definir o caminho a seguir, foi vivenciar as dificuldades pelas quais os pesquisadores passam na busca de referências e objetivos, e nessa linha foi feita uma extensa procura por resumos.

"A produção de resumos na universidade é uma das maneiras através das quais o estudante, além de registrar sua leitura de textos acadêmicos, manifesta sua compreensão de conceitos e do saber fazer em sua área de conhecimento." (MATENCIO, 2002).

A metodologia de pesquisa, incluindo a diagramação do estudo, teve o desenvolvimento de artigos com os princípios de análise e raciocínio. Especial atenção foi dada para a seleção de tipos de artigos, para cada etapa do raciocínio de algoritmo incluindo os critérios de inclusão de corpus, argumentos e estruturação de programação orientada a objetos, levando em consideração éticas relevantes.

No estudo sobre corpus considerou-se a possibilidade de localizar em bibliotecas digitais universais e especializadas (em termos de obtenção da base de recursos) e pesquisar todos os textos relacionados ao tema de estudo do pesquisador, e armazená-los para a extração e análise do conhecimento de contexto na área específica (LYAPIN et al.,).

No estudo sobre argumentos viu-se que não há necessidade de ter todos os artigos relacionados ao tema que se pesquisa, apenas os de real relevância, que serão identificados através da desmontagem do texto em palavras e consequente processo que será visto nesta tese. Em tal estudo sobre as consistências dos argumentos (SANOKI; VEGA, 2018a), em relação a itens como Resumo, Introdução, Metodologia e Conclusão de um texto acadêmico, serão incluídos a descrição dos métodos e outros trabalhos de grupo utilizados (LIUMBRUNO et al., 2016).

Para buscar a ideia principal do texto em análise, foi realizado estudo sobre a estruturação do texto orientada a objetos (SANOKI; VEGA, 2019) (no prelo).

1.9 Fundamentação teórica

Com muita frequência textos são escritos "como se quer", sem levar em consideração o público que o lerá, escutará ou interpretará. É como escrever para si mesmo.

Sendo que o título e o tema de um texto acadêmico vêm a ser um objetivo descrito por objeto de uma descoberta (uma nova ideia), de outra proposição de uma descoberta (contraposição a uma ideia), de uma redescoberta (a variação de uma ideia), ou ainda de uma nova alternativa proposta em algum outro artigo, nem tudo que está escrito tem base teórica na ciência ou na comunicação para sustentar uma redação científica (L.FIORIN,

2018).

Para a identificação da ideia principal ou do conceito central de um texto, buscam-se os argumentos na parte introdutória, e as justificativas em partes como Resultados e Conclusões. Uma das formas de realizar isso é utilizando a técnica de um avaliador de texto acadêmico, e fazendo a "desconstrução" desse texto, conforme:

> A verdade relativa, aquela obtida da relação do pesquisador com o objeto de sua investigação, deve estar expressa na Introdução. A evidência, expressada na forma da interpretação dos resultados e do diálogo com a comunidade científica, deve estar contida na Discussão. A certeza deve ser expressa na forma de Conclusão (FERREIRA; CASSIOLATO; GONZALEZ, 2009).

Para Charles Tayler, o argumento é a parte mais importante na montagem de uma questão, e o seu tipo indica uma forma de pensamento (TAYLER, 2019).

"A teoria da Argumentação na Língua" (ANSCOMBRE; DUCROT apud BARBISAN, 2004, p. 22), cunhada por Anscombre e Ducrot, baseia-se no postulado de que "a argumentação é o elemento essencial para a apreensão do sentido do enunciado, e que esse sentido é argumentativo, construído a partir da língua, e que a argumentação está na língua" (DAVID, 2007).

Foi realizado um trabalho sistemático de pesquisa nas bases de dados SCOPUS, JSTOR, e Google Scholar.

2 FUNDAMENTOS

2.1 Origem da palavra “texto” e seu significado

A origem da palavra “texto” vem do latim *textus* do século XIV e conforme dicionário etmológico (FARIAS, 2000b) significa: tecer, construir, entrelaçar os fios. Entende-se “texto” como uma “sequência de palavras e frases articuladas, escritas sobre qualquer suporte” (CAMARGO; BELLOTTO, 1996). Podendo-se entender em sua definição como uma manifestação da ideia de um pesquisador.

Um texto precisa ter sentido, tem que ter coerência relacionada à compreensão da escrita, e “a coesão textual é a relação, a conexão entre as palavras, expressões ou frases do texto” (PLATÃO, F. and FIORIN, J. L., 1996).

Sendo uma sequência de palavras, o texto precisa apresentar-se na forma de períodos gramaticais para que estes estabeleçam relações entre si, e requer um esquema lógico na descrição do seu tema ou objeto para que, por sua vez, haja uma coerência com a descrição dos pré-requisitos. Também, nas descrições de períodos gramaticais, há sequenciadores tipo conjunções ou operadores lógicos e argumentativos, que são as palavras que tem por finalidade unir orações de mesmo valor gramatical para dar-lhes sentido completo (GERALDI; GUIMARÃES; ILARI, 2012).

2.2 Paradoxo

A palavra ”paradoxo”vem do latim *paradoxum* e do grego *paradoxos*. O prefixo *“para”* quer dizer contrário a, ou oposto de, e o sufixo *“doxa”* quer dizer opinião. A etimologia da palavra paradoxo pode ser traçada a textos que remontam à aurora da Renascença, um período de acelerado pensamento científico na Europa e Ásia que começou por volta do ano de 1500. As primeiras formas da palavra tiveram por base a palavra latina *paradoxum*, mas também são encontradas em textos em grego como *paradoxon* (entretanto, o latim é fortemente derivado do alfabeto grego e, além do mais, o português é também derivado do latim romano, com a adição das letras ”s”e ”n”).

Um paradoxo é uma declaração aparentemente verdadeira que leva a uma con-

tradição lógica, ou a uma situação que contradiz a intuição comum. Em termos simples, um paradoxo é o oposto do que alguém pensa ser a verdade, ou o que é contrário a uma opinião admitida como válida. Ele consiste em uma ideia incrível, inversa ao que se espera. Também pode representar a ausência de nexo ou lógica. O paradoxo muitas vezes depende de uma suposição da linguagem falada, visual ou matemática, porque modela a realidade descrita. A identificação de um paradoxo baseado em conceitos aparentemente simples e racionais tem, por vezes, auxiliado significativamente o progresso da ciência, da filosofia e da matemática [1].

Para Quine, um paradoxo é um argumento aparentemente bem-sucedido, tendo como conclusão uma afirmação ou proposição que parece obviamente falsa ou absurda. Essa conclusão ele chama de "a proposição do"paradoxo em questão (LYCAN, 2010).

2.3 Classes de paradoxos

W. V. Quine distingue três classes de paradoxos: os verídicos, os falsídicos, e aqueles que não pertencem às classes anteriores (QUINE, 1962).

Os paradoxos verídicos produzem um resultado que parece absurdo embora seja demonstravelmente verdadeiro. Assim, o paradoxo do aniversário de Frederic na opereta *The Pirates of Penzance* (GILBERT; SULLIVAN, 1968) estabelece o fato surpreendente de que uma pessoa pode ter mais do que N anos em seu N-ésimo aniversário [2]. Da mesma forma, o teorema da impossibilidade de Arrow (NETO, 2011) envolve o comportamento de sistemas de votação que é surpreendente mas, ainda assim, verdadeiro [3].

Os paradoxos falsídicos estabelecem um resultado que não somente parece falso, como também o é demonstravelmente. Há uma falácia da demonstração pretendida. As várias provas inválidas (por exemplo, que $1 = 2$) são exemplos clássicos, geralmente dependendo de uma divisão por zero despercebida. Outro exemplo é o Paradoxo do Cavalo, que diz: "Todos os cavalos são da mesma cor." [4] (HANSEN, 1976).

Um paradoxo que não pertence a nenhuma das classes acima pode ser uma antinomia, uma declaração que chega a um resultado autocontraditório aplicando apropriada-

[1] Fontes : https://www.significados.com.br/paradoxo/ - Acesso em 09 jul. 2019 e https://pt.wikipedia.org/wiki/Paradoxo - Acesso em 09 jul. 2019

[2] Alguém que nasceu no dia 29 de fevereiro, logo, num ano bissexto, e assim, tecnicamente, tem um aniversário a cada quatro anos. Fonte: https://en.wikipedia.org/wiki/The_Pirates_of_Penzance - Acesso em 09 jul. 2019

[3] O Teorema diz que a soma das racionalidades individuais não produz uma racionalidade coletiva e é atribuído ao economista ganhador do Prêmio Nobel de 1972 Kenneth Arrow. Fonte: https://osmatematicosdojoaocruz.blogspot.com/2012/10/paradoxos.html - Acesso em 09 jul. 2019.

[4] Indução matemática: vale para conjunto de apenas um elemento. Fonte: https://osmatematicosdojoaocruz.blogspot.com/2012/10/paradoxos.html - Acesso em 09 jul. 2019.

mente meios aceitáveis de raciocínio. Por exemplo, o paradoxo de Grelling-Nelson aponta problemas genuínos na nossa compreensão das ideias de verdade e descrição [5] (ALVES, 2013).

2.3.1 Lista de tipo de paradoxos

[6]

Paradoxos verídicos: são os paradoxos que dão resultados contra-intuitivos baseados em um raciocínio lógico correto.

Matemáticos / Lógicos

- Paradoxo do enforcamento inesperado: um paradoxo sobre as expectativas de uma pessoa sobre o momento de um evento futuro (por exemplo, um prisioneiro a ser enforcado ou um teste de escola) que ocorrerá em um momento inesperado.
- Paradoxo da implicação: premissas inconsistentes sempre resultam em argumentos válidos.
- Paradoxo de distribuição: alguns sistemas de distribuição de representantes ou deputados podem dar resultados contra-intuitivos.
- Paradoxo do paradoxo: um paradoxo que se contradiz a si mesmo, em contradição ao conceito de paradoxo que diz ser o oposto da verdade. Logo se um paradoxo contradiz um paradoxo, ele contradiz o contrário da verdade, ou seja, ele contradiz o contraditório já contradito pelos seus argumentos contrários.
- Médias: o conceito matemático de média, seja definido como média ou mediana, leva a aparentes resultados paradoxais, por exemplo, é possível que ao mover um artigo da Wikipedia para o Wiktionary, o tamanho médio de um entrada aumente em ambos os sites – o fenômeno Will Rogers.
- Teorema da impossibilidade de Arrow / Paradoxo da votação / Paradoxo de Condorcet: você não pode ter todos os atributos ideais de um sistema de votação ao mesmo tempo.
- Paradoxo de Banach-Tarski: corte uma esfera em cinco partes, monte as peças, e obtenha duas esferas, ambas do mesmo tamanho da primeira.

[5] Paradoxo do Barbeiro: somente quem barbeia raspa tudo. Fonte: https://osmatematicosdojoaocruz.blogspot.com/2012/10/paradoxos.html - Acesso em 09 jul. 2019.

[6] Fonte: https://pt.wikipedia.org/wiki/Lista_de_paradoxos - Acesso em 21 jul. 2019.

- Paradoxo do aniversário: em uma sala com 23 pessoas, a chance de que pelo menos duas tenham a mesma data de aniversário é maior que 50
- Paradoxo de Borel: funções de densidade probabilística condicional não são invariantes sob transformações de coordenadas.
- Paradoxo de Burali-Forti: se os números ordinais formassem um conjunto, ele seria um número ordinal menor do que ele próprio.
- Paradoxo do elevador: combinando as observações de um morador da cobertura com um morador do térreo a respeito de um mesmo elevador, chega-se à conclusão de que "os compartimentos"deste estão sendo construídos no meio do prédio e destruídos na cobertura e no térreo.
- Paradoxo de Galileu: embora a maioria dos números não sejam quadrados, não há mais números que quadrados.
- Corneta de Gabriel ou trompete de Torricelli: um objeto simples com volume finito mas com uma área de superfície infinita. Igualmente, o conjunto de Mandelbrot e vários outros fractais têm área finita, mas perímetro infinito.
- Paradoxo de Hausdorff: há um subconjunto contável C de uma esfera S tal que S C é equidecomponível em duas cópias de si mesmo.
- Paradoxo do Grand Hotel de Hilbert: mesmo que um hotel com infinitos quartos esteja completamente cheio, ele ainda pode receber mais hóspedes.
- Problema de Monty Hall: uma consequência contra-intuitiva da probabilística condicional.
- Problema de Monty Hell: lucros positivos diários tendem a ativos nulos no limite.
- Paradoxo do corvo (ou Os Corvos de Hempel): observar um não corvo vermelho aumenta a probabilidade de que todos os corvos sejam negros.
- Paradoxo de Richard: uma lista completa de definições dos números reais não existe.
- Paradoxo de Simpson: uma associação em subpopulações pode estar revertida na população em si. Aparentemente, quando dois conjuntos de dados suportam separadamente a mesma hipótese, unidos eles suportam a hipótese inversa.
- Paradoxo da Bela Adormecida: metade (ou um terço?) dos participantes do rec.puzzles é incapaz de concordar sobre a probabilidade resultante desse problema...

- Paradoxos estatísticos: é bem possível tirar conclusões erradas de correlações. Por exemplo, cidades com um número maior de igrejas em geral possuem uma taxa maior de criminalidade. Ambos resultam de uma população maior. Uma organização profissional uma vez descobriu que economistas com um PhD tinham um salário menor do que os somente com um bacharelado, o que realmente acontecia era que os economistas com PhD geralmente trabalhavam no meio acadêmico, onde os salários são comparativamente menores.

- Paradoxo do divórcio: a reta real pode ser decomposta em um conjunto de medida zero e um conjunto magro.

Psicológicos / Filosóficos

Paradoxos da ação

- Paradoxo de Abilene: pessoas agem em contradição com o que realmente querem fazer e, portanto, acabam removendo a chance de conseguir o que queriam em primeiro lugar.

- Asno de Buridan: como uma escolha racional pode ser feita entre duas possibilidades de igual valor?

- Paradoxo de Condorcet: agentes racionais podem tomar decisões coletivas irracionais.

- Paradoxo do controle: o homem nunca pode estar livre de controle já que ser livre de controle é ser controlado por si mesmo.

- Paradoxo do poeta: o poeta não escreve vendo e, sim, sentindo.

- Paradoxo do hedonismo: quando alguém persegue a felicidade, esse alguém é miserável; mas, quando alguém persegue outro objetivo, ele atinge a felicidade.

- Paradoxo da exceção: "Toda regra tem uma exceção."Se considerarmos isso uma regra, então ela deve ter uma exceção. Se ela tem exceção então haverá regra sem exceção.

- Paradoxo oposto: "Dá para ter certeza sobre tudo."Dá para ter certeza também que esta afirmação está errada, então não dá para ter certeza sobre tudo. "Não dá para ter certeza sobre nada". Nem mesmo ter certeza se esta frase está certa, então dá para ter certeza sobre algo. Assim, os paradoxos opostos demonstram que sobre algumas coisas dá para se ter certeza e que sobre outras não se dá. Obs.: este paradoxo oposto é aplicável para alguns outros.

- Paradoxo do Perfeccionismo: "Se uma pessoa é perfeccionista, ela nunca será perfeita, pois o perfeccionismo já é considerado um defeito."

Paradoxos epistemológicos

- Paradoxo da loteria: uma pessoa pode acreditar, de cada número, que o mesmo não será sorteado na loteria, ao mesmo tempo em que acredita que um número destes será sorteado na loteria.
- Paradoxo de Moore: não há contradição entre acreditar que P e afirmar que não-P. "Está chovendo, mas eu não acredito que está".
- Paradoxo do avô: viaja-se no tempo para o passado e impede-se que esta mesma viagem seja feita, ou elimina-se a existência de quem viajou.

Paradoxos metafísicos

- Paradoxo de Epicuro: a existência do mal é incompatível com a existência de um Deus bondoso e, ao mesmo tempo, onipotente (ver Teodicéia).
- Paradoxo da pedra: pode Deus criar uma pedra que não consiga levantar?

Paradoxos físicos

- Paradoxo de Braess: algumas vezes, adicionar capacidade extra a uma rede pode aumentar a sua performance geral.
- Paradoxo dos raios cósmicos: a teoria física prevê um limite superior para a energia possível de raios cósmicos, mas energias além do limite já foram observadas.
- Paradoxo de D'Alembert: um líquido sem viscosidade não produz arrasto.
- Paradoxo de Einstein-Podolsky-Rosen: até que distância eventos podem influenciar um ao outro em mecânica quântica?
- Paradoxo de Gibbs: em um gás ideal, é a entropia uma variável extensiva?
- Paradoxo de Loschmidt: por que há um aumento inevitável de entropia quando as leis físicas são invariantes sob reversão temporal?
- Paradoxo de Mpemba: água quente, sob determinadas condições, pode congelar mais rápido do que água gelada, mesmo que tenha que passar pela temperatura mais baixa rumo ao congelamento.

- Paradoxo dos gêmeos: quando um gêmeo que saiu de viagem retorna, ele está mais novo do que o seu irmão que ficou.
- Paradoxo da informação em buracos negros.
- Paradoxo do Cone Duplo: dois cones unidos pela base, soltos numa rampa inclinada, tendem a subir e não a descer, como era esperado.

Paradoxos falsídicos: são os paradoxos que dão resultados incorretos baseados em um raciocínio sutilmente falso.

- Paradoxo de Epiménides: um cretense diz: "Todos os cretenses são mentirosos."(Mas veja também o paradoxo do mentiroso, uma antinomia.)
- Paradoxo dos cavalos: todos os cavalos são da mesma cor.
- Paradoxos de Zeno: quando você chegar ao local onde a tartaruga está, ela já terá avançado um pouco, de modo que você nunca será capaz de alcançá-la.

