PUCRS

FACULDADE DE INFORMÁTICA
PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO
MESTRADO EM CIÊNCIA DA COMPUTAÇÃO

CARLOS ALBERTO DOS SANTOS

UMA ANÁLISE COMPARATIVA ENTRE AS ABORDAGENS LINGUÍSTICA E ESTATÍSTICA PARA EXTRAÇÃO AUTOMÁTICA DE TERMOS RELEVANTES DE CORPORA

Porto Alegre
2018

PÓS-GRADUAÇÃO - *STRICTO SENSU*

Pontifícia Universidade Católica
do Rio Grande do Sul

Ficha Catalográfica

S237a Santos, Carlos Alberto dos

Uma análise comparativa entre as abordagens linguística e estatística para extração automática de termos relevantes de corpora / Carlos Alberto dos Santos . – 2018.

99 p.

Dissertação (Mestrado) – Programa de Pós-Graduação em Ciência da Computação, PUCRS.

Orientadora: Profa. Dra. Renata Vieira.

1. Extração de termos. 2. Mineração de texto. 3. Métricas estatísticas. 4. Extração estatística. 5. Extração linguística. I. Vieira, Renata. II. Título.

Elaborada pelo Sistema de Geração Automática de Ficha Catalográfica da PUCRS com os dados fornecidos pelo(a) autor(a).

Bibliotecária responsável: Salete Maria Sartori CRB-10/1363

Carlos Alberto dos Santos

Uma análise comparativa entre as abordagens linguística e estatística para extração automática de termos relevantes de corpora

Dissertação apresentada como requisito parcial para obtenção do grau de Mestre em Ciência da Computação do Programa de Pós-Graduação em Ciencia da Computação, Escola Politécnica da Pontifícia Universidade Católica do Rio Grande do Sul.

Aprovado em 27 de abril de 2018.

BANCA EXAMINADORA:

Prof. Dr. Leandro Krug Wives (INF/UFRGS)

Profa. Dr. Rafael Heitor Bordini (PPGCC/PUCRS)

Profa. Dra. Renata Vieira (PPGCC/PUCRS - Orientadora)

"No meio da confusão, encontre a simplicidade. A partir da discórdia, encontre a harmonia. No meio da dificuldade reside a oportunidade."
(Albert Einstein)

AGRADECIMENTOS

Obrigado primeiramente a Deus, que me criou e capacitou a atingir meus objetivos acadêmicos. Meu muito obrigado a minha família que tanto amo, João, Ivanete, Isabel, Sammis, Tiago, Marcina, Luis Manoel, Luciane, Augustus, Arthur, Afonso, Darci, Zilá, Francisco, Vera, Daniela, Alessandro, Daiana, Ivana, Fábio, Débora, Ana, Maria Helena, Luis Alberto, Eunice, Wilson, Vitor, Camila, Jarbas e Luiza.

Gratidão aos amigos de todas as horas, Erich, Rodney, Williams, Oldoni, Maria Tereza, Camila, Fran, Josi, Marina, André, Ebina, Igor, Adri, Leandro, Marcelo, Letícia, Paulo, Ramon, Ricardo, Eduardo, Léo, Denise, Bruno, Alex, Alberto, Silvana.

Um obrigado especial a Paulo, Lucelene, Sandro, Débora, Heloísa, e minha querida orientadora Renata.

Agradeço a minha namorada Débora pelo companheirismo, compreensão e amor durante esses dois anos dedicados ao mestrado.

Obrigado aos membros da minha banca examinadora, à FACIN, e à CAPES pelos recursos financeiros.

UMA ANÁLISE COMPARATIVA ENTRE AS ABORDAGENS LINGUÍSTICA E ESTATÍSTICA PARA EXTRAÇÃO AUTOMÁTICA DE TERMOS RELEVANTES DE CORPORA

RESUMO

Sabe-se que o processamento linguístico de *corpora* demanda grande esforço computacional devido à complexidade dos seus algoritmos, mas que, apesar disso, os resultados alcançados são melhores que aqueles gerados pelo processamento estatístico, onde a demanda computacional é menor. Esta dissertação descreve uma análise comparativa entre os processos linguístico e estatístico de extração de termos. Foram realizados experimentos através de quatro *corpora* em língua inglesa, construídos a partir de artigos científicos, sobre os quais foram executadas extrações de termos utilizando essas abordagens. As listas de termos resultantes foram refinadas com o uso de métricas de relevância e *stop list*, e em seguida comparadas com as listas de referência dos *corpora* através da técnica do *recall*. Essas listas, por sua vez, foram construídas a partir do contexto desses *corpora* e com ajuda de pesquisas na Internet. Os resultados mostraram que a extração estatística combinada com as técnicas da *stop list* e as métricas de relevância pode produzir resultados superiores ao processo de extração linguístico refinado pelas mesmas métricas. Concluiu-se que a abordagem estatística composta por essas técnicas pode ser a opção ideal para extração de termos relevantes, por exigir poucos recursos computacionais e por apresentar resultados superiores àqueles encontrados no processamento linguístico.

Palavras Chave: extração de termos; mineração de texto; lista de referência; *stop list*; métricas estatísticas; extração linguística; extração estatística.

A COMPARATIVE ANALYSIS BETWEEN THE STATISTICAL AND LINGUISTIC APPROACHES TO AUTOMATIC EXTRACTION OF RELEVANT TERMS OF CORPORA

ABSTRACT

It is known that linguistic processing of corpora demands high computational effort because of the complexity of its algorithms, but despite this, the results reached are better than that generated by the statistical processing, where the computational demand is lower. This dissertation describes a comparative analysis between the process linguistic and statistical of term extraction. Experiments were carried out through four corpora in English idiom, built from scientific papers, on which terms extractions were carried out using the approaches. The resulting terms lists were refined with use of relevance metrics and stop list, and then compared with the reference lists of the corpora across the recall technical. These lists, in its turn, were built from the context these corpora, whith help of Internet searches. The results shown that the statistical extraction combined with the stop list and relevance metrics can produce superior results to linguistic process extraction using the same metrics. It's concluded that statistical approach composed by these metrics can be ideal option to relevance terms extraction, by requiring few computational resources and by to show superior results that found in the linguistic processing.

Keywords: term extraction; text mining; reference list; stop list; statistical metrics; linguistic extraction; statistical extraction.

LISTA DE FIGURAS

LISTA DE TABELAS

LISTA DE SIGLAS

CSV – Comma-separated values

DC – Domain coherence

EXATO – Extrator de Termos para Ontologias

HTML – HyperText Markup Language

TF – Term frequency

TF-DCF – term frequency, disjoint *corpora* frequency

TF-IDF – Term frequency and inverse document frequency

TF-IDF – Term frequency, inverse domain frequency

TDS – Term domain specificity

THD – Termhood

NSP – Ngram Statistics Package

XML – eXtensible Markup Language

SUMÁRIO

1. INTRODUÇÃO

Uma das aplicações do Processamento de Linguagem Natural (PLN) é a extração de informações a partir de textos. Desde a difusão da Internet a partir dos anos 1990, e o aumento considerável de textos disponíveis nesse meio, os esforços de PLN passaram a se concentrar mais nas tarefas de extração, com o objetivo de estruturar a informação disponível nos textos, e assim facilitar o acesso a essas fontes.

Uma das importâncias desse tipo de processamento está relacionada ao apoio na construção de ontologias, que podem ser entendidas como um conjunto de conceitos organizados hierarquicamente, um conjunto de relações, e um conjunto de atributos [19]. Ontologia, segundo Gruber [13], é uma forma de estruturar informações para representar conhecimento. No entanto, isso torna-se um grande desafio frente à quantidade enorme de dados textuais a serem estruturados.

A utilização de ontologias está bastante difundida nos esforços para organizar e catalogar a informação disponível na Web. Segundo a W3C [7], a Web semântica, nome dado a esse processo, busca organizar a informação disponível na Web, de forma a possibilitar acesso padronizado e rápido aos recursos disponíveis. Há diversos padrões e linguagens que foram definidos para possibilitar essa tarefa, e todos são possíveis de serem adaptados para aplicação em outros cenários.

Dentre os tipos de recuperação de informação em textos, a busca por termos em *corpora* de domínio é uma das principais aplicações de PLN [19]. Para essa tarefa há diversas técnicas estatísticas [5, 14, 17, 21, 25] e heurísticas linguísticas [4, 15, 21] capazes de apoiar a busca por termos relevantes. Softwares disponíveis no mercado [22, 26] implementam esse arcabouço de técnicas a fim de facilitar o processamento dos textos.

De maneira geral, há dois caminhos que podem ser seguidos para extração automática de termos para apoiar a construção de ontologias. São eles: estatístico e linguístico. O primeiro utiliza-se de modelos matemáticos e métricas para calcular, por exemplo, a frequência com que os termos são repetidos em um *corpus*. Já o segundo, apoia-se em recursos gramaticais para classificar as palavras e então filtrar os sintagmas que possam ser mais relevantes para o contexto.

Para Kurdi [18], o uso de bons recursos linguísticos é indispensável para o desenvolvimento de sistemas de processamento de linguagem natural. Devido à enorme quantidade de informação a ser processada, a utilização de métodos estatísticos se fez necessária para produzir resultados mais robustos. O uso combinado dos métodos linguístico e estatístico no início dos anos 2000 desfez a ruptura histórica entre essas abordagens, e ainda gerou como consequência a adoção de algoritmos de aprendizado de máquina que atualmente vêm produzindo resultados cada vez melhores de extração de informações em textos [36].

1.1 Motivação

O processo de extração de termos pode ser feito de maneira manual ou automatizada. Graças aos avanços da computação hoje há uma gama de softwares capazes de processar textos tanto de maneira estatística quando linguística, minimizando assim a participação de conhecedores da língua ou do domínio dos textos durante o processo de levantamento de conceitos de *corpora*.

Para que seja possível a execução da extração linguística é essencial que o software seja capaz de interpretar o idioma do texto. Ou seja, o sistema precisa estar programado para interpretar gramaticalmente as entradas de texto, da mesma forma como um ser humano o faria se estivesse lendo esse mesmo texto.

O fato é que esse tipo de processamento computacional demanda muitos recursos, pois necessita que o software utilize inúmeras heurísticas textuais para resolver problemas de interpretação, ambiguidade lexical e combinações.

Por outro lado, há os softwares que fazem a extração estatística de termos, onde não existe dependência de idioma, já que o processamento leva em consideração a contagem dos termos de maneira probabilística. Os resultados desse tipo de processamento costumavam ter qualidade inferior ao processamento linguístico, já que muitos termos com pouca importância linguística acabavam sendo considerados como importantes no contexto.

Alguns trabalhos recentes mostram que ferramentas estatísticas de extração de contextos (como *Word2Vec*, por exemplo) podem retornar bons resultados quando utilizadas com o objetivo de extração de termos. Porém, para essa dissertação, foi desconsiderado o uso de ferramentas de extração de contextos, focando apenas em ferramentas que fazem a *tokenização* de caracteres estatisticamente.

Para o processamento estatístico a demanda por recursos computacionais é inferior. Desta forma não há necessidade que o usuário tenha conhecimento acerca do idioma que está processando, bem como a participação de linguistas no processo pode ser minimizada.

O uso de métricas de relevância, como as detalhadas na seção 2.4, pode auxiliar no processo de refinamento de conceitos de *corpora*. Essas medidas podem ser aplicadas a listas de termos e suas frequências, buscando eliminar termos que não tenham relevância para o domínio do texto.

A principal motivação deste trabalho é apresentar dados que comprovem, ou não, que a extração estatística pode apresentar resultados semelhantes à extração linguística quando refinada por uma métrica de relevância. De maneira complementar, busca-se comparar as listas de termos geradas por ambas as abordagens, com e sem o uso das métricas citadas.

Através da pesquisa do referencial teórico foi possível perceber que há inúmeros métodos estatísticos disponíveis para PLN mas poucos trabalhos com o objetivo de compará-los. Da mesma forma, não foi localizado nenhum trabalho atual capaz de comparar as extrações realizadas pelas abordagens linguística e estatística a partir de um mesmo conjunto de *corpora*.

1.2 Objetivos

O objetivo deste trabalho é analisar a eficácia das métricas de relevância sobre a extração de termos, observando os resultados obtidos da sua aplicação em abordagens estatística e linguística.

Alguns trabalhos indicam [20] que a abordagem puramente linguística é capaz de apresentar melhores resultados que a abordagem estatística quando o objetivo é identificar conceitos em *corpora* [23]. Ou seja, a lista de termos resultantes da extração linguística é mais próxima qualitativamente da lista de referência de um determinado *corpora*.

O que se pretende com este trabalho é aplicar as métricas de relevância nas listas de termos já extraídas pelas abordagens linguística e estatística, procurando avaliar a quantidade e a qualidade dos termos resultantes antes e depois do uso dessas métricas.

Com base na leitura de trabalhos anteriores espera-se que a aplicação das medidas de relevância seja capaz de melhorar a qualidade da lista de termos relevantes. A fim de ampliar o estudo foi aplicada também a extração de termos com apoio de *stop list*, por se tratar de uma prática comum nos casos de extração de termos relevantes de corpora para abordagens estatísticas. Foram geradas análises, relatórios e gráficos qualitativos e quantitativos para todas as extrações realizadas: linguística (L), estatística (E), estatística com *stop list* (E+*stop*), linguística com métricas (L+M) e estatística com métricas (E+M).

As quatro primeiras extrações (E, E+*stop*, L e L+M) são obtidas a partir dos experimentos realizados com auxílio dos trabalhos de referência. Já a quinta e última extração (E+M) é a principal hipótese de pesquisa que está sendo testada neste trabalho. Foram aplicadas também, as mesmas métricas sobre as listas estatísticas filtradas pela *stop list*, gerando novas listas (E+*stop*+M).

A expectativa é que a combinação da análise estatística com as métricas de relevância possa apresentar uma qualidade bastante próxima da extração linguística com as mesmas métricas. É exatamente este processo que foi investigado de forma qualitativa e quantitativa em relação aos demais processos.

Nesse sentido, para realizar o objetivo geral desta dissertação foi necessário atingir os seguintes objetivos específicos:

- mostrar baseado na literatura e experimentos que a relação das quatro primeiras extrações (E, E+*stop*, L e L+M) é verdade para os *corpora* em estudo;
- avaliar se a extração estatística com métricas (E+M) apresenta o comportamento esperado;
- avaliar o comportamento da aplicação das métricas de relevância sobre as listas estatísticas filtradas por *stop list* (E+*stop*+M).

1.3 Metodologia

Sob o enfoque metodológico, este trabalho segue o padrão de pesquisa exploratória, cuja análise foi empírica, ou seja, ocorreram uma série de experimentos seguidos de conclusão. Foram percorridas duas etapas fundamentais para o alcance do objetivo proposto.

Na primeira etapa, realizou-se uma revisão bibliográfica sobre a extração de termos com valor conceitual em *corpora*, procurando apresentar teoricamente as abordagens linguísticas e estatísticas que apoiam essa atividade (capítulo 2).

Para o processo linguístico foram detalhados os níveis de análise linguística e as contribuições da computação para o processamento de textos (seção 2.3). Também foi importante apresentar os trabalhos de Lopes *et al.* [23] que deram origem ao software *ExATO* (ferramenta de extração automática de termos) baseado em heurísticas com fundamentação linguística (seção 2.3.5).

Para o processo estatístico foram destacados os desafios para a extração estatística, suas vantagens e os detalhes sobre as métricas de relevância que podem ser aplicadas (seção 2.4). Foi apresentado também um overview sobre o software NSP (*Ngram Statistics Package*) a fim de preparar o leitor para melhor compreender os detalhes do processo de extração estatístico (seção 2.4.5).

Na segunda etapa, de cunho prático, foi inicialmente detalhado o experimento da pesquisa, definindo os *corpora* objeto deste estudo, e as listas de referência construídas a partir do domínio e dos documentos desses *corpora*. Essas listas serviram de base comparativa para todos os processamentos realizados nesta pesquisa (capítulo 4).

Ainda na etapa prática da pesquisa, foram apresentados os resultados obtidos das extrações linguística e estatística sem o uso das métricas de relevância (capítulo 5). Foi apresentada a lista de termos relevantes obtida pela extração linguística realizada pelo *parser* textual e o software *ExATO*, e em seguida os termos identificados na extração estatística com o software NSP (seção 4.1).

De posse desses resultados, foram aplicadas as métricas de relevância (*tf-idf*, *tf-dcf*) nestas listas procurando localizar termos mais aderentes ao domínio do *corpora*. Em seguida foi também utilizada a técnica da *stop list* nas listas de termos geradas pela extração estatística. Assim, foram aplicadas as mesmas métricas sobre essas novas listas filtradas.

Os resultados obtidos foram compilados e discutidos a fim de se avaliar a melhora ou piora na qualidade das listas de termos após a aplicação das métricas de relevância, buscando detalhar qual abordagem apresenta melhores resultados e sobre quais circunstâncias (seção 5.3).

Nota-se que a avaliação dos resultados obtidos pela aplicação das métricas foram medidos com base na comparação realizada com a lista de referência gerada para cada corpus (APÊNDICE B, APÊNDICE C, APÊNDICE D e APÊNDICE E).

De forma sequencial, estas foram as atividades realizadas:

- construção das listas de termos de referência para os *corpora* escolhidos;

- construção da *stop list* (APÊNDICE A);
- apresentação dos resultados dos processos linguístico e estatístico realizados sobre um conjunto de *corpora* de domínio específico;
- aplicação da *stop list* sobre as listas geradas pelo processamento estatístico;
- aplicação das métricas de relevância nas listas de termos geradas pela etapa inicial de extração, procurando observar qual medida é capaz de fornecer melhores resultados na comparação com a lista de referência de cada corpus;
- produção da análise quantitativa dos termos resultantes com cada abordagem, procurando comparar o uso das abordagens linguística e estatística com ou sem as métricas de relevância e *stop list*.
- conclusão se o uso das métricas de relevância e *stop list* pode melhorar ou piorar a qualidade dos termos extraídos, buscando eleger a melhor abordagem (linguística ou estatística) para identificar termos relevantes de um *corpus*.

2. CENÁRIO E CONTEXTUALIZAÇÃO

A partir deste capítulo serão apresentados os conceitos necessários para contextualização deste trabalho. Para tanto, inicia-se com um breve resumo sobre algumas definições importantes (2.1). Em seguida, na subseção 2.2 apresenta-se o processo de extração de termos de *corpora*, sua finalidade e os desafios em conseguir resultados satisfatórios com esse processo; posteriormente são discutidas as abordagens linguística e estatística utilizadas como ferramentas pelo processo de extração de termos.

Para o processo linguístico, subseção 2.3, será detalhada a maneira como as sentenças textuais são analisadas, e a forma como os softwares de *parser* funcionam. Em seguida, na subseção 2.4 será apresentada a extração estatística de termos, procurando também listar os softwares estatísticos utilizados para extração. Serão discutidas também as métricas utilizadas para o refinamento de conceitos de domínio; enfim, na subseção 3, serão listados e discutidos os trabalhos relacionados que são referência para o desenvolvimento desta pesquisa.

2.1 Definições importantes para o trabalho

Para que haja um melhor entendimento sobre o trabalho é importante definir alguns conceitos que foram utilizados durante a pesquisa. Ao longo do texto é abordada a extração de termos em corpora. Mas o que são termos? Para responder de maneira formal a essa pergunta recorreu-se ao Dicionário Priberam da língua portuguesa, procurando uma definição inicial. Para o dicionário, uma das definições da palavra "termo", sendo essa a que cabe ao contexto desta pesquisa, é a de "palavra ou vocábulo" [30].

Alguns autores, [10,19], classificam os termos em simples e compostos. Sendo os primeiros termos com uma única palavra, e os compostos com duas ou mais palavras. Os esforços iniciais em PLN para extração de termos resultaram na obtenção de termos simples [14]. Anos mais tarde, durante a década de 1980 os primeiros resultados da extração de termos compostos começaram a aparecer [33]. Atualmente, muitas pesquisas ainda são realizadas para aperfeiçoar esse processo, e todas dividem-se entre meios linguísticos e estatísticos.

Ainda nessa linha de orientar o leitor sobre as definições básicas para o trabalho, é importante compreender o significado da palavra "conceito", e qual sua relação com termos, simples ou compostos. Segundo o dicionário [29], conceito pode ser entendido como a concepção compreendida numa palavra que designa características e qualidades de uma classe de objetos, abstratos ou concretos. Ou seja, um termo que é conceito está sempre relacionado a algo, com o intuito de caracterizar essa coisa, abstrata ou concreta.

O processo de extrair termos de corpora busca localizar termos que são conceitos, e que podem ser utilizados para os mais diversos fins, como a construção de ontologias. Os meios utilizados para realizar essa extração são diversos, e esta pesquisa procurou avaliar alguns deles. A fim de

possibilitar uma comparação entre eles é indispensável comparar os termos extraídos com uma lista de termos de referência. Nesse contexto, esses termos de referência são os conceitos, levantados inicialmente com o intuito de validar as técnicas de extração estudadas. Num cenário real de extração não haveria essa lista referencial para comparação, e o resultado poderia ser a adoção dos termos levantados pela extração como termos conceituais.

2.2 Extração de Termos de Corpora

A extração dos termos de *corpora* pode se dar de duas formas: utilizando a abordagem linguística (subseção 2.3) ou estatística (subseção 2.4). Segundo Lopes [19], o uso dessas técnicas envolve uma certa divergência por parte da comunidade. Pois uma parcela dos cientistas optou pelo uso de linguística teórica e outros optaram por métodos estatísticos. Infelizmente, cada uma dessas partes rechaçava os métodos da outra parte, prejudicando a integração dessas duas abordagens.

A partir dos anos 2000 ocorreu uma integração dessas abordagens com o objetivo de viabilizar o processamento do grande volume de conteúdo não estruturado produzido e armazenado na Internet. Utilizar somente meios linguísticos para processar esse conteúdo se tornara inviável pela alta complexidade e demanda de recursos computacionais. Nesse sentido, a adoção de meios estatísticos passou a permitir fazer o processamento de grandes volumes de textos em menor tempo, porém com qualidade inferior.

Uma das formas de se combinar as abordagens linguística e estatística para fazer a extração de termos em texto é através da técnica de contraste de *corpora*, que consiste em comparar as frequências de ocorrência dos termos entre os documentos. Os corpora podem ser anotados por um software analisador sintático (descrito na seção 2.3.4), seus termos extraídos desse resultado e então aplicada uma métrica estatística de relevância (seção 2.4) para obter uma lista de termos candidatos a conceito para cada domínio.

Considerando que se deseja identificar os termos relevantes de um corpus *c1* pertencente a um domínio qualquer como o da Biologia utiliza-se então um ou mais *corpora* contrastantes de outros domínios (*c2* sobre política e *c3* sobre esportes, por exemplo) a fim de identificar os termos específicos daquele primeiro corpus *c1*. Utilizar um grande número de *corpora* contrastantes pode não produzir um resultado ideal, uma vez que prejudica o refinamento dos termos.

As seções seguintes procuram apresentar os detalhes das abordagens de extração de termos (linguística e estatística), destacando seus pontos fortes e fracos, bem como apresentando os softwares disponíveis na comunidade que podem auxiliar nas suas implementações.

2.3 Abordagem Linguística

As subseções 2.3.1, 2.3.2 e 2.3.3 detalham brevemente as subáreas da linguística, importantes para contextualização dos diferentes níveis de processamento em PLN. Já na subseção 2.3.5 apresenta-se o software ExATO, concebido por Lopes *et al.* [22], utilizado para extração de termos de *corpora* em inglês ou português.

2.3.1 Sintaxe

A sintaxe é o estudo dos princípios e processos pelos quais as sentenças são construídas em determinados idiomas [8]. Sua função é estudar o arranjo, combinação ou disposição das palavras na frase, a fim de construir uma gramática que pode ser vista como um dispositivo para produção de frases na língua de análise.

Esta foi a primeira área da linguística a receber atenção da comunidade da computação. Como consequência, é hoje o campo mais maduro de estudo da computação na linguística, tendo inúmeras publicações e softwares disponíveis em diversos idiomas para processamento de sentenças [3].

2.3.2 Semântica

A semântica é a área da linguística que estuda o significado das palavras e da interpretação das frases e dos enunciados, ou seja, a análise da significação das línguas naturais. Segundo Bates *et al.* [3] no livro publicado em 1993, o progresso dos estudos no campo da semântica em PLN tem ocorrido com foco em aplicações restritas a um domínio. Ao se considerar que uma palavra pode ter significados distintos dependendo do contexto onde é inserida, fica notável o desafio da computação em resolver esse problema.

Mais atualmente muitos trabalhos procuram resolver o problema da semântica através de aprendizado de máquina, utilizando-se da construção de redes neurais por exemplo. O trabalho de Bowman *et al.* [6] apresenta a construção de uma rede neural capaz de aprender lógica semântica de linguagem natural. Os trabalhos de Socher *et al.* [31, 35] apresentam o uso da inteligência artificial na tentativa de resolver os problemas de ambiguidade natural entre as sentenças. As pesquisas sobre o assunto mostram que este é um campo de estudo presente no estado da arte em PLN.

2.3.3 Pragmática

Enquanto a semântica e a sintaxe constituem a construção teórica da língua, a pragmática engloba o estudo da linguagem comum e o uso concreto da linguagem. Em outras palavras, a pragmática estuda os significados linguísticos determinados não exclusivamente pela semântica proposicional ou frásica, mas aqueles que se deduzem a partir de um contexto extralinguístico.

Para Bates et al. [3], ao contrário das linguagens de programação, onde a escrita define um único caminho a ser seguido, o contexto tudo permeia e tem muito poder em linguagens naturais. Dessa forma, o torna fundamental para comunicar informações substanciais com poucas palavras. Da mesma forma que a semântica, a pragmática também é uma área que tem recebido a atenção dos pesquisadores em PLN.

2.3.4 *Analisadores Sintáticos*

Os softwares que fazem a análise sintática (*parsers*) são os responsáveis pela tarefa de anotação sintática em textos. Isso significa que esses aplicativos têm por função descrever a estrutura gramatical das frases, identificando as palavras de acordo com suas funcionalidades sintáticas. O desenvolvimento desses softwares foi um dos maiores avanços no processamento de linguagem natural na década de 1990.

Para textos em língua portuguesa, a tarefa de anotação textual pode ser realizada com o auxílio da ferramenta PALAVRAS, com formato de saída XML, concebida por Bick no ano 2000 [4]. Os trabalhos de Bick têm muita relevância nos estudos de processamento da língua portuguesa. A ferramenta desenvolvida por ele cumpre muito bem o papel de anotação sintática em textos em português.

Para o inglês existe uma gama de softwares capazes de fazer a anotação sintática textual, cada qual com suas heurísticas que diferenciam um dos outros. Para este trabalho, as anotações de textos em língua inglesa foram realizadas com apoio da ferramenta *Stanford Parser*, versão 3.5.2, com formato de saída TXT, [34], o qual possibilita anotação de sintagmas nominais, assim como *PoS-tagging*. Esse software foi desenvolvido por uma equipe de pesquisa[1] em processamento de linguagem natural na Universidade de Stanford nos Estados Unidos.

Um exemplo de anotação da ferramenta para o inglês pode ser visto na Figura 1. O período *I need to talk to you* foi introduzido para que fosse realizado o processamento sintático. O resultado pode ser dividido em duas etapas: (i) *tagging*, que corresponde à identificação da funcionalidade de cada palavra no texto e (ii) *parse*, que corresponde à estrutura e dependência gramatical entre as palavras da frase.

[1] nlp.stanford.edu

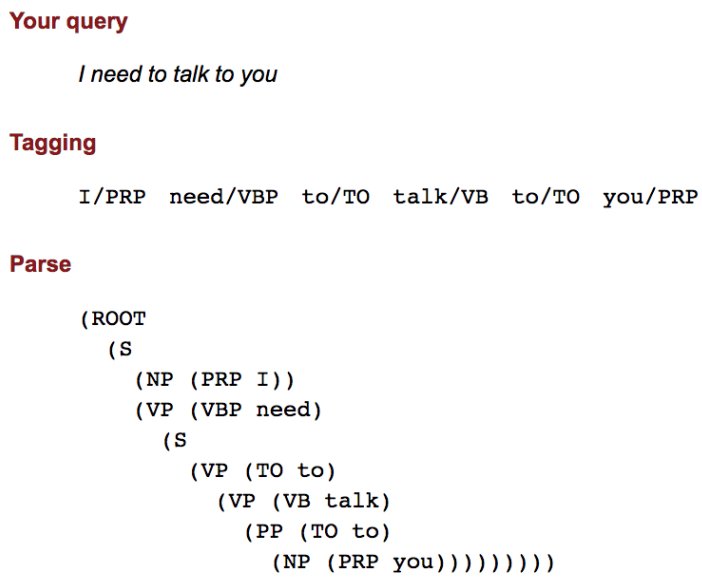


Figura 1 – Exemplo de uso da ferramenta *Stanford Parser* (Fonte: [12])

A técnica utilizada para identificar os termos no texto é baseada em regras e padrões pré-definidos que são diferentes para cada idioma. No caso dessa ferramenta, os padrões adotados são para o idioma inglês, mas a ferramenta de *parser* chinês desenvolvida pelo mesmo grupo de estudos utiliza padrões específicos para esse outro idioma [11].

2.3.5 *ExATO*

O software ExATO (Extrator de Termos para Ontologias) foi concebido por Lopes *et al.* [22] em 2009 com objetivo de gerar automaticamente uma lista de termos significantes e opcionalmente algumas medidas numéricas a partir de *corpora* em língua portuguesa. Mais recentemente, no ano de 2015, foi lançada uma nova versão desse programa, capaz de realizar também a tarefa de extração em *corpora* em língua inglesa. É imperativo para ambas línguas que o texto de entrada seja anotado sintaticamente por um *parser* gramatical.

A ferramenta ExATO foi utilizada como base neste trabalho para apoiar o processo de extração linguística de termos. A Figura-2 ilustra os processos de extração e conversão feitos pela ferramenta.

A primeira etapa de execução do sistema consiste na extração de sintagmas nominais, e é dividida em três etapas: (i) detecção dos sintagmas nominais no XML de entrada; (ii) aplicar as regras de descarte para considerar somente termos que são adequados para extração; (iii) remoção de eventuais palavras dos sintagmas nominais de acordo com as regras de transformação.

As regras de descarte são aplicadas com o intuito de retirar do contexto da extração os termos que poderiam não ter relação com o domínio do corpus analisado, por serem expressões de

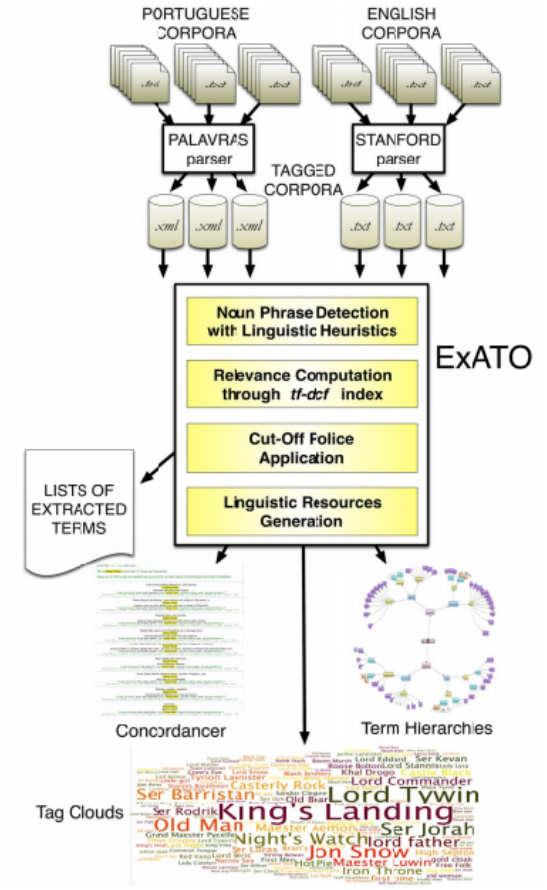


Figura 2 – Estrutura da ferramenta Exato. Fonte: [22].

uso comum. Exemplo de termos descartados podem ser numerais ou caracteres especiais que não tem relação com a língua portuguesa. [22].

Da mesma forma que as regras de descarte, as regras de transformação têm também o objetivo de refinar a extração de termos. Nesse caso, o objetivo é "reparar"de forma a mantê-los com a melhor significação possível. Foram aplicadas especialmente três Regras de Transformação: (i) remoção de artigos; (ii) remoção de pronomes no início de um sintagma nominal; (iii) remoção de conjunções no final de sintagmas nominais [22].

A segunda etapa executada pela ferramenta é a computação da frequência dos termos. Trata-se de um tratamento estatístico que conta quantos termos foram extraídos, e quais suas frequências absolutas e relativas. A métrica estatística utilizada pelo sistema é o *tf-dcf* (*term frequency, disjoint corpora frequency*), discutida na subseção 2.4.3, pois verificou-se em trabalhos anteriores [21] que ela apresenta melhores resultados qualitativos se comparada à métrica de frequência absoluta do termo (subseção 2.4.1), e melhores também que a métrica frequência do termo–inverso da frequência nos documentos (subseção 2.4.2).

Para uma melhor organização do resultado o sistema faz também uma classificação dos termos. O procedimento adotado foi classificá-los de acordo com o número de conjunto de caracteres, ou seja, 1-gram (unigramas), 2-gram (bigramas), 3-gram (trigramas), e assim por diante.

Com os termos ordenados e classificados, torna-se possível aplicar pontos de corte para então eliminar termos conceitualmente pouco relevantes. Esta é a terceira etapa do processo, capaz de permitir ao usuário selecionar qual o ponto de corte desejado para o procedimento que está executando.

A quarta e última etapa da extração, o pós-tratamento ou geração de recursos linguísticos, oferece a possibilidade de salvar a lista de termos extraídos em diversos formatos, variando de uma simples lista de termos até formas mais sofisticadas.

Outra funcionalidade que a ferramenta possui é o concordanciador, capaz de varrer os *corpora* pelo contexto de um termo específico informado pelo usuário. O formato de saída é um arquivo HTML composto por todas as ocorrências do termo e o respectivo contexto em cada corpus; através de um outro formato de saída é possível também gerar *tag clouds* e estruturas mais complexas de hierarquização de termos.

Para este trabalho foi utilizada a saída padrão da ferramenta, onde podem ser consultados os termos extraídos, em ordem decrescente por frequência, sem ponto de corte. Esta lista foi comparada com os resultados obtidos pela execução do processo estatístico em cada métrica de relevância.

2.4 Abordagem Estatística

A abordagem estatística baseia-se em modelos matemáticos e medidas estatísticas para cálculo e identificação dos termos a serem extraídos de um corpus. Dessa forma, nesta seção são detalhadas duas métricas de relevância cujos resultados de suas implementações foram apresentados neste trabalho (*tf*, *tf-idf* e *tf-dcf*).

Sobre as métricas de relevância pode-se dizer que são a ferramenta utilizada pela extração estatística para refinamento de conceitos em *corpora*. É através dessas métricas que o processo estatístico busca identificar termos que possuem relevância matemática no corpus, e que por sua vez são candidatos a conceito.

Como essas métricas são geralmente aplicadas sobre uma lista de termos, e não apenas sobre textos ou ontologias, pode-se aproveitá-las para refinar listas geradas através da extração linguística. As subseções seguintes apresentam as métricas, procurando mostrar suas formulações e algumas de suas vantagens.

2.4.1 Frequência do termo (*Term frequency*, tf)

Calcular a frequência que um termo se repete em um texto é a forma mais direta e prática de se estimar sua relevância no contexto. A popularidade e a facilidade de implementação dessa métrica são os pontos positivos do seu uso.

A fragilidade da frequência absoluta como índice de relevância é um fato conhecido, uma vez que os métodos puramente estatísticos exigem uma lista de termos frequentes conhecida como *stop list* [21] (seção 4.1.4). Sem essa lista os métodos estatísticos podem fornecer termos com muita baixa significância para o domínio, como artigos, preposições, nomes próprios e expressões que são de uso geral.

A formalização matemática do cálculo das frequências pode ser expressa por:

$$\mathbf{tf}_t^{(c)} = \sum_{\forall d \in \mathcal{D}^{(c)}} tf_{t,d}$$

onde $tf_{t,d}$ é o número de ocorrências do termo *t* no documento *d* que pertence ao conjunto $\mathcal{D}^{(c)}$ de documentos que compõem o corpus *c*.

2.4.2 Frequência do termo–inverso da frequência nos documentos (*Term frequency and inverse document frequency*, tf-idf)

Os resultados alcançados com o uso da frequência absoluta podem ser bastante precários, pois termos muito frequentes, sem relevância para o domínio, podem ter frequências absolutas muito altas, sendo então considerados importantes de forma equivocada. Nesse sentido, Spärck–Jones [14] apresenta em seus trabalhos a importância de considerar também aqueles termos que não são tão frequentes entre os documentos do corpus.

Os trabalhos de Croft e Harper [9], e posteriormente Robertson e Walker [32] propuseram a formulação de um índice que leva em consideração positivamente a frequência de um termo *(tf)*; e negativamente o número de documentos do corpus onde o termo aparece ao menos uma vez *(idf)*. A fórmula abaixo foi adaptada por Bell *et al.* [37], e representa matematicamente o índice tf-idf.

$$\mathbf{tf\text{-}idf}_{t,d} = (1 + log(tf_{t,d})) \times log\left(1 + \frac{|\mathcal{D}^{(c)}|}{|\mathcal{D}_t^{(c)}|}\right)$$

2.4.3 Frequência do termo e disjunção de corpora (*Term frequency, disjoint corpora frequency*, tf-dcf)

O último índice listado como objeto de estudo é o *tf-dcf*, apresentado por Lopes [21] em 2012, e recentemente expandido nos estudos em [19]. O desenvolvimento deste índice foi guiado por estudos sobre as demais métricas estatísticas apresentadas neste capítulo.

Em termos práticos, sua principal função é penalizar termos que aparecem em *corpora* contrastante proporcionalmente ao seu número de ocorrência; considerando o número de ocorrências dos termos no *corpora* contrastante pode-se afirmar que esta métrica combina aspectos encontrados no *tds* (subseção 2.4.4) e *thd* (subseção 2.4.4); da mesma forma se for considerado o número de

corpora contrastantes que o termo aparece, nota-se semelhanças com o índice *TF-IDF* (subseção 2.4.4).

A lógica da métrica é considerar a frequência absoluta de um termo como indicação primária de relevância do termo *(tf)*, então penalizar os termos que aparecem nos *corpora* contrastantes pela divisão de sua frequência absoluta no corpus do domínio pela composição geométrica de sua frequência absoluta em cada um dos *corpora* contrastantes *(dcf)*. Isso é definido pela fórmula abaixo:

$$\textbf{tf-dcf}_t^{(c)} = \frac{tf_t^{(c)}}{\prod_{\forall g \in G} 1 + log(1 + tf_t^{(g)})}$$

2.4.4 Outras métricas

Nesta seção são apresentadas outras métricas estatísticas para refinamento de listas de termos, porém as métricas abaixo não foram utilizadas neste trabalho. Por relevância acadêmica desses índices e com o objetivo de apresentar por completo os resultados da pesquisa essas métricas foram citadas no trabalho.

Coerência de domínio (*Domain coherence*, dc)

Os trabalhos de Bordea *et al.* [5] conduziram pesquisas sobre uma métrica capaz de extrair termos relevantes a partir de uma modelagem de domínio. Essa modelagem, segundo os autores, é um vetor de palavras que contribui para determinar o domínio que o corpus está inserido.

Do ponto de vista prático o processo é divido em duas etapas: primeiro são selecionadas as palavras que vão compor o modelo do domínio; em seguida é calculado o índice *dc* para cada termo (composto de no mínimo duas palavras) utilizando como referência o vetor de termos do domínio. A fórmula abaixo expressa o cálculo do índice *dc*.

$$\textbf{dc}_t^{(c)} = \sum_{u \in \theta} log\left(\frac{P(u,t)}{P(u)^{(c)}P(t)^{(c)}}\right)$$

Especificidade do domínio do termo (*Term domain specificity*, tds)

Os autores Park *et al.* [25] têm reconhecimento pelos trabalhos que tratam do contraste de *corpora*. O índice apresentado por eles, o *tds*, expressa essa relevância através da razão entre a probabilidade de ocorrência de um termo *t* em um corpus de domínio *c* e a probabilidade de ocorrência deste mesmo termo em um corpus genérico *g*. A expressão matemática para esta métrica é definida abaixo, onde $p_t^{(c)}$ expressa a probabilidade de ocorrência do termo *t* no corpus *c*; e $N^{(c)}$ o número total de termos do corpus *c*:

$$\textbf{tds}_t^{(c)} = \frac{p_t^{(c)}}{p_t^{(g)}} = \frac{\frac{tf_t^{(c)}}{N^{(c)}}}{\frac{tf_t^{(g)}}{N^{(g)}}}$$

Termhood, thd

O índice *thd* também pode ser utilizado para contrastar *corpora* e revelar conceitos. Ele leva em consideração que os termos relevantes de um domínio são mais frequentes em corpus de domínio que em outros *corpora*. A diferença deste índice para os demais fica por conta da construção do vocabulário de referência (conjunto de todos os termos de um corpus), ao invés da probabilidade de ocorrência do termo [21].

A fórmula do índice *thd* foi proposta por Kit e Liu's [17], onde *t* é o termo de um corpus *c*, e *g* é o corpus genérico. O vocabulário do corpus *c* é expresso por $V^{(c)}$; $|V^{(c)}|$ é a cardinalidade do conjunto de todos os termos encontrados no corpus *c*; e $r_t^{(g)}$ é o valor da ordenação do termo *t*.

$$\textbf{thd}_t^{(c)} = \frac{r_t^{(c)}}{|V^{(c)}|} - \frac{r_t^{(g)}}{|V^{(g)}|}$$

Frequência do termo–inverso da frequência no domínio (*Term frequency, inverse domain frequency*, TF-IDF)

A métrica *Term frequency, inverse domain frequency* (TF-IDF) considera a ideia original do *Term frequency and inverse document frequency* (tf-idf). Kim *et al.* [16] destacam que o primeiro é na verdade uma releitura do segundo, já que entende que todo conjunto de documentos do corpus é um único documento, e que cada corpus é um documento diferente.

Conforme proposto por Kim *et al.*, o índice TF-IDF é formalmente definido por:

$$\textbf{TF-IDF}_t^{(c)} = \frac{tf_t^{(c)}}{\sum_{\forall t'} tf_{t'}^{(c)}} \times log\frac{|G^*|}{|G_t^*|}$$

onde $tf_t^{(c)}$ é a frequência absoluta do termo *t* no corpus *c*; G^* é o conjunto de todos os *corpora* contrastantes no corpus *c*; e G_t^* é o subconjunto de G^* onde o termo *t* aparece ao menos uma única vez.

2.4.5 Software de análise estatística (*Ngram Statistics Package*)

O NSP, ou *Ngram Statistics Package*, é um software capaz de identificar palavras e caracteres Ngram que aparecem em *corpora* utilizando testes estatísticos como ferramenta de busca. Alguns dos testes implementados por ele são *Likelihood-ratio test*, *Fisher's exact test* e *Pearson's*

chi-squared test, entre outros. O software foi desenvolvido em linguagem Perl por *Ted Pedersen* e sua equipe na Universidade de Minnesota, nos Estados Unidos [26].

Por definição, um Ngram é uma sequência contínua de *n* itens de uma dada sequência de texto ou fala coletados de um corpus. Os itens podem ser fonemas, sílabas, letras, palavras ou pares de bases, de acordo com a aplicação. Um Ngram de tamanho "1"refere-se a um "unigrama", tamanho "2"um "bigrama", e assim por diante [26].

NSP consiste em dois principais programas: o primeiro, *count.pl*, considera como entrada um conjunto de arquivos de texto e gera uma lista de todos os Ngrams que ocorrem naqueles arquivos. A saída do software é composta pelos Ngrams e suas frequências, em ordem descendente pela frequência; o segundo item do pacote é o programa *statistic.pl*, cuja entrada é sempre uma lista de Ngrams e suas frequências (no formato gerado pelo programa *count.pl*), e então executa um dos testes estatísticos selecionado pelo usuário, com a finalidade de calcular o "score"para cada Ngram. A saída deste programa será sempre a lista de Ngram acompanhada de seu score, em ordem descendente por score [26].

Para este trabalho foi utilizado o programa *count.pl* para identificar os Ngram e frequência presentes no texto, e com isso poder aplicar as métricas estatísticas descritas nas seções 2.4.1 a 2.4.3 no resultado gerado. Foi desconsiderado o uso do programa *statistic.pl* para essa tarefa porque as métricas estudadas não estavam implementadas em seu contexto, bem como nenhuma dessas métricas do pacote leva em consideração a tarefa de contrastar *corpora*.

O formato de saída do programa *count.pl* é composto pelo Ngram e sua frequência. A forma como o software gera esse resultado é através de *tokens*. Um token é uma string de caracteres alfanuméricos, ou uma marca prória de pontuação. Em particular, os autores Banerjee e Pedersen [2] definem *token* como sendo uma sequência contínua de caracteres que correspondam a um conjunto de expressões regulares. O programa *count.pl* fornece a opção de o usuário informar o formato que deseja gerar *tokens* através do uso de expressões regulares. Caso o usuário não informe uma expressão, o próprio programa encarrega-se de utilizar uma padrão.

Um exemplo de token pode ser *Barack<>Obama<>*, gerado pelo termo *Barack Obama*. Concatenado ao *token*, o software apresenta as frequências de ocorrência do termo em si, e em seguida para cada palavra isoladamente. Dessa forma, o mesmo *token* poderia resultar no seguinte: *Barack<>Obama<>27<>134<>463*, onde o número 27 representa a quantidade de ocorrências do token *Barack<>Obama<>*, *Obama<>* tem 463 ocorrências, e *Barack<>*, por sua vez, 134.

Os *tokens* são gerados um abaixo do outro, como num arquivo de formato *.csv*. Com isso, o usuário tem flexibilidade para trabalhar livremente com a lista de *tokens* e frequências, podendo inclusive criar filtros para gerar outros arquivos a partir deste.

3. TRABALHOS RELACIONADOS

Esta seção foi dividida em duas partes para apresentar detalhadamente os trabalhos que foram fundamentais para sustentar esta pesquisa, e também para mostrar aqueles que se relacionam com os objetivos propostos.