Paradoxos aritméticos: são provas de coisas absurdas usando aritmética (e errando). Por exemplo, provar que $1 = 2$ escrevendo uma expressão enorme e dividindo por outra expressão que é igual a zero.

- Paradoxo do Burro: "Um burro bom e barato é raro. Tudo que é raro é caro. Um burro bom e barato é caro."

Antinomias: paradoxos que mostram falhas em raciocínio aceito, axiomas ou definições. Note que muitos deles são casos especiais ou adaptações do paradoxo de Russell.

- Paradoxo do barbeiro: o barbeiro que faz a barba de todos os homens que não fazem as próprias barbas e ninguém mais.
- Paradoxo de Berry: qual é o "primeiro número não nomeável com menos de dez palavras"?
- Paradoxo de Curry: "Se eu não estou enganado, o mundo acabará no meio da semana."
- Paradoxo de Grelling-Nelson: a palavra "heterológica", que quer dizer "não aplicável a si mesma", é ela mesma heterológica?
- Paradoxo do mentiroso: "Esta sentença é falsa."

- Paradoxo de Quine: "Resulta em uma falsidade quando anexado a sua própria citação."
- Paradoxo de Russell: há um conjunto de todos os conjuntos que não contêm a si mesmos?
- Paradoxo de São Paulo: uma cidade que nunca dorme, nunca acorda.
- Dilema do Crocodilo: um crocodilo promete ao pai de uma criança não devorá-la caso ele preveja com exatidão o que acontecerá.

Antinomias de definição: são os paradoxos que dependem de definições ambíguas.

- Navio de Teseu / Machado de George Washington: quando cada componente de um navio foi trocado pelo menos uma vez, o navio ainda é o mesmo?
- Paradoxo sorites: em que ponto um monte de areia deixa de ser um monte de areia à medida que eu tiro grãos de areia?
- Paradoxo de Richard.

Paradoxos condicionais: são paradoxos somente se certas premissas especiais são assumidas. Alguns deles mostram que as premissas são falsas ou incompletas, outros caem em classes diferentes de paradoxos.

- Paradoxo de Fermi: se há muitas outras espécies sencientes no universo, onde elas estão? A presença delas não deveria ser óbvia?
- O paradoxo GZK: raios cósmicos de alta energia foram observados e aparentemente violam o limite de Greisen-Zatsepin-Kuzmin, que é uma conseqüência da Relatividade Especial.
- Paradoxo de Jevons: em economia, aumentos de eficiência levam a aumentos ainda maiores em demanda.
- Paradoxo da mera adição: uma população maior vivendo sob condições meramente toleráveis é melhor do que uma população pequena e feliz?
- Paradoxo de Newcomb: como você joga um jogo contra um oponente onisciente?
- Paradoxo niilista: se a verdade não existe, a declaração "a verdade não existe"não é uma verdade, provando-se, portanto, incorreta.

- Paradoxo de Olbers: se o universo é infinito, possuindo um número infinito de estrelas luminosas uniformemente distribuídas, o céu deveria ser inteiramente luminoso porque haveria estrelas em qualquer direção.
- Paradoxo da onipotência: um ser onipotente seria capaz de criar um rocha pesada demais para levantar? Uma força irresistível pode mover um objeto imovível?
- Paradoxo da predestinação: um homem viaja no tempo e engravida sua própria tetravó. O resultado é uma linha de descendentes, incluindo o próprio homem e seus pais. Portanto, a menos que ele faça a viagem temporal, ele nunca existirá.
- Paradoxo de São Petersburgo: pessoas oferecerão somente uma taxa modesta por uma recompensa de valor infinito.

Outros

- Paradoxo de Giffen: é possível que aumentos progressivos no preço do pão resultem em mais pessoas pobres comendo do mesmo?
- Paradoxo da toxina de Kavka: é possível que alguém tenha somente a intenção de beber uma toxina não letal, se a intenção é tudo o que é preciso para conseguir um prêmio?
- Paradoxo de nascimentos de baixo peso: bebês com baixo peso no nascimento têm uma taxa de mortalidade maior. Bebês de mães fumantes têm uma probabilidade maior de nascer com baixo peso. Entretanto, bebês de baixo peso nascidos de mães fumantes tem uma taxa de mortalidade menor do que outros bebês nascidos com peso inferior ao normal.
- Paradoxo da área desaparecida.
- Paradoxo do Pinóquio: seria impossível o personagem Pinóquio dizer “meu nariz vai crescer agora”.
- Paradoxo do Grupo no Facebook: em novembro de 2010, o historiador da ciência português Bruno Almeida sugeriu um grupo dos utilizadores que não pertencem a qualquer grupo no Facebook, em cujo não é possível criar este grupo já que qualquer grupo tem que ter pelo menos um membro, o seu criador. No entanto, o criador de tal grupo não pode ele próprio ser membro do grupo, uma vez que não pode pertencer a qualquer grupo.

- Paradoxo da multa pela infração não cometida: alguém que não tem carta de condução vai conduzir um carro em plena via pública. Esta pessoa senta-se no carro, põe o cinto, põe o carro a trabalhar, engrena uma velocidade e imediatamente no momento em que ia arrancar aparece-lhe um agente da autoridade que lhe pede a carta de condução. Como o nosso indivíduo não tem carta, acaba por apanhar uma multa (por conduzir sem carta) apesar de nunca ter chegado a conduzir.

2.4 Corpus

Um corpus pode significar uma reunião de textos ou documentos sobre um assunto. Coletânea ou conjunto de documentos sobre determinado tema, repertório da obra científica, técnica ou artística de uma pessoa, ou a ela atribuída. Aquilo que registra toda a obra de um autor. Em linguística, são os registros orais que, colhidos no momento da fala, são utilizados para análise.[7]

Quando o primeiro computador servidor "corpus", o *Corpus Brown*, estava sendo criado, no início dos anos 1960, a gramática generativa dominou a linguística, e foi pouca a tolerância para com abordagens de estudos linguísticos que não aderiram ao que era pregado por gramáticos generativos, considerados detentores de práticas linguísticas aceitáveis. Como consequência, os criadores do corpus, Brown, W. Nelson Francis e Henry Kueera apenas atualmente são considerados pioneiros e visionários, e seus esforços para criar um corpus legível por máquina de inglês não foram calorosamente aceitos por muitos membros da comunidade da linguística à época. W. Nelson Francis conta a história de um gramático generativo, líder em seu tempo, que caracterizava a criação do Corpus Brown como "uma empresa inútil e imprudente" porque "a única fonte legítima de conhecimento gramatical" sobre uma língua veio de intuições do falante nativo, e isso não poderia ser obtido a partir de um corpus (FRANCIS, 1992, p. 28). Embora alguns linguistas ainda detenham essa crença, muitos estão agora consideravelmente mais abertos à ideia de usar um banco de dados (corpus) para estudos descritivos e teóricos da linguagem (MEYER, 2002).

2.5 Semântica

Segundo Mariana Rigonatto, no site Mundo Educação[8]

[7] Fontes: https://www.significados.com.br/?s=corpus - Acesso em 09 jul. 2019. https://www.dicionarioinformal.com.br/corpus/ - Acesso em 09 jul. 2019.

[8] Fonte: https://mundoeducacao.bol.uol.com.br/gramatica/relacoes-semanticas-entre-as-palavras.htm - Acesso em 09 jul. 2019.

> A semântica é a área da Linguística que estuda o significado das palavras, das frases, das expressões, etc., em um contexto específico. Assim, algumas vezes, os significados podem receber diferentes interpretações por estarem em situações comunicativas também diferentes. Essa relação entre os diversos significados de uma palavra é chamada de relação de sentido.

A semântica contrapõe-se com frequência à sintaxe, caso em que a primeira se ocupa do que algo significa, enquanto a segunda se debruça sobre as estruturas ou padrões formais do modo como esse algo é expresso. Dependendo da concepção de significado que se tenha, têm-se diferentes semânticas. A semântica formal, a semântica da enunciação ou argumentativa, e a semântica cognitiva, descrevem o mesmo fenômeno, mas com conceitos e enfoques diferentes. [9]

2.6 Conceitos sobre sentidos semânticos

- A denotação é o uso de uma palavra com o seu sentido original, portanto não há sentido semântico.
- A conotação é o uso de uma palavra com um significado diferente do seu original, em um contexto comunicativo que pode criar diferentes relações de sentido entre elas.
- Sinonímia é a relação que ocorre quando diferentes palavras e expressões possuem um sentido semelhante, ou seja, são sinônimas.
- Antonímia é a relação que ocorre quando diferentes palavras e expressões possuem sentidos opostos, ou seja, são antônimas.
- Hiperonímia e Hiponímia – Hiperonímia é a relação que ocorre quando uma palavra possui um sentido que engloba um conjunto de outras palavras com sentido mais específico. As outras palavras que possuem um sentido mais estrito são seus hiponômios. [10]
- Homonímia é a relação entre duas ou mais palavras que, apesar de possuírem significados diferentes, possuem a mesma estrutura fonológica, ou seja, os homônimos.

[9] Fonte: https://pt.wikipedia.org/wiki/Semântica - Acesso em 09 jul. 2019.
[10] Exemplo de hiperonímia é “fruta”, enquanto “banana”, “maçã”, “goiaba” são seus hiponômios.

3 TENDÊNCIAS SEMÂNTICAS

A ideia principal de um artigo científico indica as inclinações do texto a partir dos tipos de semântica presentes e, para tal, é preciso identificar a importância dessa ideia, quais são suas tendências semânticas, e como atribuir indagações sobre ela.

3.1 Indagação sobre a ideia principal a partir dos tipos de semântica

A Metodologia para atribuir uma indagação à ideia principal parte do desenvolvimento de um aplicativo computacional que faça, automaticamente, consultas em mídias de repositórios científicos buscando todos os textos, artigos, livros, periódicos, e publicações disponíveis a partir de palavras-chave. Esse aplicativo de pesquisa difere de outros, pois depois de encontrar todos os textos, os mesmos ficarão armazenados em um banco de dados, assim como seus resumos gerados pelo próprio aplicativo, formando um corpus especifico (SANOKI, 2017).

Ter resumos de natureza como livro, artigo, jornal, auxiliam na tomada de decisão dessa leitura (GIL, 2000).

"A produção de resumos na universidade é uma das maneiras através das quais o estudante, além de registrar sua leitura de textos acadêmicos, manifesta sua compreensão de conceitos e do saber fazer em sua área de conhecimento."(MATENCIO, 2002).

3.2 Indagação sobre a ideia principal com conceito alargado

No mundo atual, normalmente se faz pesquisa de forma linear ou indexada, porém verticalmente, ou seja, de cima para baixo, ou pontualmente. Qualquer que seja o gerador de um aplicativo, hoje ele trabalha de forma hierárquica. O que se intenta é montar um aplicativo com visão de lateralidade, que se dará com a escolha da opção que se quer deslocar.

Pesquisar envolve a realização de um processo manual repetitivo que é a busca de textos sobre determinado tema a partir de palavras-chave, em bibliotecas, virtuais ou não.

O desenvolvimento de um aplicativo computacional que execute os comandos mecânicos é necessário para a demonstração do algoritmo e de como ele funciona, obedecendo a critérios específicos para tal aplicativo, para que então se atinja o objetivo que é entregar um texto científico ou acadêmico que se encontrará em uma mídia eletrônica, sendo decomposto em palavras, estas classificadas de acordo com o sentido gramatical.

Este é um processo definido, o algoritmo caminha um passo após o outro, conforme determinado. Assim, se outrem segue a mesma sequência de passos, alcançará o mesmo objetivo. Dessa forma também se dá o processo para obtenção de resultado quando se adquire outro artigo.

Nessa abordagem, apenas o algoritmo não é o suficiente para a autoexecução de todos os passos descritos, por isso ele contará com o apoio de uma base de dados que contenha as mesmas características do texto a ser analisado e demonstrado.

Uma vez que o algoritmo tem acesso às características em comum entre o texto e a base de dados, é possível construir uma relação que possibilite a geração de questões.

Na redação dos Objetivos Específicos empregam-se verbos com menos interpretações ou de sentido fechado. Por exemplo: adquirir, aplicar, apontar, classificar, comparar, conceituar, caracterizar, enumerar, reconhecer, formular, enunciar, diferenciar, mobilizar, coletar, etc.

A identificação da ideia principal revela as ideias secundárias que a suportam, e com os Objetivos Específicos isso é alcançável em menor tempo e explicita desempenhos observáveis, operacionalizando o Objetivo Principal. Portanto o Objetivo Principal e os Específicos estão inter-relacionados entre si, bem como, com o tema do artigo e com o que o pesquisador escreveu.

4 INOVAÇÃO DA TESE

4.1 Inovação da tese

O algoritmo, quando aplicado a um texto acadêmico através de um mecanismo computacional, apontará, em outro texto digital resultante da delineação da estrutura de um artigo, qual parte do reconhecimento do título vem acompanhada de um tema. Esse tema então será denominado objeto desse texto e, sendo um objeto, será seguido de características que o descrevem e que são descritas como propriedades do mesmo. Essas propriedades podem ser vistas no texto original como o item Introdução e, parcialmente, também no item Metodologia.

A partir do item Metodologia obtém-se a descrição da função e das ações do artigo em análise, tornando assim possível descobrir quais movimentos esse objeto exerce e como cada função executa os processos quando invocada por outro objeto ou função. A descrição desse algoritmo segue uma estrutura sintática interna das expressões nominais no português do Brasil (GONZAGA, 1997).

O algoritmo desta tese, em linhas gerais, utiliza-se de escritas gramaticais para o reconhecimento de sequenciadores de períodos gramaticais, tais como: conjunções, operadores lógicos e argumentativos (MARTINO, 2013).

Porém, no artigo *"Argumentative scheme in academic texts: Algorithmic for validating generated questions"*, conclui-se que nem todos os períodos possuem tais sequenciadores (SANOKI; VEGA, 2018a). Toda essa forma de reconhecimento gramatical está descrita no artigo sobre Esquema Argumentativo (JULIA, 2014). Outra verificação a ser realizada é de que as palavras-chave não são localizadas em sua totalidade num texto a ponto de demonstrar o número de vezes em que se repetem nas partes de Introdução, Metodologia, e Resultado.

E, o mais importante, espera-se que os argumentos vindos do resumo estejam coerentes com as demais partes, principalmente com a introdução e a metodologia do texto acadêmico.

4.2 Abordagem

O ponto inicial da escrita é o tema ou o título que descreve o objeto, e isso é essencial, sendo inclusive o objeto principal deste artigo. O tema mostra se o objeto é concreto e palpável, ou filosófico, ou ainda virtual, uma vez que é necessário que haja um processo computacional que corrobore a existência desse algoritmo. Isso se dá com a execução de todos os passos de forma virtual, não visível ou física.

Nesse processo de identificação do tema como objeto é preciso descrever o que ele é, ou seja, descrever suas qualidades, quantificação, existência real ou virtual, e como ele existe no mundo real. Essa descrição pode ser definida como a propriedade desse objeto.

Com a localização de tipos de ações e, consequentemente, do argumento em si, o algoritmo fará uso de um aplicativo computacional que buscará os períodos armazenados em um banco de dados relacional (CODD, 1970).

A pesquisa segue uma forma descritiva e exploratória para abordar como o processo de mecanização é aceito como uma metodologia alternativa em projetos desenvolvidos, a princípio, de forma tradicional.

Neste tema, segundo Kermen e outros (1976), na tarefa entre o início e o término, pode-se construir uma estrutura de dados chamada de “mapa trapezoidal”, e, com o uso dessa estrutura acha-se um caminho de tarefas sem restrições ou obstáculos que impeça uma execução livre, ou sem que se tenha que recorrer a alternativas de “ajustes”.

Para atender ao objetivo deste projeto, os procedimentos da descrição do mesmo seguem estabelecidos pela Associação Brasileira de Normas Técnicas – ABNT. Pelas normas NBR 14724: trabalhos acadêmicos – apresentação - NBR 10520: informação e documentação - apresentação de citação de documentos - NBR 6023: informação e documentação – referências, elaboração.

A metodologia descreve de forma clara o tipo de pesquisa que será realizada. O pesquisador estará respondendo às perguntas: Com quem? Onde, quando e como irei realizar o projeto de pesquisa? Quais os instrumentos selecionados para a coleta de dados? Como registrarei e apresentarei os dados coletados?

5 DESENVOLVIMENTO DO SOFTWARE

Software para execução de algoritmo através de aplicativo para decompor um texto e gerar indagações que auxiliarão a identificação mais rápida e precisa das ideias de tal texto.

5.1 Utilização de Aplicativo Computacional

O algoritmo utilizará um mecanismo de coleta de textos em repositórios acadêmicos disponibilizados de forma manual e escolhida pelo interessado, conforme descrito no Apêndice 10 - fonte em C#:

```
System.Windows.Forms.OpenFileDialog();
```

tendo como finalidade criar uma base de dados que será chamada de *Corpus* Bibliográficos(BEHR; MORO; ESTABEL, 2008) e conterá todos os textos correlatos presentes na Referência Bibliográfica do texto principal. Esse "repositório particular" indicará se os textos acadêmicos encontrados a partir das palavras-chave em outros periódicos ou repositórios são pertinentes à sua pesquisa. Terminada a fase da coleta de textos, será acionada a etapa que as palavras-chave informadas serão relacionadas com o *Corpus*.

Neste *Corpus* estarão todos os termos relacionados às palavras-chave estabelecidas, e que serão classificados segundo os Tipos de Indagações de Compreensão (MARTINO, 2013) descritos para gerar questionários sem a inferência semântica, podendo ajudar a sugerir materiais escritos para apresentação da sua monografia, dissertação, ou tese, assim como para a demonstração de uma afirmação ou de uma argumentação contrária, isso a partir de um documento em conformidade com os passos exigidos pela Academia (SANOKI, 2017).