3.1 Literatura de base

A obra *Challenges in Natural Language Processing* de Bates *et al.* [3] tem sua importância na compreensão dos desafios da complexidade lexical da abordagem linguística computacional. O livro traz uma revisão histórica sobre a evolução no campo de PLN, e inicia sua discussão tratando dos sistemas de pergunta e respostas, que hoje são representados por *chat bots*, largamente utilizados por companhias aéreas e seguradoras no mundo todo.

Na mesma linha de Bates, Kao *et al.* [15] trazem sua contribuição com os meios linguísticos através do livro *Natural Language Processing and Text Mining*, mais focado na tarefa de mineração de textos, procurando unir tecnologias antigas e novas em volta da tarefa de *text mining*. A leitura desse livro contribui para uma visão geral sobre o processo de extração de informação em textos.

Os trabalhos de Bick [4] sobre seu analisador sintático para textos em português ajudam não só a compreender sua ferramenta proposta, o PALAVRAS, mas também contribui valorosamente para evolução das técnicas e heurísticas de compreensão textual para a língua portuguesa. Ainda que os *corpora* da pesquisa não estivessem em português o trabalho de Bick foi adotado por se tratar de uma referência na área.

O grupo de estudos de PLN da *Universidade de Stanford* é internacionalmente reconhecido pelas suas pesquisas em volta dos analisadores sintáticos. O maior produto de suas pesquisas, a ferramenta *Stanford Parser* [12] é mundialmente utilizada para anotação em textos de vários idiomas, inclusive chinês. Ela foi utilizada para apoiar a anotação linguística dos *corpora* desta pesquisa, e os trabalhos publicados pelo grupo orientam os usuários na compreensão das heurísticas e técnicas de anotação sintática [11, 34].

Os trabalhos de Lopes *et al.* [22–24] contribuem para o desenvolvimento das técnicas de construção de heurísticas em língua portuguesa para o tratamento de sintagmas nominais, e têm sua maior contribuição no desenvolvimento da ferramenta *ExATO*, cujo objetivo é realizar a extração de termos para ontologias a partir de textos anotados sintaticamente, ordenando os termos por medidas de relevância e efetuando a classificação de termos relevantes. O formato de implementação adotado no desenvolvimento do software *ExATO* foi repetido nos *scripts* desenvolvidos nesta pesquisa.

Pedersen *et al.* são os responsáveis pelo desenvolvimento da ferramenta NSP (*Ngram Statistics Package*) e sua disponibilização no site Metacpan[1]. A mais recente atualização do *core* da ferramenta foi entregue em 2015. Seus trabalhos tratam principalmente de definir os problemas

[1] https://metacpan.org/pod/Text::NSP

e soluções adotadas no processamento estatístico [27], bem como apresentar casos reais de uso da sua ferramenta [2, 26].

As métricas de relevância estatística adotadas neste trabalho são apresentadas com base nos trabalhos de seus autores. Para métrica *tf-idf* foram utilizados os trabalhos de Spärck–Jones [14], em *dc* Bordea *et al.* [5], *tds* Park *et al.* [25], *thd* Kit e Liu's [17], para *TF-IDF* Kim *et al.* e por fim [21] para *tf-dcf*. Através da leitura desses trabalhos foi possível conhecer mais sobre as diversas formas de se contrastar *corpora*, além de ter tido a oportunidade de traçar um comparativo entre as métricas utilizadas por cada autor.

3.2 Trabalhos similares

Lopes *et al.* [19, 21] estuda métricas estatísticas de extração de termos em *corpora* de domínio. Nesses trabalhos, autores buscam traçar as diferenças e semelhanças entre essas métricas, como também apresentam sua proposta de formulação (métrica *tf-dcf*) para resolver os problemas de identificação de termos relevantes. O comparativo dos resultados dos seus experimentos realizados com *corpora* de referência garante a eficácia da sua métrica e abrem caminho para outras pesquisas em torno da extração de conceitos.

Em outro trabalho [20], Lopes faz um comparativo entre as abordagens linguística e estatística para extração de termos utilizando a técnica do *recall*, assim como foi realizado nesta pesquisa. Os resultados foram alcançados através do uso do *ExATO* para o processamento linguístico, e o NSP para estatístico, acompanhado de uma *stop list* para remoção das palavras funcionais. Nessa pesquisa elaborada por Lopes *et al.* observou-se que o processamento estatístico acompanhado da remoção das palavras funcionais não obteve qualidade igual ou superior a extração feita pela ferramenta *ExATO*.

Em seus trabalhos futuros foi orientado o uso de métricas de relevância, como *tf-idf* e *tf-dcf*, sobre o processamento estatístico para verificar o comportamento da extração utilizando métricas, exatamente como foi realizado nesta dissertação. Seu objetivo com isso seria observar como a extração se comporta quando são aplicadas essas métricas sobre ambas as abordagens.

Conrado *et al.* fizeram uma importante contribuição para área de extração de termos em língua portuguesa em seu artigo [10] que compara diversas pesquisas realizadas nesse segmento. Os autores compararam os resultados alcançados por diversos autores a respeito das abordagens linguística, estatística e híbrida (quando ambas são combinadas) para língua portuguesa, e detalharam os processos utilizados em cada trabalho. Ainda que sua pesquisa tenha se desenvolvido sobre a língua portuguesa, algumas conclusões são genéricas para todos os idiomas.

Os autores concluem que houve um aumento considerável no número de trabalhos científicos que utilizam a abordagem estatística para extração de termos. Principalmente por sua característica de independer de idioma, como também exigir menos recursos computacionais. Em contrapartida, os resultados desse trabalho mostram que uma abordagem puramente estatística

ainda não garante uma extração qualitativamente superior àquela realizada com recursos linguísticos. Isso se observa principalmente em contextos onde não se tem conhecimento prévio de domínio, e então não se pode remover termos dispensáveis através de *stop lists*.

Nesse sentido, a pesquisa de Conrado *et al.* conclui que a melhor abordagem até então é a combinação dos meios linguístico e estatístico para extração de termos, ou seja, a abordagem híbrida, por agregar os recursos linguísticos necessários para apoiar a remoção de termos sem aderência com o domínio.

São também correlatas a este trabalho, as pesquisas realizadas atualmente que utilizam métricas de relevância em seu contexto, como a apresentada por Almeia *et al.* [1] que aplicaram a métrica *tf-idf* para identificar e monitorar grupos de ódio em redes sociais através da extração de características dos comentários retirados dessas redes. Outro trabalho realizado nessa mesma linha foi a pesquisa de Ábia *et al.* [38] que estudaram a polaridade de *tweets* referentes ao processo de impeachment da então presidente do Brasil. Para isso, cada palavra dos *tweets* foi representada pelos seus valores de *tf-idf*.

Para a abordagem linguística especificamente, o trabalho de Perna *et al.* [28] se correlaciona a este, pois utilizou-se do software *ExATO* para processar dissertações e teses de mais de dez anos de produção acadêmica do Programa de Pós-Graduação em Linguística da PUC-RS. Os resultados desse processamento serviram para tomar conhecimento dos principais temas estudados nesse período, como também impulsionam novas pesquisas em diferentes campos até então não explorados, ou não bem estabelecidos dentro da tradição linguística. Para esse trabalho não foram aplicadas métricas de relevância, nem comparadas as listas geradas com nenhum padrão de referência.

4. EXPERIMENTOS

Os experimentos realizados nesta pesquisa foram conduzidos inicialmente com o uso dos softwares NSP (seção 2.4.5) e ExATO (seção 2.3.5), buscando extrair os termos e suas frequências utilizando as abordagens puramente estatística e linguística. O resultado dessa etapa gerou listas de termos ordenados pela frequência (métrica *tf*, seção 2.4.1) que serviram de entrada para a aplicação da técnica de contraste de *corpora*, utilizando as métricas *tf-dcf* (seção 2.4.3) e *tf-idf* (seção 2.4.2). Adicionalmente, com o objetivo de aperfeiçoar a extração, foi utilizada uma *stop list* (seção 4.1.4), buscando obter ganhos na desclassificação de termos sem valor terminológico.

Como resultado dos experimentos, obteve-se nove listas de termos geradas para cada corpus, onde foi possível tecer comparações contra listas de referência (detalhadas na seção 4.2), desenvolvidas a partir dos corpora e de pesquisas na Internet. Os *corpora* escolhidos para os experimentos foram quatro conjuntos de artigos de periódicos científicos publicados de 2010 a 2015:

- TACCESS[1] - ACM Transactions on Accessible Computing, 72 artigos do volume 2 ao 7;
- ToCT[2]- ACM Transactions on Computation Theory, 76 artigos do volume 1 ao 7;
- TKDD[3] - ACM Transactions on Knowledge Discovery from Data, 157 artigos do volume 4 ao 10;
- TOSEM[4] - ACM Transactions on Software Engineering and Methodology, 158 artigos do volume 19 ao 25.

Para ter acesso a esses periódicos, foi necessário utilizar a rede da Universidade PUC-RS e buscá-los no portal da ACM (*Association for Computing Machinery*), uma vez que não estão disponíveis abertamente na Web; os periódicos foram acessados e o download realizado manualmente, abrindo a página de cada volume e acessando os artigos disponíveis. Os arquivos baixados foram organizados em diretórios: um para cada periódico e um subdiretório para cada ano de publicação, de 2010 a 2015.

Como todos os arquivos estavam em formato PDF, foi necessário transformá-los em texto (TXT). Para isso, foi utilizado o comando *pdftotext*, disponível no pacote *poppler-utils* no *Debian*. A fim de facilitar a mudança de formato dos arquivos, já que os diretórios contavam com muitos arquivos, foi desenvolvido um *batch* de processamento capaz de aplicar esse comando para todos os arquivos; esse mesmo *batch*, após a conversão em TXT, executou então outro programa, escrito em PHP, para realizar a limpeza do texto gerado, removendo itens de cabeçalho e rodapé que não faziam parte do contexto de cada artigo, além de palavras e frases muitos longas que indicassem algum defeito da conversão.

[1] https://dl.acm.org/citation.cfm?id=J1156
[2] https://toct.acm.org/
[3] https://tkdd.acm.org/
[4] https://tosem.acm.org/

Através dos experimentos, percebeu-se que a conversão dos arquivos PDF para TXT ocasionava a quebra de tabelas e outros recursos textuais, resultando em conjuntos de caracteres muito extensos e sem valor terminológico. Por se tratar de periódicos da computação, alguns textos continham também formulações matemáticas e códigos-fonte. O resultado da conversão desses recursos também foi removido, pois não agrega valor conceitual para o corpus.

A Tabela 1 apresenta a contagem de palavras dos corpora após a execução das etapas de conversão e limpeza dos arquivos. Pode-se notar que os corpora são de tamanhos distintos para favorecer uma melhor análise sobre a aplicação das abordagens de extração de termos.

Tabela 1 – Total de palavras por corpus.

Corpus	Domínio	Total de linhas	Total de palavras	Total de caracteres
TACCESS	Accessible Computing	9409	703.330	4.454.139
ToCT	Computation Theory	15.113	885.994	4.584.123
TKDD	Knowledge Discovery from Data	328.788	2.317.234	13.909.054
TOSEM	Software Engineering and Methodology	351.673	2.847.501	17.814.448

Para contar as palavras foi feito uso do comando *wc*[5] da distribuição *Debian*. O comando foi executado quatro vezes, uma para cada corpus, indicando sempre o diretório onde os arquivos de texto estavam armazenados. O resultado do comando é composto por três dados: número de linhas no texto, número de palavras e o número de caracteres.

4.1 Organização do processo

Para cada um dos quatro *corpora* descritos, foram executadas extrações estatística via NSP e linguística via ExATO. Dessa forma, para cada corpus foram produzidas duas listas de termos classificadas segundo a frequência absoluta (*tf* - seção 2.4.1):

- E - uma lista resultante da extração por abordagem estatística (NSP);
- L - uma lista resultante da extração por abordagem linguística (ExATO).

É importante destacar que os processos puramente estatístico e linguístico somente são capazes de computar o *tf*, uma vez que isso representa a contagem simples dos termos de acordo com cada processo. O resultado de cada lista foi composto de um termo por linha, acompanhado de sua frequência, como por exemplo *participant,2752* e *the,46435*, onde o primeiro item é o termo, e o segundo a frequência. Para melhor exemplificar, a Tabela 2 apresenta os cinco primeiros resultados das extrações para um dos corpora estudados na pesquisa.

Mesmo observando apenas os cinco primeiros itens de cada lista é importante perceber como elas são diferentes. Enquanto a extração linguística ranqueou apenas substantivos que podem

[5] https://www.computerhope.com/unix/uwc.htm

Tabela 2 – Exemplo das extrações linguística e estatística para o corpus *TACCESS*.

TACCESS	
Extração linguística	Extração estatística
participant,2752	the,46435
users,2142	of,22293
studies,1431	to,19436
results,1063	and,18314
task,988	a,14336

ter ou não valor terminológico, a extração estatística somente revelou palavras funcionais, sem qualquer valor conceitual. A próxima seção apresenta as etapas percorridas para realizar o processamento estatístico.

4.1.1 Processamento estatístico

Como comentado anteriormente, o processamento estatístico dos corpora foi realizado com o auxílio do software NSP (*Ngram Statistics Package*, seção 2.4.5). O NSP é um software capaz de identificar palavras e caracteres Ngram que aparecem em corpora utilizando testes estatísticos como ferramenta de busca.

Para esta pesquisa, procurou-se filtrar todos os termos desde uni até pentagramas. Dessa forma, o programa *count.pl* do NSP responsável pela extração dos termos precisou ser executado até cinco vezes para cada corpus. Para cada processamento foi necessário informar o diretório onde estavam localizados os arquivos de texto, e o número de Ngram a ser extraído (iniciando em 1 e indo até 5).

Para facilitar esse processamento, foi desenvolvido um *batch* Linux para realizar as chamadas ao NSP e ainda aplicar um filtro para gerar um único arquivo CSV final com o resultado de toda etapa. Dessa forma, o *batch* permitiu ao usuário informar até quantos itens Ngram ele gostaria de extrair do texto.

O filtro PHP desenvolvido foi executado pelo *batch* Linux ao término de cada uma das extrações com NSP. Ou seja, se o *batch* fosse configurado para extrair de uni até trigramas, seriam realizadas três chamadas ao NSP, e mais três ao filtro, uma para cada lista de *tokens* gerada. O resultado de cada iteração foi um arquivo intermediário composto por uma lista de termos ordenada pela frequência acrescentado (comando *cat*) num arquivo final, esse, por sua vez, composto por todos os Ngram gerados (de uni até pentagrama - ou o limite que fosse configurado).

Uma das atribuições desse filtro foi a transformação de todas as letras dos *tokens* gerados em minúsculas. Uma vez que isso foi feito, a lista passou a ter *tokens* repetidos, já que um termo que antes começava com letra maiúscula passou a ter todas as letras minúsculas. E na ocasião que houvesse outro termo igual em minúsculo, suas frequências deveriam ser somadas, e uma das

entradas eliminada da lista. Esse procedimento agregou uma precisão melhor para a extração, já que a capitalização de um termo não define necessariamente seu valor conceitual no texto.

Outra função desse filtro foi a remoção dos caracteres que separavam uma palavra da outra (<>) no arquivo gerado pelo NSP. Esse caractere não tem valor funcional para as extrações que foram realizadas posteriormente utilizando as métricas estatísticas, portanto foi substituído por um espaço em branco. A frequência foi incluída ao final do termo, separando-se dele por uma vírgula. Nesse sentido, seguem as Tabelas 3, 4 e 5 que demonstram respectivamente parte do resultado do processamento estatístico, total de termos extraídos pela abordagem e tempo de execução dos *scripts*.

Tabela 3 – Resultado do processamento estatístico.

Extração estatística				
#	TACCESS	ToCT	TKDD	TOSEM
1	the,46435	the,41100	the,126505	the,162007
2	of,22293	of,24400	of,67589	of,88121
3	to,19436	a,23356	and,54404	and,76012
4	and,18314	is,18035	in,44102	a,62078
5	a,14336	and,17197	a,43312	to,56705
6	in,13699	that,15021	to,38578	in,56553
7	for,9479	in,14316	is,33004	is,38850
8	that,8143	to,13788	for,24830	for,32954
9	with,7443	for,12353	we,24104	that,30823
10	is,6901	we,11172	that,21168	of the,23802

A Tabela 3 apresenta o resultado da extração estatística para os quatro corpora estudados. São listados os dez primeiros termos gerados a partir de cada corpus, a fim de exemplificar a saída do *batch* desenvolvido. É natural que os termos mais frequentes sejam unigramas. O capítulo que trata dos resultados da pesquisa detalhará melhor a qualidade dessa extração.

Tabela 4 – Total de termos extraídos com a abordagem estatística.

Total de termos extraídos via abordagem estatística				
Ngram	TACCESS	ToCT	TKDD	TOSEM
1	8.605	6.961	15.883	19.371
2	41.585	40.500	101.681	130.156
3	29.219	45.080	104.142	125.093
4	11.829	24.108	55.025	61.991
5	5.022	12.148	28.478	30.210
total	**96.260**	**128.797**	**305.209**	**366.821**

A Tabela 4 apresenta o total de termos extraídos pela abordagem estatística, e separa-os de acordo com o número de Ngrams para dar ao leitor uma visão melhor sobre o processo executado. É importante diferenciar esses números alcançados daqueles exibidos na Tabela 1. Na tabela apresentada no início do capítulo são exibidos os totais de palavras por corpus, considerando

ainda as repetidas; nessa tabela construída após a extração estatística são exibidos termos, que são compostos por uma ou mais palavras, desconsiderando ainda aqueles que são repetidos ao longo do texto (inclusive termos iguais com capitalização distinta).

Tabela 5 – Tempo de execução da extração puramente estatística para cada corpus

Execução da extração estatística	
Corpus	Tempo de execução
TACCESS	00:03:40
ToCT	00:03:59
TKDD	00:12:36
TOSEM	00:15:09

Por fim, a Tabela 5 apresenta o tempo de execução percorrido para cada *corpus* durante a extração estatística que utilizou o filtro desenvolvido com NSP. O tempo total de processamento desse *script* foi de pouco mais de meia hora para todos os *corpora* juntos.

Na seção seguinte são apresentados os detalhes da execução do processamento linguístico, como a construção de *scripts*, tempos de execução e resultados, da mesma forma como foi feito nesta seção para o processamento estatístico.

4.1.2 Processamento linguístico

O processamento linguístico foi realizado com o auxílio de duas principais ferramentas: *Stanford Parser* e ExATO (seção 2.3.5). Como mencionado no início do capítulo, todos os corpora selecionados para a pesquisa estavam escritos em inglês, fazendo-se o uso de um *parser* textual para esse idioma. A ferramenta de extração linguística escolhida, o ExATO, recebe como entrada o texto anotado, ou seja, já processado pela ferramenta *Stanford Parser*.

A ferramenta *Stanford Parser* pode ser baixada do site do grupo de estudos em PLN da Universidade de Stanford. Trata-se de um arquivo compactado (.ZIP) composto por diversos arquivos em formatos JAVA, XML entre outros. Para executar o *parser* é necessário utilizar a linha de comando do sistema operacional, chamando o programa *lexparser.sh*, e passando como parâmetro o local onde se encontra o arquivo texto que se deseja processar.

Cada um dos corpora continha inúmeros arquivos em formato texto, e que deveriam ser processados um a um pela ferramenta. A fim de facilitar essa execução foi escrito um *script* PHP capaz de unificar numa mesma *string* o nome de todos os arquivos acompanhados do executor *lexparser.sh*, assim com uma única chamada via linha de comando pode-se acionar *n* vezes o algoritmo do *Stanford Parser*.

O resultado desse processamento para cada arquivo texto foi outro arquivo em formato de texto (.TXT) contendo o texto anotado conforme detalhado na seção 2.3.4. O processo todo pode levar inúmeras horas dependendo do volume de arquivos, e do tamanho do texto processado. Em

certas ocasiões, o que demanda ainda mais tempo são frases muito longas que necessitam de mais processamento, e que aumentam a complexidade do algoritmo.

Dentre todas as fases da pesquisa, pode-se dizer que essa foi a mais dispendiosa e cansativa, pois ocorreu inúmeras vezes a interrupção do processamento em virtude de *loops* ou estouro de memória da máquina utilizada. Isso deveu-se principalmente por causa da natureza do texto que foi submetido ao processamento. A Tabela 6 apresenta o tempo total percorrido para execução do *parser* textual. Para fins de comparação de desempenho, foi utilizado para o processamento um notebook com processador Intel Core I5 de 1.7GhZ, com 6Gb de memória RAM.

Tabela 6 – Tempo de execução da extração puramente linguística para cada corpus

Execução da extração linguística	
Corpus	Tempo de execução
TACCESS	08:13:06
ToCT	09:32:11
TKDD	37:47:02
TOSEM	42:55:54

Mesmo com a conversão de PDF para TXT e a limpeza de caracteres indesejados nos corpora, algumas frases ainda permaneceram longas demais para a interrupta execução da ferramenta. As frases que estavam nessas características foram ajustadas e o processamento iniciava-se novamente de onde havia parado.

É importante salientar que esses ajustes não causaram nenhum dano à integridade dos corpora, pelo contrário, ajudaram o *parser* a entender melhor o que o texto queria dizer. Um exemplo clássico de ajuste foi em frases que continham letras gregas e/ou letras correspondentes a variáveis matemáticas. Esse tipo de frase foi desconsiderado, pois não correspondia a nenhum valor terminológico ao corpora.

Tabela 7 – Características dos *corpora*

corpus	total de sentenças	total de palavras	termos extraídos
TACCESS	31,679	703,33	90,656
ToCT	38,895	911,338	74,752
TKDD	31,803	2,347,315	80,237
TOSEM	71,792	3,125,588	181,693

Com toda etapa de anotação de texto com a ferramenta *Stanford Parser* concluída, foi executado então o ExATO [22]. Os arquivos resultantes da fase de anotação foram organizados em diretórios com o nome do corpus correspondente. Dessa forma, a ferramenta ExATO foi acionada dentro de cada um desses diretórios. O comando utilizado correspondeu à saída padrão da ferramenta, onde se informa apenas o diretório com os arquivos anotados e configura-se para saída padrão. O resultado para cada corpus foi um arquivo CSV contendo a lista de termos extraídos de 1-gram até 10-gram ordenados pela frequência (*tf* 2.4.1).

O tempo de execução do ExATO para cada corpus foi desconsiderado na análise, uma vez que a execução se dá de forma muito rápida, o que somaria alguns poucos segundos ao tempo de processamento já apresentado na Tabela 6. Para fins de comparação entre os processos linguístico e estatístico é apresentada a Tabela 7 contendo o resultado da extração pelo processo linguístico. Pode-se notar que o processo em questão foi capaz de retornar um número menor de termos, justamente por conta dos termos dispensados, sem valor conceitual, discutidos na seção 2.3.5.

Tabela 8 – Resultado do processamento linguístico.

	Extração linguística			
#	TACCESS	ToCT	TKDD	TOSEM
1	participant,2752	g,3049	number,1777	example,2531
2	users,2142	k,2126	algorithm,1413	number,2503
3	studies,1431	y,2116	data,1241	results,2459
4	results,1063	i,1961	method,1046	set,2191
5	task,988	n,1904	results,1022	method,2096
6	system,859	x,1871	set,1009	model,2001
7	time,848	h,1840	value,942	approach,1792
8	examples,784	p,1508	t,923	case,1678
9	number,783	sets,1487	dataset,908	value,1623
10	people,758	v,1482	example,868	system,1532

A Tabela 8 apresenta os dez primeiros resultados de cada lista gerada pela extração linguística em cada corpus. Essa tabela tem o mesmo formato da Tabela 3, onde foram apresentados os resultados da abordagem estatística. Pode-se notar a diferença entre as abordagens logo nos primeiros termos extraídos nas respectivas listas. Enquanto uma lista apresenta repetidamente termos com baixo valor conceitual, a outra exclui preposições, pronomes e verbos, revelando apenas substantivos que possam estar alinhados com os conceitos do texto.

A próxima seção apresenta como foi realizada a aplicação das métricas de refinamento sobre as listas de termos extraídos geradas pelos processos linguístico e estatístico.

4.1.3 Aplicação das métricas de refinamento

O principal mecanismo de execução dos índices *tf-dcf* e *tf-idf* é o contraste entre os *corpora*, ou seja, a comparação entre eles. O processo de contrastar listas de termos é uma técnica que procura penalizar termos que são muito frequentes em todos os corpora, e pontuar melhor aqueles que são específicos do *corpus* que se pretende realizar a extração.

Há inúmeras formas de aplicar os algoritmos que fazem o contraste de *corpora*, inclusive aplicando diretamente sobre os textos, de maneira estatística, contando inicialmente a ocorrência de cada palavra, e em seguida fazendo o contraste em tempo de execução. Nesse sentido, a biblioteca *Python TextBlob* [6] é capaz de efetuar esse processo com poucas linhas de código, recebendo

[6] https://textblob.readthedocs.io/en/latest/

diretamente os textos para extração, porém, há algumas desvantagens, como a contagem apenas de unigramas e o tempo de processamento dispendioso para textos muito grandes.

Como esta pesquisa investigou a aplicação das abordagens de extração de termos sobre Ngrams, optou-se pelo uso das ferramentas de extração de Ngrams (como o NSP e o ExATO), com o objetivo de não contar apenas palavras, mas sim termos compostos com melhor valor terminológico (bigramas, trigramas, etc). Dessa forma, foi possível isolar o processo de contagem dos termos (*tf*) do restante do processo de contraste, já que o *input* desse processo são as listas de termos com *tf* ordenadas de forma decrescente.

Portanto, em cima das listas (E e L) foram aplicadas as métricas de refinamento *tf-dcf* e *tf-idf* resultando em mais quatro listas para cada corpus, sendo elas (i) E + *tf-dcf*, (ii) L + *tf-dcf*, (iii) E + *tf-idf*, (iv) L + *tf-idf*. Em resumo, para cada processo foram aplicadas as duas métricas citadas, gerando até aqui um total de seis listas de termos para cada corpus:

- E - uma lista resultante da extração por abordagem estatística (NSP);
- L - uma lista resultante da extração por abordagem linguística (ExATO);
- E + *tf-dcf* - uma lista resultante da extração por abordagem estatística (NSP) combinada com a métrica tf-dcf;
- L + *tf-dcf* - uma lista resultante da extração por abordagem linguística (ExATO) combinada com a métrica tf-dcf;
- E + *tf-idf* - uma lista resultante da extração por abordagem estatística (NSP) combinada com a métrica tf-idf;
- L + *tf-idf* - uma lista resultante da extração por abordagem linguística (ExATO) combinada com a métrica tf-idf;

A execução das métricas sobre os corpora se deu através de *scripts* PHP que foram desenvolvidos para realizar essa tarefa de contraste. Foram escritos dois programas: tf-idf.php e tf-dcf.php que recebiam dois parâmetros: (i) o nome do arquivo de saída; (ii) um conjunto de listas de termos que se desejava contrastar. Os programas executavam a comparação das listas e calculavam a métrica em questão e o resultado era apresentado no arquivo de saída informado pelo usuário no primeiro parâmetro.

Todas as listas geradas foram comparadas com o padrão de referência construído para cada domínio dos corpora (melhor detalhado na seção 4.2). Essa comparação com a lista de referência é o processo que mede a qualidade de cada extração e foi descrito na seção de resultados do trabalho (seção 5.3).

A próxima seção detalha como foi aplicada a *stop list* (APÊNDICE A) construída para esses corpora sobre a lista de termos gerada pelo processamento estatístico de cada *corpus*.

4.1.4 Stop list

Adotou-se inicialmente como *stop list* a lista disponível no site *Ranks.nl* [7] que reúne *stop lists* para vários idiomas, inclusive português, espanhol e até russo. A lista era composta por unigramas e bigramas diversos como os citados acima, num total de 173 termos.

Para aplicar a *stop list* foi desenvolvido um *script* em linguagem PHP que recebia dois arquivos como parâmetro: a *stop list* e uma lista de termos arbitrária que precisava-se filtrar. O resultado do *script* foram dois arquivos: o primeiro contendo a lista final filtrada sem os termos dispensáveis, e outro contendo uma lista com os termos dispensados para fins de análise.

O processo de comparação para definir se um termo deve ser dispensado ou não se fez da seguinte forma: buscou-se pelo termo de parada no início e no fim do termo analisado, e então se houvesse ocorrência o termo todo era dispensado. Um exemplo foi o bigrama *number of*, que foi dispensado na lista gerada pelo processamento estatístico (E) do corpus *TACCESS*. Por si só, o unigrama *number* pode ter algum valor conceitual para o corpus, mas quando acompanhado do sufixo *of*, o bigrama é então dispensado. Outro exemplo é *the participants*, dispensado pela presença do prefixo *the*. Todas as comparações foram realizadas ignorando a capitalização das letras (*non case sensitive*).

O que se notou com a aplicação sobre as listas geradas estatisticamente foi que alguns termos sem valor conceitual continuaram bem ranqueados na lista final. Percebeu-se que esses termos eram frequentes em todos os corpora, como *ibidem* ou *et. al*. Acontece que esses termos são expressões latinas, frequentemente utilizadas em textos científicos, e que por essa razão optou-se por incluí-las na *stop list* a fim de removê-los das listas finais e assim melhorar a qualidade da extração.

Ainda nessa linha de análise, notou-se que a lista estatística filtrada do corpus *TOCT* (*ACM Transactions on Computation Theory*) era composta inicialmente por muitos unigramas alfabéticos, como a, b, c até z. Isso ocorria devido à natureza textual do corpus, uma vez que havia inúmeras fórmulas matemáticas em seu contexto, geralmente representadas por letras do alfabeto.

Para resolver também essa questão, optou-se por adicionar na *stop list* todos os unigramas alfabéticos do A ao Z do alfabeto inglês. Com isso, todas as letras que estavam sendo classificadas como importantes nas listas geradas passaram a ser desconsideradas, e, como consequência, a lista final gerada passou a ter um valor terminológico mais interessante.

O processo de construção da *stop list* foi apoiado pela análise recorrente dos resultados gerados pela execução do *script* extrator descrito anteriormente. Ou seja, foi um processo iterativo, onde a cada extração realizada foi importante avaliar o resultado gerado e procurar entender a razão que levou um determinado termo ser classificado como terminologicamente relevante para o corpus. Para tal, foram sempre analisados os 100 termos melhor ordenados após o processamento das listas

[7] https://www.ranks.nl/stopwords

geradas pela abordagem estatística. A lista final ficou composta por 217 termos caracterizados por não terem relevância terminológica para os corpora (APÊNDICE A).

De certa forma, pode-se dizer que a *stop list* construída não é totalmente genérica, pois sofreu algumas mudanças para se adequar à natureza contextual dos corpora, a fim de possibilitar resultados mais fiéis às listas de referência pesquisadas. O processo completo de pesquisa e processamento para criação dessa *stop list* demandou em torno de quatro horas de trabalho. Ainda que a execução da extração seja rápida computacionalmente, o trabalho maior está em adquirir conhecimento suficiente sobre os corpora para eleger então os termos de parada mais interessantes para um resultado final com maior valor terminológico.

Dependendo do contexto da extração, a construção de uma lista de termos de parada pode levar inúmeras horas, além de geralmente envolver conhecedores do domínio dos textos para garantir que nenhum termo importante terminologicamente esteja sendo dispensado.

A próxima seção detalha como foi realizado o refinamento das listas estatísticas de termos com o uso dessa *stop list* construída.

4.1.5 Extração usando a *stop list*

O processo de extração das listas utilizando a *stop list* foi realizado com apoio do *script* PHP descrito anteriormente. Foram selecionadas para essa etapa as listas geradas pela abordagem puramente estatística. Na nomenclatura adotada no trabalho, a lista E de cada corpus, geradas a partir do NSP.

O processo de aplicação da lista de parada à lista de termos gerada pelo NSP deu origem a uma nova lista, nomeada de E *stop list*. Essa lista foi gerada para todos os corpora da pesquisa, cada qual com base em sua lista E.

A fim de avaliar o resultado do uso da técnica *stop list* combinada com as métricas de relevância *tf-idf* e *tf-dcf* (apresentadas na seção 2.4.2 e 2.4.3), foi realizada a aplicação das métricas sobre as listas E *stop list*, gerando outras duas listas para cada corpus: *E + stop list + tf-dcf* e *E + stop list + tf-idf*. O processo de aplicação das métricas foi o mesmo que aquele relatado na seção 4.1.3, porém o *input* do processo foram as listas já refinadas pela *stop list*.

Sendo assim, deu-se origem a três novas listas de termos: *E + stop list*, *E + stop list + tf-idf* e *E + stop list + tf-dcf*. Abaixo, são novamente apresentadas as listas citadas anteriormente, adicionadas as geradas pela aplicação da *stop list*, totalizando nove listas resultantes.

- E - uma lista resultante da extração por abordagem estatística (NSP);
- L - uma lista resultante da extração por abordagem linguística (ExATO);
- E + *tf-dcf* - uma lista resultante da extração por abordagem estatística (NSP) combinada com a métrica tf-dcf;

- L + *tf-dcf* - uma lista resultante da extração por abordagem linguística (ExATO) combinada com a métrica tf-dcf;
- E + *tf-idf* - uma lista resultante da extração por abordagem estatística (NSP) combinada com a métrica tf-idf;
- L + *tf-idf* - uma lista resultante da extração por abordagem linguística (ExATO) combinada com a métrica tf-idf;
- E + *stop list* - uma lista resultante da aplicação da stop list sobre a lista E (gerada pelo NSP);
- E + *stop list* + *tf-dcf* - uma lista resultante da aplicação da métrica tf-dcf sobre a lista E + stop list (gerada pela aplicação da stop list sobre a lista E);
- E + *stop list* + *tf-idf* - uma lista resultante da aplicação da métrica tf-idf sobre a lista E + stop list (gerada pela aplicação da stop list sobre a lista E).

A seção abaixo apresenta a construção das listas de referência dos *corpora*, utilizadas nessa pesquisa para validar os resultados de cada extração realizada.

4.2 Listas de Referência

Como pode ser observado no início do capítulo, cada um dos *corpora* pertence a um domínio específico da ciência da computação. Cada um desses domínios possui termos específicos (conceitos) que se repetem ao longo do corpus e que remetem à ideia central de cada texto. Usualmente um conhecedor da área, ou então um linguista, saberia dizer quais termos estão alinhados com o contexto do texto. Um leitor que nunca tenha tido contato com o tema certamente não poderá identificar facilmente esses mesmos termos.

Sendo assim, a fim de construir um melhor conhecimento sobre o domínio de cada *corpora*, foi gerada uma lista de termos de referência (ou lista padrão ouro) para cada um dos corpora. Ou seja, foram pesquisadas as palavras-chave relacionadas a cada uma das áreas listadas na Tabela 9. Essas listas foram construídas a partir de pesquisas nas Internet.

Tabela 9 – Total de termos de referência por domínio - listas da Internet

Corpus	Domínio	Total de termos de referência
TACCESS	Accessible Computing	144
ToCT	Computation Theory	104
TKDD	Knowledge Discovery from Data	96
TOSEM	Software Engineering and Methodology	255

Esse esforço resultou em quatro listas de termos, com quantidades de ocorrências variadas. Cada uma dessas listas foi utilizada para tecer comparações com as listas de termos geradas

pela aplicação das métricas de relevância em cima das abordagens linguística e estatística, melhor detalhado na seção 5.3.

Para comprovar a aderência dos *corpora* com as listas de referência foi realizada a busca por cada termo da lista de referência no seu respectivo corpus. Esse processo de contagem foi capaz de comprovar o quão alinhados estavam os *corpora* com suas listas de referência levantadas na pesquisa, conforme demonstrado na Tabela 10

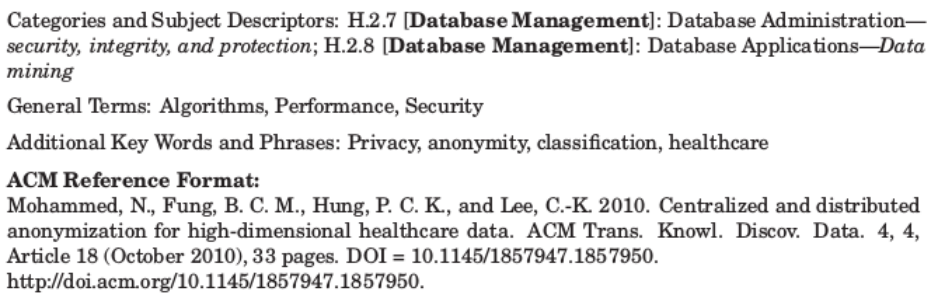
Tabela 10 – Aderência dos corpora às suas listas de referência pesquisadas na Internet

Corpus	Domínio	Total de termos de referência	Ocorrências no corpus
TACCESS	Accessible Computing	144	49
ToCT	Computation Theory	104	34
TKDD	Knowledge Discovery from Data	96	68
TOSEM	Software Engineering and Methodology	255	168

A Tabela 10 apresenta o resultado dessa busca. A lista de referência do domínio *Knowledge Discovery from Data* contém 96 termos, e o corpus de estudo pertencente a esse domínio, o *TKDD*, utiliza 68 desses 96 termos em seu contexto. Do mesmo modo, para o corpus *TOSEM*, foram identificados 168 termos aderentes à lista de referência do domínio *Software Engineering and Methodology*, composta por 255 itens.

A partir dessa análise inicial, pode-se notar que muitos termos encontrados nas listas de referência não poderiam ser identificados nas extrações linguística e estatística realizadas, uma vez que não estão presentes nos corpora de estudo. Com isso, observou-se que no melhor caso, poder-se-ia localizar apenas os termos da quarta coluna da Tabela 10, pois representam aqueles que pertencem ao domínio do corpus, e estão também presentes no contexto do texto.

Na tentativa de melhorar a aderência dos corpora com as listas, buscou-se incrementar as listas de referência com novas pesquisas, porém, nada de muito útil e confiável pode ser localizado na Internet. Sendo assim, partiu-se para outra abordagem: procurar por palavras-chave dentro dos arquivos de cada corpus, conforme demonstrado na figura abaixo:



Categories and Subject Descriptors: H.2.7 [**Database Management**]: Database Administration—*security, integrity, and protection*; H.2.8 [**Database Management**]: Database Applications—*Data mining*

General Terms: Algorithms, Performance, Security

Additional Key Words and Phrases: Privacy, anonymity, classification, healthcare

ACM Reference Format:
Mohammed, N., Fung, B. C. M., Hung, P. C. K., and Lee, C.-K. 2010. Centralized and distributed anonymization for high-dimensional healthcare data. ACM Trans. Knowl. Discov. Data. 4, 4, Article 18 (October 2010), 33 pages. DOI = 10.1145/1857947.1857950.
http://doi.acm.org/10.1145/1857947.1857950.

Figura 3 – Cabeçalho de um dos artigos do corpus TKDD

Como os corpora são compostos por textos científicos, em formato de artigo, todos tinham no cabeçalho palavras-chave que são conceituais para cada texto. Sendo assim, iniciou-se um

trabalho para buscar por todas essas palavras-chave de cada corpus e anotá-las numa lista. A Figura 3 ilustra o cabeçalho de um dos artigos do corpus TKDD, de onde as palavras-chave foram retiradas (seções *General Terms* e *Additional Key Words and Phrases*). Para o artigo apresentado na figura, foram adicionadas à lista de referência os termos *Algorithms*, *Performance*, *Security*, *Privacy*, *anonymity*, *classification* e *healthcare*.

Por fim, para evitar que algum termo se repetisse na lista, e também para padronizar a capitalização dos termos foi desenvolvido mais um *script* PHP, capaz de processar as listas. Esse procedimento recebia como parâmetro uma lista de referência e executava três operações: (i) remoção dos termos repetidos, (ii) transformação em minúsculas e (iii) ordenação alfabética. As listas de referência resultantes após esse processo cresceram de tamanho, e notou-se uma aderência melhor dos corpora às respectivas listas. O resultado pode ser visualizado na Tabela 11. Essas listas podem ser encontradas na seção apêndice desse documento: *TACCESS* (APÊNDICE B), *ToCT* (APÊNDICE C), *TKDD* (APÊNDICE D), *TOSEM* (APÊNDICE E).

Tabela 11 – Aderência dos corpus às suas listas de referência completas

Corpus	Domínio	Total de termos de referência	Ocorrências no corpus
TACCESS	Accessible Computing	291	235
ToCT	Computation Theory	356	243
TKDD	Knowledge Discovery from Data	521	490
TOSEM	Software Engineering and Methodology	810	714

Para obter um melhor conhecimento sobre os termos que estavam presentes nas listas de referência, foi desenvolvido um novo script PHP capaz de varrer uma lista e contar os Ngrams, ou seja, foram contados quantos unigramas, bigramas, trigramas, etc existiam em cada lista. A Tabela 12 traz os números encontrados para cada corpus.

Tabela 12 – Contagem de Ngram por lista de referência

	Contagem de Ngram			
Ngram	TACCESS	TOCT	TKDD	TOSEM
1-gram	111	87	133	258
2-gram	140	173	291	378
3-gram	27	61	78	142
4-gram	10	26	18	26
5-gram	2	8	0	4
6-gram	0	1	1	1
7-gram	1	0	0	0
8-gram	0	0	0	0
9-gram	0	0	0	1
total	**291**	**356**	**521**	**810**

É importante destacar que para esse experimento todos os termos têm o mesmo valor conceitual para o domínio. Esse processo não garante que um termo seja mais importante que

outro, pois não existe uma medida capaz de determinar sua relevância. Por isso optou-se pela ordenação alfabética. Listas de referência ordenadas por relevância podem ser geradas por linguistas com apoio de especialistas de domínio, mas no caso dessa pesquisa não foi possível alcançar esse nível de conhecimento sobre os corpora.

Para este trabalho procurou-se construir as listas de referência a partir dos próprios textos processados, e também com a ajuda de pesquisas na Internet. Esse processo não resulta em listas de referência absolutamente precisas e completas, mas segue um caminho capaz de proporcionar listas com um padrão adequado de construção. Acredita-se que as palavras-chave dos artigos deva ter relação forte com os conceitos apresentados por cada texto.

5. RESULTADOS

Os resultados apresentados neste capítulo referem-se às comparações realizadas com as listas de termos processadas e apresentadas no capítulo anterior. O objetivo é dissertar com foco principal numa análise quantitativa, a fim de localizar quantos termos de cada uma das listas pode ser encontrado na lista de referência de cada corpus. Em algum momento o leitor poderá observar também comparações de cunho mais qualitativo entre as listas de termos.

Recapitulando, para um melhor entendimento, para cada corpus foram extraídas listas de termos geradas pelos processamentos linguístico e estatístico com e sem métricas e *stop list*, totalizando um conjunto de nove listas de termos. A fim de avaliar o desempenho de cada uma dessas listagens, uma lista de referência foi igualmente construída para cada corpus (seção 4.2), a partir de uma metodologia desenvolvida sobre medida para esse conjunto de corpora.

Para realizar essa análise quantitativa foi utilizada como métrica a revocação (ou *recall*), com o objetivo de encontrar uma medida de relevância para cada lista processada. Essa ferramenta busca compreender a fração de termos relevantes que são recuperados em cada uma das listas processadas. Em outras palavras, isso significa buscar a razão entre o número de termos relevantes que são retornados em cada lista processada e o total de termos relevantes existentes (presentes na lista de referência).

Nesse sentido, foi adotado como total de termos relevantes existentes o somatório entre *1-gram* e *2-gram* das listas de referência de cada corpus. Esta pesquisa preocupou-se em comparar apenas termos compostos por até duas palavras, por serem aqueles que estão mais presentes nessas listas, e que nesse sentido podem apresentar melhor valor conceitual para o corpus.

A Tabela 13, conforme pode-se observar abaixo, apresenta o total de termos relevantes existentes que foram levantados a partir das listas de referência de cada corpus. No caso do corpus *TACCESS* por exemplo, a lista referencial continha 111 termos 1-gram e 140 2-gram, totalizando então 251 termos relevantes existentes para esse corpus. Todas as listas de termos processadas a partir desse corpus foram comparadas com a listagem de referência, onde se buscou localizar esses 251 termos.

Tabela 13 – Target da pesquisa (denominador da revocação)

Total de termos relevantes existentes (1-gram e 2-gram)				
Qtde palavras	TACCESS	ToCT	TKDD	TOSEM
1-gram	111	87	133	258
2-gram	140	173	291	378
Total	**251**	**260**	**424**	**636**

Para viabilizar e facilitar a execução do algoritmo de revocação foi desenvolvido um filtro através da linguagem PHP com objetivo de remover das listas processadas os termos que não eram unigramas ou bigramas. Além disso, esse filtro possibilitou a aplicação de cortes nas listas, que foram úteis para viabilizar a execução da revocação em listas de termos de tamanhos distintos.

Esse filtro recebia três parâmetros: (i) a lista de termos que desejava-se filtrar, (ii) o número de N-gram (no caso 2, para manter apenas unigramas e bigramas na saída), e (iii) o número de corte. A Figura 4 traz um exemplo de execução desse *script*, conforme abaixo:

```
php get_ate_ngram.php taccess.csv 2 1000
```

Figura 4 – Exemplo de execução do *script* PHP para aplicação do filtro e corte nas listas.

Nesse exemplo apresentado, o trecho *get-ate-ngram.php* é o nome do *script*; *taccess.csv* o nome da lista a ser filtrada; *2* o número N de N-gram a ser mantido na lista; e *1000* o número de corte da lista processada. O resultado desse processamento gerou a lista *taccess-ate2-gram-1000.csv*, composta por 1.000 unigramas e bigramas ordenados de forma decrescente pela frequência (*tf*, *tf-dcf* ou *tf-idf*). Esse processo foi executado para todas as listas de todos os corpora.

Com isso, todas as listas de termos processadas inicialmente foram filtradas e geraram novas listas, com um total de 1.000 unigramas e bigramas. A aplicação do algoritmo de revocação foi realizado com o uso das listas filtradas, tendo como referência de busca os termos apresentados na tabela 13. Os resultados alcançados com essas listas foram descritos na seção 5.1. A fim de avaliar também amostras menores de termos, foi aplicado novamente o algoritmo de corte para gerar listas com apenas 100 itens. Os resultados dessa análise podem ser encontrados mais adiante na seção 5.2

Para implementar o algoritmo de revocação foi desenvolvido um novo *script* PHP. Esse programa recebia dois parâmetros: (i) a lista de referência do corpus, e (ii) a lista processada que se pretendia fazer a busca (resultado do filtro descrito anteriormente). O resultado desse processamento foi apresentado na tela do terminal, exibindo todos os termos localizados e um contador final. A Figura 5 ilustra um exemplo de execução desse *script*, conforme abaixo:

```
php recall.php listas_referencia/Aderencia/ordered-taccess2.txt E/taccess-ate-2gram_1000.csv
```

Figura 5 – Exemplo de execução do *script* PHP para aplicação do algoritmo de revocação.