Alternativas para isso são justamente a reunião e a organização dos textos em uma infraestrutura em meio digital, com o dimensionamento do seu espaço, a estruturação de um banco de dados que suportará o corpus de textos, e a organização da forma de armazenamento de dados. E, nesse algoritmo em particular, isso se inicia com a montagem do corpus de textos, que por sua vez depende da obtenção de todos os textos das áreas específicas nos repositórios acadêmicos disponíveis em endereços eletrônicos (*Word Wide*

Web).

Cada texto então será particionado em períodos gramaticais e frases classificadas gramaticalmente para o armazenamento, este podendo ser opcional, o que viabilizará o posterior processo de montagem de períodos simples, compostos, ou subordinados, e o uso pelo algoritmo.

No artigo *"Generating key questions from academic texts"* é descrita uma estratégia em várias etapas (SANOKI, 2017). Na primeira etapa do algoritmo há interação com o pesquisador, que deverá informar o texto e o objetivo da sua pesquisa através de palavras-chave. A segunda etapa será a inserção desse texto em um meio computacional (caso ele ainda não esteja inserido) e, a partir dos itens da Referência Bibliográfica, serão procurados artigos referenciados e correlatos em periódicos ou repositórios disponíveis para que, então, se crie uma base de dados que se chamará Corpus Bibliográfico e que dará suporte ao próprio algoritmo.

Para conectar as indagações geradas no Resumo a outras partes, sempre tendo como base as conjunções ou os operadores lógicos e argumentativos contidos nos sequencializadores (MARTINO, 2013), é preciso que o Resumo seja também particionado em palavras, e que cada uma delas seja rotulada em sua classe gramatical (SANOKI; VEGA, 2017). Tem-se a possibilidade de classificar os tipos de períodos gramaticais existentes em tais textos, tanto simples quanto rebuscados, partindo de algumas premissas (SANOKI; VEGA, 2018a).

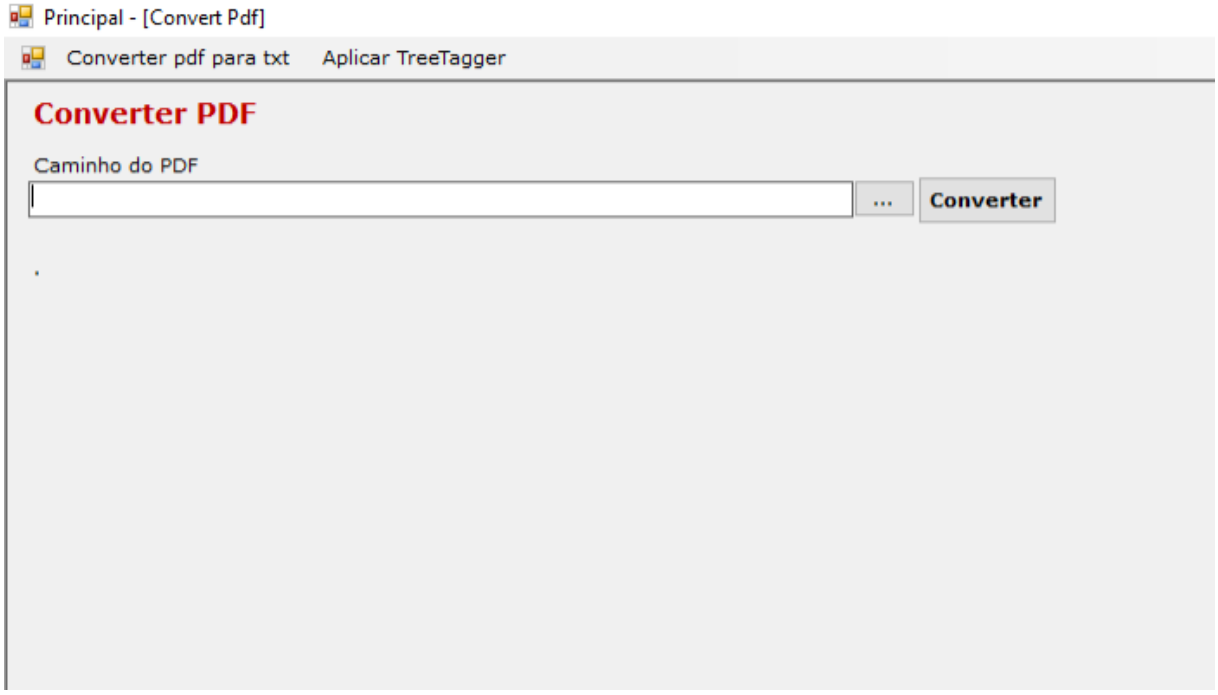
5.2 Algoritmo Gerador

Quando se tem o texto, é possível executar o algoritmo que gerará várias indagações a partir das classes gramaticais.

O processo começa a partir dos artigos científicos ou acadêmicos escolhidos que passam pela etapa de pré-processamento, na qual são preparados para o processo computacional, utilizando-se a técnica case folding (WITTEN et al., 1994) , que coloca todas as letras em caixa baixa (minúsculas), além de outros cuidados, como o descarte de todas as figuras, tabelas e marcações existentes. Esses artigos, para a fase de pré-processamento, devem estar no formato Portable Document Format (Formato Portátil de Documento), conhecido como PDF, como informado nas Figuras 1 e 2, vide Apêndice 10 – fonte em C# :

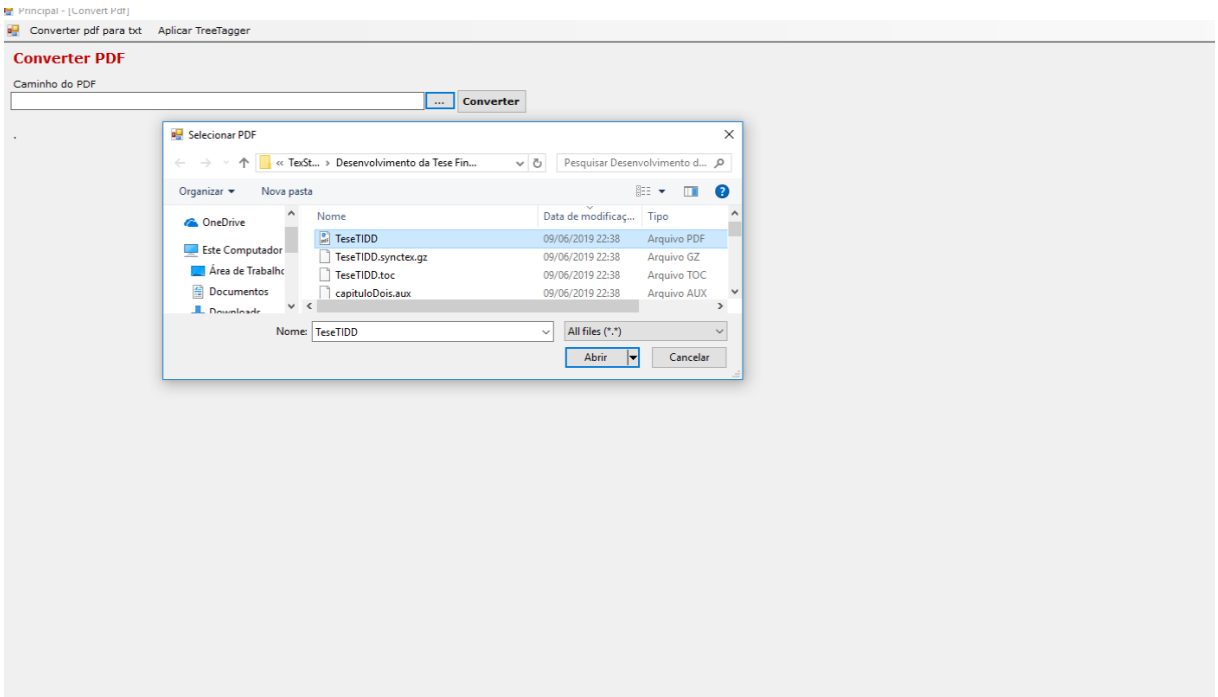
```
comando nessa fonte:

pdf.ConvertePdfEmTexto

(txtCaminhoPDF.Text, txtCaminhoPDF.Tag.ToString());
```

Figura 1 – Tela que Informa o diretório PDF



Fonte: C:/ProjetoTreeTagger

Figura 2 – Tela que escolhe PDF para converter em texto



Fonte: C:/ProjetoTreeTagger

que passam pela etapa de pré-processamento, na qual são preparados para o processo computacional, utilizando-se a técnica case folding (WITTEN et al., 1994), que coloca todas as letras em minúsculas, além de outros cuidados, como descarte de todas as figuras,

tabelas e marcações existentes.

Após os descartes, ficam somente os alfabetos, números, e alguns caracteres especiais, e esses textos encontram-se então em formato compatível para serem interpretados. Nessa etapa, é usado o processo de sumarização, cuja finalidade é diminuir o número de palavras, viabilizando o processamento.

Com a sumarização, obtém-se a parte mais importante, ou seja, a ideia principal do texto fonte, através da criação de um resumo com as palavras mais significativas. O sumário, além de ser mais conciso do que o texto fonte, tem um número muito menor de atributos. Essa redução possibilita o uso de um espaço amostral que consegue atenuar a questão da alta dimensionalidade e dos dados esparsos.

A sumarização também consegue viabilizar a permanência das stop words, o que possibilita que o modelo Cassiopeia seja independente do idioma (GUELPELI, 2012). Terminada a etapa de pré-processamento, começa a de processamento, que faz uso do agrupamento de textos hierárquicos, e um algoritmo para juntar os textos por similaridade.

Os agrupamentos criados nesta etapa têm um vetor de palavras que representam os centroides desses e que são de alta relevância para cada agrupamento, além de pertinentes em relação à frequência média das palavras nos textos agrupados.

À medida que novos textos são agrupados ocorre o reagrupamento, podendo surgir agrupamentos, subagrupamentos ou até mesmo a fusão desses (LOH, 2001).

5.3 Softwares

Antes de utilizar o software deve-se também saber qual é o conceito do texto a ser analisado (Lopes Pinheiro; ANDRÉA; PEREIRA, 2012). A utilização de um software que faz análise de dados textuais auxilia na produção de uma lista de sintaxes, e será feita de acordo com os perfis indicados pelo algoritmo (CAMARGO; JUSTO, 2013).

5.3.1 Software resumidor

O software *Intellexer Summarizer* tem uma resposta a apresentar quando se trata de sumarização. Sumarização Automática de Textos é o ramo da área de recuperação de informação que utiliza técnicas e algoritmos para gerar ou identificar e coletar sentenças relevantes a partir de documentos textuais. Claramente, o uso de Processamento de Linguagem Natural (PLN) revela-se benéfico ao processo de sumarização, principalmente quando se processam documentos sem nenhuma estrutura ou padrão definido (CABRAL et al., 2015).

5.3.2 Software de estatística

Software *"Interface de R pour les Analyses Multidimensionnelles de Textes et de Questionnaire"* (IRAMUTEQ), desenvolvido por Pierre Ratinaud (2009), interagindo com o software R (uma linguagem e ambiente para computação estatística e gráficos da *"The R Foundation for Statistical Computaing"* – Viena, Áustria), que alcança diferentes análises estatísticas sobre corpus textuais.

Camargo conclui:

> IRAMUTEQ pode trazer importantes contribuições aos estudos que envolvam dados textuais. O processamento de dados permitido pelo software viabiliza o aprimoramento das análises, inclusive em grandes volumes de texto (pesquisa o vocabulário e reduz as palavras com base em suas raízes, técnica chamada de lematização). Pode-se fazer uso das análises lexicais sem que se perca o contexto em que a palavra aparece, tornando possível a integração de níveis quantitativos e qualitativos à análise, trazendo maior objetividade e avanços às interpretações dos dados de texto (CAMARGO; JUSTO, 2013).

5.3.3 Software que determina a classe gramatical

Software "TreeTagger" – desenvolvido por Laurent Pointal, de uma organização CNRS - LIMSI com direitos de cópia CNRS - 2004-2019, sob licença GNU- GLP – versão 4 ou mais, onde se utiliza a versão 2.3 para buscar os verbos e palavras com sentidos semânticos que serão aplicados nesta tese (SCHMID; HELMUT, 2009).

6 DESCONSTRUÇÃO DO TEXTO

Como funciona o processo de decomposição do texto em palavras e termos.

6.1 Indagações 5W2H (pronomes interrogativos)

Será utilizada a técnica 5W2H (ENCARNATION, 1999) para gerar indagações, pois esta é uma ferramenta que consiste em estruturar o pensamento de forma bem organizada e materializada antes que se efetue uma tarefa, e, nesse contexto, o motivo é a realização da leitura de um texto. O "5W" corresponde às palavras em inglês para fazer perguntas: *who*, *what*, *when*, *where* e *why* (que, o que, quando, onde, por que, respectivamente), e o "2H" às palavras *how* e *how much* (como e quanto). A utilização dessa ferramenta de indagação objetiva gerar perguntas sobre a ideia principal com uso de uma, ou todas, 5W2H.

Essa técnica, da área de humanas, requer a combinação dos pronomes interrogativos e advérbios interrogativos utilizados na construção de uma pergunta. As definições de pronome interrogativo e de advérbio interrogativo estão de acordo com Avram Noam Chomsky, um linguista, filósofo, ativista, autor e analista político estadunidense, cuja tese de doutorado foi sobre a análise transformacional (1955), elaborada a partir das teorias de Z. Harris, de quem foi discípulo (CHOMSKY, 1964).

6.2 Identificação das palavras no texto

Com a utilização do software "TreeTagger" o texto será em separado em palavras, e com elas já identificadas em sua classe gramatical, o algoritmo efetuará então o processo de montagem de períodos simples, compostos ou subordinados. Após isso há o armazenamento das palavras e períodos gramaticais, identificados com ID, em outro item do Corpus de Indagações, um banco de dados estruturado em linhas e coluna (CODD, 1970).

A localização dos períodos em artigos apresentados em sua forma física, ou seja, impressos em papel, dar-se-ia com o ato de copiar cada sentença em outro papel, mas como

normalmente busca-se o artigo em repositórios universitários que estão digitalizados, os textos serão separados em palavras com a utilização de um aplicativo computacional com uma função específica para tal (CODD, 1970).

Pronomes relativos sempre têm as palavras que, qual, quem, quanto, ou cujo onde se relacionam com o termo (nome) da oração anterior, substituindo os que se repetem. Observando a regência, requerem preposição, que deve ser colocada obrigatoriamente antes do pronome. Pronomes interrogativos são quando as palavras que, qual, quem, quanto, ou cujo não fazem referência ao termo (nome) da oração anterior, independente de a oração ser ou não terminada com um símbolo de interrogação [1].

Das associações entre os períodos e os operadores argumentativos da Tabela 1, se obtêm a extração automática dos termos com ações (SANOKI; VEGA, 2017), o que irá determinar os termos mais importantes de um texto pela frequência adjacente das palavras que o compõem. Com a consequente localização das ações, passa-se para a remontagem das orações, como descrito na Figura 3. Com os períodos remontados torna-se possível a construção de um esquema para o desenvolvimento de questionamentos sobre as ações utilizadas. Inicialmente, será utilizada a técnica 5W2H na construção de questões.