Para essa execução, o trecho *recall.php* é o *script* de processamento da revocação; *listas-referencia/* o caminho onde estavam localizadas as listas de referência dos corpora; *ordered-taccess2.txt* a lista referencial do corpus *TACCESS*; *L+tf-dcf/* o caminho onde estavam localizadas as listas geradas pelo processamento linguístico com *tf-dcf*; e *taccess-ate2-gram-1000.csv* a lista gerada a partir do filtro descrito anteriormente.

Após esse processamento os resultados obtidos foram organizados em planilhas no Google Drive e cada corpus foi representado por uma planilha contendo várias abas: uma aba para cada lista de termos processada composta por 1.000 termos; outra para lista de referência do corpus; e uma última aba contendo o resultado do processamento da revocação. Assim, foi possível organizar os resultados e compartilhá-los com quem mais tiver interesse em acessá-los. Os links de acesso

para as planilhas dos corpora TACCESS [1], ToCT [2], TKDD [3] e TOSEM [4] podem ser encontrados no rodapé do documento.

As próximas duas seções apresentam os cortes aplicados nas listas de termos resultantes, com o objetivo de avaliar a qualidade das listas de acordo com o volume de termos presentes em cada corte.

5.1 *Recall* com corte de 1.000 termos

Conforme descrito anteriormente, o algoritmo de revocação foi inicialmente aplicado sobre listas compostas por 1.000 termos, filtradas pelo *script* de remoção de Ngrams e aplicação de cortes. Esse processamento resultou como saída um único valor inteiro que representa quantos termos da lista processada estão presentes na lista referencial do corpus.

As tabelas seguintes apresentam os resultados compilados para todos os corpora. Cada linha das tabelas representa uma lista processada, e a última coluna apresenta o percentual de revocação da lista diante dos termos de referência.

A Tabela 14, conforme demonstrado abaixo, apresenta os resultados do *recall* para o corpus *TACCESS*. A lista referencial de unigramas e bigramas desse corpus possui 251 termos, e foram esses termos procurados nas listas processadas. O percentual de termos encontrados, ou índice de revocação pode ser lido na última coluna da tabela.

Tabela 14 – Recall do corpus *TACCESS* nas listas 1.000+.

Recall TACCESS 1.000+ (1^2-gram)				
Lista	**Recall**	**Total Corpus**	**Total Lista**	**%**
E	41	251	1000	16,33
E+stop	52	251	1000	20,72
E+tf-dcf	75	251	1000	29,88
E+tf-idf	77	251	1000	30,68
E+stop+tf-dcf	77	251	1000	30,68
E+stop+tf-idf	83	251	1000	33,07
L	64	251	1000	25,50
L+tf-dcf	70	251	1000	27,89
L+tf-idf	76	251	1000	30,28

O que se nota inicialmente é a baixa qualidade da lista puramente estatística (E), que foi capaz de localizar apenas 41 dos 251 termos de referência para o corpus, totalizando um percentual de 16,33%. Esse comportamento já era esperado, pois sabe-se de antemão que o processamento

[1] https://goo.gl/DYmF7A
[2] https://goo.gl/ZsVx8P
[3] https://goo.gl/9VMutG
[4] https://goo.gl/dw1bhC

puramente estatístico sempre resulta em termos com baixo valor conceitual, mantendo no topo da lista artigos, conjunções e pronomes que se repetem inúmeras vezes durante o texto.

É importante perceber também o reflexo gerado pela aplicação da *stop list* sobre a lista estatística (E + *stop list*). O processo de remoção das palavras indesejadas e com baixo valor terminológico resultou na melhora do índice de revocação da lista estatística. O índice que era antes de pouco mais de 16,00% passou para 20,72%, totalizando 52 termos, ou seja, 11 a mais que na lista puramente estatística.

Embora tenha melhorado a extração estatística, o uso da *stop list* não foi capaz de revelar o mesmo número de termos candidatos a conceito como revelou a lista puramente linguística. Observando ainda o corpus *TACCESS*, nota-se que índice de revocação da lista linguística foi de 25,50%, frente à 20,72% da lista estatística com o uso da *stop list*. Esse mesmo comportamento se repetiu para os demais *corpora*, e foi relatado por Lopes *et al.* em um dos seus trabalhos [20] que utilizou a métrica *recall* para avaliar a qualidade das suas extrações.

O uso de *stop lists* para refinamento é bastante usual em PLN, e não poderia ser deixado de fora nesta pesquisa. Utilizá-lo foi fundamental para comprovar a eficácia desse processo frente às listas puramente estatísticas, e ainda gerou novos pontos de comparação com a aplicação das métricas de relevância. O uso do *stemming* foi desconsiderado, pois seu uso poderia implicar na perda de significância de alguns termos, em decorrência da remoção de partes do termo.

As listas geradas pelo processamento estatístico combinado com as métricas de relevância *tf-dcf* e *tf-idf* apresentaram um percentual de revocação bastante superior quando comparadas às listas estatística (E) e estatística com *stop list* (E + *stop list*). Para o *tf-dcf* (E + *tf-dcf*), foi encontrado um índice de revocação de 29,88%, com 75 termos localizados. Já para a lista E + *tf-idf* houve uma ligeira melhora, totalizando 77 termos, com 30,68% de revocação, conforme apresentado na Figura 6.

A avaliação da aplicação das métricas de relevância sobre listas processadas estaticamente é um dos objetivos desta pesquisa. O que se pode perceber até o momento é a destacável capacidade que as métricas têm de refinar as listas em questão. A melhora nos índices de revocação das listas (*recall*) é não só perceptível, como também confirma uma das suspeitas iniciais de que seu resultado do seu uso pode ser melhor que os resultados alcançados pelo uso de *stop lists*.

A fim de investigar mais a fundo os uso das métricas de relevância e de *stop lists* foi realizado o cálculo da revocação para as listagens que combinam as duas técnicas. Para ambas foi possível perceber uma melhora ainda maior em relação às listas já discutidas. Para aquela que combina o uso de *stop list* com *tf-dcf*, foi observado um índice de revocação de 30,68%, ou 77 termos. Ou seja, o mesmo resultado que aquele encontrado pela métrica *tf-idf* sem o uso da *stop list*.

Entretanto, a melhora fica ainda maior quando observada a revocação da lista que combina a *stop list* com *tf-idf*. O índice de revocação atingiu 33,07%, totalizando 83 termos e foi o melhor resultado não só para as listas já discutidas, como também para toda a amostra, o que indica que o

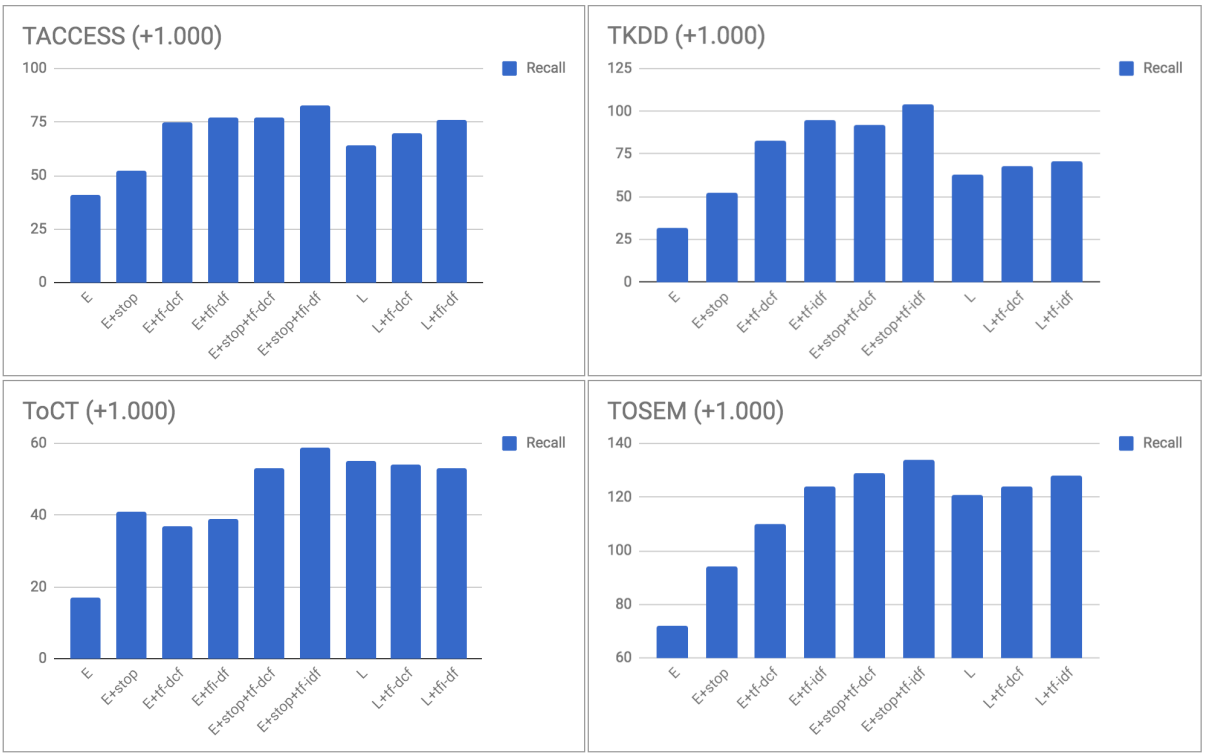


Figura 6 – Gráficos das tabelas que apresentam os resultados da revocação para o corte dos 1.000+.

uso combinado das duas técnicas pode potencializar o processo de refinamento das listas de termos extraídas pelo processamento estatístico.

A listagem gerada pelo processo puramente linguístico retornou uma revocação de 25,50%, ou 64 termos. Esse índice é inferior aos índices gerados pelas listas estatísticas que combinam as métricas de relevância, e principalmente inferior àquelas que combinam as mesmas métricas com a técnica da *stop list*. Nesse ponto da pesquisa pode-se comprovar que o uso das métricas de relevância pode não só melhorar o resultado da extração estatística, mas também se torna uma opção mais vantajosa em relação ao processamento puramente linguístico.

Para que fosse possível gerar uma comparação mais justa, foram avaliados os índices de revocação alcançados pela lista linguística processada pelas métricas de relevância. A percepção inicial é de que as métricas puderam retornar melhores resultados quando comparadas à lista puramente linguística. A listagem L + *tf-dcf* retornou um total de 70 termos, ou 27,89%. Enquanto a métrica *tf-idf* foi um pouco melhor, retornando 76 termos, ou 30,28%. Sendo assim, ficou evidente que o uso das métricas de relevância pode melhorar o refinamento das listas de termos geradas por ambas as abordagens (linguística e estatística).

Reforçando a percepção de que as métricas de relevância aplicadas ao processamento estatístico podem evoluir os resultados dessa abordagem, fica claro perceber que o processamento linguístico, com o uso das mesmas métricas, retornou resultados equivalentes. Mais interessante ainda foi perceber que entre as listas geradas pela aplicação das métricas sobre ambas as abordagens, a que teve melhor desempenho foi aquela que combinou a abordagem estatística com *tf-idf*.

O ponto alto da comparação entre os processos linguístico e estatístico ficou por conta da lista E + *stop* + *tf-idf*, com resultados superiores mesmo em relação ao processo linguístico com a mesma métrica. Essa avaliação indicou que mesmo o processamento linguístico sendo mais custoso computacionalmente, e em teoria retornando resultados mais precisos linguisticamente, diante de uma comparação quantitativa apoiada pela técnica da revocação, os resultados se mostram inferiores ao processamento estatístico em situações que são combinadas técnicas de refinamento. Esse comportamento, de forma geral, foi identificado também nos demais corpora processados nessa pesquisa, para as listas de termos com o mesmo número de corte. O único que apresentou resultados pouco diferentes foi o *ToCT*, mas todos possíveis de serem justificados.

A Tabela 15 apresenta, no mesmo formato, os resultados para o corpus *ToCT*. De todos os corpora processados esse foi o que apresentou resultados diferentes na aplicação das métricas sobre a lista linguística, onde notou-se que o uso das métricas não melhorou a performance das listas geradas, e sim, evidenciou-se uma piora.

Tabela 15 – Recall do corpus *ToCT* nas listas 1.000+.

Recall ToCT 1.000+ (1^2-gram)				
Lista	**Recall**	**Total Corpus**	**Total Lista**	**%**
E	17	260	1000	6,54
E+stop	41	260	1000	15,77
E+tf-dcf	37	260	1000	14,23
E+tfi-df	39	260	1000	15,00
E+stop+tf-dcf	53	260	1000	20,38
E+stop+tf-idf	59	260	1000	22,69
L	55	260	1000	21,15
L+tf-dcf	54	260	1000	20,77
L+tfi-df	53	260	1000	20,38

Apesar dessas divergências, a lista com menor índice de revocação também foi aquela gerada pelo processo puramente estatístico, resultando em 17 termos localizados dentre os 260 unigramas e bigramas de referência (6,54%). Da mesma forma, a listagem com melhor índice de revocação foi a E+*stop*+*tf-idf*, retornando 59 dos 260 termos referenciais (22,69%). Pode ser percebida também a evolução com o uso da *stop list* no refinamento da lista estatística. A listagem E + *stop* retornou 41 termos (15,77%), 24 termos a mais que na lista E.

Importante destacar que o processo puramente linguístico conseguiu retornar mais resultados que o processo estatístico combinado com a lista de parada (E + *stop list*). Esse mesmo comportamento foi percebido no corpus *TACCESS*, denotando a importância de se combinar as métricas com a *stop list* para poder alcançar melhores resultados estatísticos que os gerados pelo processo linguístico.

A fim de buscar explicações para o cenário divergente da aplicação das métricas nesse corpus, procurou-se investigar melhor as listas processadas, buscando identificar os termos que foram melhor ranqueados em cada processo. Percebeu-se que as listas estatísticas geradas pela aplicação

das métricas de relevância apresentaram unigramas do alfabeto (a, b, c em diante, geralmente usados em formulações matemáticas) como termos relevantes (posicionados no topo da lista). Durante a construção da *stop list*, esses termos foram adicionados a essa lista sobre a justificativa de que não teriam valor terminológico (discutido na seção 4.1.4). Nesse sentido, o efeito da aplicação da *stop list* sobre essa lista estatística foi mais perceptível que em outros corpora, pois o único corpus que continha esses termos era o *ToCT*.

Esses mesmos unigramas alfabéticos foram bem ranqueados na lista linguística, o que consequentemente resultou num baixo efeito sobre a aplicação das métricas de relevância, já que esses mesmos termos também foram bem ranqueados nas listas geradas pela aplicação das métricas, e não estão presentes na lista referencial do corpus *ToCT*.

O comportamento observado no corpus *TACCESS* pode ser observado também nos outros dois corpora: *TKDD* e *TOSEM*. A Tabela 16 evidencia o *recall* do corpus *TKDD* nas listas 1.000+, conforme a seguir:

Tabela 16 – Recall do corpus *TKDD* nas listas 1.000+.

Recall TKDD 1.000+ (1^2-gram)				
Lista	**Recall**	**Total Corpus**	**Total Lista**	**%**
E	32	424	1000	7,55
E+stop	52	424	1000	12,26
E+tf-dcf	83	424	1000	19,58
E+tf-idf	95	424	1000	22,41
E+stop+tf-dcf	92	424	1000	21,70
E+stop+tf-idf	104	424	1000	24,53
L	63	424	1000	14,86
L+tf-dcf	68	424	1000	16,04
L+tf-idf	71	424	1000	16,75

Para o corpus *TKDD*, a principal diferença fica por conta das listas E + *stop* + *tf-dcf* e E + *tf-idf*. Enquanto que no corpus TACCESS (Tabela 14) observou-se um empate no índice de revocação das duas listas, para o corpus *TKDD* o processamento estatístico com a métrica *tf-idf* foi melhor que o processamento estatístico combinado com a *stop list* e a métrica *tf-dcf* (diferença de três termos entre eles). Importante notar também que a lista E + *stop* + *tf-idf* foi bastante superior a todas as demais listas processadas para esse corpus, atingindo 104 termos dos 424 da listagem referencial (24,53%).

O mesmo ocorreu para o corpus *TOSEM*, conforme pode-se observar na Tabela 17, onde a melhor extração foi aquela representada pela lista E + *stop* + *tf-idf*. Para esse corpus, o total de termos localizados foi de 134, dos 636 termos de referência (21,07%). Na outra ponta da análise, a lista E foi a que apresentou pior performance, totalizando apenas 72 dos 636 termos referenciais (11,32%), comportamento esse também encontrado em todos os corpora estudados com corte de 1.000 termos nas listas processadas.

Tabela 17 – Recall do corpus *TOSEM* nas listas 1.000+.

Recall TOSEM 1.000+ (1^2-gram)				
Lista	**Recall**	**Total Corpus**	**Total Lista**	**%**
E	72	636	1000	11,32
E+stop	94	636	1000	14,78
E+tf-dcf	110	636	1000	17,30
E+tf-idf	124	636	1000	19,50
E+stop+tf-dcf	129	636	1000	20,28
E+stop+tf-idf	134	636	1000	21,07
L	121	636	1000	19,03
L+tf-dcf	124	636	1000	19,50
L+tf-idf	128	636	1000	20,13

O efeito da aplicação das métricas de relevância *tf-dcf* e *tf-idf* sobre as listas geradas por ambas as abordagens é o mesmo que aquele identificado nos corpora *TACCESS* e *TKDD*. Pode-se notar uma melhora considerável na taxa de revocação das listas que usaram essas métricas frente aos métodos puros. Isso demonstra como a técnica do contraste pode evoluir consideravelmente os resultados da extração, eliminando termos dispensáveis e que aparecem repetidamente em outros corpora. A combinação dessa técnica com a *stop list* para o processamento estatístico gerou um resultado ainda superior em relação aos métodos linguísticos, o que corrobora para a tese de que a abordagem estatística, combinada com outras técnicas pode substituir a abordagem linguística quando o objetivo for a extração automática de termos. Resumidamente, através da análise realizada nas listas processadas com corte de 1.000 termos, pode-se compreender que:

- a abordagem puramente estatística (E) é a menos eficiente quando se pretende extrair termos relevantes de corpora;
- a abordagem puramente linguística (L) é bastante superior à puramente estatística (E);
- a aplicação das métricas de relevância pode evoluir os resultados de ambas as abordagens (E + *tf-dcf* e E + *tf-idf*);
- o uso das métricas na extração estatística (E + *tf-dcf* e E + *tf-idf*), em alguns casos, pode retornar resultados superiores àqueles encontrados na abordagem puramente linguística (L);
- o uso de *stop lists* evolui os resultados da extração estatística (E + *stop lists*), mas não o suficiente para ser melhor que a lista linguística (L);
- a combinação da *stop list* com métricas de extração foi a abordagem com melhor desempenho (E + *stop* + *tf-dcf* e E + *stop* + *tf-idf*);
- os números das extrações podem variar um pouco para cada corpus, mas no geral os resultados seguem sendo os mesmos.

5.2 Recall com corte de 100 termos

Após analisar os resultados alcançados nas listas com 1.000 termos, houve interesse também em avaliar o comportamento do índice de revocação em listas menores. Para isso, foi eleito o número 100 como novo corte nas listas processadas, cujo objetivo principal foi avaliar se o comportamento apresentado na primeira análise se mantinha para listas menores.

Em termos gerais os resultados foram semelhantes, mas alguns pontos merecem destaque: para o *corpus TACCESS* por exemplo, pode-se notar que a lista E + *tf-dcf* (5,18%) não apresentou igual desempenho como na análise anterior - apesar de ter melhorado os resultados da abordagem estatística (E), houve uma piora em relação à lista estatística filtrada pela *stop list* (5,58%); já para lista que combinou o uso da métrica *tf-idf*, o desempenho foi bastante superior, batendo os 10,36% de revocação.

Vale destacar também que o desempenho das métricas sobre a lista linguística desse corpus se manteve como apresentado no corte dos 1.000 termos, ou seja, as métricas foram capazes de melhorar consideravelmente a extração linguística. Embora na lista estatística a métrica *tf-dcf* tenha tido um resultado inferior à *tf-idf*, ao analisar a aplicação sobre a lista linguística o cenário se inverte, retornando uma revocação pouco maior, 5,18%, contra 3,98%. Neste sentido, a Tabela 18 demonstra o *recall* do corpus *TACCESS*, conforme demonstrado abaixo.

Tabela 18 – Recall do corpus *TACCESS* nas listas 100+.

Recall TACCESS 100+ (1^2-gram)				
Lista	**Recall**	**Total Corpus**	**Total Lista**	**%**
E	3	251	100	1,20
E+stop	14	251	100	5,58
E+tf-dcf	13	251	100	5,18
E+tf-idf	26	251	100	10,36
E+stop+tf-dcf	29	251	100	11,55
E+stop+tf-idf	30	251	100	11,95
L	7	251	100	2,79
L+tf-dcf	13	251	100	5,18
L+tf-idf	10	251	100	3,98

Para o corpus *ToCT*, os resultados da aplicação do novo corte mostraram que o uso das métricas de refinamento tiveram desempenho superior à lista filtrada pela *stop list*, chegando a 3,85% de revocação para a métrica *tf-idf* e 2,69% para *tf-dcf*, frente a 2,31% da lista filtrada pela lista de parada. Isso mais uma vez demonstra o poder da aplicação das métricas, resultando em resultados positivos mesmo em listas menores. A combinação das métricas com a *stop list* para esse corpus não apresentou uma melhora tão significativa, mostrando até que para esse cenário, a métrica *tf-idf* sozinha foi capaz de atingir o mesmo resultado que aquele apresentado pela combinação com a lista de parada (3,85%). Os gráficos que ilustram os resultados do corte dos 100+ pode ser encontrado na Figura 7.

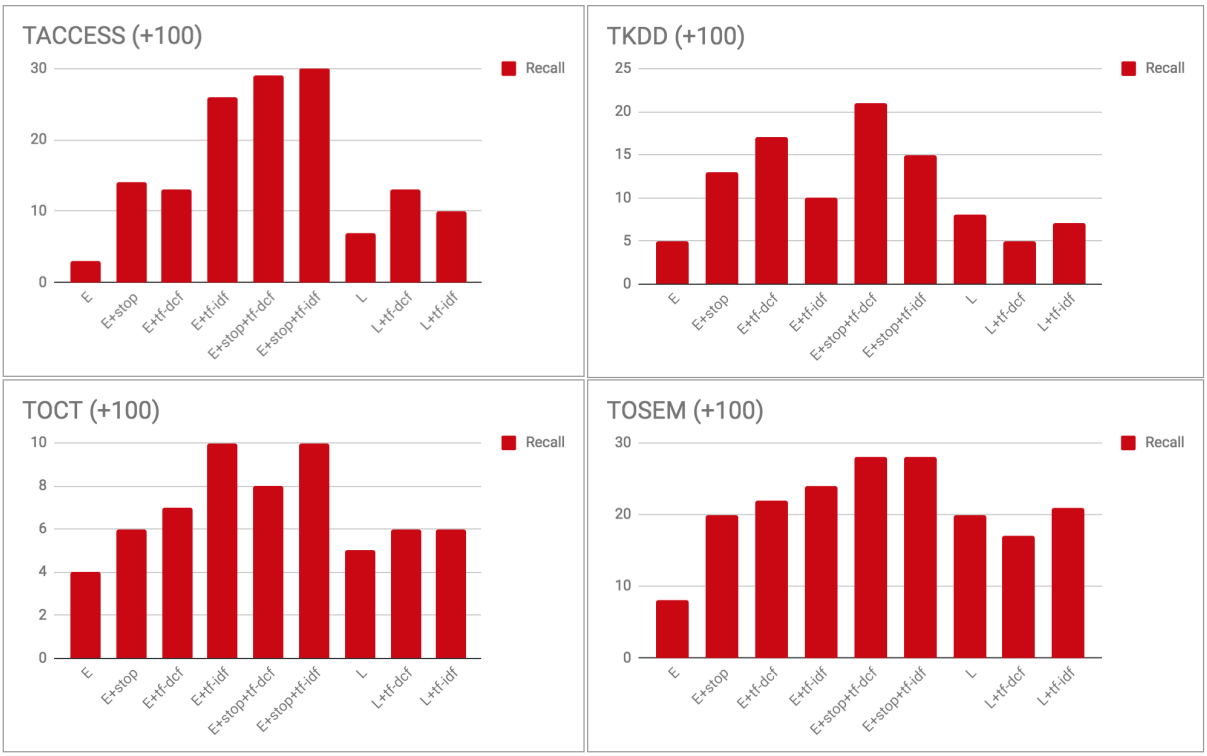


Figura 7 – Gráficos das tabelas que apresentam os resultados da revocação para o corte dos 1.00+.

Na outra ponta, mais uma vez o método linguístico perdeu para o estatístico combinado com as técnicas de refinamento. O resultado da aplicação das métricas sobre as listagens linguísticas retornou resultados inferiores (2,31%) àqueles apresentados pela aplicação das mesmas métricas sobre a lista estatística. Em termos mais práticos, pode-se dizer que os resultados são semelhantes, pois a diferença chega a ser de um termo entre as listas, porém a expectativa era de que o método linguístico apresentasse uma eficácia melhor. Por outro lado, o uso das métricas na lista linguística foi capaz de melhorar a revocação da lista revelando um termo a mais para as duas métricas.

Tabela 19 – Recall do corpus *ToCT* nas listas 100+.

Recall ToCT 100+ (1^2-gram)				
Lista	**Recall**	**Total Corpus**	**Total Lista**	**%**
E	4	260	1000	1,54
E+stop	6	260	1000	2,31
E+tf-dcf	7	260	1000	2,69
E+tf-idf	10	260	1000	3,85
E+stop+tf-dcf	8	260	1000	3,08
E+stop+tf-idf	10	260	1000	3,85
L	5	260	1000	1,92
L+tf-dcf	6	260	1000	2,31
L+tf-idf	6	260	1000	2,31

Para o corpus *TKDD* (Tabela 20), a aplicação da métrica *tf-idf* (2,36%) sobre a lista estatística teve um comportamento diferente dos demais corpora. Embora tenha evoluído os re-

sultados do método puramente estatístico (1,18%), ela não foi capaz de ser superior ao método estatístico combinado com o uso da *stop list* (3,07%). Por outro lado, a métrica *tf-dcf* cumpriu seu papel retornando um resultado superior (4,01%) àquele apresentado na lista E + *stop list*. Esse acontecimento reforça mais uma vez a importância de combinar os métodos para se obter resultados eficazes no refinamento de listas estatísticas.

Esse *corpus*, quando analisado com um corte de 100 termos nas listas, mostrou um desempenho inferior relevante da sua lista puramente linguística (1,89%) comparada à estatística com *stop list* (3,07%). Todos os demais *corpora* e cortes fizeram o inverso, colocando a extração linguística como uma melhor opção, mas para esse *corpus* não foi seguido esse padrão.

O uso combinado das métricas de refinamento com o filtro da *stop list*, mais uma vez mostrou ser a abordagem campeã para a extração automática de termos. A métrica *tf-dcf* combinada com a *stop list* foi capaz de retornar um índice de 4,95% de revocação, ou seja, o melhor resultado para esse corpus. A métrica *tf-idf* combinada também com o filtro da lista de parada retornou 3,54%, nesse caso resultado um pouco inferior que a aplicação do *tf-dcf* sobre a lista estatística (4,01%).

Para abordagem linguística, o comportamento das métricas de refinamento não foi dos melhores. Infelizmente elas não conseguiram melhorar a qualidade da extração, e acabaram retornando um número menor de termos que a listagem puramente linguística, também observado no corpus *TOSEM* para apenas uma das métricas (*tf-dcf*). Para garantir que o processamento foi realizado corretamente, a aplicação das métricas sobre o processamento linguístico para esse corpus foi refeito, mas ainda assim os resultados seguiram sendo os mesmos. Infelizmente não foi possível encontrar uma explicação concreta para esse resultado, já que outros dois corpora tiveram comportamento diferente para listas de 100 termos, onde percebeu-se uma melhora na extração linguística realizada.

Tabela 20 – Recall do corpus *TKDD* nas listas 100+.

Recall TKDD 100+ (1^2-gram)				
Lista	**Recall**	**Total Corpus**	**Total Lista**	**%**
E	5	424	1000	1,18
E+stop	13	424	1000	3,07
E+tf-dcf	17	424	1000	4,01
E+tf-idf	10	424	1000	2,36
E+stop+tf-dcf	21	424	1000	4,95
E+stop+tf-idf	15	424	1000	3,54
L	8	424	1000	1,89
L+tf-dcf	5	424	1000	1,18
L+tf-idf	7	424	1000	1,65

Para o quarto e último corpus analisado, *TOSEM* (Tabela 21), a lista com corte de 100 termos apresentou resultados equivalentes ao primeiro corte dos 1.000 termos para as listas estatísticas que combinaram técnicas de refinamento. Novamente foi possível notar que as métricas

estatísticas melhoraram a qualidade da extração final (3,46% para *tf-dcf* e 3,77% para *tf-idf*) ao ponto de serem superiores à extração com *stop list* (3,14%). Nesse corpus, a combinação das duas técnicas mostrou um resultado interessante. O índice de revocação resultante foi o mesmo para as duas listas (4,40% para E + *stop list* + *tf-dcf* e E + *stop list* + *tf-idf*), sendo inclusive o melhor resultado de extração para esse corpus.

Para extração linguística, obteve-se índice de revocação de 3,14%, com destaque para a métrica *tf-idf* que evoluiu os resultados para 3,30%. Infelizmente o índice *tf-dcf* não apresentou resultado satisfatório, penalizando o refinamento, resultando em 2,67% de revocação.

Tabela 21 – Recall do corpus *TOSEM* nas listas 100+.

Recall TOSEM 100+ (1^2-gram)				
Lista	**Recall**	**Total Corpus**	**Total Lista**	**%**
E	8	636	100	1,26
E+stop	20	636	100	3,14
E+tf-dcf	22	636	100	3,46
E+tf-idf	24	636	100	3,77
E+stop+tf-dcf	28	636	100	4,40
E+stop+tf-idf	28	636	100	4,40
L	20	636	100	3,14
L+tf-dcf	17	636	100	2,67
L+tf-idf	21	636	100	3,30

Da mesma forma como foi apresentado para o corte dos 1.000 termos, esses são os pontos da análise dos 100 primeiros termos que merecem destaque:

- novamente a lista puramente estatística (E) apresentou o pior desempenho entre todas;
- a extração puramente linguística (L) foi melhor que a puramente estatística (E);
- o uso da técnica da *stop list* mais uma vez evoluiu os resultados da extração estatística (E + *stop list*);
- a aplicação das métricas sobre as listas estatísticas evoluiu os resultados dessa extração, mas em dois corpora, ao menos uma das métricas não conseguiu superar a limpeza feita com a *stop list*;
- a combinação das métricas de relevância com a técnica da *stop list* mais uma vez mostrou-se ser a melhor abordagem (E + *stop* + *tf-dcf* e E + *stop* + *tf-idf*);
- ambas as métricas não puderam evoluir os resultados da extração linguística em um dos corpora; e em outro o *tf-dcf* apresentou também esse resultado;
- de forma geral, pode-se dizer que os resultados do corte 100+ são equivalentes ao corte 1.000+.

5.3 Resultado Final

Para apoiar a compreensão final dos resultados alcançados na pesquisa, foram geradas mais quatro tabelas (APÊNDICE F e APÊNDICE G), acompanhadas de gráficos, ilustrando a combinação dos resultados das listas de 1.000 e 100 termos dos quatro corpora processados. Dessa forma, fica possível perceber visualmente os nuances do índice de revocação resultante em cada uma das listas geradas. Os resultados das listas de 1.000 termos podem ser encontrados na cor azul nos gráficos seguintes, enquanto os resultados descobertos para as listas de 100 termos estão em vermelho.

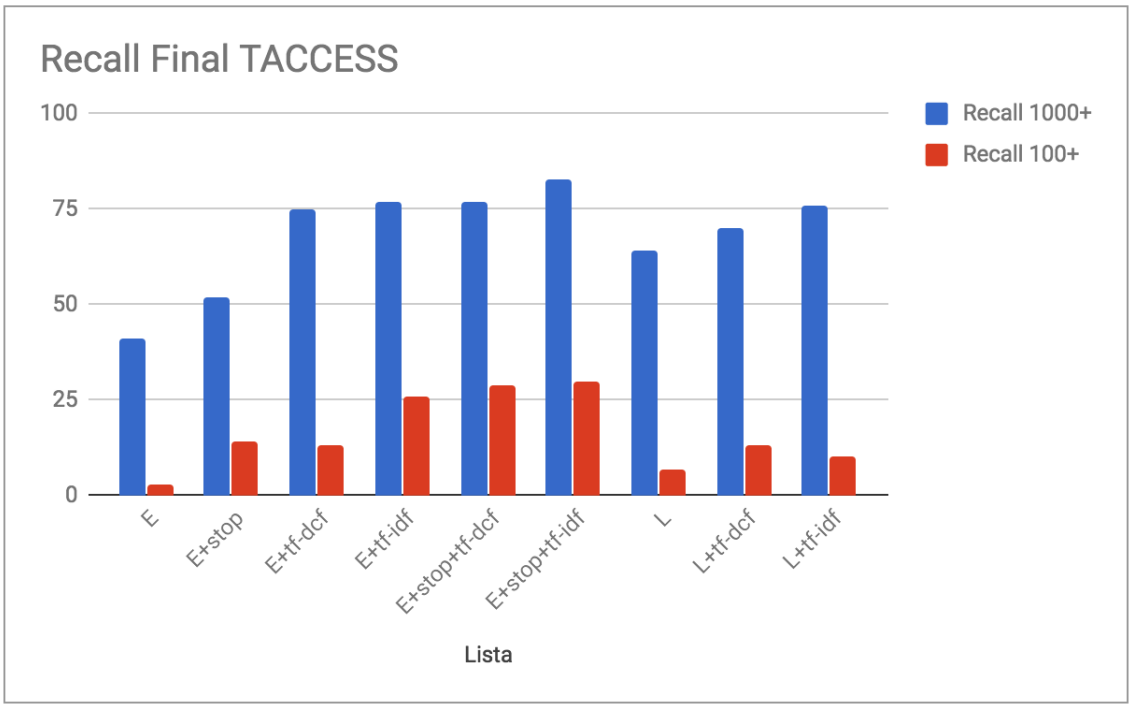


Figura 8 – Gráfico da Tabela 23 apresentando o resultado dos dois cortes (100+ e 1.000+) para o corpus *TACCESS*.

Para o corpus *TACCESS* por exemplo (Figura 8), pode-se notar o crescimento desigual da barra da lista E + *tf-dcf* composta por 100 termos (em vermelho). Ao invés de acompanhar o crescimento da barra azul, indicando uma melhora da aplicação da métrica frente ao uso da *stop list*, houve uma diminuição do índice de revocação. O comportamento inverso pode ser observado no gráfico do corpus *ToCT* (Figura 9), representado pela Figura 9, onde a aplicação da métrica *tf-dcf* não surtiu efeito positivo quando comparada à lista E + *stop*. Nessa mesma linha, pode-se perceber a diferença entre os resultados alcançados pela aplicação da métrica *tf-idf* na lista linguística do corpus *TACCESS*. Para o corte de 1.000 termos houve uma melhora em relação à aplicação da métrica *tf-dcf*, porém quando observado o resultado sobre o corte de 100 termos, percebe-se uma piora. Os gráficos dos *corpora* TKDD e TOSEM podem ser conferidos nas Figuras 10 e 11, respectivamente.

Apesar de algum ou outro resultado divergente para os cortes realizados, deve-se destacar que a abordagem que apresentou melhor índice de revocação (E + *stop list* + *tf-idf*) é nitidamente

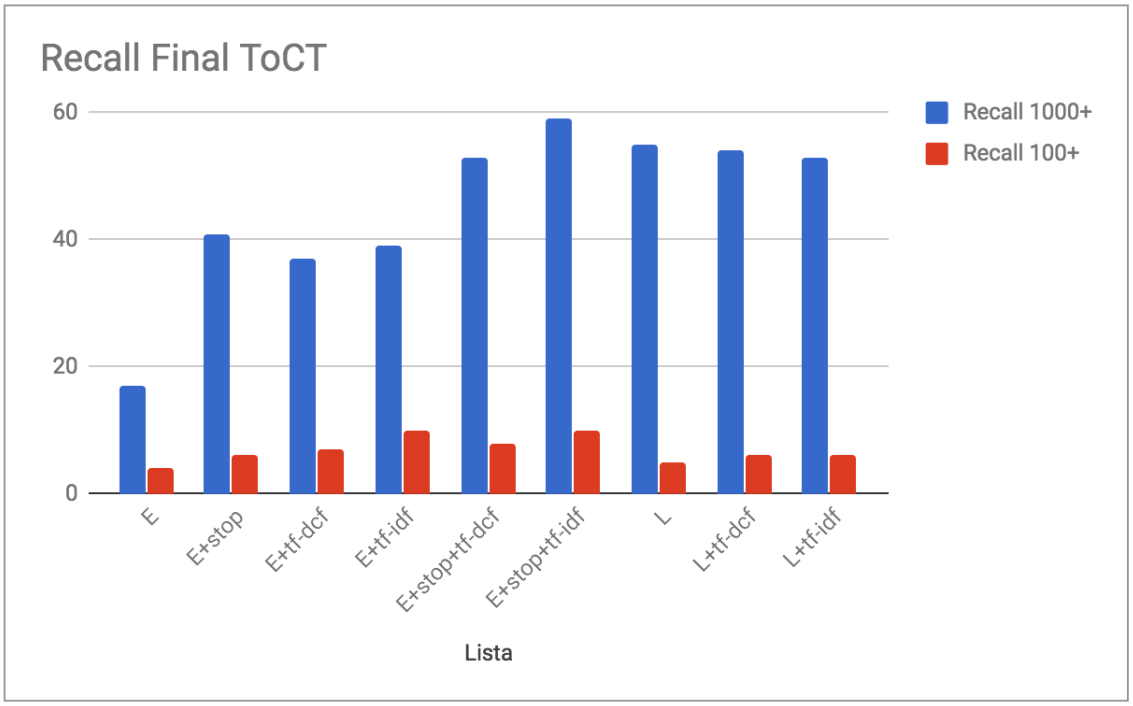


Figura 9 – Gráfico da Tabela 24 apresentando o resultado dos dois cortes (100+ e 1.000+) para o corpus *ToCT*.

percebida nos gráficos. Da mesma forma, pode-se notar que a abordagem puramente estatística tem a menor barra entre todas.

Um ponto questionado durante a aplicação dos cortes foi como os termos relevantes (presentes também na lista de referência do corpus) estão distribuídos nas listagens, a partir da seguinte questão: proporcionalmente, as listas com 1.000 termos revelaram mais ou menos termos relevantes que as listas de 100 termos? Para tentar responder a essa pergunta foi realizada uma última análise, tentando encontrar um índice que representasse a proporcionalidade da revocação em cada corte das listas.

O cálculo foi realizado dividindo o número de termos revelados no *recall* de cada lista pelo número do corte aplicado. Neste sentido, foi calculada a média desse número para todas as listas e o intervalo de valores possíveis ficou entre 0 e 1. Quanto mais próximo de 1, mais completa por termos de referência a lista está, quando próximo de 0, menos termos de referência foram encontrados. Esse índice foi chamado de índice de revocação proporcional. Abaixo está a formulação desse cálculo:

$$\mathbf{rp} = \frac{\sum \frac{R_n}{c}}{|L|}$$

onde R é o índice de revocação para uma lista n, dividido pelo número de corte aplicado c. O resultado desse somatório é dividido pelo número total de listas processadas L, no caso dessa pesquisa $|L| = 9$.

Por exemplo, a lista estatística (E) do corpus *TKDD* apresentou revocação de 32 termos para o corte 1.000, assim calculou-se 32/1.000, resultando 0,032. Esse processo foi aplicado para

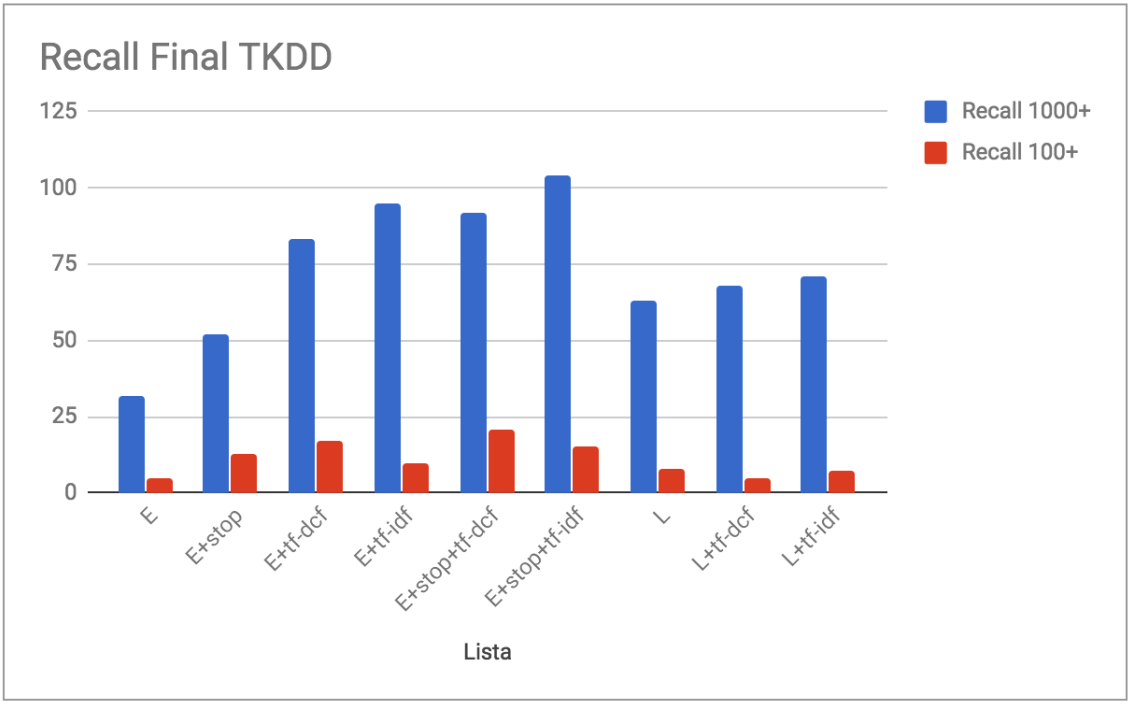


Figura 10 – Gráfico da Tabela 25 apresentando o resultado dos dois cortes (100+ e 1.000+) para o corpus *TKDD*.

as demais listas processadas do referido corte, e então calculada a média para obter o índice de revocação proporcional desse corte para o corpus *TKDD*. Isso foi repetido para o corte dos 100, e também para os demais corpora. A fim de facilitar a leitura o resultado final foi multiplicado por 100. O resultado desses cálculos pode ser conferido na Tabela 22.

Tabela 22 – Média da revocação proporcional para todos os cortes.

Média da revocação proporcional (%)				
Corte	TACCESS	ToCT	TKDD	TOSEM
100	16,10	6,80	11,20	20,80
1.000	6,80	4,50	7,30	11,50

Foi possível notar que as listas com 100 termos apresentaram um índice de revocação proporcional maior que as listas de 1.000 termos. Isso indica que os termos referenciais, ou conceitos, estão presentes mais na parte de cima das listas, ou seja, eles são melhor ordenados que aqueles que não estão presentes na lista de referência dos corpora.

Para o corpus *TACCESS* por exemplo, nas listas filtradas com 100 termos o índice proporcional de revocação foi de 16,10%, enquanto que nas listas compostas por 1.000 termos foi de apenas 6,80%. Esta breve análise indica que as abordagens de extração de termos aplicadas nessa pesquisa são capazes de penalizar corretamente termos dispensáveis do conceito central de cada corpus, aplicando uma medida de relevância inferior. Elas também ajudam e revelar termos que são aderentes a esse contexto (termos ditos conceitos), pontuando-os para que fiquem bem ranqueados nessas listas.

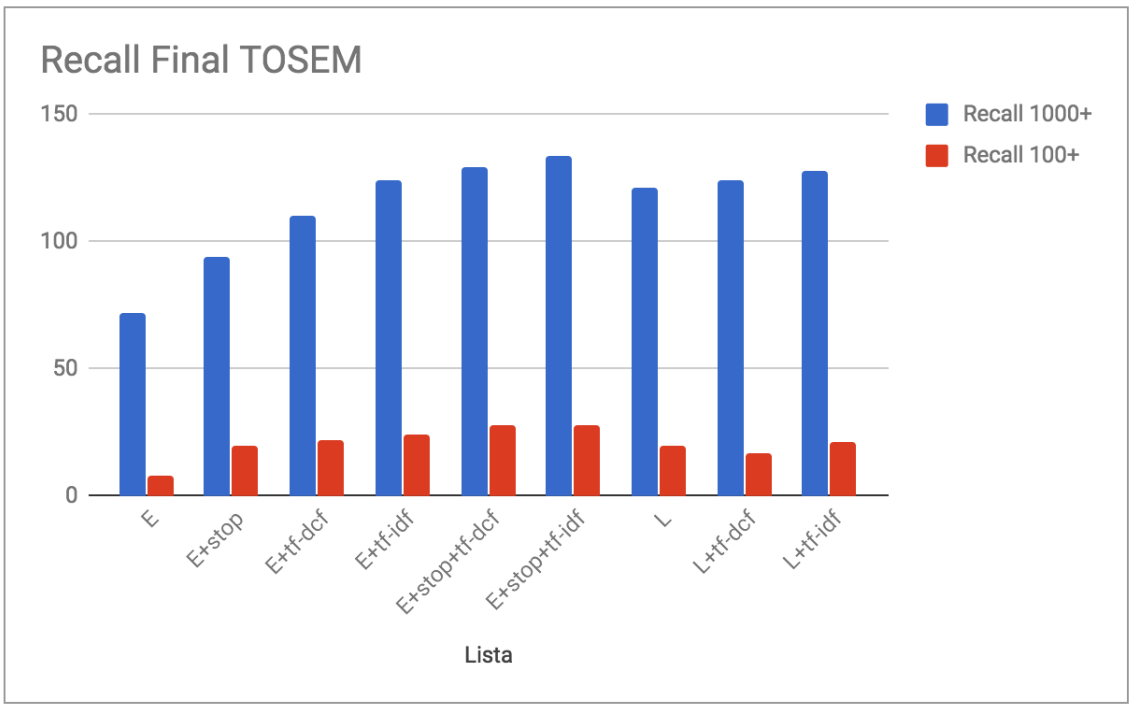


Figura 11 – Gráfico da Tabela 26 apresentando o resultado dos dois cortes (100+ e 1.000+) para o corpus *TOSEM*.