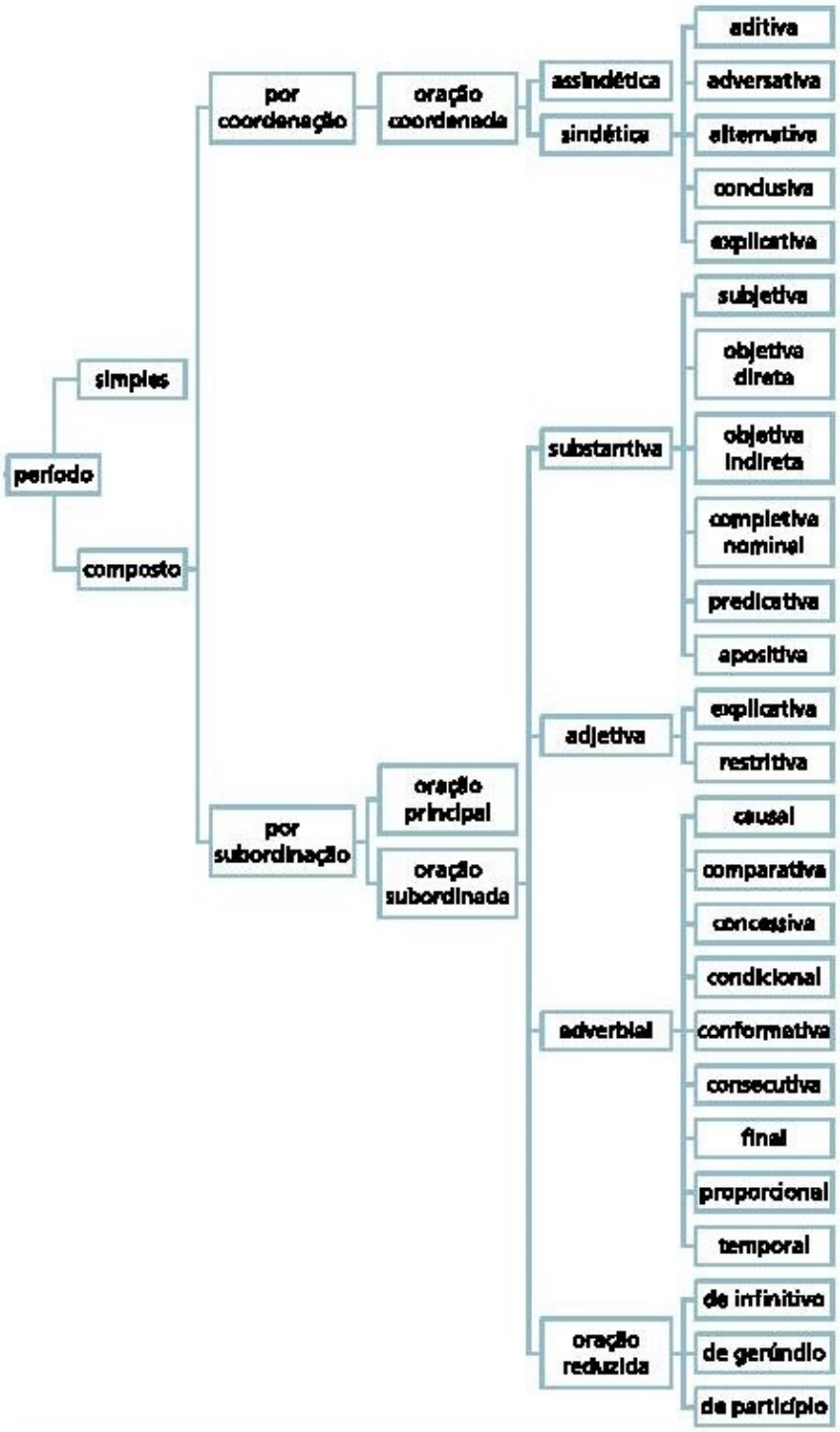
[1] Fonte: https://www.significados.com.br/pronome/ – Acesso em 09 jul. 2019

Tabela 1 – Sentidos semânticos - Sinônimos

	Sinônimos
Importante	significativo, grave, sério, considerável, respeitável, interessante, relevante, marcante, notável, superior, valoroso, solene, meritório, elogiável, momentoso, ponderoso, vultoso, indispensável, imprescindível, essencial, fundamental, básico, crucial, primordial, principal, elementar, necessário, preciso, determinante, útil, conveniente, pertinente, oportuno, proveitoso. .
Necessidade	imposição, obrigação, inevitabilidade, indispensabilidade, imprescindibilidade, primordialidade, conveniência, utilidade, dever, urgência, premência, precisão, instância, emergência, exigência, aperto, ânsia, vontade, desejo, mister, apuro.
Objetivo	propósito, fim, finalidade, objeto, alvo, meta, destino, intenção, intuito, intento, tenção, escopo, fito, mira, desígnio, querer, sonho, imparcial, isento, neutro, justo, apartidário, desapaixonado, reto, equânime, neutral.
Processo	método, procedimento, técnica, maneira, metodologia, modo, meio, recurso, norma, sistema, processamento, ordem, regime, desenvolvimento, andamento, evolução, progresso, crescimento, marcha, movimento.
Conhecimento	percepção, lucidez, compreensão, cognição, discernimento, entendimento, apreensão, razão, clareza, informação, informe, notícia, aprendizagens, alicerces, rudimentos, fundamentos, noções, base.
Analisar	investigar, pesquisar, averiguar, explorar, sondar, indagar, inquirir, verificar, ler, ver, observar, examinar, considerar, estudar, ponderar, refletir, pensar, pesar, criticar, comentar, julgar, apreciar, avaliar, aferir, dissecar, decompor, esmiuçar, esquadrinhar, minuciar.
Consequente	portanto, assim, assim sendo, por conseguinte, isto posto, como resultado, de modo consequente, em consequência.

Fonte: https://mundoeducacao.bol.uol.com.br/gramatica/relacoes-semanticas-entre-as-palavras.htm

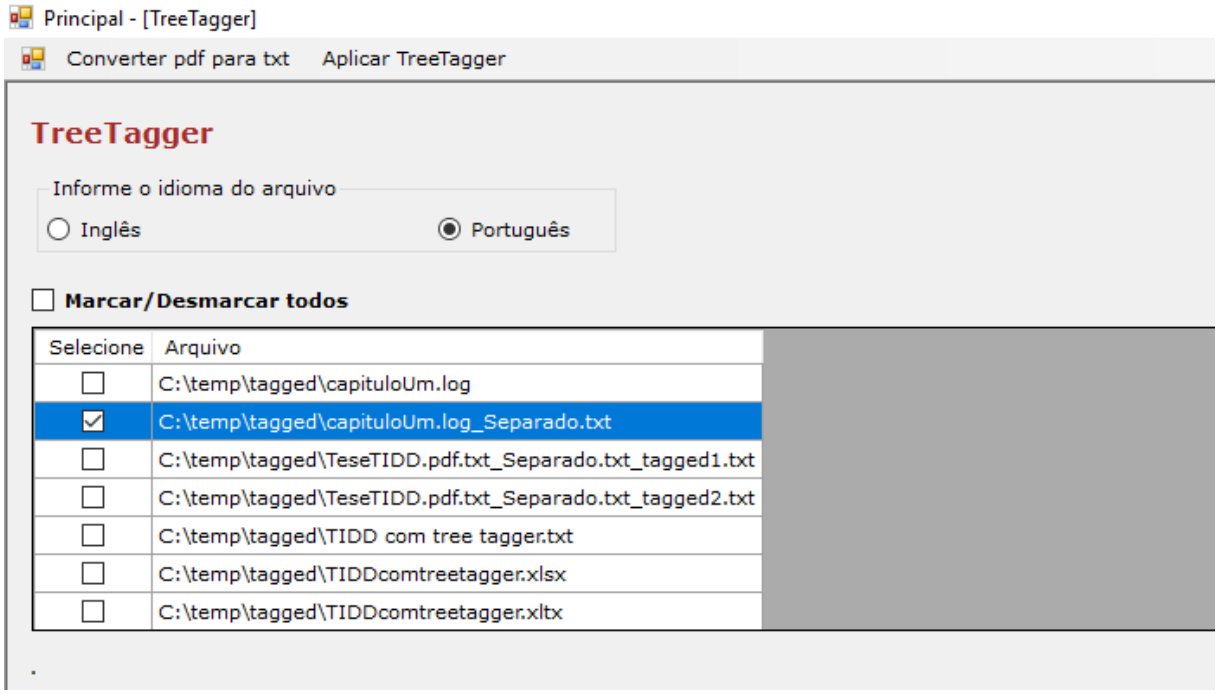
Figura 3 – Períodos simples e compostos



Fonte: https://www.significados.com.br/classe-gramatical

A utilização desse software acontece a partir de um aplicativo computacional desenvolvido para dar suporte a esta tese, e que vincula a escolha do texto à mídia computacional onde automaticamente as palavras do texto serão separadas para a realização de uma "etiquetagem", que é a atribuição da classe gramatical a cada uma delas.Então será feita a escolha dos verbos que darão indícios de como formular as indagações. A etiquetagem também é realizada pelo aplicativo "TreeTagger", conforme Figura 4 - vide Apêndice C – COMO EXECUTAR O SOFTWARE TreeTagger: –, que ao atribuir uma classe gramatical a cada palavra, confere a ela característica ou função morfológica e semântica. A escolha da classe gramatical "verbo" (vide Tabela 2), de maior incidência, conjugada a outras classes gramaticais e associada à técnica 5W2H (pronomes interrogativos), dará indícios de como formular as indagações.

Figura 4 – Tela que aplica no TreeTagger



Fonte: C:/ProjetoTreeTagger

Tabela 2 – Cclasses gramaticais

Classe	**Função ou Característica**
Substantivo	Palavras que nomeiam as coisas, os seres, os lugares. Podem ser flexionados em "gênero", "número"e "grau".
Verbo	Palavras que indicam uma ação, estado, ocorrência ou um fenômeno. Podem ser flexionados em "número", "voz", "pessoa", "modo", "tempo"e "aspecto".
Adjetivo	Palavras que caracterizam ou qualificam os substantivos. Podem ser flexionados em "gênero", "número"e "grau".
Advérbio	Palavras que alteram um verbo, um adjetivo ou outro advérbio. São invariáveis, mas alguns advérbios podem ser flexionados em "grau".
Pronome	Palavras que substituem, acompanham, determinam ou modificam alguns substantivos nas frases. Podem ser flexionados em "gênero", "número"e "pessoa".
Preposição	Palavras que estabelecem conexões entre dois termos de uma oração.
Artigo	Palavras que antecedem os substantivos, definindo ou não os mesmos. Podem ser flexionados em "gênero"e "número".
Interjeição	Palavras que exprimem emoções, sensações ou estados de espírito. São invariáveis.
Conjunção	Palavras que servem de elementos de ligação entre duas orações ou termos de uma mesma oração. São invariáveis.
Numeral	Palavras que indicam quantidades, sejam de pessoas, coisas e etc. Podem ser flexionados em "gênero"e "número".

Fonte: https://www.significados.com.br/classe-gramatical/

Considerando que todos os textos acadêmicos possuem título e palavras-chave, depois de já "etiquetados" há como montar indagações em consonância com o "Lembrar", dos Tópicos Adequados para Classificação das Indagações, e com os Tipos de Indagações de Compreensão, descritos nos tópicos 6.6 e 6.7 desta tese, respectivamente. As indagações serão montadas logo após ter sido realizada a etiquetagem.

Já o item Resumo de um texto acadêmico tem outra conotação, pois envolve o "Entender", dos Tópicos Adequados para Classificação das Indagações, e os Tipos de Indagações de Compreensão.

O item Referências Bibliográficas é a indicação da utilização de ideias de outros autores, ali apontados. Para poder criar relações com as ideias do autor referenciado no texto acadêmico, se fazem necessários os tópicos "Entender", "Analisar" e "Avaliar", dos Tópicos Adequados para Classificação das Indagações, para então gerar os Tipos de Questões de Compreensão.

Pode-se contar as palavras e atribuir a maior incidência a partir do software que gera um relatório estatístico sobre um texto, e isso é realizado por um aplicativo on-line , conforme Figura 5.

Figura 5 – tela que aplica na linguística

Fonte: http://linguistica.insite.com.br/corpus.php

6.3 Identificação dos tipos de características nos períodos

Em uma estruturação orientada a objetos, as características são as classes gramaticais das palavras existentes nos períodos. Gonzaga faz alguns apontamentos em relação às características. De acordo com ele, os primeiros advérbios detectados num texto ocupam uma posição dependente do verbo que os seleciona. Seguindo o raciocínio de que o advérbio modifica o adjetivo quando faz definições no período, haveria uma modificação dessa característica, passando de um adjetivo para um advérbio, verbo ou substantivo. Outros advérbios adjuntos podem respeitar algumas restrições, mas não ocupam um lugar na frase temática do verbo, por isso dispõem de uma mobilidade grande que constitui, de certo modo, o objeto de análise do trabalho do autor (e desta tese também).

O segundo ponto que o autor apresenta é uma distinção que é essencialmente sintática e consiste em separar, dentro do grupo dos adjuntos, os advérbios de verbo principal dos advérbios de frase. O fator de separação dessas duas classes incide basicamente

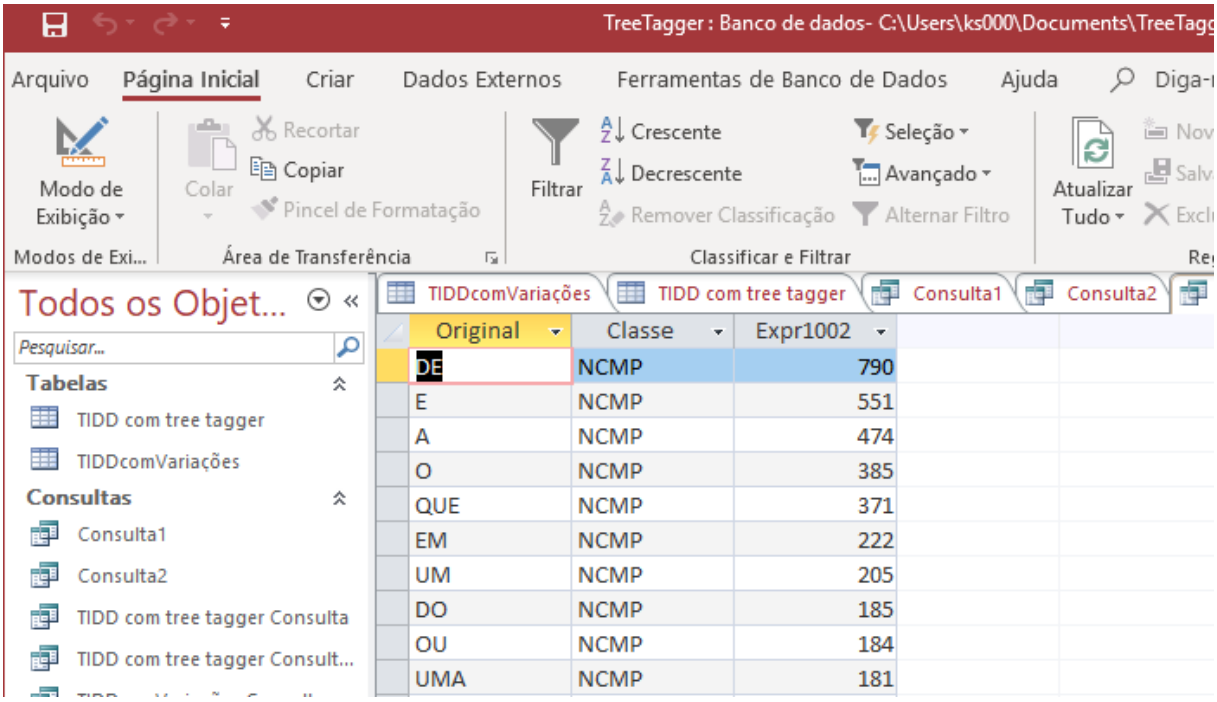
sobre o seu movimento e interpretação. Assim, pode-se observar que quando há movimento, este está caracterizando uma função ou ação que o autor descreve sobre o tema. Os advérbios de frase ocupam posições cujo alcance é toda a frase, condicionando também a interpretação da mesma. Os advérbios do verbo principal ocupam, tipicamente, posições no âmbito do verbo principal, o que, em termos de interpretação, pode condicionar apenas o verbo (enquanto cabeça da projeção), ou qualquer categoria ou elemento gerado dentro da oração (sujeito e complemento, nomeadamente).

O terceiro ponto é uma classificação fundamentalmente semântica, onde se define um "estudo do significado" (VOLPATO, 2015). Gonzaga considera a distinção entre advérbio de frase e de oração, e dentro de cada uma das duas classes encontra subclasses semânticas. Assim, subdivide os advérbios de frase em quatro modais: advérbios orientados para o sujeito, advérbios de tempo, de lugar e advérbios especulais (que determinam a realidade). E os advérbios da oração são subdivididos em advérbios orientados para o verbo, orientados para o objeto, orientados para o sujeito, e especulais/quantificadores (GONZAGA, 1997).

Na tentativa de definir o estatuto lexical, categorial e funcional dos advérbios, Gonzaga defende que os mesmos constituem uma categoria com a capacidade de projetar uma categoria máxima, embora, dado o estatuto de operador, a maioria dos advérbios não possa ser subcategoria (BARROS, 2013).

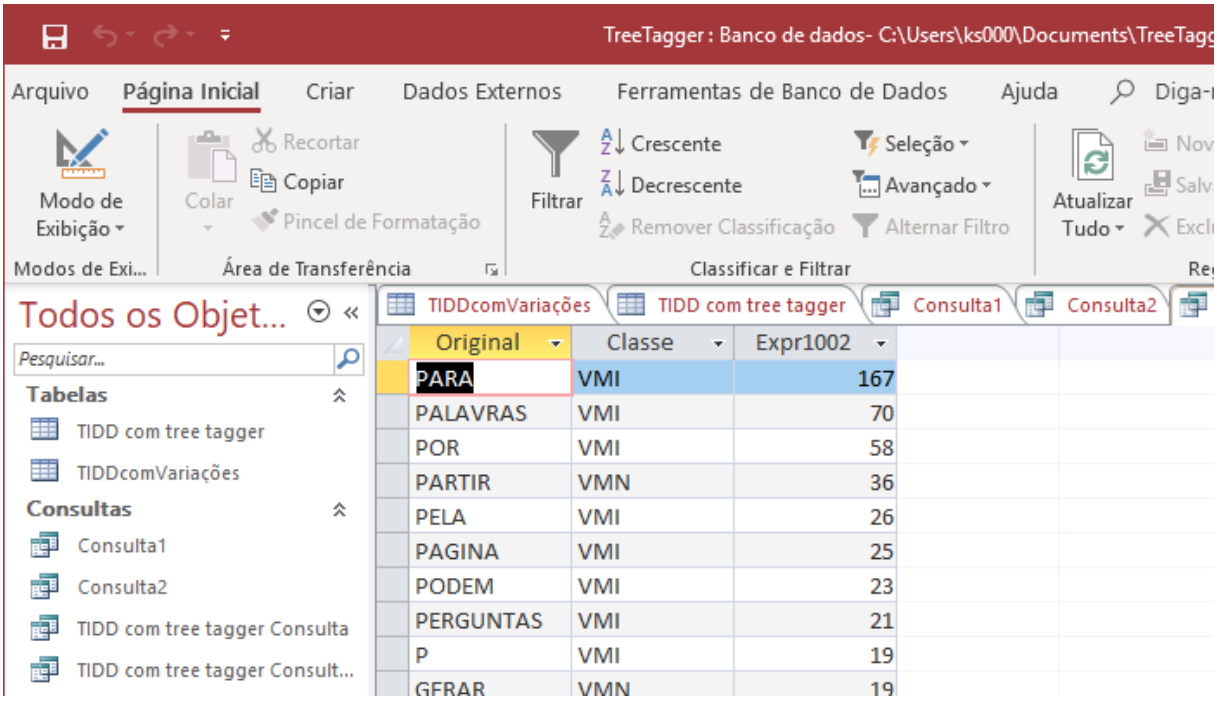
Obtendo o resultado da análise dos tipos de períodos no texto e a etiquetagem das classes gramaticais conforme a Figura 6, assim como as classes de indagações e ações de interesse (verbos), como aparece na Figura 7, o algoritmo acrescentará indagações aos períodos, independente de quais sejam as sentenças (LIU, 2012).