Em resumo, a utilização dessas técnicas de refinamento pode produzir resultados bastante satisfatórios na geração de listas de conceitos. A aplicação do corte ideal na lista gerada é fundamental para obter o melhor número proporcional de conceitos possível. Alguns autores, como Lopes *et.al.* [19] apresentam sua própria metodologia de corte nas listas de termos processadas, a fim de obter o melhor das extrações realizadas.

6. CONCLUSÃO

O objetivo central desta dissertação foi realizar a comparação entre as abordagens linguística e estatística para extração automática de termos de *corpora*, procurando avaliar qual processo é capaz de revelar maior número de termos aderentes ao domínio de cada *corpus*.

Esse objetivo foi alcançado e experimentado em cima de quatro *corpora* de domínios diferentes da computação, construídos a partir de artigos científicos de revistas eletrônicas do Portal ACM[1]. Juntos, esses corpora totalizaram mais de 6 milhões de palavras. Todas as listas de termos geradas pelos experimentos foram comparadas com a lista de referência de cada *corpus* utilizando a técnica do *recall*.

Para alcançar esse objetivo foi inicialmente realizada a fundamentação teórica apresentada no Capítulo 2 deste documento, onde se discursou sobre a importância da extração de termos em *corpora* de domínio. Foram apresentadas as abordagens linguística e estatística, descrevendo o funcionamento das ferramentas utilizadas por elas. Foram também apresentadas as métricas de relevância adotadas para refinamento das listas extraídas. Enfim, ainda nesse capítulo, os trabalhos relacionados foram discutidos para melhor justificar e apoiar a execução dessa pesquisa.

No Capítulo 3 foram apresentados os detalhes sobre a execução dos experimentos, abordando a construção dos *scripts* desenvolvidos sob medida para apoiar o uso das ferramentas de extração. No Capítulo 4 os resultados alcançados foram apresentados, procurando detalhar o índice de revocação encontrado em cada um dos cortes aplicados nas listas, de 100 e 1.000 termos, e as diferenças entre eles. Por fim, apresentou-se um resumo desses resultados num subcapítulo chamado Resultado Final.

6.1 Contribuição Científica

Na busca pelo objetivo desta dissertação foram constatados comportamentos que já eram esperados por alguns trabalhos [3, 8, 19, 24], mas também desenvolvidos alguns avanços científicos:

- foi constatado que a extração puramente linguística é bastante superior à extração puramente estatística;
- a aplicação de métricas de relevância pode não só evoluir os resultados da abordagem estatística, como também podem torná-la superior aos resultados alcançados pela extração puramente linguística;
- uma *stop list* construída com pesquisas na Internet e melhorada com algum conhecimento sobre os textos pode produzir resultados bastante satisfatórios;

[1] https://dl.acm.org/

- a utilização de *stop list* combinada com métricas de relevância para abordagem estatística pode produzir resultados superiores àqueles encontrados na extração linguística com as mesmas métricas;

- a métrica *tf-idf* teve um resultado superior à *tf-dcf* numa abordagem de comparação quantitativa (utilizando-se o *recall*).

Os resultados alcançados pelos trabalhos de Lopes *et.al.* [19, 21] no desenvolvimento e aplicação da métrica *tf-dcf* mostram extrações com qualidade superior ao *tf-idf* e às demais métricas, como as citadas na seção 2.4.4. No caso desta pesquisa, avaliou-se o resultado das extrações tendo um foco mais quantitativo, onde aparentemente a métrica *tf-dcf* teve um desempenho inferior ao *tf-idf*, revelando menos termos nas listas apresentadas. Como não foi realizada uma análise posicional dos termos individualmente não foi possível comprovar qual das duas métricas apresentou um resultado qualitativo melhor nessa ocasião.

6.2 Trabalhos Futuros

No contexto desta pesquisa, seria interessante revalidar com especialistas as listas de referência que foram geradas para cada *corpus*. Com isso, seria possível rodar novamente os algoritmos de extração e contagem da revocação das listas, a fim de comparar os resultados com aqueles que foram apresentados neste documento. Não que se questione o método que foi utilizado para compor as listas de referência, mas acredita-se que essa comparação seria interessante para obter uma avaliação mais qualitativa dos resultados.

Para fornecer um resultado mais qualitativo do uso das métricas utilizadas neste trabalho seria interessante avaliar posicionalmente os termos nas listas geradas por essas métricas, tentando eleger aquela que melhor posicionou conceitos no topo das listas resultantes, outra opção seria diminuir o número de corte das listas.

Outro ponto que pode ser explorado seria a aplicação de outras métricas de relevância, como as que foram apresentadas na seção 2.4.4, a fim de traçar novos comparativos com os resultados que já foram obtidos até aqui. A utilização de outras ferramentas para extração estatística de termos é uma opção de pesquisa e aplicação também.

Portanto, há muito que pode ser feito para seguir testando novas abordagens de extração de termos. Acredita-se que refinar as listas de referência com especialistas de cada área e utilizar diferentes métricas de relevância no processo de refinamento poderiam ser os primeiros passos para continuidade desse trabalho.

A tarefa de extração de termos de corpora é uma das áreas de PLN que oferece inúmeros desafios para comunidade acadêmica, e por isso vem sendo bastante pesquisada. Atualmente, uma

gama de softwares analisadores de texto vêm sendo desenvolvidos para inúmeros idiomas [2], assim como novas métricas de relevância seguem sendo testadas para apoiar a extração de termos.

O que se pretende num âmbito mais geral seria o acompanhamento dessas novidades que seguem surgindo, principalmente de ferramentas que utilizam a linguagem *Python* e suas bibliotecas próprias para tarefas de PLN, por serem de fácil implementação e customização.

[2] https://nlp.stanford.edu/software/lex-parser.htmlHistory

REFERÊNCIAS BIBLIOGRÁFICAS

[1] Almeida, T.; Nakamura, F.; Nakamura, E. "Uma abordagem para identificar e monitorar haters em redes sociais online". In: Anais do XXIII Simpósio Brasileiro de Sistemas Multimídia e Web: Workshops e Pôsteres, 2017, pp. 41–46.

[2] Banerjee, S.; Pedersen, T. "The design, implementation, and use of the ngram statistics package". In: Proceedings of the 4th International Conference on Computational Linguistics and Intelligent Text Processing, 2003, pp. 370–381.

[3] Bates, M.; Weischedel, R. "Challenges in Natural Language Processing". Cambridge University Press, 2006, 312p.

[4] Bick, E. "The parsing system Palavras: automatic grammatical analysis of Portuguese in a constraint grammar framework". Aarhus University Press, 2000, 411p.

[5] Bordea, G.; Buitelaar, P.; Polajnar, T. "Domain-independent term extraction through domain modelling". In: Proceedings of the 10th International Conference on Terminology and Artificial Intelligence, 2013, pp. 9.

[6] Bowman, S. R.; Potts, C.; Manning, C. D. "Recursive neural networks for learning logical semantics", *Computing Research Repository*, vol. abs/1406.1827, Julho 2015, pp. 12–21.

[7] Brazil, W. "Web semântica". Capturado em: http://www.w3c.br/Padroes/WebSemantica, Abril 2018.

[8] Chomsky, N. "Syntactic Structures". Mouton, 1957, 117p.

[9] Croft, W.; Harper, D. "Using probabilistic models of document retrieval without relevance information", *Journal of Documentation*, vol. 35–4, Abril 1979, pp. 285–295.

[10] da Silva Conrado, M.; Felippo, A.; Pardo, T. S.; Rezende, S. O. "A survey of automatic term extraction for brazilian portuguese", *Journal of The Brazilian Computer Society (Online)*, vol. 20, Dezembro 2014, pp. 1–28.

[11] de Marneffe, M.-C.; MacCartney, B.; Manning, C. D. "Generating typed dependency parses from phrase structure parses". In: Proceedings of the International Conference on Language Resources and Evaluation, 2006, pp. 449–454.

[12] Group, T. S. N. L. P. "Stanford parser online". Capturado em: http://nlp.stanford.edu:8080/parser/index.jsp, Março 2018.

[13] Gruber, T. R. "Toward principles for the design of ontologies used for knowledge sharing", *International Journal of Human-Computer Studies*, vol. 43–5-6, Dezembro 1995, pp. 907–928.

[14] Jones, K. S. "A statistical interpretation of term specificity and its application in retrieval", *Journal of Documentation*, vol. 28, Maio 1972, pp. 11–21.

[15] Kao, A.; Poteet, S. R. "Natural Language Processing and Text Mining". Springer Publishing Company, 2006, 265p.

[16] Kim, S. N.; Baldwin, T.; Kan, M.-Y. "Extracting domain-specific words - a statistical approach". In: Proceedings of the Australasian Language Technology Association Workshop, 2009, pp. 94–98.

[17] Kit, C.; Liu, X. "Measuring mono-word termhood by rank difference via corpus comparison", *Terminology. International Journal of Theoretical and Applied Issues in Specialized Communication*, vol. 14–2, Dezembro 2008, pp. 204–229.

[18] Kurdi, M. Z. "Natural Language Processing and Computational Linguistics: Speech, Morphology and Syntax". John Wiley e Sons Incorporated, 2015, 296p.

[19] Lopes, L. "Extração automática de conceitos a partir de textos em língua portuguesa", Tese de Doutorado, Pontifícia Universidade Católica do Rio Grande do Sul, Porto Alegre, Brasil, 2012, 156p.

[20] Lopes, L.; de Oliveira, L. H. M.; Vieira, R. "Portuguese term extraction methods: Comparing linguistic and statistical approaches". In: Proceedings of the International Conference on Computational Processing of the Portuguese Language, 2010, pp. 6.

[21] Lopes, L.; Fernandes, P.; Vieira, R. "Estimating term domain relevance through term frequency, disjoint corpora frequency - tf-dcf", *Knowledge-Based Systems*, vol. 97, Abril 2016, pp. 237 – 249.

[22] Lopes, L.; Fernandes, P.; Vieira, R. "Exato – high quality term extraction for portuguese and english". In: Proceedings of the International Conference on Web Intelligence, 2016, pp. 1–6.

[23] Lopes, L.; Fernandes, P.; Vieira, R.; Fedrizzi, G. "Exatolp an automatic tool for term extraction from portuguese language corpora". In: Proceedings of the 4th Language and Technology Conference, 2009, pp. 427–431.

[24] Lopes, L.; Vieira, R. "Improving portuguese term extraction". In: Proceedings of the 10th International Conference on Computational Processing of the Portuguese Language, 2012, pp. 85–92.

[25] Park, Y.; Patwardhan, S.; Visweswariah, K.; Gates, S. C. "An empirical analysis of word error rate and keyword error rate". In: Proceedings of the 9th Annual Conference of the International Speech Communication Association, 2008, pp. 22–26.

[26] Pedersen, T. "Usage - nsp". Capturado em: http://search.cpan.org/dist/Text-NSP/doc/USAGE.pod, Janeiro 2010.

[27] Pedersen, T.; Banerjee, S.; McInnes, B. T.; Kohli, S.; Joshi, M.; Liu, Y. "The ngram statistics package (text::nsp): A flexible tool for identifying ngrams, collocations, and word associations". In: Proceedings of the Workshop on Multiword Expressions: From Parsing and Generation to the Real World, 2011, pp. 131–133.

[28] Perna, C.; Lopes, L.; Rollsing, L. "Português para fins acadêmicos sob o aporte da linguística de corpus e do processamento de linguagem natural", vol. 11, Abril 2017, pp. 379–393.

[29] Priberam, D. "Definição da palavra conceito". Capturado em: https://www.priberam.pt/dlpo/conceito, Abril 2018.

[30] Priberam, D. "Definição da palavra termo". Capturado em: https://www.priberam.pt/dlpo/termo, Abril 2018.

[31] Richard Socher and Eric H. Huang and Jeffrey Pennington and Andrew Y. Ng and Christopher D. Manning. "Dynamic Pooling and Unfolding Recursive Autoencoders for Paraphrase Detection". In: *Advances in Neural Information Processing Systems 24*, Neural Information Processing Systems Foundation, 2011, pp. 1–9.

[32] Robertson, S. E.; Walker, S. "On relevance weights with little relevance information", *Proceedings of the 20th annual international conference on Research and development in information retrieval*, vol. 31–SI, Julho 1997, pp. 16–24.

[33] Smadja, F. "Retrieving collocations from text: Xtract", *Computational Linguistics*, vol. 19–1, Março 1993, pp. 143–177.

[34] Socher, R.; Bauer, J.; Manning, C. D.; Ng, A. Y. "Parsing with compositional vector grammars". In: Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics, 2013, pp. 455–465.

[35] Socher, R.; Huval, B.; Manning, C. D.; Ng, A. Y. "Semantic Compositionality Through Recursive Matrix-Vector Spaces". In: Proceedings of the 2012 Conference on Empirical Methods in Natural Language Processing, 2012, pp. 11.

[36] Witten, I. H.; Frank, E.; Hall, M. A. "Data Mining: Practical Machine Learning Tools and Techniques". Morgan Kaufmann Publishers Incorporated, 2011, 558p.

[37] Witten, I. H.; Moffat, A.; Bell, T. C. "Managing Gigabytes: Compressing and Indexing Documents and Images". Morgan Kaufmann Publishers Incorporated, 1999, 550p.

[38] Ábia, B.; Almeida, T.; Menezes, A.; Nakamura, F.; Figueiredo, C.; Nakamura, E. "For or against?: Polarity analysis in tweets about impeachment process of brazil president". In: Proceedings of the 22nd Brazilian Symposium, 2016, pp. 335–338.

APÊNDICE A – STOP LIST

a, about, above, after, again, against, all, am, an, and, any, are, aren't, as, at, be, because, been, before, being, below, between, both, but, by, can't, cannot, could, couldn't, did, didn't, do, does, doesn't, doing, don't, down, during, each, few, for, from, further, had, hadn't, has, hasn't, have, haven't, having, he, he'd, he'll, he's, her, here, here's, hers, herself, him, himself, his, how, how's, i, i'd, i'll, i'm, i've, if, in, into, is, isn't, it, it's, its, itself, let's, me, more, most, mustn't, my, myself, no, nor, not, of, off, on, once, only, or, other, ought, our, ours, ourselves, out, over, own, same, shan't, she, she'd, she'll, she's, should, shouldn't, so, some, such, than, that, that's, the, their, theirs, them, themselves, then, there, there's, these, they, they'd, they'll, they're, they've, this, those, through, to, too, under, until, up, very, was, wasn't, we, we'd, we'll, we're, we've, were, weren't, what, what's, when, when's, where, where's, which, while, who, who's, whom, why, why's, with, won't, would, wouldn't, you, you'd, you'll, you're, you've, your, yours, yourself, yourselves, Supra, Passim, Opus citatum, op. cit., Loco citado, loc. cit., Ipsis verbis, Ipsis litteris, In, Idem, Id, ibid, Ibidem, Et. al., Et, al, Apud, a, b, c, d, e, f, g, h, i, j, k, l, m, n, o, p, q, r, s, t, u, v, w, x, y, z.

APÊNDICE B – LISTA DE REFERÊNCIA - TACCESS

aac, aba, ability-based design, accessibility, accessibility guidelines, accessibility research, accessibility technology, accessibility technology for people who are deaf, accessibility technology for the deaf, accessible bus stops, accessible computing, accessible graphs, accessible i.t., access technology, ada, adaptive interface, adaptive technology, adaptive user interfaces, aging, als, alttyping, alzheimer 's disease, ambient intelligence, ambiguous keyboard, american sign language, animation, aphasia, articulation disorders, ascii, asd, asl, asr accuracy prediction, assistive devices, assistive technologies, assistive technology, auditory display, auditory menus, augmentative and alternative communication, authentication, autism, automated speech analysis, automatic assessment of pronunciation, automatic speech processing, automatic speech recognition, baseline, battery power, behavioral strategies, bit rate, blind, blind and visually impaired, blind children, blind community, blindness, blind users, brain-based interfaces, brain-computer interface, bus stop auditing, calibration, captcha, care staff, cast, childhood apraxia of speech, children, cognitive, cognitive disabilities, cognitive disability, collaboration, collaborative games, color differentiation, color vision deficiency, communication, communication model, communication rate, comprehension, computer access, computer mouse, computer use, computing methodologies, computing milieux, constructionism, context-aware interaction, cooperation, coping framework, crowdsourcing, crowdsourcing accessibility, cyberglove, deaf, deaf community, dementia, design, design for all, direct-brain interface, disabilities, disability, disordered speech, distance learning, dmd, doi, doj, dol, dot, down syndrome, dredf, dwell-free, dysarthria, eeoc, eit, email, evaluation, expanding targets, expansion, experimentation, eye-gaze, eye-tracking, eye typing, factor analysis, fcc, fctd, federal region ix, field evaluation, filteryedping, frame rate, framework, friedreich 's ataxia, functional nearinfrared, galvanic skin response, game accessibility, gaze, gestures, goodness of pronunciation, google street view, government web sites, graphical user interfaces, gui, haptic, haptic and audio interfaces, haptic information perception, haptic interfaces, haptics, hard of hearing, hco, health information seeking, hierarchical task analysis, hip, hta, hud, human computation, human-computer interaction, human–computer interaction, human evaluation, human factors, image description, inclusion, inclusive design, universal design, inclusive education, individual characteristics, information search, information systems, information visualization, intelligibility, intelligibility assessment, ipad, ivectors, keyboard replacement, learning, lexical simplification, locationawareness, low-vision and blind users, magnitude estimation, measurement, measures, mechanical turk, metadata, metadata extraction, mobile computing, mobile device, mobile phone, mobile robotics, mobility, motion-based games, motion capture, motion-capture glove, motor disability, mouse emulator, multimodal interaction, multimodal interfaces, natural language generation, navigation, nca, ncd, neuromuscular disease, niddr, nmf, nonvisual charts navigation, nonvisual interaction, objective detection of acoustic anomalies, odep, ofccp, older adults, on-body interaction, open source component, optimization, pdf, pen-based computing, people with visual impairments, performance, person-centered care, persons with mobility disabilities, point and click, pointing performance, prior experience, prior knowledge,

readability, real-time captioning, real-world data collection, remote data collection, representative users, research methods, residential care, resna, screen design, section 508, security, sentence prediction, sign language, situation-specific modeling, smart home, smartphone, social, social computing, social skills, sonification, spanish, speech, speech disorders, speech-language therapy, speech therapy, spindex, standardization, stroke, stroke rehabilitation, sts, supervised classification, syndrome, syntactic simplification, tactile communication, target selection, tdd, technology experience, tel, text simplification, text simplification corpus, therapy, title i, title ii, title iii, touchscreen, transcoding, troubleshooting, trs, tty, universal design, universal usability, user evaluation, user interfaces, user interfaces for all, user profiles, user study, vco, video analysis, video compression, video games, video sharing, virtual characters, virtual environments, virtual peers, visual impairments, visually impaired, voice command, voice conversion, voice reconstruction, vrs, w3c, wai, wcag, web, web accessibility, web survey, whispers, whisper-to-speech conversion, word prediction.

APÊNDICE C – LISTA DE REFERÊNCIA - TOCT

abc conjecture, acyclic graph, advice, algebra, algorithmic mechanism design, algorthms, alphabet, alternation, ambiguity in context free grammar, approximate certificate complexity, approximate decision tree complexity, approximate privacy, approximation, approximation algorithms, approximation resistance, arithmetic circuits, arthur-merlin games, auxiliary push down automata, average-case analysis, average-case complexity, average preimage size, bilinear program, binary sequences, bipartite graph, bipartite graphs, boolean hidden shift problem, bounded-depth frege, bpp, branching programs, branching program size lower bounds, cactus graph, canonical derivations, cell-probe model, cellular automata, certificate complexity, chomsky normal form, circuit complexity, circuit games, class np, class p, coalitional games, coin-weighing problems, collusion, communication complexity, compact representations, comparator circuits, competitive analysis, complete bipartite graph, complete graph, complexity, complexity classes, computable analysis, computational and structural complexity, computational complexity, computational game theory, computations on polynomials, concatenation of strings, concise games, conjunctive queries, connected graph, co-nondeterminism, constraint satisfaction problem, constraint satisfaction problems, constraint satisfactory problems linear, context free grammar, context sensitive grammar, correlation bounds, counting classes, counting complexity, counting modulo 2, counting problems, cut problems, cyclic graph, cyk algorithm, data processing inequality, data streams, dead states, decision tree complexity, decoding, dense model theorem, derandomization, derviation tree, design, determinant, deterministic finite automata, dfa, dictatorship test, directed graph, directed graph reachability, distortion, distributed computing, dpll elimination rules, e, economics, edge clique cover, edge completion, effective convergence, effective fractal dimensions, embedding, empty or null string, entangled games, entropy, error-correcting codes, evolvability, exp, expander codes, extended l-system, extended turing machines, external-memory algorithms, extractors, fiecursive language, finite automata, fixed parameter tractability, fourier analysis of boolean functions, frequency computation, frequency moments, fully homomorphic encryption, game theory, gaussian stability, grammars, graph, graph algorithms, graph coloring, graph decompositions, graph games, graph homomorphism, graph homomorphisms, graph packing, graphs, graphs on surfaces, greibach normal form, grothendieck inequality, hamming distance, hardness amplification, hardness of approximation, hard sequences, hereditary graph properties, higher-type complexity, homomorphism duality, homomorphism problems, influence of a boolean function, information theory, interactive proofs, intractable problem, ip, isolating lemma, isomorphism problems, iterative compression, kernelization, kernelization lower bounds, kernels, kolmogorov complexity, label cover, languages, lang-waldschmidt conjecture, large alphabet sets, lattices, ldpc codes, learning with errors, leftmost derivation, left recursion, length of a string, limited nondeterminism, linear bounded automata, linear programming, linear programming, multiway cut, lipschitz, list approximations, local filter, locality-sensitive hashing, locally decodable codes, locally samplable sources, logarithmic forms, logdcfl, log space, log-space reduction, low-degree polynomials, lower bounds, lr(k) grammars, lsh, l-systems, ma, markov algorithm, matrix

fractional program, matrix multiplication, max-2lin, max-3lin, mealy machine, measure hypothesis, minimal indices, mixed derivation, modification, modified post correspondence problem, monotonicity, moore machine, motzkin-straus theorem, multidimensional turing machines, multigraph, multi head tufiing machine, multiset, multistack machines, multitape tufiing machine, mutual information, nash equilibrium, nc, ndfa, ne, nexp, nl, noise sensitivity, noise stability, non determininistic finite automata, nondeterminism, non-deterministic pushdown automata, non deterministic tufiing machines, nonlinear optimization, np, np-completeness, np-hardness, nucleolus, null productions, numberings, numerical solutions of odes, off-line turing machines, one-way function, online algorithms, online learning, optimal algorithms, optimal proof systems, optimization, oracles, overlap, palindrome, parallel-repetition, parameterized algorithms, parameterized complexity, parameterized logarithmic space, parity, parity complexity dichotomy, partial derivation tree, partially observable markov decision process, pathwidth, p-completeness, pcp, pebbling games, perfect matching, performance, permanent, pigeonhole principle, planar graphs, planarity, planner graph, polynomial, polynomial parametric transformations, polynomials, post correspondence problem, post markov thue (pmt), power of an alphabet, power set of a set, p/poly, prediction, price of anarchy, privacy, privacy trade-off, probabilistic parsing, programming, proof circuits, proof complexity, property testing, propositions or statements, pseudo graph, public-coin protocols, pumping lemma for context free languages, pumping lemma for regular language, pushdown automata, quantum algorithms, quantum communication, quantum metropolis sampling, query complexity, ramsey theory, random bipartite graph, randomness, read/write streams, recursively enumerable language, reducibility, regular expressions, rejection sampling, relations, resolution, restricted backtracking algorithm, restricted turing machines, reverse of a string, rewriting systems, rightmost derivation, rp, satisfiability, satisfiability problem, second-order polynomials, security, selfish routing, sentential form, set, set packing, sleeping experts, small depth proofs, smoothed analysis, solution concepts, soundness error, sparse recovery, stay option turing machine, stochastic context-free grammars, stochastic controller, storage management, string rewriting system, strings, subexponential parameterized complexity, sublinear query approximation algorithms, subset of a set, substitution rule, substring, subtree of derivation tree, succinct representation of numbers, sum of square roots, sum-of-square-roots problem, symmetric chains, term rewriting system, the kleene star, theory, the unique games conjecture, threshold predicates, time-space trade-offs, transition table, tree, tree graph on tree, trees, turing machine, turing machine with semi-infinite tape, two way infinite tape, undecidable problems, unique games conjecture, unit productions, universal algebra, universal turing machines, unreachable state, unrestricted grammar, usefulness, useless productions, venn diagrams, vertex deletion problems, vickrey auction, weighted graph, well-quasi-ordering, worst-case prediction, xor games, yield of derivation tree, zk.