Figura 6 – Tela do Accces da classes das palavras mais utilizadas



Fonte: C:/ProjetoTreeTagger

Figura 7 – Tela do Access de classes especiais das palavras utilizadas



Fonte: C:/ProjetoTreeTagger

6.4 Associações dos advérbios e pronomes no *corpus*

O alcance da composição do *Corpus* tem três dimensões. A primeira é o número de palavras, uma medida da representatividade do *Corpus*, no sentido de que quanto maior o número de palavras, maior será a chance do mesmo contê-las na baixa frequência que forma a maioria das palavras de um idioma.

A segunda dimensão é o número de textos a que se aplica a *Corpus* específicos. Um número maior garante que o tipo textual, gênero, ou registro, estejam representados de forma mais adequada.

A terceira é o número de gêneros, registros ou tipos textuais criados para representar um idioma como todo, e a esta dimensão se aplica a *Corpus* variados. Aqui, um número maior de textos com vários tipos permite uma maior abrangência do espectro genérico do idioma (SARDINHA, 2000).

6.5 Indagações geradas

Após a etiquetagem e a escolha dos pronomes demonstrativos que indicam a posição de algo, situado no espaço e tempo: este, isso, aquilo... (LOPES-ROSSI, 1996), são feitas perguntas que darão indícios de como formular indagações (SANOKI, 2017).

Os períodos gramaticais, por si só, muitas vezes não são o suficiente para gerar indagações, e para isso o algoritmo contará com o apoio de um Corpus Bibliográfico que contenha as mesmas características do texto em questão.

As indagações serão iniciadas com os "pronomes interrogativos (qual, o que, como, porque, para que, como) ou de tipos de argumentos + o primeiro verbo do período + o predicado que precede o verbo". A composição das indagações tem como base as ideias principais encontradas, portanto, se elas não forem encontradas, serão mostradas apenas as definições.

Os resultados são uma série de indagações, consoantes, ou não, com a descrição sintática da estrutura interna das expressões nominais no português do Brasil (DAVID, 2007). (SANOKI; VEGA, 2018b).

6.6 Tópicos adequados para classificação das indagações

As indagações também levam em consideração a estrutura do processo cognitivo na taxonomia de Bloom (FERRAZ; BELHOT et al., 2010), da mais fácil à mais elaborada, que é Lembrar, Entender, Aplicar, Analisar, Avaliar e Criar.

A taxonomia origina-se de uma palavra grega que significa ordenação. A taxonomia de Bloom, embora formulada na década de 50, tem sido revisitada por pesquisadores que reconhecem nela mais do que uma ferramenta para a avaliação do processo ensino/aprendizagem, mas um instrumento útil e eficaz no planejamento e implementação de aulas, na organização e criação de estratégias de ensino. Níveis da taxonomia revisada e seus respectivos verbos:

- Lembrar – Recuperar conhecimento relevante da memória de longo termo, ou reconhecer informações, ideias e princípios de maneira aproximada do que foi aprendido.
- Entender – Construção de significados através de linguagem oral, escrita ou gráfica, usando para isso a interpretação, exemplificação, classificação, sumarização, inferência e explicação, com base em um conhecimento prévio.
- Aplicar – Aplicar, computar, demonstrar, manipular, modificar, produzir, resolver, selecionar, transferir e utilizar princípios para resolver o problema ou tarefa, com um mínimo de supervisão.
- Analisar – Distinguir, classificar e relacionar pressupostos, hipóteses, evidências ou estruturas de uma declaração ou questão.
- Avaliar – A avaliação pode ser definida como a realização de julgamentos baseados em critérios e padrões específicos.
- Criar – O ato de juntar elementos para formar um todo coerente e funcional, integrando e combinando ideias num produto, plano ou proposta nova.

6.7 Tipos de indagações de compreensão

A montagem se baseia em um ou mais tipos de indagações para identificação de relações que possam existir entre os termos. A utilização de um aplicativo computacional se torna necessária para classificar tais tipos de questões, que serão formuladas, como (MARTINO, 2013):

- Questões Óbvias: são perguntas não muito frequentes e de perspicácia mínima, respondidas pela própria formulação. Assemelham-se às indagações do tipo: “Qual a cor do cavalo branco de Napoleão?”.
- Questões Cópias: são perguntas que sugerem atividades mecânicas de transcrição de frases ou palavras. Verbos frequentes aqui são: copie, retire, aponte, indique, transcreva, complete, assinale, identifique.

- Questões Objetivas: são perguntas sobre conteúdos objetivamente inscritos no texto (o quê, quando, por que, onde, como, quem e quanto) numa atividade de pura decodificação.

- Questões Inferenciais: estas perguntas são as mais complexas. Exigem, entre outros, conhecimentos textuais, sejam eles pessoais, contextuais, ou enciclopédicos, bem como exigem regras inferenciais e análise crítica para a busca de respostas.

- Questões Globais: são perguntas que levam em conta o texto como um todo e aspectos extratextuais, envolvendo processos inferenciais complexos.

- Questões Subjetivas: estas perguntas em geral têm a ver com o texto de maneira apenas superficial, sendo que a resposta fica por conta do pesquisador e não há como testá-la em sua validade.

- Questões Vale-tudo: são perguntas sobre questões que admitem qualquer resposta, não havendo possibilidade de equívoco. A ligação com o texto é apenas um pretexto sem base alguma para a resposta.

- Questões Impossíveis: estas perguntas exigem conhecimentos externos ao texto e só podem ser respondidas com base em conhecimentos enciclopédicos. São questões antípodas às de cópia e às objetivas.

- Questões Metalinguísticas: são perguntas sobre indagações formais, geralmente da estrutura do texto ou do léxico, bem como de partes textuais.

6.8 Noção de ações descritivas

Segundo Márcia Cançado (1978), "a noção de argumento tem sua origem na lógica de predicados, em que uma constituinte central, o predicado, que não tem seu sentido completo, ou seja, insaturado, pede um determinado número de ações ou características que lhe completem ou saturem o sentido, usando o termo proposto pelo lógico Gottlieb Frege".

Nesse sentido, de forma semântica, a noção de argumento de Márcia será a descrição da ação na sentença, ou a justificativa que o autor defende em sua descritiva.

6.8.1 Tipos de ações

São os parâmetros necessários para detecção de ações ou funções de uma sentença, e tem como base a montagem de um algoritmo de indagações específicas:

- Ações de autoridade: ideias de quem, reconhecidamente, domina a matéria de que trata – o especialista –, citações de uma obra, de uma instituição.
- Ações por comparação ou analogia: baseadas em fatores de semelhança, justamente por ser uma causa ou consequência dos dados.
- Ações dedutivas: implicam em uma dedução e particularização.
- Ações por evidências: são justificadas por meio de evidências de que se aplicam os dados considerados.
- Ações exemplares: comportamentos e personalidades vistos como virtude ou exemplo a ser seguido, ou os quais, por si sós, já são suficientes para justificar as ações.
- Ações experienciais: experiências já vivenciadas.
- Ações históricas: exemplos da tradição e experiência histórica.
- Ações indutivas: quando se recorre a generalizações, previsões ou probabilidades.
- Ações de princípios: baseadas numa constatação (lógica, científica, ética, estética...) aceita como verdadeira e de validade universal.
- Ações proverbiais ou de sabedoria popular: citações da voz e consciência comuns.
- Ações de singularidade: quando algo ou alguém é apresentado por sua singularidade, por sua diferença.
- Ações universais: saberes universalmente aceitos porque foram demonstrados factual ou cientificamente.

7 DESCRIÇÃO DO ALGORITMO

Detalhamento de como funciona o algoritmo.

7.1 Heurística e execução de tarefas

De acordo com o dicionário de significados, "a heurística pode ser considerada um 'atalho mental' usado no pensamento humano para se chegar aos resultados e indagações mais complicadas de modo rápido e fácil, mesmo que estes sejam incertos ou incompletos." [1] De certa forma é uma motivação para dar seguimento à demonstração de como pensar para apresentar um projeto de sistemas de forma alternativa, não tão convencional, descrito na literatura de planejamento em áreas como engenharia, sistemas e métodos.

Existem convívios entre uma nova forma de pensar e o método tradicional de desenvolvimento de projetos, que fazem uso de sistemas tanto manuais quanto mecanizados. A motivação para apresentar alternativas se caracteriza pela participação no desenvolvimento de projetos de sistemas mecanizados que são implementados para substituir atividades manuais ou semiautomáticas.

Quando esses projetos são terminados e implantados, surgem novas perspectivas que não são perceptíveis quando ainda se está no estágio embrionário. Ou seja, no projeto inicial, a análise do processo manual não permite que algumas lacunas desse processo sejam vistas, porque essas pequenas rotinas entre estágios muitas vezes não estão descritas em quaisquer manuais ou procedimentos de treinamento, o que prejudica muito a correta interpretação de um processo.

O relevante é que essas pequenas lacunas são exatamente o mais importante detalhe executado entre as rotinas, isso dificulta o entendimento do desenvolvimento de um aplicativo computacional. É exatamente a heurística que mostra como se realiza o processo fora da codificação de instruções computacionais.

Em artigos científicos apresentados em simpósios, o autor Natan Costa Lima e o orientador Carlos Eduardo Ferreira descrevem "Algoritmos e estrutura de dados para problemas de deslocamento no plano", onde analogicamente pode-se adaptar problemas

[1] Fonte: https://www.significados.com.br/heuristica/ – Acesso em 09 jul. 2019.

de desenvolvimento de processos que normalmente são executados de forma bidimensional, ou seja, realiza-se uma tarefa por vez, ou duas ou mais tarefas em paralelo, de forma que ao término de uma tarefa (ou tarefas em paralelo) inicia-se outra.

Esses autores se manifestam sobre o deslocamento entre dois pontos. Neste estudo é mostrado que há outras abordagens, onde os pontos seriam as tarefas iniciais e finais e que muitas vezes não se restringem a apenas duas, mas a múltiplas tarefas. Em processos tradicionais sempre se leva em consideração a filosofia que diz que "o caminho mais curto entre dois pontos é uma reta", deixando de ter em conta os obstáculos existentes, ou não, tornando assim o desenvolvimento de um algoritmo um processo moroso.

Interessante observar que no desenvolvimento de um algoritmo, para a obtenção de um caminho mais curto para se fazer "n" tarefas, há a solução de problemas não apenas em duas tarefas lineares, mas em tarefas trapezoidais. Demonstrando justamente que no desenvolvimento de um projeto de processos mecanizados há outras formas de solução que não são utilizadas, porém, por serem consideradas pela heurística tridimensional como "atalhos" . Tal se deve, por um lado, à não existência de uma agenda de investigação definida que aborde sistematicamente a temática da realidade heurística e que crie um enquadramento teórico necessário à compreensão do fenômeno trapezoidal.

7.2 Funções do algoritmo

A função inicial do algoritmo é desmembrar em palavras o texto a ser analisado, palavras tais que serão identificadas por classe gramatical e semântica, e armazenadas para posterior contagem. Serão indicadas dez classes de palavras: substantivo, verbo, adjetivo, pronome, artigo, numeral, preposição, conjunção, interjeição e advérbio.

A função seguinte é montar o corpus de referência desse texto, podendo ser invocada quantas vezes sejam necessárias sem que se tenha que recorrer às funções anteriores, exceto quanto há alterações nas especificações, como um novo endereço dos repositórios ou das palavras que direcionam as buscas.

Outra função é juntar as palavras agrupadas duas a duas, depois agrupadas três a três, até cinco a cinco, a serem armazenadas com a utilização de um aplicativo disponibilizado de forma interativa.

A principal função é montar a indagação em itens agrupados e classificados de acordo com as cinco maiores incidências na classe gramatical "verbo", e então acrescentar o pronome interrogativo (5W2H).

Há também a função de buscar, a partir dos predicados dos verbos, uma correspondência na figura da linguagem (caso haja alguma), o que é a construção de um tipo

paradoxo, conforme Lista de Tipos de Paradoxos, do item 2.3.1 desta tese.

A última função desse algoritmo é criar um questionário que mostra o tipo de indagação e a sua consistência científica, tendo como base os conteúdos informados no texto.

7.3 Algoritmos de agrupamentos de texto

Na visão algorítmica, os textos são analisados em agrupamentos, e é preciso que grupos constituídos por eles tenham certa coesão entre si, sendo esse o grande dificultador para os algoritmos, por ser bastante complexa a análise textual, que envolve uma série de indagações, de âmbito linguístico, cultural, social, situacional, político, como coerência e coesão dos textos, e/ou por estarem diretamente relacionadas com o autor e o momento em que o texto foi escrito (FáVERO, 1991).

Dessa forma, obter agrupamentos com textos muito diferentes não seria admissível, pela falta de coesão, de significado ou de sentido entre eles. Os algoritmos de agrupamento estão intimamente relacionados aos métodos de organização escolhidos.

Um algoritmo computacional que busca a ideia principal de um texto atua em três macroetapas (pré-processamento, processamento e pós-processamento) (GUELPELI, 2012), como propõe o modelo Cassiopeia, e agrupa textos de forma hierárquica, com novo método para definição do corte de Luhn [2].

Para o método hierárquico aglomerativo, Kowalski (1997, p. 282), Wives (2004) e Larose (2004) citam quatro algoritmos: o de ligação simples (*single linkage*), o de ligação completa (*complete linkage*), o de ligação mediana ou de valor médio (*average linkage*), e o Ward.

O algoritmo de ligação simples utiliza-se do critério de vizinho mais próximo, no qual a distância entre dois grupos é determinada pela distância do par de documentos mais próximos, cada um pertencente a um desses grupos. Esse algoritmo, cuja característica é a união de grupos, apresenta um problema conhecido como "efeito da corrente", quando ocorre a união indevida de grupos, influenciada pela presença de ruídos na base de dados (LAROSE, 2014).

No algoritmo de ligação completa, ao contrário do de ligação simples, o critério utilizado é o de vizinho mais distante. A distância entre dois grupos é a maior entre um par de documentos, e cada documento pertence a um grupo distinto. Esse método dificulta a

[2] O Algoritmo de Luhn foi criado por Hans Peter Luhn (1896 -1964), cientista da computação que trabalhou na IBM. Fonte: http:pipeless.blogspot.com.br200810o-algoritmo-de-luhn.html – Acesso em 09 jul. 2019.

formação do "efeito da corrente", e tende a formar grupos mais compactos e em formatos esféricos (LAROSE, 2014).

O algoritmo de ligação mediana, ou de valor médio, apresenta a definição da distância entre dois grupos. Mostra a média das distâncias entre todos os pares de documentos em cada grupo, e cada par é composto por um documento de cada grupo. Esse método elimina muitos problemas relacionados à dependência do tamanho dos grupos, mantendo próxima a variabilidade interna entre eles (LAROSE, 2014).

No Ward, há uma variação dos anteriores, buscando menor variância entre os agrupamentos, juntando os elementos cuja soma dos quadrados, ou o erro dessa soma, seja mínima (ALDENDERFER e BLASHFIELD, 1984)[3]. Segundo Wives (2004), este algoritmo gera grupos hiperesféricos de tamanhos semelhantes.

O método hierárquico divisivo tem dois tipos de categoria de algoritmos, os monotéticos e os politéticos. Os monotéticos apresentam apenas uma variável, testada a cada divisão do agrupamento, que ocorre na presença ou ausência de cada uma das variáveis. Essas variáveis são binárias, registrando, assim, a existência ou não de um objeto. O resultado apresenta um agrupamento cujos elementos têm os mesmos valores para uma variável. Essa abordagem divisiva monotética também é conhecida como análise de associações. Já nos algoritmos da categoria politéticos, para cada repetição do método divisivo, todas as variáveis entram no processo de cálculo de distância (LAROSE, 2014).

Para o método particionado iterativo, o algoritmo mais utilizado e conhecido é o k-médias (*k-means*), que faz uma comparação entre o valor de cada linha, por meio da distância, para gerar os agrupamentos, utilizando geralmente a distância euclidiana para calcular o quão distante uma ocorrência está da outra. A maneira de calcular essa longinquidade vai depender da quantidade de atributos. Após o cálculo das distâncias, o algoritmo calcula centroides para cada um dos agrupamentos, e conforme vai iterando, o valor de cada centroide é refinado pela média dos valores de cada atributo, e de cada ocorrência que pertence a esse centroide. Com isso, o algoritmo gera "k" centroides e coloca as ocorrências em uma matriz, de acordo com a distância dos centroides. Wives (2004) explica que o grande problema vem da necessidade de o usuário definir o número de agrupamentos. Há de se ressaltar que, nos agrupamentos textuais, Fan (2006), e Alsumait e Domeniconi (2007) consideram os processos como não interativos, ou seja, mostram que não existe a possibilidade de se prever o número de agrupamentos, contrário ao caso do algoritmo k-médias, que necessita definir, previamente, um número de agrupamentos.

No método de busca em profundidade, os algoritmos podem ser divididos em duas categorias, os de ligação simples, discutidos anteriormente, e os de distribuições multivari-

[3] ALDENDERFER, M. S.; BLASHFIELD, R. K. Cluster Analysis. Beverly Hills, CA:Sage, 1984.

adas de probabilidade. Esses últimos são baseados em um modelo estatístico que assume serem membros de grupos diferentes, portadores de distribuições de probabilidades diferentes para cada variável.

No método de fator analítico, o cálculo deve ser realizado, a princípio, através de algum coeficiente de similaridade. Os objetos são então alocados em agrupamentos de acordo com a sua carga em cada fator (fator *loading* [4]).

Já com o método de amontoamento, os algoritmos permitem sobreposição, ou seja, podem associar os objetos a um ou mais grupos. Esse amontoamento mostra que um objeto está em um ou mais grupos ou, ainda, que pertence a todos os grupos com um grau de pertinência/probabilidade (WITTEN e FRANK, 2005)[5].

Dois são os principais algoritmos do método grafoteorético: os cliques, descrito formalmente no algoritmo 3, e o estrela (*star*), com suas variações que, segundo Wives (2004), é a melhor estrela (*best star*), e estrelas completas (*full stars*). Nessa última variação, os agrupamentos têm forma similar a de uma estrela, ou seja, um elemento central que possui relação com diversos outros ligados a ele, formando as pontas. Os elementos nas extremidades não têm relação com os outros, sendo esse fato um dos problemas desse algoritmo, pois eles podem não ser similares. Porém existem variações para atenuar esse contratempo. O algoritmo melhor estrela aloca um elemento à estrela mais similar, pois não ignora os elementos já adicionados a outras estrelas, conseguindo realocá-los. Nas estrelas completas, os elementos são alocados em mais de um agrupamento (GUELPELI, 2012).

É possível a utilização de sumarizadores agrupados com os textos de similaridade na montagem do corpus, a partir do texto escolhido pelo leitor. Os agrupamentos criados nessa etapa têm um vetor de palavras de alta relevância para cada agrupamento (DELGADO; DIAS; GUELPELI, 2013).

7.4 Medida de similaridade em agrupamento de texto

Para definir a similaridade entre os textos, utiliza-se uma "medida" de similaridade que, definida, vai mensurar os valores dos atributos de cada texto, ou seja, quanto menor a distância entre tais valores, mais similares são esses textos. Os tipos de medida de similaridade utilizados na literatura são as de distância, os coeficientes de correlação, os de associação, e as medidas probabilísticas de similaridade (ALDENDERFER e BLASHFIELD, 1984).

[4] Fator loading são os coeficientes de correlação entre as variáveis e fatores.

[5] WITTEN I. H., FRANK, E. Data Mining: Practical Machine Learning Tools and Techniques, 2nd ed. San Francisco, CA: Morgan Kaufmann, 2005

As medidas de distância são: a Euclidiana, a mais usual (ALDENDERFER e BLASHFIELD, 1984), função cosine (SALTON e MACGILL, 1983)[6] e a distância Manhattan, ou métrica city block (KAUFMAN e ROUSSEEUW, 1990)[7].

Dentre os coeficientes de correlação, os mais importantes são: o de Pearson, o de Jaccard, o de associação simples (*simple matching coeficiente*) e o de Gower. Para medida probabilística, não existe um cálculo, mas um aferimento direto, no dado bruto. Já para a medida *fuzzy*, Wives (2004) cita o seu trabalho, no qual usa as funções, inclusão simples (*set theoretic inclusion*) de Cross (1994), e a média por operadores difusos, de Oliveira (1996).

No processo de recuperação de informação, os documentos podem ser descritos por um conjunto de termos representativos do corpus em questão (vocabulário do domínio). O grau de similaridade entre dois documentos é calculado em função da distância entre o conjunto de termos de cada um (GUELPELI, 2012).

[6] SALTON, G. e MACGILL, J. M. Introduction to Modern Information Retrieval. New York: McGraw-Hill, 1983.

[7] KAUFMAN, L. and ROUSSEEUW, P. Finding Groups in Data: An Introduction to Cluster Analysis. New York: Wiley Interscience, 1990.

8 RESULTADO

Com os textos armazenados em banco de dados, o pesquisador terá a sua disposição todos os resumos, de modo que haverá opção para fazer combinações ou junções dos mesmos. O aplicativo fará combinação desses resumos em pontos que são congruentes e mostrará o resultado em um novo resumo. O aplicativo, fazendo consulta nos repositórios científicos disponíveis e acessíveis, buscará todos os textos, artigos, livros, periódicos, e publicações a partir das palavras-chave. Todo esse material consultado será guardado em um banco de dados.

8.1 Resultado que o algoritmo não alcançará

Existem muitos artigos que foram escritos em um tempo distante da data de hoje, existem outros tantos que o autor descreve sem uma estruturação conhecida, o que faz com que o algoritmo não reconheça a Introdução, a Metodologia ou os Resultados buscados. Assim, torna-se deficitária a sua identificação enquanto artigo científico ou acadêmico.

Em se tratando de algoritmos, regularmente há uma recorrência similar: sem que se possa reconhecer quais são as características que o tema ou o título possuem, não é possível dar suporte à sua existência, real ou virtual. Da mesma forma que, não havendo como reconhecer uma metodologia nas descrições das funções ou das ações exercidas sobre o tema, não há como mostrar quais os resultados desse objeto para que se atinja o objetivo onde as ações discursivas põem em jogo determinados "dispositivos" existentes no idioma, esses designados operadores argumentativos e conjunções. São ainda os operadores argumentativos que permitem o encadeamento dos atos ilocutórios que, como os elos de uma cadeia, constituem o discurso.

Segundo Ducrot, o ato ilocutório opera um tipo especial de transformação: "trata-se sempre de uma transformação de ordem jurídica, da criação de direitos ou de deveres para os participantes do ato de fala." (Ducrot, 1984b: 445).

8.2 Resultado que o algoritmo alcançará

Com o uso do algoritmo serão formuladas indagações sobre a ideia principal do texto analisado, baseadas nas relações entre as características do texto e o corpus, e as possíveis combinações com alguns tipos de advérbios para a formulação da sentença. Com isso, o resultado mostrará o objeto principal de tal texto, que é o seu tema, o assunto que o descreve e, em alguns casos, o próprio título. Os itens Introdução e Conclusão de um artigo em análise fazem uma complementação com os argumentos e justificativas sobre o tema. E, no item Metodologia, onde estão descritas ou demonstradas ações, que por sua vez podem ser exemplos ou fórmulas, ficarão explicitados os objetivos a serem alcançados no artigo (FARIAS, 2000a).

O algoritmo desenvolvido identificará as características das ideias pertinentes ao texto acadêmico a partir das orações subordinadas e segundo os operadores argumentativos, que serão acrescidos da técnica 5W2H, que podem ser os pronomes interrogativos ou advérbios interrogativos, a partir das relações entre os períodos gramaticais e o texto (LOPES-ROSSI, 1996).

A composição do resultado para mostrar as indagações será composta de "pronome interrogativo (5W2H) + verbo de maior incidência + complementos de maior incidência neste verbo". Exemplo: "Para que + utilização do Caminhar em 3D (mapa trapezoidal) no projeto de mecanização de quaisquer processos..." ou "Quais as + utilização do Caminhar em 3D (mapa trapezoidal) no projeto de mecanização de quaisquer processos...".

Lobo diz:

> Uma das características distintivas das interrogativas parciais é o fato de apresentarem um constituinte interrogativo que marca o foco da interrogação. Os constituintes interrogativos apresentam, na sua maior parte, um morfema "qu" (quem, quando, que...), em português, ou um morfema "wh" (who, what, where...), em inglês (quem, o que, onde, respectivamente). Por essa razão, são designados por constituintes "wh". Essas estruturas têm a particularidade de apresentarem alterações à ordem básica de palavras... (LOBO, 2018, p. 226). (C., 2018).

Espera-se também que as ações sejam congruentes com as palavras-chave apresentadas no texto. Caso as indagações não estejam adequadas, a sugestão é que se faça a revisão do tema, das características e das ações.

A concretização deste empreendimento está na demonstração de que com a utilização do mapa trapezoidal no projeto de mecanização de quaisquer processos, obter-se-á:

1. A análise de um processo manual, identificado com uma tarefa de início e fim, procurando as funções e as rotinas, assim como o ponto comum entre elas. Com isso pode-se ver que existem pontos que estão fora do processo das funções e rotinas, e isso agilizará o processo de execução.

2. O entendimento de que das análises de processos, uma vez não compreendidas, resulta o surgimento de outra rotina para contornar este obstáculo. É o que esse projeto mostra: ao invés de contornar o obstáculo, deve-se achar o espaço livre entre as rotinas.

8.3 Interpretação do principal resultado

A pesquisa de Sousa e Silva (l988) acrescenta ainda que o tipo de regras de redução semântica utilizado e a eficiência no domínio dessas regras são determinados não só pela escolaridade do aluno, conforme sugerem Brown e Day, e pelo grau de dependência em relação ao texto-base, conforme defendem Kleiman e Terzi, mas, principalmente, pela capacidade do leitor/resumidor de tomar decisões adequadas em relação a certas etapas características do processamento da escrita (solução de problemas de planejamento, tradução e revisão).

Para comprovar essa hipótese, a autora em questão analisou um resumo produzido por um aluno universitário do Curso de Letras. A tarefa foi realizada em dois momentos. No primeiro, o aluno produziu o resumo do texto, sendo-lhe facultada a consulta a este. No segundo, o aluno deveria revisar o resumo, porém sem recorrer ao texto-base, observando sua adequação à audiência, a clareza de sua estrutura global e a eficiência de suas marcas formais para a leitura.

A autora buscou analisar tanto o produto da leitura (mediante análise de resumo escrito pelo sujeito) quanto o seu processo (mediante análise de protocolo verbal). O desempenho do sujeito do experimento foi analisado em três fases do processamento da escrita planejamento, tradução e revisão (conforme modelo de Hayes e Flower, apud Sousa e Silva) (FARIAS, 2000a).

8.4 Ressalvas

O processo do algoritmo desenvolvido, inicialmente, consiste em relacionar a morfologia da palavra com os tipos de advérbios e pronomes relativos. Essa relação pode ser ampliada com decorrer das evoluções de novas relações, e isso implica diretamente no resultado, uma vez que a adição de uma nova relação gerará novas combinações.

No decorrer do desenvolvimento do algoritmo e pela complexidade de análise sintática de todas as classes gramaticais, como o objetivo da tese é montar indagações optou-se por levar em consideração somente a classe gramatical "verbo", uma vez que essa classe determina a ação ou a definição de um texto.

O algoritmo deve ser reavaliado para não apresentar equívocos nos resultados e incongruências no que se refere às indagações geradas.

9 CONCLUSÃO

9.1 Conclusão

Como apresentado no artigo "Generating key questions from academic texts" (SANOKI, 2017), um algoritmo computacional deve ser redesenhado a cada evolução das novas formas de apresentação. Uma vez projetado e totalmente desenvolvido o algoritmo, se executado em diversas aplicações, mostrará o resultado esperado, mas há sempre a necessidade de implementação ou alteração de processo, pois é alto o grau de obsolescência da forma inicial do desenho de um algoritmo, e isso traz muitas dificuldades.

O algoritmo computacional é responsável pela da ideia principal para gerar perguntas da classe paradoxal, utilizando um modelo de programação estruturado a partir da adaptação e da conversão de uma escrita tradicional de artigos, em que o esquema parte do título e do tema do artigo científico ou acadêmico para elucidar de forma mais rápida o que o autor quer demonstrar com uma determinada pesquisa, teórica ou prática, e o que o resultado pode alcançar, tornando assim possível saber exatamente o que investigar.

O algoritmo busca nas premissas descritas no item Introdução do texto todas as propriedades e características que compõem e dão suporte ao tema e título, sem as quais não é possível o delineamento do resultado que se quer demonstrar utilizando processos de manuseio de tais premissas. Ele então busca nos processos explicitados no item Metodologia, quais as funções e ações descritas que propõem cumprir o tema, e para isso analisa e monta métodos com a descrição do passo a passo da sua experiência, mostrada nos exemplos ou fórmulas.

O resultado é a descrição de um objeto com suas propriedades, e os métodos descritos em ações como função, fazendo uso dessas propriedades. Nessa descrição da metodologia do artigo, quando a função está bem clara, nos é proporcionado com maior exatidão o que realmente se quer demonstrar, e o que está sendo feito com tais propriedades. E então o tema encerra o intuito que quer revelar.

Um dos ganhos na aplicação desse algoritmo é que ele apresenta não somente as indagações baseadas em objetos, propriedades e métodos, mas também as coerências das características do texto, e possíveis combinações com alguns tipos de advérbios para a

formulação de sentenças, uma vez que para analisar esse objeto que é o tema com sua propriedade e suas funções, o algoritmo formula indagações que podem corresponder ao resultado esperado, ou não. Encontrando-se correspondência entre o descrito pelo algoritmo e o que tema propõe, é gerado um estímulo à curiosidade que somente a leitura do texto poderá saciar, e cada leitor, então, fará sua própria interpretação.

As informações veiculadas nesta tese representam apenas uma introdução ao modelo de compreensão de Kintsch e Van Dijk e às formulações de tal modelo acerca da natureza e dos processos envolvidos na tarefa de resumo de textos, servindo como sugestão de um ponto de partida para o aprofundamento dessas questões. As considerações de caráter experimental aqui contidas tentam, por sua vez e entre outros aspectos, destacar alguns dos fatores (presentes ou não nos estudos de Kintsch e Van Dijk) que interferem na tarefa de resumo (a maturidade do leitor, as condições da tarefa em si, a habilidade no processamento da escrita...) e que devem ser considerados com mais atenção no meio escolar, onde a leitura e produção de textos (resumos ou não) ainda têm sido propostas de forma um tanto aleatória (SANOKI; VEGA, 2018b).

E, recorrendo ao artigo "Argumentative scheme in academic texts: Algorithmic for validating generated questions" (SANOKI; VEGA, 2018a), a aplicação das premissas descritas na Introdução será a base para que o algoritmo monte as funções no sentido do objeto enquanto questões pertinentes a partir da adequação e compreensão morfológica e semântica do tema de um texto acadêmico, de suas palavras-chave, de seu Resumo e de suas Referências Bibliográficas.

A montagem de um algoritmo computacional leva em consideração o encadeamento das conjunções de uma oração à outra, ou dos operadores lógicos e argumentativos, e como eles se encadeiam no texto. Quando houver coesão na construção de argumentos há como montar os aspectos conceituais destes períodos gramaticais algoritmicamente, na forma de objetos, propriedades e métodos (PLATÃO, F. and FIORIN, J. L., 1996).

REFERÊNCIAS

ALVES, L. M. e C. R. Paradoxos I – Considerações Iniciais. *Cognitio: Revista de Filosofia*, v. 8, n. 2, p. 249–264, 2013. ISSN 2316-5278. Disponível em: <https://revistas.pucsp.br/cognitiofilosofia/article/view/12229>. Citado na página 23.

BARROS, M. G. *Reescrita da seção justificativa do projeto de pesquisa por escritores iniciantes*. Tese (Doutorado) — www. teses. ufc. br, 2013. Citado na página 49.

BEHR, A.; MORO, E. L.; ESTABEL, L. B. *Gestão da biblioteca escolar: metodologias, enfoques e aplicação de ferramentas de gestão e serviços de biblioteca*. [S.l.]: Ciência da Informação, 2008. v. 37. Citado na página 36.

C., L. M. e S.-J. *Aquisição de língua materna e não materna: Questões gerais e dados do português*. 225–248: Berlin: Language Science Press, 2018. Disponível em: <https://run.unl.pt/bitstream/10362/35521/1/Interrogativas_relativas_e_clivadas.pdf>. Citado na página 62.

CABRAL, L. et al. Automatic Summarization of News Articles in Mobile Devices. In: *Proceedings of the 2015 Fourteenth Mexican International Conference on Artificial Intelligence (MICAI)*. Washington, DC, USA: IEEE Computer Society, 2015. (MICAI '15), p. 8–13. ISBN 978-1-5090-0323-5. Disponível em: <http://dx.doi.org/10.1109/MICAI.2015.8>. Citado na página 39.

CAMARGO, A. M.; BELLOTTO, H. L. *Dicionario de terminologia arquivistica*. 1996. Citado 2 vezes nas páginas 17 e 21.

CAMARGO, B. V.; JUSTO, A. M. Tutorial para uso do software de analise textual IRAMUTEQ. *Florianopolis-SC: Universidade Federal de Santa Catarina*, 2013. Citado 2 vezes nas páginas 39 e 40.

CHEN, D. et al. Reading Wikipedia to Answer Open-Domain Questions. *Proceeding of the 55th Annual Meeting of the Association for Computation Linguistics*, v. 1: Long Papers, p. 1870–1879, 2017. Disponível em: <https://doi.org/10.18653/v1/P17-1171>. Citado na página 14.

CHOMSKY, N. *Aspects of the Theory of Syntax*. [S.l.: s.n.], 1964. v. 11. Citado na página 41.

CODD, E. F. A relational model of data for large shared data banks. *Communications of the ACM*, ACM, v. 13, n. 6, p. 377–387, 1970. Citado 3 vezes nas páginas 35, 41 e 42.

DAVID, K. A. Sintaxe das expressões nominais no português do Brasil: uma abordagem computacional. www. teses. ufc. br, 2007. Disponível em: <http://repositorio.ufc.br/handle/riufc/8770>. Citado na página 20.

DELGADO, C. C. N.; DIAS, H. D.; GUELPELI, M. V. C. Utilização de somários Humanos no modelo Cassiopéia. *Anais do Computer on the Beach*, p. 258–267, 2013. Disponível em: <http://siaiap32.univali.br/seer/index.php/acotb/article/view/6222>. Citado na página 59.

ENCARNATION, D. J. *Japanese Multinationals in Asia: Regional Operations in Comparative Perspective*. Oxford University Press, 1999. (Japan Business and Economics Series). ISBN 9780195353013. Disponível em: <https://books.google.com.br/books?id=bvcihMdMYwC>. Citado na página 41.

FARIAS, W. S. de. Compreensão e resumo de textos: alguns aspectos teóricos e experimentais. v. 1, n. 22, 2000. Citado 2 vezes nas páginas 62 e 63.