APÊNDICE D – LISTA DE REFERÊNCIA - TKDD

accuracy, active learning, aggregating method, aggregation, algorithm automatic recommendation, algorithms, ann, anomaly detection, anomaly trajectory, anonymity, antibiotic resistance, api, application programming interface, artificial intelligence, artificial neural network, ascos++, association, association discovery, association rule, association rule mining, association rule quality, association rules, asymptotic convergence, attributed graph clustering, attribute similarity, automorphic equivalence, back propagation, bayesian method, bayesian models, bayesian networks, behavioral modeling, behavioral targeting, belief propagation, bic, binary tree, binning, bioacoustics, birthday paradox, blocking, block minimization methods, blogs, bootstrap, bootstrapping, bound estimation, bregman divergence, brute force algorithm, candecomp, cardinality, cart, cascade svm, categorical data, causal modeling, chaid, chernoff distance, chi-square automatic interaction detector, citation analysis, classification, classification and regression trees, classification rules, class separability, click log analysis, closed subsequence, cluster alignment, cluster indicator matrix, clustering, clustering coefficient, clustering coefficients, clustering quality, coauthor network, co-clustering, coherence, collaboration, collaborative filtering, collaborative tagging, collective intelligence, collinearity, column generation simplex, communications network, community detection, community discovery, community evolution, community recommendation, community structure, comparative document summarization, complex network, complex networks, composite social network, computational journalism, conditional probability, conductance, confidence, confidence weighted learning, connected components, context-aware, context recognition, correlated change, correlation-based clustering, cost-sensitive learning, coverage, crm, cross-correlation, cross-language classification, cross-media analysis, cross validation, customer relationship management, cyber-physical system, data, database, database applications, database management, database management system, data cleaning, data integration, data mining, data mining algorithms, data publishing, data selection, data streams, data transformation, data visualization, dbms, decision model, decision tree learning, decision trees, deduplication, density-based clustering, design, detection algorithms, dimensionality reduction, discrimination, discriminative sentence selection, disparate clustering, distance learning, distributed classification, distributed clustering, distributed data mining, diverse clustering, document clustering, document clustering and classification, dominant pattern, dpchain, dual averaging method, dyadic data, dynamical systems, dynamic graph mining, dynamic networks, dynamic social network analysis, ego networks, embedded data mining, empirical risk minimization, ensemble methods, ensembles, ensemble selection, entity matching, entropy, error rate, evolution, evolutionary clustering, evolving pattern mining, expectation maximization, experimental evaluation, experimentation, expert system, exploratory data analysis, factor analysis, factorization, fault detection, feature evaluation and selection, feature selection, field, filtering, fkt, frequent itemset mining, frequent itemsets, frequent patterns, front office, fuzzy logic, fuzzy set, fuzzy system, gap-constraint, gene expression pattern, generalized linear model, general local region algorithm, genetic algorithm, genetic operator, gibbs, gibbs sampling, gini metric, gleam, global/local outliers, graph augmentation, graph

clustering, graphical models, graph matching, graph mining, graph model, graph sampling, graph streams, grid computing, hadi, hadoop, hdp-htm, healthcare, hebbian learning, heterogeneous information network, heterogeneous information networks, hidden markov model, hidden markov models, hierarchical and nonhierarchical clustering, hierarchical ensemble clustering, hierarchical transition matrix, hierarchy, higher-order, higher-order hmm, high-performance and terascale computing, hill climbing search, hmm with duration, hypothesis testing, id3, ihms, image annotation, imbalanced data, inclusion-exclusion, independence, inference, information, information cascades, information extraction, information filtering, information geometry, information propagation, information propagation pattern, information search and retrieval, instance annotation, intelligent agent, invariant networks, inverse problem, isolation, isolation forest, item-response theory, itemset discovery, itemset screening, iterative scaling, k-anonymity, kernel methods, knowledge communities, knowledge discovery, knowledge discovery in graphs, knowledge presentation, knowledge transfer, kohonen networks, label relationship, landmarks, large graphs, large-scale, large-scale data, large-scale learning, latent variable models, lda, l-diversity, leakage, learning, least squares loss, lift, likelihood ratio test, linear classification, link analysis, link mining, link prediction, local region information, lower bounds, low-rank and sparse patterns, low rank approximation, machine learning, manifold, mapreduce, marginal probability distribution, massive-graph computing, massive networks, matrix perturbation theory, maximal frequent subgraph mining, maximum entropy, maximum-likelihood estimation, maximum mean discrepancy, mbr, mcmc, mdl, mean reversion, measurement, meme-tracking, memory-based reasoning, mercer kernel, message cascade, message-passing, metafac, metagraph factorization, meta-path selection, metric anomaly ranking, microblogging services, minimal hypergraph transversals, minimum description length, minimum description length principle, mining, mixing time, mixture models, mobile phone, model, model-based clustering, mogs, motifs, multidimensional databases, multidocument summarization, multi-instance, multi-label, multi-label classification, multilabel classification, multilabel k nearest neighbors, multilabel learning, multilevel patterns, multimodal data, multiplayer online games, multiple comparison procedure, multiresolution, multiscale, multisource domain adaption, multitask, multitask clustering, multitask learning, name disambiguation, nearest neighbor, nearest neighbor rule, negative transfer, netflix prize, network alignment, network engineering, network evolution, network motifs, networks, network sampling, networks of diffusion, network structure index, neural network, news media, nmf, nominal categorical predictor, nonnegative matrix, nonnegative matrix factorization, nonnegative matrix factorization with given bases, nonnegative matrix trifactorization, nonnegative tensor factorization, nonredundant clustering, occam 's razor, occupancy, olap, on-line algorithm, on-line analytical processing, online learning, online social networks, ordinal categorical predictor, orthogonalization, outlier analysis, outlier detection, overfitting, p2p networks, parafac, parafac decomposition, parallel algorithms, parallel data mining, pattern evaluation, pattern sets, performance, physiological signals, portfolio selection, prediction, predictive model, predictive modeling, predictor, primacy and recency effects, principle components analysis, prior probability, privacy, privacy preservation, probabilistic model, probability density functions, prominence, propagation, propagation tree, radial basis function networks, radius plot, randomized algorithms, random projections, random tree ensemble, random walks, ranking,

rationality analytics, rdb, reciprocal relationship, recommendation, recommender systems, record, record linkage, regression, reinforcement learning, relational, relational classification, relational database, relational hypergraph, repetitive subsequence, representative and discriminative, resource discovery, response, retrieval, robust, role, role similarity, rule mining, sampler, sampling, segmentation, semiautomated text classification, semisupervised learning, sensitivity analysis, sensor network, sequence database, sequence mining and modeling, sequential pattern, sequential patterns, shared subspace, shortcuts, shortest path distance, similarity search, simrank, simulated annealing, singular value decomposition, skyline query, small web, social annotation, social circles, social influence, social information processing, social media, social network, social network analysis, social networks, social relationship, spam, sparsity, spatial anomaly, spectral, sports analytics, sql, statistical evaluation, statistical methods, streaming algorithms, streaming graphs, structural proximity, structured query language, subgradient, subgraph patterns, subject-based variability, sufficient statistics, suffix arrays, summarization, supervised classification, supervised learning, support, support vector machine, support vector machines, surface electromyogram, sybil attacks, targeted marketing, temporal analysis, tensor decomposition, tensor factorization, tensors, test set validation, text classification, text mining, theory, time series, time-series, time-series analysis, time series database, time-series forecasting, topic model, topic modeling, trace norm, traffic classification, trajectory, transaction data, transductive learning, transfer, transfer learning, transitivity, triangle counting, triangle listing, trust network, trust prediction, tucker, twitter, ultra-metric, uncertain data, unsupervised and semisupervised clustering, unsupervised learning, user behavior, user-community-topic model, user-guided clustering, user identification, user interaction, user modeling, user profiling, user recommendation, variable-order hmm, variable selection, vector field, vertex similarity, visualization, web computing, web page categorization, web search, weighted networks.

APÊNDICE E – LISTA DE REFERÊNCIA - TOSEM

abstract dynamic slicing, acceptance criteria, acceptance process, acceptance testing, accuracy, active learning, activity, activity diagram, actor, adaptive changes, adaptive random testing, additional statement coverage, additional strategy, advice dispatch, aes, agent organizations, agent-oriented software engineering, agile method, algebra, algorith, algorithms, alignment, alloy, alloy calculus, analysis model, analysis models, anomaly, aop, api discovery, api migration, application, architectural views, array invariants, aspect-oriented intermediate-languages, aspect oriented programming, aspect-oriented programming, aspect-oriented virtual machines, assertion checkers, assessment, assumption, attribute, audit, augmented constraint network, authentication, authorship, automated repair, automated test generation, automated testing, automated verification, automatic addition of fault tolerance, automatic test case generation, automatic test data generation, automatic test generation, automatic workarounds, automation, bandwidth, baseline, baseline model, batch processing, behavioral modeling, benchmark, bi-infinite time, boa, boolean specifications, bounded model checking, bpel, branch coverage, bugdetection, bug report triage, business process complexity, business processes, business process maturity, business process model, business rule, call graphs, cardiac pacemakers, case study, centralised testing, certification and accreditation, cfj, check-point selection, checkpoint selection, cide, class cohesion, class diagram, class hierarchy constraints, class hierarchy structure, class size, client, cluster analysis, clustering, code, code completion, code generation, code recommender, code review, code smells, coincidental correctness, co-installability, collaborative software development, collaborative systems, combinatorial testing, comparison, compatiblity and replaceability analysis, component, component-based software engineering, comprehensibility, computer-aided software engineering, computer software, concept location, conceptual modeling, conceptual queries, concerns, concolic testing, concurrency, concurrency bugs, conditional compilation, configuration assistance, configuration management, conflicts, conformance, confounding effect, consistency, consistent configuration, constraint, constraint programming, constraints, constraint satisfaction techniques, container profiling, content assist, context-aware, context-aware computing, context-aware software engineering, context diagram, context-free languages, contingency, continuous asm, control, controlled experiment, controller synthesis, conversion, cookies, copy profiling, correct-by-construction control software synthesis, cost-benefit analysis, cots components, countermeasure, coverage-based fault localization, coverage criteria, critical success factors, cryptographic software, csfs, csl, ct, ctmc, customer resources, data, database, database administrator, database application testing, database management system, database object, data dictionary, data element, data entity, data flow diagram, data mining, data models, data type, dbms, debugging, decision procedure, declarative programming, default, defect detection, defects, degree-of-interest, degree-of-knowledge, deliverable, demand-driven, dependencies, deployed analysis, design, design analysis, design aspect, design decisions, design document, design element, design modeling, design pattern, design patterns, design rules, design stage, design structure matrix, detection and cause identification, developer skills, development environment, development stage, diagnosis, differen-

tial power analysis, distributed testing, documentation, domain-specific language, domain-specific modelling languages, dtmc, dynamic adaption of service-oriented applications based on standard apis, dynamic analysis, dynamic checking, dynamic symbolic execution, ease of use, embedded software development, empirical evaluation, empirical software engineering, empirical studies, empirical study, emulation, enabledness abstractions, encryption, end-user development, end-user programming, end-user review, engineering, ensembles of learning machines, entity, entity-relationship diagra4, equational reasoning, equivalence checking, european train control system, evaluation, event-driven programming, evolution, evolutionary testing, exception handling, exception handling in component-based software systems, executable, execution, existing test cases, experimentation, expertise, expert opinion, explicit announcement, explicit join points, extensible middleware, external entities, external interface, external interface complexity, extreme programming, failure diagnosis, false warnings, family of experiments, fault, fault-based test case prioritization, fault-based testing, fault class, fault localization, faults, feasibility study, featherweight java, feature identification, field, field failures, finite satisfiability, flags, focus, foreign key, formal concept analysis, formal iteration process, formal languages, formal method, formal methods, formal specification, formal specification languages, forms, framework, functional area, functional element, functional size, functional testing, function point analysis, fuzzing, fuzzy sets, generalization set constraints, generative middleware, generative programming, genetic algorithms, geometric invariant inference, gilligan, graph matching, group, gui testing, hardware, hazard, healing connectors, healing patterns, heap assertions, heuristic searches, hierarchical menu, hmm, html, human-computer interaction, human factors, hybrid systems, hypertext markup language, identification graph, ifdef, implementation element, implementation stage, implicit announcement, implicit invocation, incremental development, index, informal iteration process, information, information flow analysis, information foraging, information needs, information retrieval, information system, information theory, inheritance, initial data load, initial risk, inner query, inner source, insertion point, inspection, in- stage assessment process, installation and acceptance stage, intege, integer overflow, integer wraparound, integrated development environments, integrated development methodology, integration and test stage, integrity, integrity constraints, intellectual property, interaction overview diagrams, interfac, interfaces, interface testing, inter-object approach, intersection, invariant generation, invocation, iterative development model, jad, java, join, join point polymorphism, joint application design, junit, kb, key field, key process area, kickoff process, kilobyte, knowledge base, knowledge engineering, knowledge management, knowledge sharing, labeled transition systems, language design, languages, latent semantics, leaking confidence, legacy study, lifecycle, lightweight process and tooling, lightweight specification, live sequence charts, location-aware, logical design document (ldd), low commitment, lower barrier to entry, low-level design, machine learning, mac-keccak, maintainability, maintenance, management, master table, mathematical computation, mb, mbt, mdd, mde, megabyte, megahertz, memory leaks, merge, message sequence charts, metadata, metamodelling, metamodelling patterns, method, method-method interaction, methodology, metrics, mfs, mhz, middleware, milestone, minimal failure-causing schema, mining, mining software repositories, mobile application development, mobile applications, mobile computing, mock classes, model, model-based design of control soft-

ware, model-based product-line engineering, model-based testing, model checking, model-driven development, model-driven engineering, model merging, modifiability, modularity, module, module and runtime views, module testing, monitors, mts, multilevel modelling, multi-objective evolutionary algorithms, multiplicity constraints, multitolerance, multiuser, mutation analysis, natural language, natural language processing, next release problem, non-adequate test suites, nonlinear invariants, non-zenoness, normalization, nu, oas, object code, object-oriented software quality, obliviousness, ocl, odbc, office automation system, olap, oltp, onboarding, online analytical processing, online community, online transaction processing, ontology-based domain modeling, open database connectivity, open source, open-source development practices, open-source software, open source software development, operating system, operational data area, operational transaction load, opportunistic development, options, organization, origin analysis, orphan, outer join, outer query, package management, pagerank, parameter, partial behavior models, partial program analysis, pat, path explosion, path segments, path-sensitive analysis, pattern composition, pctl, pdr, peer revier, peer review, per, perfect masking, performance, permissions, pervasive computing, pip, place-aware, place recognition, planning, planning stage, pragmaticreuse plan enactment, pragmatic-reuse plans, pragmatic software reuse, prediction, primary developer representative, primary end-user representative, primary key, privacy, privileges, procedure, process calculi, process risk, product configuration, production, production initiation plan, productivity, program analysis, program comprehension, program differencing, programming stage, program spectra, project, project manager, project plan, prototyping, provenance, pseudocode, publish/subscribe, pvs, quality, quality assurance assessment, quality of service, query, query processing, questions, random prioritization, rapid prototypin, rdbms, real-time systems, reasoning rules, recommendation, recommendation system, record, refactoring, referential integrity, refinement, regression testing, regular languages, reinforcement learning, relational database management system, relational database schemas, relational data model, relational topic modeling, release version, reliability, reliability of service-oriented architectures, reliability prediction, remodularization, replicated experiments, replications, representation, required behavior, requiremen, requirement engineering, requirements specification, requirements stage, requirements traceability, requirements validation, requirements visualization, restriction rules, retirement, reusability, reverse engineering, review, rfbi, rigorous design and development, risk, risk evaluation formulas, risk management, risk measurement, role-based modeling, rule, runtime bloat, runtime monitoring, runtime verification, safety-critical applications, satisfiability modulo theory, scala, scalable, scenario-based programming, scenario-based requirements, scenarios, schema testing, scientific workflows, screen mockups, sdd, sdlc, search archetypes, search-based software engineering, search-based software testing, security, self-healing, self-join, semantic code search, semantic relatedness, semantics, semantic web, semi-supervised learning, sensing, sensitivity analysis, sequence diagram, sequencing and path properties, service contracts, service-oriented architecture, service-oriented computing, side-channel attack, similarity function, simultaneous users, small world networks, sme, smt, smt solvers, soa, social networks, software, software and system safety, software architecture, software changes, software code smells, software configuration management, software debugging, software design document, software design methodology, software development, software development lifecycle, software eco-

nomics, software effort estimation, software engineering, software evolution, software evolution and maintenance, software maintenance, software metrics, software modeling, software models, software modularization, software process assessment, software product line engineering, software product lines, software/program verification, software project management plan, software quality, software quality assurance, software requirements document, software requirements engineering, software reuse, software safety, software testing, solvability of linear inequality system, sops, source code, source-code analysis, source-code comprehension, source code investigation, source-code search, source-code validation, specification, spectrum-based fault localization, spiral development model, sqa, sql, srd, stage, stage exit process, stakeholders, standard operating procedures, standards, state equivalence detection, stateful timed csp, state machine synthesis, static analysis, static checking, static program analysis, static race detection, static verification, statistical debugging, stress testing, string analysis, string constraints, strong and weak conditional commitments, strong coincidental correctness, structured analysis, structured query language, stubs, subject matter expert, subquery, subtyping, supply chain, support activities, support data area, symbolic, symbolic analysis, symbolic execution, synchronization, synthesized database interactions, system, systematic literature review, system design stage, system owner, systems analysis, systems analyst, system software, system testing, table, tailoring, task, task assignment, task deadline, task models, taxonomy, team formation, technical feasibility, technical review, technology, temporal constraints, temporal logic, temporal verification, testability, testability transformation, test amplification, testbed, test case, test case generation, test case minimization, test case prioritization, test case selection, test generation, testing, testing stage, test item, test plan, test results document, test strategies, test transformation, theorem proving, theory, three-tier, timed automata, timed business protocols, timestamp, tool support, total strategy, traceability, trace semantics, transaction, transaction analysis, transformation, transformed linear model, type hierarchy, typestate, type system, typing, uml, uml class diagram, uml/ocl, uml sequence diagrams, undefined behavior, unified modeling language, unit, unit testing, unit tests, unsat-cores, usability, use case, use case modeling, use case models, use cases, use case template, user, user-collaboration, user interface, user manual, user studies, validation, validation criterion, value-based software engineering, variability, variation, verification, verification and validation plan, violation handling point selection, virtual machines, virtual organization, visual formalisms, visual programming, vvp, walk-through, waterfall development model, wbs, weak coincidental correctness, weaving, web api, web applications, web application security, web application testing, web security, web service, web service aggregation, web services, white box reuse, whyline, widow, word equations, work breakdown structure, workflows, work product, xp, yawl, zone abstraction.

APÊNDICE F – RECALL FINAL 100+ - TACCESS E TOCT

Tabela 23 – Recall Final do corpus TACCESS (100+ e 1.000+)

Recall Final TACCESS (1^2-gram)			
Lista	**Recall 1000+**	**Recall 100+**	**Total Corpus**
E	**41**	**3**	251
E+stop	52	14	251
E+tf-dcf	75	13	251
E+tf-idf	77	26	251
E+stop+tf-dcf	77	29	251
E+stop+tf-idf	**83**	**30**	251
L	64	7	251
L+tf-dcf	70	13	251
L+tf-idf	76	10	251

Tabela 24 – Recall Final do corpus ToCT (100+ e 1.000+)

Recall Final ToCT (1^2-gram)			
Lista	**Recall 1000+**	**Recall 100+**	**Total Corpus**
E	**17**	**4**	260
E+stop	41	6	260
E+tf-dcf	37	7	260
E+tf-idf	39	**10**	260
E+stop+tf-dcf	53	8	260
E+stop+tf-idf	**59**	**10**	260
L	55	5	260
L+tf-dcf	54	6	260
L+tf-idf	53	6	260

APÊNDICE G – RECALL FINAL 100+ - TKDD E TOSEM

Tabela 25 – Recall Final do corpus TKDD (100+ e 1.000+)

Recall Final TKDD (1^2-gram)			
Lista	**Recall 1000+**	**Recall 100+**	**Total Corpus**
E	**32**	**5**	424
E+stop	52	13	424
E+tf-dcf	83	17	424
E+tf-idf	95	10	424
E+stop+tf-dcf	92	**21**	424
E+stop+tf-idf	**104**	15	424
L	63	8	424
L+tf-dcf	68	**5**	424
L+tf-idf	71	7	424

Tabela 26 – Recall Final do corpus TOSEM (100+ e 1.000+)

Recall Final TOSEM (1^2-gram)			
Lista	**Recall 1000+**	**Recall 100+**	**Total Corpus**
E	**72**	**8**	636
E+stop	94	20	636
E+tf-dcf	110	22	636
E+tf-idf	124	24	636
E+stop+tf-dcf	129	**28**	636
E+stop+tf-idf	**134**	**28**	636
L	121	20	636
L+tf-dcf	124	17	636
L+tf-idf	128	21	636