FARIAS, W. S. de. Compreensão e resumo de textos: alguns aspectos teóricos e experimentais. v. 1, n. 22, 2000. Citado 3 vezes nas páginas 15, 16 e 21.

FERREIRA, H.; CASSIOLATO, M.; GONZALEZ, R. Como elaborar Modelo Lógico de programa: um roteiro básico. *Nota Técnica, IPEA*, 2009. Disponível em: <http://scholar.google.com/scholar?hl=en&btnG=Search&q=intitle:Como+Elaborar+Modelo+L?g Citado na página 20.

FáVERO, L. L. *Coesão e coerência textuais*. [S.l.]: Ática, 1991. Citado na página 57.

GERALDI, J. W.; GUIMARÃES, E. R.; ILARI, R. Operadores de argumentação e diálogo. *Cadernos de Estudos Linguísticos*, v. 9, 2012. Citado na página 21.

GIL, A. C. Como elaborar projetos de pesquisa. v. 5, p. 61, 2000. Citado 2 vezes nas páginas 17 e 32.

GILBERT, W. S.; SULLIVAN, A. S. *The Pirates of Penzance*. Alfred Music, 1968. (Kalmus Edition). ISBN 9781457481543. Disponível em: <https://books.google.com.br/books?id=dl294_Te6FMC>. Citado na página 22.

GONZAGA, M. *Aspectos da sintaxe dos advérbios em Português*. Tese (Doutorado) — MA dissertation, University of Lisbon, 1997. Citado 2 vezes nas páginas 34 e 49.

GUELPELI, M. V. C. *CASSIOPEIA: UM MODELO DE AGRUPAMENTO DE TEXTOS BASEADO EM SUMARIZAÇÃO*. Tese (Doutorado) — Universidade Federal Fluminense, 2012. Citado 4 vezes nas páginas 39, 57, 59 e 60.

HANSEN, C. D. Mass Nouns and A White Horse Is Not a Horse. *Philosophy East and West*, University of Hawaii Press, v. 26, n. 2, p. 189–209, 1976. ISSN 0031-8221. Disponível em: <http://www.jstor.org/stable/1398188>. Citado na página 22.

JULIA. Dicionário Informal. ago. 2014. Disponível em: <¡https://www.dicionarioinformal.com.br/texto/¿.>. Citado na página 34.

LAROSE, D. T. *Discovering knowledge in data: an introduction to data mining*. [S.l.]: John Wiley & Sons, 2014. Citado 2 vezes nas páginas 57 e 58.

L.FIORIN, J. *Elementos de Análise do Discurso*. [S.l.: s.n.], 2018. ISBN 978-85-7244-294-7. Citado na página 20.

LI, Y. et al. Setence similarity based on semantic nets and corpus statistics. *IEEE Transactions on Knowledge and Data Engineering*, n. 18(8), p. 1138–1150, 2006. Citado na página 15.

LIU, H. e. a. BioLemmatizer: a lemmatization tool for morphological processing of biomedical text. *Journal of biomedical semantics*, vol. 3 3., abr. 2012. ISSN 2041-1480. Disponível em: <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC3359276/>. Citado na página 49.

LIUMBRUNO, G. M. et al. How to write a scientific manuscript for publication. v. 11, n. 2, p. 217–26, 2016. Citado na página 19.

Lopes Pinheiro, C.; ANDRÉA, J.; PEREIRA, M. O RESUMO ACADÊMICO: TEXTUALIDADE E ENSINO AcAdemic summAry: textuAlity And teAching. *Natal/RN Número Especial*, v. 14, n. 14, p. 117–130, 2012. Citado na página 39.

LOPES-ROSSI, M. A. G. *A sintaxe diacronica das interrogativas-q do portugues*. Tese (application/pdf) — Universidade Estadual de Campinas. Instituto de Estudos da Linguagem, jul. 1996. Disponível em: <http://repositorio.unicamp.br/jspui/handle/REPOSIP/269151>. Citado na página 62.

LYAPIN, S. K. et al. SCIENTIFIC AND TECHNICAL. v. 15, n. 1, p. 155–162. Citado na página 19.

LYCAN, W. G. What, exactly, is a paradox? *Analysis*, v. 70, n. 4, p. 615–622, 07 2010. ISSN 0003-2638. Disponível em: <https://doi.org/10.1093/analys/anq069>. Citado na página 22.

MACHADO, A. R.; BEZERRA, M. A. Gêneros Textuais & Ensino. 2002. Citado na página 15.

MARTINO, A. *Portugues esquematizado: gramatica, interpretacao de texto, redacao oficial,redacao discursiva*. [S.l.]: 2014, 2013. v. 3. ISBN 978-85-02-21376-0. Citado 4 vezes nas páginas 34, 36, 37 e 52.

MATENCIO, M. d. L. M. Atividades de (re) textualiza{ç}{ã}o em pr{á}ticas acad{ê}micas: um estudo do resumo. v. 6, n. 11, p. 109–122, 2002. Citado 3 vezes nas páginas 15, 19 e 32.

MEYER, C. F. *English Corpus Linguistics: An Introduction*. Cambridge University Press, 2002. (Studies in English Language). ISBN 9780521004909. Disponível em: <https://books.google.com.br/books?id=jVspcHe54uUC>. Citado na página 30.

NETO, R. G. MINIMALISMO SCHUMPETERIANO, TEORIA ECONÔMICA DA DEMOCRACIA E ESCOLHA RACIONAL. *Revista de Sociologia e Política*, v. 19, n. 38, p. 2–3, 2011. ISSN 1678-9873. Disponível em: <https://revistas.ufpr.br/rsp/article/view/31666>. Citado na página 22.

PLATÃO, F. and FIORIN, J. L. *Para entender o texto: leitura e redação*. [S.l.]: Atica, 1996. Citado 2 vezes nas páginas 21 e 66.

QUARESMA, P. Um sistema de Pergunta-Resposta para uma base de Documentos. . *Letras de Hoje*, v. 41, n. 2, 2006. ISSN 1984-7726. Citado na página 16.

QUINE, W. V. PARADOX. *Scientific American*, Scientific American, a division of Nature America, Inc., v. 206, n. 4, p. 84–99, 1962. ISSN 0036-8733. Disponível em: <http://www.jstor.org/stable/24937290>. Citado na página 22.

RAJPURKAR, P. et al. SQuAD: 100,000+ Questions for Machine Comprehension of Text. In: *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*. Austin, Texas: Association for Computational Linguistics, 2016. p. 2383–2392. Disponível em: <https://www.aclweb.org/anthology/D16-1264>. Citado na página 15.

RINO, L. H. M. *modelagem de discurso para o tratamento da concisão e preservação da ideia central na geração de dados*. Tese (Doutorado) — Instituto de Física de São Carlos, 1996. Disponível em: <http://www.teses.usp.br/teses/disponiveis/76/76132/tde-07012009-095918/en.php>. Citado na página 18.

SANOKI, K. Generating key questions from academic texts. 2017 12th Iberian Conference on Information System and Tecnologies (CISTI), p. 1–4, 2017. ISBN 978-989-96247-4-0. Citado 6 vezes nas páginas 18, 32, 36, 37, 51 e 65.

SANOKI, K.; VEGA, I. S. Questions to encourage the reading of an Academic Text. XXXVII International Sodebras Congress, v. 12, n. 141, p. 95–98, 2017. ISSN 1809-3957. Citado 2 vezes nas páginas 37 e 42.

SANOKI, K.; VEGA, I. S. ARGUMENTATIVE SCHEME IN ACADEMIC TEXTS: ALGORITHMIC FOR VALIDATING GENERATED QUESTIONS. XXXIX International Sodebras Congress, v. 13, n. 152, p. 61–63, maio 2018. ISSN 1809-3957. Disponível em: <http://www.sodebras.com.br>. Citado 5 vezes nas páginas 16, 19, 34, 37 e 66.

SANOKI, K.; VEGA, I. S. ARGUMENTATIVE SCHEME IN ACADEMIC TEXTS: ALGORITHMIC IDENTIFICATION OF CONCEPT. XXXVIII International Sodebras Congress, v. 13, n. 147, p. 95–98, mar. 2018. ISSN 1809-3957. Citado 2 vezes nas páginas 51 e 66.

SANOKI, K.; VEGA, I. S. ARGUMENTATIVE SCHEME IN TEXTS: FIND CONCEPTUAL DEFINITION IN THE ACADEMIC TEXT AND GENERATE QUESTIONS THAT JUSTIFY IT - no prelo. XL International Sodebras Congress, v. 14, n. 159, dez. 2019. Disponível em: <http://www.sodebras.com.br>. Citado na página 19.

SARDINHA, T. B. Lingüística de corpus: histórico e problemática. *Delta*, SciELO Brasil, v. 16, n. 2, p. 323–367, 2000. Citado na página 51.

SCHMID; HELMUT. TreeTagger - a language independent part-of-speech tagger. 01 2009. Citado na página 40.

TAYLER, C. *Argumentos filosoficos*. [S.l.]: editora loyola, 2019. ISBN 9788515018956. Citado na página 20.

VOLPATO, G. L. O método lógico para redação científica. *Revista Eletrônica de Comunicação, Informação e Inovação em Saúde*, v. 9, n. 1, mar. 2015. ISSN 1981-6278. Citado na página 49.

10 APÊNDICE - FONTE EM C#

```
private void InitializeComponent()
{
this.ofd1 = new System.Windows.Forms.OpenFileDialog();
this.txtCaminhoPDF = new System.Windows.Forms.TextBox();
this.btnBuscaPDF = new System.Windows.Forms.Button();
this.btnConverter = new System.Windows.Forms.Button();
this.label42 = new System.Windows.Forms.Label();
this.label44 = new System.Windows.Forms.Label();
this.lblMsg = new System.Windows.Forms.Label();
this.SuspendLayout();
this.ofd1.FileName = "openFileDialog1";
this.txtCaminhoPDF.BorderStyle =
System.Windows.Forms.BorderStyle.FixedSingle;
this.txtCaminhoPDF.Font =
new System.Drawing.Font("Verdana", 8.25F,
System.Drawing.FontStyle.Regular,
System.Drawing.GraphicsUnit.Point, ((byte)(0)));
this.txtCaminhoPDF.Location = new System.Drawing.Point(12, 57);
this.txtCaminhoPDF.Name = "txtCaminhoPDF";
this.txtCaminhoPDF.Size = new System.Drawing.Size(477, 21);
this.txtCaminhoPDF.TabIndex = 299;
this.txtCaminhoPDF.Tag = ;
this.btnBuscaPDF.Font =
```

```
new System.Drawing.Font("Verdana", 8.25F,
System.Drawing.FontStyle.Regular,
System.Drawing.GraphicsUnit.Point, ((byte)(0)));
this.btnBuscaPDF.Location = new System.Drawing.Point(489, 56);
this.btnBuscaPDF.Name = "btnBuscaPDF";
this.btnBuscaPDF.Size = new System.Drawing.Size(36, 22);
this.btnBuscaPDF.TabIndex = 300;
this.btnBuscaPDF.Text = "...";
this.btnBuscaPDF.UseVisualStyleBackColor = true;
this.btnBuscaPDF.Click +=
new System.EventHandler(this.btnBuscaPDF_Click);
this.btnConverter.Font =
new System.Drawing.Font("Verdana", 8.25F,
System.Drawing.FontStyle.Bold,
System.Drawing.GraphicsUnit.Point, ((byte)(0)));
this.btnConverter.Location = new System.Drawing.Point(526, 54);
this.btnConverter.Name = "btnConverter";
this.btnConverter.Size = new System.Drawing.Size(81, 28);
this.btnConverter.TabIndex = 302;
this.btnConverter.Text = "Converter";
this.btnConverter.UseVisualStyleBackColor = true;
this.btnConverter.Click += new System.EventHandler(this.btnConverter_Click);
this.label42.AutoSize = true;
this.label42.Font =
new System.Drawing.Font("Verdana", 8.25F,
System.Drawing.FontStyle.Regular,
System.Drawing.GraphicsUnit.Point, ((byte)(0)));
this.label42.Location = new System.Drawing.Point(12, 41);
this.label42.Name = "label42";
```

```
this.label42.Size = new System.Drawing.Size(102, 13);
this.label42.TabIndex = 304;
this.label42.Text = "Caminho do PDF";
this.label44.AutoSize = true;
this.label44.Font =
new System.Drawing.Font("Verdana", 12F,
System.Drawing.FontStyle.Bold,
System.Drawing.GraphicsUnit.Point, ((byte)(0)));
this.label44.ForeColor = System.Drawing.Color.FromArgb(((int)(((byte)(192))))
((int)(((byte)(0)))), ((int)(((byte)(0)))));
this.label44.Location = new System.Drawing.Point(12, 9);
this.label44.Name = "label44";
this.label44.Size = new System.Drawing.Size(137, 18);
this.label44.TabIndex = 303; this.label44.Text = "Converter PDF";
this.lblMsg.AutoSize = true;
this.lblMsg.Font =
new System.Drawing.Font("Verdana", 8.25F,
System.Drawing.FontStyle.Regular,
System.Drawing.GraphicsUnit.Point, ((byte)(0)));
this.lblMsg.Location = new System.Drawing.Point(12, 100);
this.lblMsg.Name = "lblMsg";
this.lblMsg.Size = new System.Drawing.Size(11, 13);
this.lblMsg.TabIndex = 305;
this.lblMsg.Text = ".";
this.AutoScaleDimensions = new System.Drawing.SizeF(7F, 13F);
this.AutoScaleMode = System.Windows.Forms.AutoScaleMode.Font;
this.BackColor = System.Drawing.SystemColors.Control;
this.ClientSize = new System.Drawing.Size(1125, 551);
this.Controls.Add(this.lblMsg);
```

```
this.Controls.Add(this.label42);
this.Controls.Add(this.label44);
this.Controls.Add(this.btnConverter);
this.Controls.Add(this.btnBuscaPDF);
this.Controls.Add(this.txtCaminhoPDF);
this.Font = new System.Drawing.Font("Verdana", 8.25F,
System.Drawing.FontStyle.Regular,
System.Drawing.GraphicsUnit.Point, ((byte)(0)));
this.Name = "ConvertPdf";
this.Text = "Convert Pdf";
this.ResumeLayout(false);
this.PerformLayout();
}
using ProjetoTreeTagger.DAL;
using iTextSharp.text.pdf;
using iTextSharp.text.pdf.parser;
using System;
using System.IO;
using System.Text;
using System.Text.RegularExpressions;
using System.Windows.Forms;
namespace ProjetoTreeTagger
{
public class PDF
{
public event MsgPDF Msg;
public virtual void OnMsg(EventArgs e)
{
if (Msg != null)
```

```
      Msg(this, e);
      }
      public delegate void MsgPDF(object source, EventArgs e);
      private string _msgclass = ;
      public string MsgClassPDF
      {
      get
      {
      return _msgclass;
      }
      }
      public string ConvertePdfEmTexto(string caminho, string nomepdf)
      {
      try
      {
      string textoCompleto = , teste = ;
      StreamWriter arq;
      _msgclass = "Processando....";
      OnMsg(new EventArgs());
      Application.DoEvents();
      string destino = @"C:
temp
tagged";
      if (Directory.Exists(destino) == false)
      {
      Directory.CreateDirectory(destino);
      }
      using (PdfReader leitor = new PdfReader(caminho))
      {
      StringBuilder texto = new StringBuilder();
```

```
if (File.Exists(destino + "//"
+ nomepdf.Replace(, ) + ".txt") == true)
{
File.Delete(destino + "//"+ nomepdf.Replace(, ) + ".txt");
}
arq = File.AppendText(destino + "//"
+ nomepdf.Replace(, ) + ".txt");
for (int i = 1; i <= leitor.NumberOfPages; i++)
{
string thePage = PdfTextExtractor.GetTextFromPage(leitor, i);
arq.WriteLine(thePage.ToUpper());
_msgclass = "Qtd de Pagina(s): "+ i;
OnMsg(new EventArgs());
Application.DoEvents();
}
arq.Dispose();
return textoCompleto.ToString();
}
}
catch (Exception ex)
{
throw new Exception(ex.Message
+ "Método: [ConvertePdfEmTexto] / Classe: [PDF]");
}
}
public void InsereArquivosTexto
(string textoConvertido, string nomepdf)
{
SQLServer db = new SQLServer();
```

```
try
{
string Qry;
db.ConectaDB();
Qry = "insert into Tb_ArquivoTexto(cTexto,
cNomePdf, dtRegistro) "+
"values ('"+ textoConvertido + "', '"
+ nomepdf + "', getdate() ) ";
db.ExecQry(Qry);
}
catch (Exception ex)
{
throw new Exception(ex.Message
+ "Método: [InsereArquivosTexto] / Classe: [PDF]");
}
finally
{
db.FechaDB();
}
}
}
}
```

que é transformado no formato texto, conforme linhas codificação em C#

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
```

```
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;
using ProjetoTreeTagger;
using ProjetoTreeTagger.DAL;
namespace ConvertePdfTexto
{
public partial class ConvertPdf : Form
{
public ToolStripProgressBar Pbar;
private PDF pdf = new PDF();
public ConvertPdf()
{
InitializeComponent();
pdf.Msg += new PDF.MsgPDF(Msgpdf);
}
void Msgpdf(object source, EventArgs e)
{
string mensagem;
PDF Proc = (PDF)source;
mensagem = Proc.MsgClassPDF.ToString();
lblMsg.Text = mensagem;
}
private void btnBuscaPDF_Click(object sender, EventArgs e)
{
this.ofd1.Multiselect = false;
this.ofd1.Title = "Selecionar PDF";
ofd1.InitialDirectory = @"C:
dados";
```

```
ofd1.Filter = "Files (*.PDF)|*.PDF|"+ "All files (*.*)|*.*";
ofd1.CheckFileExists = true;
ofd1.CheckPathExists = true;
ofd1.FilterIndex = 2;
ofd1.RestoreDirectory = true;
ofd1.ReadOnlyChecked = true;
ofd1.ShowReadOnly = false;
DialogResult dr = this.ofd1.ShowDialog();
if (dr == System.Windows.Forms.DialogResult.OK)
{
txtCaminhoPDF.Text = ofd1.FileName;
txtCaminhoPDF.Tag = ofd1.SafeFileName;
}
}
private void btnConverter_Click(object sender, EventArgs e)
{
try
{
if (txtCaminhoPDF.Text == )
{
throw new Exception("Informe um arquivo .PDF");
}
btnBuscaPDF.Enabled = false;
btnConverter.Enabled = false;
lblMsg.Text = "Aguarde...";
pdf.ConvertePdfEmTexto(txtCaminhoPDF.Text,
txtCaminhoPDF.Tag.ToString());
lblMsg.Text = ".";
MessageBox.Show("Fim processamento.", "Converter PDF",
```

```
MessageBoxButtons.OK, MessageBoxIcon.Information);
}
catch (Exception ex)
{
MessageBox.Show(ex.Message, "Converter PDF",
MessageBoxButtons.OK, MessageBoxIcon.Information);
}
finally {
btnBuscaPDF.Enabled = true;
btnConverter.Enabled = true;
}
}
}
}
```

11 APÊNDICE - COMO EXECUTAR TREE-TAGGER

```
using System;
using System.Collections.Generic;
using System.Diagnostics;
using System.IO;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;
using System.Data.SqlClient;
using System.Data;
using ProjetoTreeTagger.DAL;
using System.Threading;
namespace ProjetoTreeTagger.BLL
{
class treetagger
{
public event Msgtreetagger Msg;
public virtual void OnMsg(EventArgs e)
{
if (Msg != null)
Msg(this, e);
}
public delegate void Msgtreetagger(object source, EventArgs e);
```

```
      private string _msgclass = ;
      public string MsgClasstreetagger
      {
      get
      {
      return _msgclass;
      }
      }
      public string aplicaTreeTagger(string input, string idioma)
      {
      try
      {
      _msgclass = input + "Processando....";
      OnMsg(new EventArgs());
      Application.DoEvents();
      string caminhoTreeTagger =
      Path.GetDirectoryName(Application.ExecutablePath);
      string outputfilepath = input + "_tagged.txt";
      string filepath = input;
      string tagCommand;
      tagCommand = -token -lemma -no-unknown -sgml -proto
      -gramotron -base -pt-with-lemma "+ caminhoTreeTagger +
      @"
TreeTagger
lib¨ idioma;
      tagCommand +=  + filepath + ;
      tagCommand +=  + outputfilepath + ;
      Process myProcess = new Process();
      myProcess.StartInfo.Arguments = tagCommand;
      myProcess.StartInfo.UseShellExecute = false;
```

```
      myProcess.StartInfo.FileName = caminhoTreeTagger +
      @"
TreeTagger
bin
tree-tagger.exe ";
      myProcess.StartInfo.CreateNoWindow = true;
      myProcess.Start();
      myProcess.WaitForExit();
      myProcess.Close();
      int qtd = 0;
      while (File.Exists(outputfilepath) == false)
      {
      if (qtd == 10)
      { throw new Exception("ERRO no retorno do arquivo: "+ input);
      }
      Thread.Sleep(1000);
      qtd++;
      }
      return outputfilepath;
      }
      catch (Exception e)
      {
      throw new Exception(e.Message + "Método: "
      + System.Reflection.MethodBase.GetCurrentMethod().ToString()
      + "Classe: "+ this.GetType().Name);
      }
      }
      public void DeletaArquivoTagged(string arquivo)
      {
      SQLServer db = new SQLServer();
```

```
db.ConectaDB();
try
{
string StrQry =
"Delete Tb_ArquivosTagged where cNomeArquivoLimpo = '"
+ arquivo + "'";
db.ExecQry(StrQry);
}
catch (Exception ex)
{
throw new Exception(ex.Message + "Método: "
+ System.Reflection.MethodBase.GetCurrentMethod().ToString()
+ "Classe: "+ this.GetType().Name);
}
finally
{
db.FechaDB();
}
}
public void DeletaOracaoPorArq(string arquivo)
{
SQLServer db = new SQLServer();
db.ConectaDB();
try
{
string StrQry = "Delete Tb_Oracao where cNomeArq = '"
+ arquivo + "'";
db.ExecQry(StrQry);
}
```

```
      catch (Exception ex)
      {
      throw new Exception(ex.Message + "Método: "
      + System.Reflection.MethodBase.GetCurrentMethod().ToString()
      + "Classe: "+ this.GetType().Name);
      }
      finally
      {
      db.FechaDB();
      }
      }
      public void moveArquivoTagged
      (string pathArquivo, string nomeArquivo)
      {
      try
      { nomeArquivo = DateTime.Now.ToString().Replace(-", ).Replace(":",
      ).Replace(,).Replace("/", ) + "_"+ nomeArquivo;
      string pathImportados = @"C:
temp
tagged
importados";
      if (Directory.Exists(pathImportados) == false)
      {
      Directory.CreateDirectory(pathImportados);
      }
      File.Move(pathArquivo, pathImportados + "
"+ nomeArquivo);
      }
      catch (Exception ex)
      {
```

```
throw new Exception(ex.Message + "Método: "
+ System.Reflection.MethodBase.GetCurrentMethod().ToString()
+ "Classe: "+ this.GetType().Name);
}
finally
{
}
}
public void ImportaArquivoTagged
(string pathArquivo, string nomeArquivoTagged,
string cNomeArquivoLimpo)
{
SQLServer db = new SQLServer();
db.ConectaDB();
int i = 1;
try
{
String Linha;
SqlDataAdapter Adp = new SqlDataAdapter
("select top 0 * from Tb_ArquivosTagged", db.Cnx);
DataSet Ds = new DataSet();
Adp.SelectCommand.Transaction = db.Trans;
Adp.FillSchema(Ds, SchemaType.Source, "AP");
DataTable dt = Ds.Tables["AP"];
DataRow row;
double QtdLidos = 0;
using (StreamReader Arq = new StreamReader(pathArquivo,
System.Text.Encoding.Default))
{
```

```
Linha = Arq.ReadLine();
do
{
for (int x = 0; x < 500000; x++)
{
if (Linha == null)
{
break;
}
row = dt.NewRow();
try
{
string[] vSplit = Linha.Split('^');
if (Linha.Split('^').Length != 3)
{
throw new Exception("Erro: Arquivo fora do layout.
Certifique se o arquivo é válido.");
}
row["cPalavra"] = vSplit[0];
row["cPos"] = vSplit[1];
row["cLema"] = vSplit[2];
row["cNomeArquivoTagged"] = nomeArquivoTagged;
row["cNomeArquivoLimpo"] = cNomeArquivoLimpo;
row["dDtImportacao"] = DateTime.Now.ToString().Trim();
dt.Rows.Add(row);
QtdLidos++;
_msgclass = "Importando: "+ QtdLidos
+ - Arquivo: "+ cNomeArquivoLimpo;
OnMsg(new EventArgs());
```

```
Application.DoEvents();
i++;
}
catch (Exception ex)
{
throw new Exception("Erro: "+ ex.Message + "N° Linha: "+ i);
}
Linha = Arq.ReadLine();
}
db.BulkInsert(dt, "Tb_ArquivosTagged", db.Trans);
dt.Dispose();
dt.Clear();
}
while (Linha != null);
}
db.Trans = null;
}
catch (Exception ex)
{
throw new Exception(ex.Message + "Método: "
+ System.Reflection.MethodBase.GetCurrentMethod().ToString()
+ "Classe: "+ this.GetType().Name);
}
finally
{
db.FechaDB();
}
}
public bool consultaArquivoImportadoAnteriormente
```

```
(string nomeArquivo)
{
SQLServer db = new SQLServer();
db.ConectaDB();
try
{
SqlDataReader DR = db.DR
("Select * from Tb_ArquivosTagged where "
+ "cNomeArquivoLimpo = '"+ nomeArquivo + "'");
if (DR.HasRows)
{
DR.Dispose();
return true;
}
else
{
DR.Dispose();
return false;
}
}
catch (Exception ex)
{
throw new Exception(ex.Message
+ "Método: [consultaArquivoImportadoAnteriormente]
Classe: [treetagger]");
}
finally
{
db.FechaDB();
```

```
      }
      }
      public string QuebraArquivoTextoPorPalavra
      (string caminho, string nomeTxt)
      {
      try
      {
      StreamWriter arq;
      string[] lines = System.IO.File.ReadAllLines(caminho);
      _msgclass = "Processando....";
      OnMsg(new EventArgs());
      Application.DoEvents();
      string destino = @"C:
temp
tagged";
      string[] palavras;
      string teste = ;
      string Oracao = ;
      if (Directory.Exists(destino) == false)
      {
      Directory.CreateDirectory(destino);
      }
      StringBuilder texto = new StringBuilder();
      if (File.Exists(destino + "
"+ nomeTxt.Replace(, )
      + "_Separado.txt") == true)
      {
      File.Delete(destino + "
"+ nomeTxt.Replace(, )
      + "_Separado.txt");
```

```
      }
      arq = File.AppendText(destino + "
"
      + nomeTxt.Replace(, ) + "_Separado.txt");
      int qtdPalavra = 0;
      int qtdLinha = 0;
      foreach (string line in lines)
      {
      qtdLinha++;
      palavras = line.Trim().Split(' ');
      foreach (var palavra in palavras)
      {
      if (palavra.Contains(".") == true
      || palavra.Contains(":") == true
      || palavra.Contains("...") == true
      || palavra.Contains("!") == true
      || palavra.Contains("?") == true
      || palavra.Contains(";") == true)
      {
      Oracao = Oracao + palavra;
      Oracao = ;
      }
      else
      {
      Oracao = Oracao + palavra + ;
      }
      teste = System.Text.RegularExpressions.Regex.Replace
      (palavra,
      @"[â-zA-ZéúíóáÉÚÍÓÁèùìòàÈÙÌÒÀõãñÕÃÑêûîôâÊÛÎÔÂëÿüïöäËYÜÏÖÄçÇ
s]
```

```
      + ?", string.Empty);
      if (teste != )
      {
      arq.WriteLine(teste.ToUpper());
      qtdPalavra++;
      _msgclass = "Qtd de Linha(s): "+ qtdLinha
      + Environment.NewLine + "Qtd de Palavra(s): "+ qtdPalavra;
      OnMsg(new EventArgs());
      Application.DoEvents();
      }
      }
      }
      arq.Dispose();
      return destino + "
"+ nomeTxt.Replace(, ) + "_Separado.txt";
      }
      catch (Exception ex)
      {
      throw new Exception(ex.Message +
      "Método: [QuebraArquivoTextoPorPalavra] / Classe: [treetagger]");
      }
      }
      public void InsereOracao(string Oracao, string nomeArquivo)
      {
      SQLServer db = new SQLServer();
      try
      {
      string Qry;
      db.ConectaDB();
```

```
Qry = "insert into Tb_Oracao(cOracao, cNomeArq, dDtRegistro) "
+ "values ('"+ Oracao + "', '"+ nomeArquivo + "',
getdate() ) ";
db.ExecQry(Qry);
}
catch (Exception ex)
{
throw new Exception(ex.Message
+ "Método: [InsereOracao] / Classe: [treetagger]");
}
finally
{
db.FechaDB();
}
}
}
}
```

12 APÊNDICE - COMO EXECUTAR O SOFTWARE TREETAGGER

/***/

/* How to use the TreeTagger */

/* */

/* Author: Helmut Schmid, CIS, Ludwig-Maximilians-Universität,

Germany */

/***/

The TreeTagger consists of two programs: the training program creates a parameter file from a fullform lexicon and a handtagged corpus. The tagger program reads the parameter file and annotates the text with part of speech and lemma information. Both programs print information about their usage when they are called without arguments.

Tagging ——-

Tagging is done with the *tree-tagger* program.

The first argument is the name of a parameter file which was generated with the train-tree-tagger program. Parameter files generated on different platforms or with older versions of train-tree-tagger will not work.

The second argument is the input file. It must be in one-word-per-line format, i.e. each line contains one token (word, punctuation character or parenthesis) and should not exceed 1000 characters. Tokens may contain blanks. It is possible to override the lexical information contained in the parameter file of the tagger by specifying a list of possible tags after the token. This list has to be preceded by a tab character and the elements are separated by tab characters. Pretagging could be used e.g. to ensure that certain text-specific expressions are tagged correctly. Clitics (like "'s", "'re", and "'d"in English or -la"and -t-elle"in French) have to be separated if they were separated in the training data. (The French and English parameter files available by ftp expect separation of clitics).

Sample input file: He moved to New York City NP .

The third argument is the name of the output file. The output is also in one-word-

per-line format. Depending on the specified options, it will contain columns with tokens, tags and lemmas. If the third argument is missing, the output will be printed to standard output. If the second argument is missing, too, input is read from standard input.

Options:

-token: Prints the token as well. -lemma: Prints the lemma as well. -sgml: Don't tag SGML annotations, i.e. lines starting with '¡' and ending with '¿'. -threshold ¡p¿: Print all tags with a probability higher than ¡p¿ times the probability of the best tag. -prob: Print tag probabilities (requires option -threshold) -no-unknown: Print the token rather than ¡unknown¿ for unknown lemmas -quiet: Don't print status messages -pt-with-lemma: If this option is specified, then each pretagging tag (see above) has to be followed by a whitespace and a lemma. -pt-with-prob: If this option is specified, then each pretagging tag (see above) has to be followed by whitespace and a tag probability value. If -pt-with-prob and -pt-with-lemma have been specified, then each pretagging tag is followed by a probability and a lemma in that order. -files f: Read the names of input and output files pairwise from the file f. The format of f is the lexicon file format described below. -lex f: Read auxiliary lexicon entries from the file f. -eos-tag ¡tag¿: The SGML tag ¡tag¿ signals the end of a sentence. This option implies the option -sgml

Some more exotic options: -proto: Print lexical information for each word The lexicon type is signalled by one of the characters f: The word was found in the full form lexicon. c: The word in lowercase was found in the lexicon h: The word contains an hyphen and the word following the hyphen was found in the full form lexicon; e.g. instead of "table-wine"only "wine"has been found. s: The word has been looked up in the suffix lexicon p: Tags have been assigned by pretagging. -gramotron: Same as -proto but with a different format -proto-with-prob: Same as -proto but with lexical tag probabilities -print-prob-tree: Print the transition probability tree and exit -eps ¡epsilon¿: Value which is used to replace zero lexical frequencies. Zero frequencies occur when a word/tag pair is contained in the lexicon but not in the training corpus. The default is 0.1. -base: Use only lexical probabilities for tagging. This option is only useful to obtain a baseline result to which the actual tagger output is compared.

Training ———

Training is done with the *train-tree-tagger* program. If the program is called without arguments, the following output is printed:

USAGE: train-tree-tagger ¡lexicon¿ ¡open class file¿ ¡infile¿ ¡outfile¿ -cl ¡context length¿ -dtg ¡min. decision tree gain¿ -ecw ¡eq. class weight¿ -atg ¡affix tree gain¿ -st ¡sent. tag¿

Description of the command line arguments: * ¡lexicon¿: name of a file which

contains the fullform lexicon. Each line of the lexicon corresponds to one word form and contains the word form itself followed by a Tab character and a sequence of tag-lemma pairs. The tags and lemmata are separated by whitespace.

Example: aback RB aback abacuses NNS abacus abandon VB abandon VBP abandon abandoned JJ abandoned VBD abandon VBN abandon abandoning VBG abandon

Important: Ordinal and cardinal numbers which consist of digits should not be included in the lexicon. Otherwise, the tagger will not be able to learn how to tag numbers which are not listed in the lexicon. Numbers with unusual tags should be added to the lexicon, however.

Remark: The tagger doesn't need the lemmata for tagging. If you do not have the lemma information or if you do not plan to annotate corpora with lemmas, you can replace the lemma with a dummy value, e.g. -".

* ¡open class file¿: name of a file which contains a list of open class tags i.e. possible tags of unknown word forms. This information is needed to estimate likely tags of unknown words. This file would typically contain adverb, adjective, noun, proper name and perhaps verb tags, but not prepositions, determiners, pronouns or numbers. * ¡input file¿: name of a file which contains tagged training data. The data must be in one-word-per-line format. This means that each line contains one token and one tag in that order separated by a tabulator. Punctuation marks are considered as tokens and must have been tagged as well.

Example: Pierre NP Vinken NP , , 61 CD years NNS

* ¡output file¿: name of the file in which the resulting tagger parameters are stored.

The following parameters are optional:

* -cl ¡context length¿: number of preceding words forming the tagging context. The default is 2 which corresponds to a trigram context. For small training corpora and/or large tagsets, it could be useful to reduce this parameter to 1. * -dtg ¡min. decision tree gain¿: Threshold - If the information gain at a leaf node of the decision tree is below this threshold, the node is deleted. The default value is 0.7. * -ecw ¡eq. class weight¿: weight of the equivalence class based probability estimates. The default is 0.15. * -atg ¡affix tree gain¿ Threshold - If the information gain at a leaf of an affix tree is below this threshold, it is deleted. The default is 1.2. * -st ¡sent. tag¿: the end-of-sentence part-of-speech tag, i.e. the tag which is assigned to sentence punctuation like ".", "!", "?". Default is "SENT". It is important to set this option properly, if your tag for sentence punctuation is not "SENT".

The accuracy of the TreeTagger usually improves a bit, if different settings of the

above parameters are tested and the best combination is chosen.